GigaDevice Semiconductor Inc.

GD32G553
Arm® Cortex®-M33 32-bit MCU

User Manual

Revision 1.2

(Aug. 2025)

Table of Contents

Table of Contents **2**

List of Figures **25**

List of Tables **35**

1. System and memory architecture **40**

1.1. Arm® Cortex®-M33 processor **40**

1.2. System architecture **41**

1.3. Memory map **43**

1.3.1. On-chip SRAM memory 48

1.3.2. On-chip Flash memory 51

1.4. Boot configuration **51**

1.5. Remap configuration **52**

1.6. System configuration controller **53**

1.7. System configuration registers **54**

1.7.1. System configuration register 0 (SYSCFG_CFG0) 54

1.7.2. System configuration register 1 (SYSCFG_CFG1) 55

1.7.3. EXTI sources selection register 0 (SYSCFG_EXTISS0) 56

1.7.4. EXTI sources selection register 1 (SYSCFG_EXTISS1) 57

1.7.5. EXTI sources selection register 2 (SYSCFG_EXTISS2) 59

1.7.6. EXTI sources selection register 3 (SYSCFG_EXTISS3) 60

1.7.7. System configuration register 2 (SYSCFG_CFG2) 61

1.7.8. System status register (SYSCFG_STAT) 62

1.7.9. System configuration register 3 (SYSCFG_CFG3) 64

1.7.10. System configuration register 4 (SYSCFG_CFG4) 65

1.7.11. System configuration register 5 (SYSCFG_CFG5) 66

1.7.12. SYSCFG TCMSRAM control and status register (SYSCFG_TCMSRAMCS) 67

1.7.13. SYSCFG TCMSRAM key register (SYSCFG_TCMSRAMKEY) 67

1.7.14. SYSCFG TCMSRAM write protection register (SYSCFG_TCMSRAMWP) 68

1.7.15. I/O Compensation cell control/status register (SYSCFG_CPSCTL) 68

1.7.16. Timer input selection register 0 (SYSCFG_TIMERCISEL0) 69

1.7.17. Timer input selection register 1 (SYSCFG_TIMERCISEL1) 70

1.7.18. Timer input selection register 2 (SYSCFG_TIMERCISEL2) 71

1.7.19. Timer input selection register 3 (SYSCFG_TIMERCISEL3) 73

1.7.20. Timer input selection register 4 (SYSCFG_TIMERCISEL4) 75

1.7.21. TIMERx configuration register 0 (SYSCFG_TIMERxCFG0, x=0, 1, 2, 3, 4, 7, 19) 77

1.7.22. TIMERx configuration register 1 (SYSCFG_TIMERxCFG1, x=0, 1, 2, 3, 4, 7, 19) 79

1.7.23. TIMERx configuration register 2 (SYSCFG_TIMERxCFG2, x=0, 1, 2, 3, 4, 7, 19) 82

1.7.24. TIMERx configuration register 0 (SYSCFG_TIMERxCFG0, x=14) 84
1.7.25. TIMERx configuration register 1 (SYSCFG_TIMERxCFG1, x=14) 86
1.7.26. TIMERx configuration register 2 (SYSCFG_TIMERxCFG2, x=14) 87
1.8. Device electronic signature 89
1.8.1. Memory density information 89
1.8.2. Unique device ID (96 bits) 89
2. Flash memory controller (FMC) 91
2.1. Overview 91
2.2. Characteristics 91
2.3. Function overview 92
2.3.1. Flash memory architecture 92
2.3.2. Error Checking and Correcting (ECC) 93
2.3.3. Read operations 94
2.3.4. Dual bank architecture with read-while-write (RWW) capability 96
2.3.5. Unlock the FMC_CTL register and FMC_OBCTL register 96
2.3.6. Page erase 97
2.3.7. Mass erase 98
2.3.8. Main flash programming 99
2.3.9. OTP programming 101
2.3.10. Option bytes 102
2.3.11. Dedicated code read protection area (DCRP) 107
2.3.12. Erase/program protection (WP) 108
2.3.13. Security protection (SPC) 109
2.3.14. Secure user area (SCR) 110
2.3.15. Disabling core debug access 111
2.3.16. Forcing boot from Flash memory 111
2.3.17. FMC interrupts 111
2.4. Register definition 112
2.4.1. Wait state register (FMC_WS) 112
2.4.2. Unlock flash mode during run mode key register (FMC_RUNKEY) 113
2.4.3. Unlock key register (FMC_KEY) 114
2.4.4. Option byte unlock key register (FMC_OBKEY) 114
2.4.5. Status register (FMC_STAT) 114
2.4.6. Control register (FMC_CTL) 116
2.4.7. ECC control and status register (FMC_ECCCS) 118
2.4.8. Option byte control register (FMC_OBCTL) 120
2.4.9. DCRP start address register 0(FMC_DCRP_SADDR0) 122
2.4.10. DCRP end address register 0(FMC_DCRP_EADDR0) 123
2.4.11. Bank0 erase/program protection area 0 register (FMC_BK0WP0) 124
2.4.12. Bank0 erase/program protection area 1 register (FMC_BK0WP1) 124
2.4.13. DCRP start address register 1(FMC_DCRP_SADDR1) 125

2.4.14. DCRP end address register 1(FMC_DCRP_EADDR1) 125
2.4.15. Bank1 erase/program protection area 0 register (FMC_BK1WP0) 126
2.4.16. Bank1 erase/program protection area 1 register (FMC_BK1WP1) 127
2.4.17. Bank0 secure user area register (FMC_BK0SCR) 127
2.4.18. Bank1 secure user area register (FMC_BK1SCR) 128
2.4.19. Product ID register (FMC_PID) 128

3. Power management unit (PMU) 130

3.1. Overview 130

3.2. Characteristics 130

3.3. Function overview 131
3.3.1. Backup domain 132
3.3.2. VDD / VDDA power domain 135
3.3.3. 1.1V power domain 139
3.3.4. Power saving modes 141

3.4. Register definition 144
3.4.1. Control register 0 (PMU_CTL0) 144
3.4.2. Control and status register 0 (PMU_CS) 146
3.4.3. Control register 1 (PMU_CTL1) 148
3.4.4. Control register 2 (PMU_CTL2) 149
3.4.5. Control register 3(PMU_CTL3) 150

4. Reset and clock unit (RCU) 152

4.1. Reset control unit (RCTL) 152
4.1.1. Overview 152
4.1.2. Function overview 152

4.2. Clock control unit (CCTL) 153
4.2.1. Overview 153
4.2.2. Characteristics 156
4.2.3. Function overview 156

4.3. Register definition 160
4.3.1. Control register (RCU_CTL) 160
4.3.2. PLL register (RCU_PLL) 162
4.3.3. Clock configuration register 0 (RCU_CFG0) 164
4.3.4. Clock interrupt register (RCU_INT) 166
4.3.5. AHB1 reset register (RCU_AHB1RST) 169
4.3.6. AHB2 reset register (RCU_AHB2RST) 170
4.3.7. AHB3 reset register (RCU_AHB3RST) 172
4.3.8. APB1 reset register (RCU_APB1RST) 172
4.3.9. APB2 reset register (RCU_APB2RST) 175
4.3.10. APB3 reset register (RCU_APB3RST) 177
4.3.11. AHB1 enable register (RCU_AHB1EN) 179

4.3.12. AHB2 enable register (RCU_AHB2EN) 180
4.3.13. AHB3 enable register (RCU_AHB3EN) 181
4.3.14. APB1 enable register (RCU_APB1EN) 182
4.3.15. APB2 enable register (RCU_APB2EN) 184
4.3.16. APB3 enable register (RCU_APB3EN) 187
4.3.17. AHB1 sleep and deep-sleep mode enable register (RCU_AHB1SPDPEN) 188
4.3.18. AHB2 sleep and deep-sleep mode enable register (RCU_AHB2SPDPEN) 190
4.3.19. AHB3 sleep and deep-sleep mode enable register (RCU_AHB3SPDPEN) 191
4.3.20. APB1 sleep and deep-sleep mode enable register (RCU_APB1SPDPEN) 192
4.3.21. APB2 sleep and deep-sleep mode enable register (RCU_APB2SPDPEN) 195
4.3.22. APB3 sleep and deep-sleep mode enable register (RCU_APB3SPDPEN) 197
4.3.23. Backup domain control register (RCU_BDCTL) 198
4.3.24. Reset source/clock register (RCU_RSTSCK) 201
4.3.25. Clock configuration register 1 (RCU_CFG1) 202
4.3.26. Clock configuration register 2 (RCU_CFG2) 204

5. EXTI introduction Interrupt / event controller (EXTI) 207

5.1. Overview 207

5.2. Characteristics 207

5.3. Interrupts function overview 207

5.4. External interrupt and event (EXTI) block diagram 212

5.5. External Interrupt and Event function overview 213

5.6. Register definition Register definition 214

5.6.1. Interrupt enable register 0 (EXTI_INTEN0) 214
5.6.2. Event enable register 0 (EXTI_EVEN0) 215
5.6.3. Rising edge trigger enable register 0 (EXTI_RTEN0) 215
5.6.4. Falling edge trigger enable register 0 (EXTI_FTEN0) 215
5.6.5. Software interrupt event register 0 (EXTI_SWIEV0) 216
5.6.6. Pending register 0 (EXTI_PD0) 216
5.6.7. Interrupt enable register 1 (EXTI_INTEN1) 217
5.6.8. Event enable register 1 (EXTI_EVEN1) 217
5.6.9. Rising edge trigger enable register 1 (EXTI_RTEN1) 217
5.6.10. Falling edge trigger enable register 1 (EXTI_FTEN1) 218
5.6.11. Software interrupt event register 1 (EXTI_SWIEV1) 218
5.6.12. Pending register 1 (EXTI_PD1) 219

6. Trigger selection controller (TRIGSEL) 220

6.1. Overview 220

6.2. Characteristics 220

6.3. Function overview 220

6.4. Internal connect 221

6.5. Register definition 233
6.5.1. Trigger selection for EXTOUT register 0 (TRIGSEL_EXTOUT_0) 233
6.5.2. Trigger selection for EXTOUT register 1 (TRIGSEL_EXTOUT_1) 233
6.5.3. Trigger selection for EXTOUT register 2 (TRIGSEL_EXTOUT_2) 234
6.5.4. Trigger selection for EXTOUT register 3 (TRIGSEL_EXTOUT_3) 235
6.5.5. Trigger selection for ADC0 register (TRIGSEL_ADC0) 235
6.5.6. Trigger selection for ADC1 register (TRIGSEL_ADC1) 236
6.5.7. Trigger selection for ADC2 register (TRIGSEL_ADC2) 237
6.5.8. Trigger selection for ADC3 register (TRIGSEL_ADC3) 237
6.5.9. Trigger selection for TIMER0_BRKIN register (TRIGSEL_TIMER0BRKIN) 238
6.5.10. Trigger selection for TIMER7_BRKIN register (TRIGSEL_TIMER7BRKIN) 239
6.5.11. Trigger selection for TIMER14_BRKIN register (TRIGSEL_TIMER14BRKIN) 240
6.5.12. Trigger selection for TIMER15_BRKIN register (TRIGSEL_TIMER15BRKIN) 240
6.5.13. Trigger selection for TIMER16_BRKIN register (TRIGSEL_TIMER16BRKIN) 241
6.5.14. Trigger selection for TIMER19_BRKIN register (TRIGSEL_TIMER19BRKIN) 242
6.5.15. Trigger selection for CAN0 register (TRIGSEL_CAN0) 242
6.5.16. Trigger selection for CAN1 register (TRIGSEL_CAN1) 243
6.5.17. Trigger selection for CAN2 register (TRIGSEL_CAN2) 244
6.5.18. Trigger selection for TIMER0_ETI register (TRIGSEL_TIMER0ETI) 244
6.5.19. Trigger selection for TIMER1_ETI register (TRIGSEL_TIMER1ETI) 245
6.5.20. Trigger selection for TIMER2_ETI register (TRIGSEL_TIMER2ETI) 245
6.5.21. Trigger selection for TIMER3_ETI register (TRIGSEL_TIMER3ETI) 246
6.5.22. Trigger selection for TIMER4_ETI register (TRIGSEL_TIMER4ETI) 247
6.5.23. Trigger selection for TIMER7_ETI register (TRIGSEL_TIMER7ETI) 247
6.5.24. Trigger selection for TIMER19_ETI register (TRIGSEL_TIMER19ETI) 248
6.5.25. Trigger selection for HPDF_ITRG register (TRIGSEL_HPDF) 248
6.5.26. Trigger selection for TIMER0_ITI14 register (TRIGSEL_TIMER0ITI14) 249
6.5.27. Trigger selection for TIMER1_ITI14 register (TRIGSEL_TIMER1ITI14) 250
6.5.28. Trigger selection for TIMER2_ITI14 register (TRIGSEL_TIMER2ITI14) 250
6.5.29. Trigger selection for TIMER3_ITI14 register (TRIGSEL_TIMER3ITI14) 251
6.5.30. Trigger selection for TIMER4_ITI14 register (TRIGSEL_TIMER4ITI14) 251
6.5.31. Trigger selection for TIMER7_ITI14 register (TRIGSEL_TIMER7ITI14) 252
6.5.32. Trigger selection for TIMER14_ITI14 register (TRIGSEL_TIMER14ITI14) 253
6.5.33. Trigger selection for TIMER19_ITI14 register (TRIGSEL_TIMER19ITI14) 253
6.5.34. Trigger selection for DAC0 register (TRIGSEL_DAC0) 254
6.5.35. Trigger selection for DAC1 register (TRIGSEL_DAC1) 255
6.5.36. Trigger selection for DAC2 register (TRIGSEL_DAC2) 255
6.5.37. Trigger selection for DAC3 register (TRIGSEL_DAC3) 256
6.5.38. Trigger selection for DAC0 extended register (TRIGSEL_EXTDAC0) 257
6.5.39. Trigger selection for DAC1 extended register (TRIGSEL_EXTDAC1) 258
6.5.40. Trigger selection for DAC2 extended register (TRIGSEL_EXTDAC2) 258
6.5.41. Trigger selection for DAC3 extended register (TRIGSEL_EXTDAC3) 259
6.5.42. Trigger selection for CLA register 0 (TRIGSEL_CLA_0) 260
6.5.43. Trigger selection for CLA register 1 (TRIGSEL_CLA_1) 261

6.5.44. Trigger selection for CLA register 2 (TRIGSEL_CLA_2) 262
6.5.45. Trigger selection for CLA register 3 (TRIGSEL_CLA_3) 262
6.5.46. Trigger selection for CLA register 4 (TRIGSEL_CLA_4) 263
7. General-purpose and alternate-function I/Os (GPIO and AFIO) 265
7.1. Overview 265
7.2. Characteristics 265
7.3. Function overview 265
7.3.1. GPIO pin configuration 267
7.3.2. External interrupt / event lines 268
7.3.3. Alternate functions (AF) 268
7.3.4. Additional functions 268
7.3.5. Input configuration 268
7.3.6. Output configuration 269
7.3.7. Analog configuration 270
7.3.8. Alternate function (AF) configuration 270
7.3.9. GPIO locking function 271
7.3.10. GPIO single cycle toggle function 271
7.3.11. I/O compensation unit 271
7.3.12. Input filtering 272
7.4. Register definition 275
7.4.1. Port control register (GPIOx_CTL, x=A..G) 275
7.4.2. Port output mode register (GPIOx_OMODE, x=A..G) 277
7.4.3. Port output speed register (GPIOx_OSPD, x=A..G) 278
7.4.4. Port pull-up/down register (GPIOx_PUD, x=A..G) 280
7.4.5. Port input status register (GPIOx_ISTAT, x=A..G) 282
7.4.6. Port output control register (GPIOx_OCTL, x=A..G) 282
7.4.7. Port bit operate register (GPIOx_BOP, x=A..G) 283
7.4.8. Port configuration lock register (GPIOx_LOCK, x=A..G) 283
7.4.9. Alternate function selected register 0 (GPIOx_AFSEL0, x=A..G) 284
7.4.10. Alternate function selected register 1 (GPIOx_AFSEL1, x=A..G) 285
7.4.11. Bit clear register (GPIOx_BC, x=A..G) 286
7.4.12. Port bit toggle register (GPIOx_TG, x=A..G) 287
7.4.13. Input filtering register (GPIOx_IFL, x=A..G) 287
7.4.14. Input filtering type register (GPIOx_IFTP, x=A…G) 288
8. Direct memory access controller (DMA) 290
8.1. Overview 290
8.2. Characteristics 290
8.3. Block diagram 291
8.4. Function overview 291
8.4.1. DMA operation 291

8.4.2. Peripheral handshake 293
8.4.3. Arbitration 293
8.4.4. Address generation 294
8.4.5. Circular mode 294
8.4.6. Memory to memory mode 294
8.4.7. Channel configuration 294
8.4.8. Interrupt 295
8.4.9. DMA request mapping 296
8.5. Register definition 297
8.5.1. Interrupt flag register (DMA_INTF) 297
8.5.2. Interrupt flag clear register (DMA_INTC) 297
8.5.3. Channel x control register (DMA_CHxCTL) 298
8.5.4. Channel x counter register (DMA_CHxCNT) 300
8.5.5. Channel x peripheral base address register (DMA_CHxPADDR) 301
8.5.6. Channel x memory base address register (DMA_CHxMADDR) 301
9. DMA request multiplexer (DMAMUX) 303
9.1. Overview 303
9.2. Characteristics 303
9.3. Block diagram 304
9.4. Function overview 304
9.4.1. DMAMUX signals 305
9.4.2. DMAMUX request multiplexer 305
9.4.3. DMAMUX request generator 308
9.4.4. Channel configurations 309
9.4.5. Interrupt 309
9.4.6. DMAMUX mapping 310
9.5. Register definition 316
9.5.1. Request multiplexer channel x configuration register (DMAMUX_RM_CHxCFG) 316
9.5.2. Request multiplexer channel interrupt flag register (DMAMUX_RM_INTF) 317
9.5.3. Request multiplexer channel interrupt flag clear register (DMAMUX_RM_INTC) 317
9.5.4. Request generator channel x configuration register (DMAMUX_RG_CHxCFG) 318
9.5.5. Request generator interrupt flag register (DMAMUX_RG_INTF) 319
9.5.6. Request generator interrupt flag clear register (DMAMUX_RG_INTC) 319
10. Cyclic redundancy checks management unit (CRC) 321
10.1. Overview 321
10.2. Characteristics 321
10.3. Function overview 322
10.4. Register definition 324
10.4.1. Data register (CRC_DATA) 324

10.4.2. Free data register (CRC_FDATA) 324
10.4.3. Control register (CRC_CTL) 325
10.4.4. Initialization data register (CRC_IDATA) 325
10.4.5. Polynomial register (CRC_POLY) 326
11. Configurable Logic Array (CLA) 327
11.1. Overview 327
11.2. Characteristics 327
11.3. Block diagram 327
11.4. Function overview 328
11.4.1. CLA input Signal Selector 328
11.4.2. LCU control 329
11.4.3. CLA Output 330
11.4.4. Interrupt 330
11.5. Register definition 332
11.5.1. Global control register (CLA_GCTL) 332
11.5.2. Interrupt enable register (CLA_INTE) 332
11.5.3. Interrupt flag register (CLA_INTF) 334
11.5.4. Status register (CLA_STAT) 335
11.5.5. Signal selection register (CLAx_SIGS) (x = 0...3) 336
11.5.6. LCU control register (CLAx_LCUCTL) (x = 0…3) 336
11.5.7. Control register (CLAx_CTL) (x=0..3) 337
12. True random number generator (TRNG) 339
12.1. Overview 339
12.2. Characteristics 339
12.3. Limitation 339
12.4. Function overview 340
12.4.1. LFSR 341
12.4.2. Post processing 341
12.4.3. Conditioning 341
12.4.4. output FIFO 341
12.4.5. Health check 342
12.4.6. NIST mode FSM 343
12.4.7. Operation flow 343
12.4.8. Error flags 344
12.4.9. Low power usage 344
12.5. Register definition 345
12.5.1. Control register (TRNG_CTL) 345
12.5.2. Status register (TRNG_STAT) 347
12.5.3. Data register (TRNG_DATA) 348

12.5.4. Health tests configure register (TRNG_HTCFG) 348
13. Cryptographic Acceleration Unit (CAU) 350
13.1. Overview 350
13.2. Characteristics 350
13.3. CAU data type and initialization vectors 351
13.3.1. Data type 351
13.3.2. Initialization vectors 352
13.4. Cryptographic acceleration processor 352
13.4.1. DES / TDES cryptographic acceleration processor 353
13.4.2. AES cryptographic acceleration processor 357
13.5. Operating modes 364
13.6. CAU DMA interface 365
13.7. CAU interrupts 366
13.8. CAU suspended mode 366
13.9. Register definition 368
13.9.1. Control register (CAU_CTL) 368
13.9.2. Status register 0 (CAU_STAT0) 370
13.9.3. Data input register (CAU_DI) 370
13.9.4. Data output register (CAU_DO) 371
13.9.5. DMA enable register (CAU_DMAEN) 371
13.9.6. Interrupt enable register (CAU_INTEN) 372
13.9.7. Status register 1 (CAU_STAT1) 373
13.9.8. Interrupt flag register (CAU_INTF) 373
13.9.9. Key registers (CAU_KEY0..3(H/L)) 374
13.9.10. Initial vector registers (CAU_IV0..1(H/L)) 376
13.9.11. GCM or CCM mode context switch register x (CAU_GCMCCMCTXSx) (x=0..7) 378
13.9.12. GCM mode context switch register x (CAU_GCMCTXSx) (x=0..7) 378
14. Trigonometric Math Unit (TMU) 379
14.1. Overview 379
14.2. Characteristics 379
14.3. Block diagram 379
14.4. Function overview 380
14.4.1. Data format and configuration 380
14.4.2. Floating-point data format 381
14.4.3. Mode configuration 382
14.4.4. TMU operation pending 394
14.4.5. Zero-overhead mode 395
14.4.6. Interrupt and DMA requests 395

14.5. Registers definition 395
14.5.1. Control and status register (TMU_CS) 396
14.5.2. Input data register (TMU_IDATA) 399
14.5.3. Output data register (TMU_ODATA) 399
15. Fast Fourier Transform (FFT) 401
15.1. Overview 401
15.2. Characteristics 401
15.3. Functional description 401
15.4. Data format 402
15.5. Radix-2 FFT 403
15.6. Butterfly unit 403
15.7. IFFT mode 404
15.8. FFT SRAM 404
15.9. FFT loop mode 404
15.10. Operation guide 405
15.11. FFT interrupts 406
15.12. Register definition 407
15.12.1. Control and status register (FFT_CSR) 407
15.12.2. Real start address register(FFT_RESADDR) 409
15.12.3. Image start address register (FFT_IMSADDR) 409
15.12.4. Window start address register (FFT_WSADDR) 409
15.12.5. Output start address register (FFT_OSADDR) 410
15.12.6. Loop length register (FFT_LOOPLEN) 410
16. Debug (DBG) 412
16.1. Overview 412
16.2. JTAG / SW function overview 412
16.2.1. Switch JTAG or SW interface 412
16.2.2. Pin assignment 412
16.2.3. JTAG 413
16.2.4. Debug reset 415
16.2.5. JEDEC-106 ID code 415
16.3. Debug hold function overview 415
16.3.1. Debug support for power saving mode 415
16.3.2. Debug support for TIMER, I2C, WWDGT, FWDGT, RTC, CAN, LPTIMER and HRTIMER 416
16.4. Register definition 417
16.4.1. ID code register (DBG_ID) 417
16.4.2. Control register0 (DBG_CTL0) 417

16.4.3. Control register1 (DBG_CTL1) 418
16.4.4. Control register2 (DBG_CTL2) 420
17. Analog-to-digital converter (ADC) 422
17.1. Overview 422
17.2. Characteristics 422
17.3. Pins and internal signals 423
17.4. Function overview 424
17.4.1. Foreground calibration function 424
17.4.2. Dual clock domain architecture 425
17.4.3. ADC enable 425
17.4.4. Single-enden and differential input channels 425
17.4.5. Routine sequence 426
17.4.6. Operation modes 427
17.4.7. Conversion result threshold monitor function 430
17.4.8. Data storage mode 430
17.4.9. Sample time configuration 431
17.4.10. External trigger configuration 431
17.4.11. DMA request 431
17.4.12. Overflow detection 431
17.4.13. ADC internal channels 432
17.4.14. Battery voltage monitoring 433
17.4.15. Using HPDF to managing the conversion results 433
17.4.16. Programmable resolution (DRES) 434
17.4.17. On-chip hardware oversampling 434
17.4.18. Gain mode 435
17.4.19. ADC convertsion signal 435
17.5. ADC sync mode 436
17.5.1. Free mode 437
17.5.2. Routine parallel mode 437
17.5.3. Follow-up mode 438
17.5.4. Use DMA in ADC sync mode 439
17.6. ADC interrupts 440
17.7. Register definition 441
17.7.1. Status register (ADC_STAT) 441
17.7.2. Control register 0 (ADC_CTL0) 442
17.7.3. Control register 1 (ADC_CTL1) 444
17.7.4. Watchdog high threshold register0 (ADC_WDHT0) 447
17.7.5. Watchdog low threshold register0 (ADC_WDLT0) 447
17.7.6. Routine sequence register 0 (ADC_RSQ0) 448
17.7.7. Routine sequence register 1 (ADC_RSQ1) 448
17.7.8. Routine sequence register 2 (ADC_RSQ2) 449

17.7.9. Routine sequence register 3 (ADC_RSQ3) 449
17.7.10. Routine sequence register 4 (ADC_RSQ4) 450
17.7.11. Routine sequence register 5 (ADC_RSQ5) 451
17.7.12. Routine sequence register 6 (ADC_RSQ6) 451
17.7.13. Routine sequence register 7 (ADC_RSQ7) 452
17.7.14. Routine sequence register 8 (ADC_RSQ8) 452
17.7.15. Routine data register (ADC_RDATA) 453
17.7.16. Oversample control register (ADC_OVSAMPCTL) 453
17.7.17. Watchdog 1 Channel Selection Register (ADC_WD1SR) 455
17.7.18. Watchdog 2 Channel Selection Register (ADC_WD2SR) 455
17.7.19. Watchdog high threshold register1 (ADC_WDHT1) 456
17.7.20. Watchdog low threshold register1 (ADC_WDLT1) 456
17.7.21. Watchdog high threshold register2 (ADC_WDHT2) 457
17.7.22. Watchdog low threshold register2 (ADC_WDLT2) 457
17.7.23. Differential mode control register (ADC_DIFCTL) 458
17.7.24. Gain configure register (ADC_GAINCFG) 458
17.7.25. Summary status register (ADC_SSTAT) 459
17.7.26. Sync control register (ADC_SYNCCTL) 460
17.7.27. Sync routine data register (ADC_SYNCDATA) 462

18. Digital-to-analog converter (DAC) 464

18.1. Overview 464

18.2. Characteristics 464

18.3. Function overview 466

18.3.1. DAC enable 466
18.3.2. DAC output buffer 466
18.3.3. DAC data configuration 466
18.3.4. DAC trigger 467
18.3.5. DAC conversion 467
18.3.6. DAC noise wave 467
18.3.7. DAC sawtooth wave 468
18.3.8. DAC output voltage 470
18.3.9. DMA request 470
18.3.10. DAC concurrent conversion 470
18.3.11. DAC reset persist mode 471
18.3.12. DAC output buffer calibration 472
18.3.13. DAC modes 473
18.3.14. DAC low-power modes 475

18.4. Register definition 476

18.4.1. DACx control register 0 (DAC_CTL0) 476
18.4.2. DACx software trigger register (DAC_SWT) 479
18.4.3. DACx_OUT0 12-bit right-aligned data holding register (DAC_OUT0_R12DH) 480
18.4.4. DACx_OUT0 12-bit left-aligned data holding register (DAC_OUT0_L12DH) 480

18.4.5. DACx_OUT0 8-bit right-aligned data holding register (DAC_OUT0_R8DH) 481
18.4.6. DACx_OUT1 12-bit right-aligned data holding register (DAC_OUT1_R12DH) 481
18.4.7. DACx_OUT1 12-bit left-aligned data holding register (DAC_OUT1_L12DH) 481
18.4.8. DACx_OUT1 8-bit right-aligned data holding register (DAC_OUT1_R8DH) 482
18.4.9. DACx concurrent mode 12-bit right-aligned data holding register (DACC_R12DH) 482
18.4.10. DACx concurrent mode 12-bit left-aligned data holding register (DACC_L12DH) 483
18.4.11. DACx concurrent mode 8-bit right-aligned data holding register (DACC_R8DH) 484
18.4.12. DACx_OUT0 data output register (DAC_OUT0_DO) 484
18.4.13. DACx_OUT1 data output register (DAC_OUT1_DO) 485
18.4.14. DACx status register 0 (DAC_STAT0) 485
18.4.15. DACx calibration register (DAC_CALR) 486
18.4.16. DACx mode control register (DAC_MDCR) 487
18.4.17. DACx sample and keep sample time register 0 (DAC_SKSTR0) 488
18.4.18. DACx sample and keep sample time register 1 (DAC_SKSTR1) 488
18.4.19. DACx sample and keep keep time register (DACx_SKKTR) 489
18.4.20. DACx sample and keep refresh time register (DACx_SKRTR) 489
18.4.21. DACx_OUT0 sawtooth register (DAC_OUT0_SAW) 490
18.4.22. DACx_OUT1 sawtooth register (DAC_OUT1_SAW) 490
18.4.23. DACx sawtooth mode register (DACx_SAWMDR) 491

19. Comparator (CMP) 493

19.1. Overview 493

19.2. Characteristic 493

19.3. Function overview 493

19.3.1. CMP clock 494
19.3.2. CMP I/O configuration 494
19.3.3. CMP hysteresis 496
19.3.4. CMP register write protection 496
19.3.5. CMP output blanking 496
19.3.6. CMP voltage scaler function 497
19.3.7. CMP interrupt 497
19.3.8. CMP reset persist mode 497

19.4. Register definition 498

19.4.1. CMP status register (CMP_STAT) 498
19.4.2. CMP interrupt flag clear register (CMP_IFC) 500
19.4.3. CMP0 control/status register (CMP0_CS) 501
19.4.4. CMP1 control/status register (CMP1_CS) 503
19.4.5. CMP2 control/status register (CMP2_CS) 505
19.4.6. CMP3 control/status register (CMP3_CS) 507
19.4.7. CMP4 control/status register (CMP4_CS) 510
19.4.8. CMP5 control/status register (CMP5_CS) 512
19.4.9. CMP6 control/status register (CMP6_CS) 514
19.4.10. CMP7 control/status register (CMP7_CS) 516

20. VREF 519
20.1. Overview 519
20.2. Characteristics 519
20.3. Function overview 519
20.3.1. VREF calibration 520
20.4. Register definition 521
20.4.1. Control and status register (VREF_CS) 521
20.4.2. Calibration register (VREF_CALIB) 522
21. Watchdog timer (WDGT) 523
21.1. Free watchdog timer (FWDGT) 523
21.1.1. Overview 523
21.1.2. Characteristics 523
21.1.3. Function overview 523
21.1.4. Register definition 526
21.2. Window watchdog timer (WWDGT) 530
21.2.1. Overview 530
21.2.2. Characteristics 530
21.2.3. Function overview 530
21.2.4. Register definition 533
22. Real time clock (RTC) 535
22.1. Overview 535
22.2. Characteristics 535
22.3. Function overview 536
22.3.1. Block diagram 536
22.3.2. Clock source and prescalers 537
22.3.3. Shadow registers introduction 537
22.3.4. Configurable and field maskable alarm 537
22.3.5. Configurable periodic auto-wakeup counter 538
22.3.6. RTC initialization and configuration 538
22.3.7. Calendar reading 539
22.3.8. Resetting the RTC 541
22.3.9. RTC shift function 541
22.3.10. RTC reference clock detection 542
22.3.11. RTC smooth digital calibration 542
22.3.12. Time-stamp function 544
22.3.13. Tamper detection 545
22.3.14. Calibration clock output 546
22.3.15. Alarm output 546
22.3.16. RTC pin configuration 547

22.3.17. RTC power saving mode management 548
22.3.18. RTC interrupts 548
22.4. Register definition 549
22.4.1. Time register (RTC_TIME) 549
22.4.2. Date register (RTC_DATE) 549
22.4.3. Control register (RTC_CTL) 550
22.4.4. Status register (RTC_STAT) 553
22.4.5. Prescaler register (RTC_PSC) 555
22.4.6. Wakeup timer register (RTC_WUT) 555
22.4.7. Alarm 0 time and date register (RTC_ALRM0TD) 556
22.4.8. Alarm 1 time and date register (RTC_ALRM1TD) 557
22.4.9. Write protection key register (RTC_WPK) 558
22.4.10. Sub second register (RTC_SS) 559
22.4.11. Shift function control register (RTC_SHIFTCTL) 559
22.4.12. Time of time stamp register (RTC_TTS) 560
22.4.13. Date of time stamp register (RTC_DTS) 560
22.4.14. Sub second of time stamp register (RTC_SSTS) 561
22.4.15. High resolution frequency compensation register (RTC_HRFC) 562
22.4.16. Tamper register (RTC_TAMP) 562
22.4.17. Alarm 0 sub second register (RTC_ALRM0SS) 566
22.4.18. Alarm 1 sub second register (RTC_ALRM1SS) 567
22.4.19. Configuration register (RTC_CFG) 568
22.4.20. Backup registers (RTC_BKPx) (x=0..31) 568
23. TIMER (TIMER) 569
23.1. Advanced timer (TIMERx, x=0, 7, 19) 571
23.1.1. Overview 571
23.1.2. Characteristics 571
23.1.3. Block diagram 572
23.1.4. Function overview 572
23.1.5. Registers definition (TIMERx, x=0, 7, 19) 629
23.2. General level0 timer (TIMERx, x=1, 2, 3, 4) 706
23.2.1. Overview 706
23.2.2. Characteristics 706
23.2.3. Block diagram 706
23.2.4. Function overview 707
23.2.5. Registers definition (TIMERx, x=1,2,3,4) 734
23.3. General level3 timer (TIMERx, x=14) 774
23.3.1. Overview 774
23.3.2. Characteristics 774
23.3.3. Block diagram 774
23.3.4. Function overview 775
23.3.5. Register definition (TIMERx, x=14) 805

23.4. General level4 timer (TIMERx, x=15, 16) 843
23.4.1. Overview 843
23.4.2. Characteristics 843
23.4.3. Block diagram 843
23.4.4. Function overview 844
23.4.5. Register definition (TIMERx, x=15, 16) 862
23.5. Basic timer (TIMERx, x=5, 6) 893
23.5.1. Overview 893
23.5.2. Characteristics 893
23.5.3. Block diagram 893
23.5.4. Function overview 893
23.5.5. Registers definition (TIMERx, x=5,6) 899
24. Low power timer (LPTIMER) 905
24.1. Overview 905
24.2. Characteristics 905
24.3. Block diagram 906
24.4. Function overview 906
24.4.1. Clock selection 906
24.4.2. LPTIMER enable 908
24.4.3. Prescaler 908
24.4.4. Input filter 909
24.4.5. External inputs high level counter 909
24.4.6. Start counting mode 910
24.4.7. External trigger mapping 911
24.4.8. Counter operating mode 911
24.4.9. Counter Reset 913
24.4.10. Output Mode 914
24.4.11. Timeout mode 915
24.4.12. Decoder mode 916
24.4.13. Register update operation 921
24.4.14. Low-power modes 921
24.4.15. Interrupts 922
24.4.16. LPTIMER debug mode 923
24.5. Register definition 924
24.5.1. Interrupt flag register (LPTIMER_INTF) 924
24.5.2. Interrupt flag clear register (LPTIMER_INTC) 926
24.5.3. Interrupt enable register (LPTIMER_INTEN) 927
24.5.4. Control register 0 (LPTIMER_CTL0) 929
24.5.5. Control register 1 (LPTIMER_CTL1) 933
24.5.6. Compare value register (LPTIMER_CMPV) 934
24.5.7. Counter auto reload register (LPTIMER_CAR) 935

24.5.8. Counter register (LPTIMER_CNT) 935
24.5.9. External input remap register (LPTIMER_EIRMP) 936
24.5.10. Input high level counter max value register (LPTIMER_INHLCMV) 937
25. High-Resolution Timer (HRTIMER) 938
25.1. Overview 938
25.2. Characteristics 938
25.3. Block diagram 938
25.4. Function overview 939
25.4.1. Master_TIMER unit 939
25.4.2. Slave_TIMERx(x=0..7) unit 947
25.4.3. DLL calibrate 987
25.4.4. Bunch mode 988
25.4.5. Synchronization input/output 992
25.4.6. External event 994
25.4.7. Fault input 995
25.4.8. Trigger to ADC 1000
25.4.9. Trigger to DAC 1003
25.4.10. Double source trigger 1004
25.4.11. Interrupt 1006
25.4.12. DMA request 1007
25.4.13. DMA mode 1008
25.4.14. Debug mode 1009
25.5. Register definition 1010
25.5.1. Master_TIMER registers 1010
25.5.2. Slave_TIMERx registers(x=0..7) 1022
25.5.3. Common registers 1137
26. Infrared interface (IFRP) 1209
26.1. Overview 1209
26.2. Characteristics 1209
26.3. Function overview 1209
27. Universal synchronous / asynchronous receiver / transmitter (USART) ... 1211
27.1. Overview 1211
27.2. Characteristics 1211
27.3. Function overview 1213
27.3.1. USART frame format 1213
27.3.2. Baud rate generation 1214
27.3.3. USART transmitter 1215
27.3.4. USART receiver 1216

27.3.5. Use DMA for data buffer access 1217
27.3.6. Hardware flow control 1219
27.3.7. Multi-processor communication 1220
27.3.8. LIN mode 1221
27.3.9. Synchronous mode 1222
27.3.10. IrDA SIR ENDEC mode 1223
27.3.11. Half-duplex communication mode 1225
27.3.12. Smartcard (ISO7816-3) mode 1225
27.3.13. ModBus communication 1227
27.3.14. Receive / Transmitter FIFO 1227
27.3.15. Wakeup from deep-sleep mode 1228
27.3.16. USART interrupts 1229
27.4. Register definition 1231
27.4.1. Control register 0 (USART_CTL0) 1231
27.4.2. Control register 1 (USART_CTL1) 1234
27.4.3. Control register 2 (USART_CTL2) 1236
27.4.4. Baud rate generator register (USART_BAUD) 1240
27.4.5. Prescaler and guard time configuration register (USART_GP) 1240
27.4.6. Receiver timeout register (USART_RT) 1241
27.4.7. Command register (USART_CMD) 1242
27.4.8. Status register (USART_STAT) 1243
27.4.9. Interrupt status clear register (USART_INTC) 1248
27.4.10. Receive data register (USART_RDATA) 1249
27.4.11. Transmit data register (USART_TDATA) 1250
27.4.12. USART coherence control register (USART_CHC) 1250
27.4.13. USART FIFO control and status register (USART_RFCS) 1251
28. Inter-integrated circuit interface (I2C) 1255
28.1. Overview 1255
28.2. Characteristics 1255
28.3. Function overview 1255
28.3.1. Clock requirements 1256
28.3.2. I2C communication flow 1257
28.3.3. Noise filter 1260
28.3.4. I2C timings configuration 1260
28.3.5. I2C reset 1262
28.3.6. Data transfer 1262
28.3.7. I2C slave mode 1264
28.3.8. I2C master mode 1269
28.3.9. SMBus support 1274
28.3.10. SMBus mode 1277
28.3.11. Wakeup from power saving mode 1278
28.3.12. Use DMA for data transfer 1279

28.3.13. I2C error and interrupts 1279
28.3.14. I2C debug mode 1280
28.4. Register definition 1281
28.4.1. Control register 0 (I2C_CTL0) 1281
28.4.2. Control register 1 (I2C_CTL1) 1283
28.4.3. Slave address register 0 (I2C_SADDR0) 1285
28.4.4. Slave address register 1 (I2C_SADDR1) 1286
28.4.5. Timing register (I2C_TIMING) 1287
28.4.6. Timeout register (I2C_TIMEOUT) 1288
28.4.7. Status register (I2C_STAT) 1289
28.4.8. Status clear register (I2C_STATC) 1292
28.4.9. PEC register (I2C_PEC) 1293
28.4.10. Receive data register (I2C_RDATA) 1293
28.4.11. Transmit data register (I2C_TDATA) 1293
28.4.12. Control register 2 (I2C_CTL2) 1294
29. Controller area network (CAN) 1295
29.1. Overview 1295
29.2. Characteristics 1295
29.3. Function overview 1296
29.3.1. Mailbox descriptor 1297
29.3.2. Rx FIFO descriptor 1302
29.3.3. Communication modes 1308
29.3.4. Power saving modes 1309
29.3.5. Data transmission 1310
29.3.6. Data reception 1314
29.3.7. Data reception in Pretended Networking mode 1322
29.3.8. CAN FD operation 1324
29.3.9. Errors and states 1327
29.3.10. Communication parameters 1330
29.3.11. Interrupts 1333
29.4. Example for a typical configuration flow of CAN 1333
29.5. CAN registers 1336
29.5.1. Control register 0 (CAN_CTL0) 1336
29.5.2. Control register 1 (CAN_CTL1) 1338
29.5.3. Timer register (CAN_TIMER) 1340
29.5.4. Receive mailbox public filter register (CAN_RMPUBF) 1340
29.5.5. Error register 0 (CAN_ERR0) 1341
29.5.6. Error register 1 (CAN_ERR1) 1342
29.5.7. Interrupt enable register (CAN_INTEN) 1345
29.5.8. Status register (CAN_STAT) 1345
29.5.9. Control register 2 (CAN_CTL2) 1347

29.5.10. CRC for classical frame register (CAN_CRCC) 1349
29.5.11. Receive FIFO public filter register (CAN_RFIFOPUBF) 1349
29.5.12. Receive FIFO identifier filter matching number register (CAN_RFIFOIFMN) 1350
29.5.13. Bit timing register (CAN_BT) 1350
29.5.14. Receive FIFO/mailbox private filter x register (CAN_RFIFOMPFx)(x=0..31) 1351
29.5.15. Pretended Networking mode control register 0 (CAN_PN_CTL0) 1352
29.5.16. Pretended Networking mode timeout register (CAN_PN_TO) 1353
29.5.17. Pretended Networking mode status register (CAN_PN_STAT) 1353
29.5.18. Pretended Networking mode expected identifier 0 register (CAN_PN_EID0) 1354
29.5.19. Pretended Networking mode expected DLC register (CAN_PN_EDLC) 1355
29.5.20. Pretended Networking mode expected data low 0 register (CAN_PN_EDL0) 1356
29.5.21. Pretended Networking mode expected data low 1 register (CAN_PN_EDL1) 1356
29.5.22. Pretended Networking mode identifier filter / expected identifier 1 register (CAN_PN_IFEID1) 1357
29.5.23. Pretended Networking mode data 0 filter / expected data high 0 register (CAN_PN_DF0EDH0) 1358
29.5.24. Pretended Networking mode data 1 filter / expected data high 1 register (CAN_PN_DF1EDH1) 1359
29.5.25. Pretended Networking mode received wakeup mailbox x control status information register (CAN_PN_RWMxCS)(x=0..3) 1359
29.5.26. Pretended Networking mode received wakeup mailbox x identifier register (CAN_PN_RWMxI)(x=0..3) 1360
29.5.27. Pretended Networking mode received wakeup mailbox x data 0 register (CAN_PN_RWMxD0)(x=0..3) 1361
29.5.28. Pretended Networking mode received wakeup mailbox x data 1 register (CAN_PN_RWMxD1)(x=0..3) 1361
29.5.29. FD control register (CAN_FDCTL) 1362
29.5.30. FD bit timing register (CAN_FDBT) 1363
29.5.31. CRC for classical and FD frame register (CAN_CRCCFD) 1364

30. Serial peripheral interface (SPI) 1365

30.1. Overview 1365

30.2. Characteristics 1365
30.2.1. SPI characteristics 1365

30.3. SPI function overview 1366
30.3.1. SPI block diagram 1366
30.3.2. SPI signal description 1366
30.3.3. SPI clock timing and data format 1367
30.3.4. Separate transmission and reception FIFO 1369
30.3.5. NSS function 1370
30.3.6. SPI operation modes 1371
30.3.7. DMA function 1380
30.3.8. CRC function 1380

30.3.9. SPI interrupts 1381
30.4. Register definition 1383
30.4.1. Control register 0 (SPI_CTL0) 1383
30.4.2. Control register 1 (SPI_CTL1) 1385
30.4.3. Status register (SPI_STAT) 1387
30.4.4. Data register (SPI_DATA) 1388
30.4.5. CRC polynomial register (SPI_CRCPOLY) 1389
30.4.6. Receive CRC register (SPI_RCRC) 1389
30.4.7. Transmit CRC register (SPI_TCRC) 1390
30.4.8. Quad-SPI mode control register (SPI_QCTL) of SPI0 1390
31. Quad-SPI interface (QSPI) 1392
31.1. Overview 1392
31.2. Characteristics 1392
31.3. Function overview 1392
31.3.1. QSPI block diagram 1392
31.3.2. QSPI command format 1393
31.3.3. QSPI signal line modes 1395
31.3.4. QSPI DDR mode 1396
31.3.5. DQS signal 1396
31.3.6. CSN and SCK 1396
31.4. Operating modes 1397
31.4.1. Normal mode 1397
31.4.2. Read polling mode 1399
31.4.3. Memory map mode 1399
31.5. QSPI configuration 1400
31.5.1. Flash configuration 1400
31.5.2. IP configuration 1400
31.6. Send instruction only once 1400
31.7. Error and interrupts 1400
31.8. QSPI register definition 1402
31.8.1. Control register (QSPI_CTL) 1402
31.8.2. Device configuration register (QSPI_DCFG) 1404
31.8.3. Status register (QSPI_STAT) 1406
31.8.4. Status clear register (QSPI_STATC) 1407
31.8.5. Data length register (QSPI_DTLEN) 1408
31.8.6. Transfer configuration register (QSPI_TCFG) 1408
31.8.7. Address register (QSPI_ADDR) 1410
31.8.8. Alternate bytes register (QSPI_ALTE) 1411
31.8.9. Data register (QSPI_DATA) 1411
31.8.10. Status mask register (QSPI_STATMK) 1412

31.8.11. Status match register (QSPI_STATMATCH) 1412
31.8.12. Interval register (QSPI_INTERVAL) 1413
31.8.13. Timeout register (QSPI_TMOUT) 1413
31.8.14. FIFO flush register (QSPI_FLUSH) 1414
32. Clock phase delay module (CPDM) 1415
32.1. Overview 1415
32.2. Characteristics 1415
32.3. Function overview 1415
32.3.1. Overview 1416
32.3.2. Operation process 1417
32.4. Register definition 1419
32.4.1. Control register (CPDM_CTL) 1419
32.4.2. Configuration register (CPDM_CFG) 1419
33. External memory controller (EXMC) 1421
33.1. Overview 1421
33.2. Characteristics 1421
33.3. Function overview 1421
33.3.1. Block diagram 1421
33.3.2. Basic regulation of EXMC access 1422
33.3.3. External device address mapping 1423
33.3.4. NOR/PSRAM controller 1424
33.4. Registers definition 1447
33.4.1. NOR/PSRAM controller registers 1447
34. High-Performance Digital Filter (HPDF) 1454
34.1. Overview 1454
34.2. Characteristics 1454
34.3. Function overview 1455
34.3.1. HPDF Block Diagram 1455
34.3.2. HPDF on-off control 1456
34.3.3. HPDF clock 1456
34.3.4. Multiplex serial data channel 1457
34.3.5. Parallel data input 1465
34.3.6. Regular group conversion 1467
34.3.7. Inserted group conversion 1468
34.3.8. Digital filter 1470
34.3.9. Integrator 1471
34.3.10. Threshold monitor 1471
34.3.11. Malfunction monitor 1474

34.3.12. Extremes monitor 1474
34.3.13. Data unit 1474
34.3.14. HPDF interrupt 1476
34.4. Register definition 1478
34.4.1. HPDF channel x registers (x=0…7) 1478
34.4.2. HPDF filter y registers (y=0…3) 1483
35. Filter arithmetic accelerator (FAC) 1498
35.1. Overview 1498
35.2. Characteristics 1498
35.3. Function overview 1498
35.3.1. General description 1498
35.3.2. Local memory and buffers 1499
35.3.3. Input buffers 1500
35.3.4. Output buffer 1502
35.3.5. Initialization functions 1503
35.3.6. Filter functions 1504
35.3.7. Fixed point data format 1506
35.3.8. Float point data format 1507
35.3.9. FIR filters 1507
35.3.10. IIR filters 1509
35.4. Register definition 1511
35.4.1. FAC X0 buffer configure register (FAC_X0BCFG) 1511
35.4.2. FAC X1 buffer configure register (FAC_X1BCFG) 1511
35.4.3. FAC Y buffer configure register (FAC_YBCFG) 1512
35.4.4. FAC Parameter configure register (FAC_PARACFG) 1513
35.4.5. FAC Control register (FAC_CTL) 1513
35.4.6. FAC Status register (FAC_STAT) 1515
35.4.7. FAC write data register (FAC_WDATA) 1516
35.4.8. FAC read data register (FAC_RDATA) 1517
36. Document appendix 1518
36.1. List of abbreviations used in registers 1518
36.2. List of terms 1518
36.3. Available peripherals 1519
37. Revision history 1520

List of Figures

Figure 1-1. The structure of the Cortex®-M33 processor 41
Figure 1-2. Series system architecture of GD32G553 series 43
Figure 1-3. ECC decoder 49
Figure 2-1. Process of page erase operation 98
Figure 2-2. Process of mass erase operation 99
Figure 2-3. Process of double-word program operation 101
Figure 3-1. Power supply overview 131
Figure 3-2. 1.1V domain supply overview 132
Figure 3-3. Waveform of the Backup domain voltage thresholds 134
Figure 3-4. Temperature thresholds 135
Figure 3-5. Waveform of the POR/PDR 136
Figure 3-6. Waveform of the BOR 137
Figure 3-7. Waveform of the LVD threshold 138
Figure 3-8. Waveform of the VAVD threshold 138
Figure 3-9. waveform of VOVD 140
Figure 3-10. waveform of VUVD 141
Figure 4-1. The system reset circuit 153
Figure 4-2. Clock tree 154
Figure 4-3. HXTAL clock source 156
Figure 4-4. HXTAL clock source in bypass mode 157
Figure 5-1. Block diagram of EXTI 212
Figure 6-1. TRIGSEL main composition example 221
Figure 7-1. Basic structure of a standard I/O port bit 267
Figure 7-2. Input configuration 269
Figure 7-3. Output configuration 269
Figure 7-4. Analog configuration 270
Figure 7-5. Alternate function configuration 271
Figure 7-6. Filtering using the sampling window 272
Figure 7-7. Input filtering clock cycle 274
Figure 8-1. Block diagram of DMA 291
Figure 8-2. Handshake mechanism 293
Figure 8-3. DMA interrupt logic 295
Figure 9-1. Block diagram of DMAMUX 304
Figure 9-2. Synchronization mode 306
Figure 9-3. Event generation 307
Figure 10-1. Block diagram of CRC calculation unit 322
Figure 11-1. Block diagram of CLA 327
Figure 11-2. CLA interrupt logic 331
Figure 12-1. TRNG block diagram 340
Figure 13-1. DATAM No swapping and Half-word swapping 351

Figure 13-2. DATAM Byte swapping and Bit swapping 352
Figure 13-3. CAU diagram 353
Figure 13-4. DES/TDES ECB encryption 354
Figure 13-5. DES/TDES ECB decryption 355
Figure 13-6. DES/TDES CBC encryption 356
Figure 13-7. DES/TDES CBC decryption 357
Figure 13-8. AES ECB encryption 358
Figure 13-9. AES ECB decryption 358
Figure 13-10. AES CBC encryption 359
Figure 13-11. AES CBC decryption 360
Figure 13-12. Counter block structure 360
Figure 13-13. AES CTR encryption/decryption 361
Figure 14-1. TMU block diagram 379
Figure 15-1. FFT module block diagram 401
Figure 15-2 8 point DIT FFT flow digram 403
Figure 15-3. 8 points DIT FFT flow digram 404
Figure 16-1. Block diagram of JTAG unit 413
Figure 17-1. ADC module block diagram 424
Figure 17-2. Single operation mode 427
Figure 17-3. Continuous operation mode 427
Figure 17-4. Scan operation mode, continuous disable 428
Figure 17-5. Scan operation mode, continuous enable 429
Figure 17-6. Discontinuous operation mode 429
Figure 17-7. 12-bit Data storage mode 430
Figure 17-8. 6-bit data storage mode 431
Figure 17-9. Schematic diagram of handshake signal between HFDF and ADC module 434
Figure 17-10. Numerical example with 5-bits shift and rounding 435
Figure 17-11. ADC convertsion signal in continuous operation mode 436
Figure 17-12. ADC sync block diagram 437
Figure 17-13. Routine parallel mode on 16 channels 438
Figure 17-14. Follow-up mode on 1 channel in continuous operation mode 439
Figure 18-1. DAC block diagram 465
Figure 18-2. DAC LFSR algorithm 468
Figure 18-3. DAC triangle noise wave 468
Figure 18-4. DAC sawtooth wave (SAWDIRx = 1) 469
Figure 18-5. DAC sawtooth wave (SAWDIRx = 0) 470
Figure 18-6. DAC sample and keep mode stage diagram 474
Figure 19-1. CMP block diagram 494
Figure 19-2. CMP hysteresis 496
Figure 19-3. The CMP outputs signal blanking 497
Figure 20-1.Precision reference connection 519
Figure 21-1. Free watchdog block diagram 524
Figure 21-2. Window watchdog timer block diagram 530
Figure 21-3. Window watchdog timing diagram 531

Figure 22-1. Block diagram of RTC 536
Figure 23-1. Advanced timer block diagram 572
Figure 23-2. Normal mode, internal clock divided by 1 573
Figure 23-3. Counter timing diagram with prescaler division change from 1 to 2 574
Figure 23-4. Timing diagram of up counting mode, PSC=0 / 2 575
Figure 23-5. Timing diagram of up counting mode, change TIMERx_CAR ongoing 576
Figure 23-6. Timing diagram of down counting mode, PSC=0 / 2 577
Figure 23-7. Timing diagram of down counting mode, change TIMERx_CAR ongoing 577
Figure 23-8. Timing diagram of center-aligned counting mode 579
Figure 23-9. Repetition counter timing diagram of center-aligned counting mode 580
Figure 23-10. Repetition counter timing diagram of up counting mode 580
Figure 23-11. Repetition counter timing diagram of down counting mode 581
Figure 23-12. Input capture logic for channel 0 582
Figure 23-13. Input capture logic for multi mode channel 0 582
Figure 23-14. Output compare logic (when MCHxMSEL = 2'b00, x=0, 1, 2, 3) 584
Figure 23-15. Output compare logic (when MCHxMSEL = 2'b11, x=0,1,2,3) 584
Figure 23-16. Output-compare in three modes 586
Figure 23-17. Timing diagram of EAPWM 587
Figure 23-18. Timing diagram of CAPWM 587
Figure 23-19. Adjustment mode: Data format and the register bit-field 588
Figure 23-20. PWM adjustment mode schematic diagram 589
Figure 23-21. Channel x output PWM with (CHxVAL < CHxCOMVAL_ADD) 592
Figure 23-22. Channel x output PWM with (CHxVAL = CHxCOMVAL_ADD) 592
Figure 23-23. Channel x output PWM with (CHxVAL > CHxCOMVAL_ADD) 592
Figure 23-24. Channel x output PWM with CHxVAL or CHxCOMVAL_ADD exceeds CARL 593
Figure 23-25. Channel x output PWM duty cycle changing with CHxCOMVAL_ADD 593
Figure 23-26. Four Channels outputs in Composite PWM mode 594
Figure 23-27. CHx_O output with a pulse in edge-aligned mode (CHxOMPSEL≠2'b00) 595
Figure 23-28. CHx_O output with a pulse in center-aligned mode (CHxOMPSEL≠2'b00) 595
Figure 23-29. Complementary output with dead time insertion 599
Figure 23-30. Complementary output with different dead time(DTDIFEN=1) 600
Figure 23-31. BREAK0 function logic diagram 601
Figure 23-32. BREAK1 function logic diagram 601
Figure 23-33. Output behavior of the channel in response to BREAK0 (the break input high active and IOS=1) 601
Figure 23-34. Output behavior of the channel outputs with the BREAK0 and BREAK1 603
Figure 23-35. BRKINx (x=0...2) pins logic with BREAK0 function 604
Figure 23-36. Example of counter operation in decoder interface mode 606
Figure 23-37. Example of decoder interface mode with CI0FE0 polarity inverted 606
Figure 23-38. Quadrature decoder signal disconnection detection block diagram 607
Figure 23-39. Example of counter operation in decoder mode 0 / 1 with CH1P=0 608
Figure 23-40. Example of counter operation in decoder mode 2 / 3 (CH0P / CH1P=0) 609
Figure 23-41. Three types of index signals 610

Figure 23-42. Counter with the index signal the same as A (INDP[1:0]=2'b11) 611
Figure 23-43. The relationship between the index signal and counter reset events 612
Figure 23-44. The counter reset events with the FINDRST bit 612
Figure 23-45. The Index error detection in up counting mode 613
Figure 23-46. Hall sensor is used for BLDC motor 614
Figure 23-47. Hall sensor timing between two timers 615
Figure 23-48. Restart mode 616
Figure 23-49. Pause mode 617
Figure 23-50. Event mode 617
Figure 23-51. Single pulse mode TIMERx_CHxCV=0x04, TIMERx_CAR=0x60 618
Figure 23-52. delayable single pulse mode with TIMERx_CHxCV=0x00, TIMERx_CAR=0x60 619
Figure 23-53. Programmable pulse output circuity 620
Figure 23-54. Programmable pulse output circuity waveform 620
Figure 23-55. CH2_O and CH3_O output the pulse at the same time 621
Figure 23-56. Trigger mode of TIMER0 controlled by enable signal of TIMER2 622
Figure 23-57. Trigger mode of TIMER0 controlled by update signal of TIMER2 622
Figure 23-58. Pause mode of TIMER0 controlled by enable signal of TIMER2 623
Figure 23-59. Pause mode of TIMER0 controlled by O0CPRE signal of TIMER2 624
Figure 23-60. Trigger TIMER0 and TIMER2 by the CI0 signal of TIMER2 625
Figure 23-61. Configurable phase method diagram 626
Figure 23-62. Phase shift diagram for three timers 626
Figure 23-63. Direction of the counter after the reset in center-aligned counting mode 627
Figure 23-64. General Level 0 timer block diagram 707
Figure 23-65. Normal mode, internal clock divided by 1 708
Figure 23-66. Counter timing diagram with prescaler division change from 1 to 2 709
Figure 23-67. Timing chart of up counting mode, PSC=0 / 2 710
Figure 23-68. Timing chart of up counting, change TIMERx_CAR ongoing 710
Figure 23-69. Timing chart of down counting mode, PSC=0 / 2 711
Figure 23-70. Timing chart of down counting mode, change TIMERx_CAR ongoing 712
Figure 23-71. Timing chart of center-aligned counting mode 713
Figure 23-72. Input capture logic 714
Figure 23-73. Output compare logic (x=0,1,2,3) 715
Figure 23-74. Output-compare under three modes 716
Figure 23-75. Timing chart of EAPWM 717
Figure 23-76. Timing chart of CAPWM 717
Figure 23-77. Adjustment mode: Data format and the register bit-field 718
Figure 23-78. PWM adjustment mode schematic diagram 719
Figure 23-79. Channel x output PWM with (CHxVAL < CHxCOMVAL_ADD) 722
Figure 23-80. Channel x output PWM with (CHxVAL = CHxCOMVAL_ADD) 722
Figure 23-81. Channel x output PWM with (CHxVAL > CHxCOMVAL_ADD) 723
Figure 23-82. Channel x output PWM with CHxVAL or CHxCOMVAL_ADD exceeds CARL 723
Figure 23-83. Channel x output PWM duty cycle changing with CHxCOMVAL_ADD 724

Figure 23-84. Four Channels outputs in Composite PWM mode 724
Figure 23-85. CHx_O output with a pulse in edge-aligned mode (CHxOMPSEL≠2'b00) 725
Figure 23-86. CHx_O output with a pulse in center-aligned mode (CHxOMPSEL≠2'b00) 726
Figure 23-87. Restart mode 729
Figure 23-88. Pause mode 729
Figure 23-89. Event mode 729
Figure 23-90. Single pulse mode TIMERx_CHxCV = 0x04, TIMERx_CAR=0x60 731
Figure 23-91. delayable single pulse mode TIMERx_CHxCV=0x00, TIMERx_CAR=0x60 732
Figure 23-92. General level3 timer block diagram 775
Figure 23-93. Normal mode, internal clock divided by 1 776
Figure 23-94. Counter timing diagram with prescaler division change from 1 to 2 777
Figure 23-95. Timing diagram of up counting mode, PSC=0 / 2 778
Figure 23-96. Timing diagram of up counting mode, change TIMERx_CAR on the go 779
Figure 23-97. Repetition timechart for up-counter 780
Figure 23-98. Input capture logic for channel 0 781
Figure 23-99. Input capture logic for multi mode channel 0 781
Figure 23-100. Output compare logic (when MCHxMSEL = 2'b00, x=0) 782
Figure 23-101. Output compare logic (when MCHxMSEL = 2'b11, x=0) 783
Figure 23-102. Output compare logic (x=1) 783
Figure 23-103. Output-compare in three modes 785
Figure 23-104. PWM mode timechart 786
Figure 23-105. Adjustment mode: Data format and the register bit-field 786
Figure 23-106. PWM adjustment mode schematic diagram 787
Figure 23-107. Channel x output PWM with (CHxVAL < CHxCOMVAL_ADD) 789
Figure 23-108. Channel x output PWM with (CHxVAL = CHxCOMVAL_ADD) 790
Figure 23-109. Channel x output PWM with (CHxVAL > CHxCOMVAL_ADD) 790
Figure 23-110. Channel x output PWM with CHxVAL or CHxCOMVAL_ADD exceeds CARL 790
Figure 23-111. Channel x output PWM duty cycle changing with CHxCOMVAL_ADD 791
Figure 23-112. CHx_O output with a pulse in edge-aligned mode (CHxOMPSEL =2'b00) 792
Figure 23-113. Complementary output with dead-time insertion 795
Figure 23-114. Complementary output with different dead time(DTDIFEN=1) 796
Figure 23-115. BREAK0 function logic diagram 797
Figure 23-116. Output behavior of the channel in response to BREAK0 (the break input high active and IOS=1) 797
Figure 23-117. BRKIN0 pin logic with BREAK0 function 799
Figure 23-118. Restart mode 800
Figure 23-119. Pause mode 800
Figure 23-120. Event mode 801
Figure 23-121. Single pulse mode TIMERx_CHxCV = 0x04 TIMERx_CAR=0x60 802
Figure 23-122. delayable single pulse mode TIMERx_CHxCV=0x00, TIMERx_CAR=0x60 803
Figure 23-123. General level4 timer block diagram 844
Figure 23-124. Normal mode, internal clock divided by 1 845
Figure 23-125. Counter timing diagram with prescaler division change from 1 to 2 845

Figure 23-126. Timing diagram of up counting mode, PSC=0/2 846
Figure 23-127. Timing diagram of up counting mode, change TIMERx_CAR on the go 847
Figure 23-128. Repetition timechart for up-counter 848
Figure 23-129. Input capture logic for channel 0 849
Figure 23-130. Input capture logic for multi mode channel 0 849
Figure 23-131. Output compare logic (when MCHxMSEL = 2'b00, x=0) 850
Figure 23-132. Output compare logic (when MCHxMSEL = 2'b11, x=0) 851
Figure 23-133. Output-compare in three modes 852
Figure 23-134. PWM mode timechart 853
Figure 23-135. Complementary output with dead-time insertion 856
Figure 23-136. Complementary output with different dead time(DTDIFEN=1) 857
Figure 23-137. BREAK0 function logic diagram 858
Figure 23-138. Output behavior of the channel in response to BREAK0 (the break input high active and IOS=1) 858
Figure 23-139. BRKIN0 pin logic with BREAK0 function 860
Figure 23-140. Single pulse mode TIMERx_CHxCV = 0x04 TIMERx_CAR=0x60 860
Figure 23-141. Basic timer block diagram 893
Figure 23-142. Normal mode, internal clock divided by 1 894
Figure 23-143. Counter timing diagram with prescaler division change from 1 to 2 894
Figure 23-144. Timing chart of up counting mode, PSC=0/1 895
Figure 23-145. Timing chart of up counting mode, change TIMERx_CAR ongoing 896
Figure 23-146. Adjustment mode: Data format and the register bit-field 896
Figure 23-147. Adjustment mode schematic diagram 897
Figure 24-1. LPTIMER block diagram 906
Figure 24-2. LPTIMER clock source selection 907
Figure 24-3. Internal clock mode1 (CKSSEL = 0 and CNTMEN = 1 and PSC[2:0] = 000) 908
Figure 24-4. Input filter timing diagram (ECKFLT=2'b01) 909
Figure 24-5. External inputs high level counter 910
Figure 24-6. LPTIMER output with SMST = 1(16-bit) 912
Figure 24-7. LPTIMER output with OMSEL = 1(16-bit) 912
Figure 24-8. LPTIMER output with CTNMST = 1(16-bit) 913
Figure 24-9. LPTIMER_O output mode with OPSEL bit(16-bit) 915
Figure 24-10. LPTIMER timeout mode(16-bit) 916
Figure 24-11. Counter operation in decoder mode 0 with rising-edge-mode 917
Figure 24-12. Counter operation in decoder mode 0 with falling-edge-mode 918
Figure 24-13. Counter operation in decoder mode 1 with non-inverted 919
Figure 24-14. Counter operation in decoder mode 1 with non-inverted(IN1EIF) 919
Figure 24-15. Counter operation in decoder mode 1 with non-inverted(IN0EIF) 920
Figure 24-16. Counter operation in decoder mode 1 with non-inverted(INRFOEIF) 920
Figure 24-17. Counter operation in decoder mode 1 with non-inverted(INHLOEIF) 920
Figure 25-1. HRTIMER block diagram 939
Figure 25-2. Master_TIMER diagram 940
Figure 25-3. Counter clock when divided by 16 941
Figure 25-4. Counter behavior in single pulse mode 942

Figure 25-5. Counter behavior in continuous mode 942
Figure 25-6. Repetition counter behavior in continuous mode 943
Figure 25-7. Repetition counter behavior in single pulse mode with CNTRSTM = 0 943
Figure 25-8. Repetition counter behavior in single pulse mode with CNTRSTM = 1 944
Figure 25-9 Reset event resynchronization when prescaling ratio is 64 944
Figure 25-10. Slave_TIMERx diagram 948
Figure 25-11. Capture 0 triggered by EXEV0 and EXEV1 952
Figure 25-12. Compare 1 behavior with STxCAR=0x8, STxCMP1V=0x02 952
Figure 25-13. Compare delayed mode chart 954
Figure 25-14. Compare 1 delayed mode 0 955
Figure 25-15. Compare 1 delayed mode 1 956
Figure 25-16. Compare delayed mode with SHWEN = 0 957
Figure 25-17. Variable frequency half mode 958
Figure 25-18. PWM waveform when IMPUD = 1 and IMUPD = 0 959
Figure 25-19. Channel output diagram 959
Figure 25-20. O0PRE wave: set on CMP0, reset on CMP1 960
Figure 25-21. OyPRE wave in center-aligned mode 962
Figure 25-22. Counter repetition value CREP[7:0] and ROVM[1:0] in center aligned mode 963
Figure 25-23. Arbitration mechanism during each t_{HRTMER_CK} period 965
Figure 25-24. Arbitration mechanism example 966
Figure 25-25. A pulse of 1 t_{HRTMER_CK} period 967
Figure 25-26. A pulse of 2 t_{HRTMER_CK} period 967
Figure 25-27. High-resolution OxPRE wave 967
Figure 25-28. OxPRE wave with CNTCKDIV[2:0] = 3'b110 968
Figure 25-29. C0OPRE wave in regular mode 969
Figure 25-30. C0OPRE and C1OPRE complementary wave with dead-time 970
Figure 25-31. Complementary wave with pulse width less than dead-time 971
Figure 25-32. Structure chart in balanced mode 971
Figure 25-33. C0OPRE and C1OPRE wave in balanced mode 972
Figure 25-34. ISO0 = 0 and CHOP = 0 in delayed IDLE 974
Figure 25-35. ISO0 = 1 and CHOP = 0 in delayed IDLE 975
Figure 25-36. ISO0 = 0 and CHOP = 1 in delayed IDLE 975
Figure 25-37. ISO0 = 1 and CHOP = 1 in delayed IDLE 976
Figure 25-38. Balanced IDEL with ISO0 = 0 and ISO1 = 0 977
Figure 25-39. STxCHy_O wave with CHyP=0 or CHyP=1 979
Figure 25-40. Carrier-signal structure diagram 980
Figure 25-41. HRTIMER output with carrier-signal mode enabled 981
Figure 25-42. Blanking mode and windowing mode 983
Figure 25-43. External event X counter 986
Figure 25-44. External event X counter when EXEVXCEN bit is 1 and EXEVXCNTTHR[5:0] = 2 986
Figure 25-45. Bunch mode timing chart 989
Figure 25-46. Regular entry for bunch mode 990

Figure 25-47. Delayed entry for bunch mode 991
Figure 25-48. Emulate bunch mode example 992
Figure 25-49. Extern event y(y=0..4) processed diagram 994
Figure 25-50. Extern event y(y=5..9) processed diagram 994
Figure 25-51. Fault input diagram 996
Figure 25-52. Fault counter when FLTxRST bit is 1, FLTRECCTL bit is 1 and FLTxCNT[3:0] = 0x03 998
Figure 25-53. Fault automatic resume when FLTxRST bit is 1 and FLTAR = 1 999
Figure 25-54. Trigger to ADC selection overview 1001
Figure 25-55. ADC trigger division in up counting mode 1002
Figure 25-56. ADC trigger division in center aligned mode 1003
Figure 25-57. Trigger to DAC selection overview 1004
Figure 25-58. Trigger to DAC when TRIG0M = 0 and TRIG1M = 0 1005
Figure 25-59. Trigger to DAC when TRIG0M = 1 and TRIG1M = 1 1006
Figure 25-60. DMA mode operation flowchart 1009
Figure 26-1. IFRP output timechart 1 1209
Figure 26-2. IFRP output timechart 2 1210
Figure 26-3. IFRP output timechart 3 1210
Figure 27-1. USART module block diagram 1213
Figure 27-2. USART character frame (8 bits data and 1 stop bit) 1214
Figure 27-3. USART transmit procedure 1216
Figure 27-4. Oversampling method of a receive frame bit (OSB = 0) 1217
Figure 27-5. Configuration step when using DMA for USART transmission 1218
Figure 27-6. Configuration step when using DMA for USART reception 1219
Figure 27-7. Hardware flow control between two USARTs 1219
Figure 27-8. Hardware flow control 1220
Figure 27-9. Break frame occurs during idle state 1222
Figure 27-10. Break frame occurs during a frame 1222
Figure 27-11. Example of USART in synchronous mode 1223
Figure 27-12. 8-bit format USART synchronous waveform (CLEN = 1) 1223
Figure 27-13. IrDA SIR ENDEC module 1224
Figure 27-14. IrDA data modulation 1224
Figure 27-15. ISO7816-3 frame format 1225
Figure 27-16. USART receive FIFO structure 1228
Figure 27-17. USART transmitter FIFO structure 1228
Figure 27-18. USART interrupt mapping diagram 1230
Figure 28-1. I2C module block diagram 1256
Figure 28-2. Data validation 1257
Figure 28-3. START and STOP condition 1258
Figure 28-4. I2C communication flow with 10-bit address (Master Transmit) 1258
Figure 28-5. I2C communication flow with 7-bit address (Master Transmit) 1259
Figure 28-6. I2C communication flow with 7-bit address (Master Receive) 1259
Figure 28-7. I2C communication flow with 10-bit address (Master Receive when HEAD10R=0) 1259

Figure 28-8. I2C communication flow with 10-bit address (Master Receive when HEAD10R=1) 1259
Figure 28-9. Data hold time 1260
Figure 28-10. Data setup time 1261
Figure 28-11. Data transmission 1263
Figure 28-12. Data reception 1263
Figure 28-13. I2C initialization in slave mode 1266
Figure 28-14. Programming model for slave transmitting when SS=0 1267
Figure 28-15. Programming model for slave transmitting when SS=1 1268
Figure 28-16. Programming model for slave receiving 1269
Figure 28-17. I2C initialization in master mode 1270
Figure 28-18. Programming model for master transmitting (N<=255) 1271
Figure 28-19. Programming model for master transmitting (N>255) 1272
Figure 28-20. Programming model for master receiving (N<=255) 1273
Figure 28-21. Programming model for master receiving (N>255) 1274
Figure 28-22. SMBus master transmitter and slave receiver communication flow 1278
Figure 28-23. SMBus master receiver and slave transmitter communication flow 1278
Figure 29-1. CAN module block diagram 1296
Figure 29-2. Transmitter delay 1327
Figure 29-3. CAN bit time 1331
Figure 30-1. Block diagram of SPI 1366
Figure 30-2. SPI0 timing diagram in normal mode 1368
Figure 30-3. SPI0 data frame right-aligned diagram 1368
Figure 30-4. SPI timing diagram in Quad-SPI mode (CKPL=1, CKPH=1, LF=0) 1368
Figure 30-5. Transmission and reception FIFO 1369
Figure 30-6. A typical full-duplex connection 1372
Figure 30-7. A typical simplex connection (Master: Receive, Slave: Transmit) 1373
Figure 30-8. A typical simplex connection (Master: Transmit only, Slave: Receive) 1373
Figure 30-9. A typical bidirectional connection 1373
Figure 30-10. Timing diagram of TI master mode with discontinuous transfer 1375
Figure 30-11. Timing diagram of TI master mode with continuous transfer 1375
Figure 30-12. Timing diagram of TI slave mode 1376
Figure 30-13. Timing diagram of NSS pulse with continuous transmit 1377
Figure 30-14. Timing diagram of quad write operation in Quad-SPI mode 1378
Figure 30-15. Timing diagram of quad read operation in Quad-SPI mode 1379
Figure 31-1 QSPI diagram 1393
Figure 31-2 QSPI command format 1394
Figure 31-3 QSPI DDR mode 1396
Figure 31-4 DDR read timing using DQS 1396
Figure 31-5 Behavior of CSN and SCK (CSNCKM = 0) 1397
Figure 32-1. Schematic diagram of CPDM 1416
Figure 32-2. CPDM delay line length configuration flowchart 1417
Figure 32-3. CPDM output clock phase configuration flowchart 1418
Figure 33-1. The EXMC block diagram 1422

Figure 33-2. EXMC memory banks 1423
Figure 33-3. Four regions of bank0 address mapping 1423
Figure 33-4. Mode 1 read access 1429
Figure 33-5. Mode 1 write access 1429
Figure 33-6. Mode A read access 1431
Figure 33-7. Mode A write access 1431
Figure 33-8. Mode 2/B read access 1433
Figure 33-9. Mode 2 write access 1433
Figure 33-10. Mode B write access 1433
Figure 33-11. Mode C read access 1435
Figure 33-12. Mode C write access 1435
Figure 33-13. Mode D read access 1437
Figure 33-14. Mode D write access 1437
Figure 33-15. Multiplex mode read access 1439
Figure 33-16. Multiplex mode write access 1439
Figure 33-17. Read access timing diagram under async-wait signal assertion 1441
Figure 33-18. Write access timing diagram under async-wait signal assertion 1441
Figure 33-19. Read timing of synchronous multiplexed burst mode 1443
Figure 33-20. Write timing of synchronous multiplexed burst mode 1445
Figure 34-1. HPDF block diagram 1455
Figure 34-2. The sequence diagram of SPI data transmission 1459
Figure 34-3. The sequence diagram of Manchester data transmission 1460
Figure 34-4. Manchester synchronous sequence diagram 1460
Figure 34-5. Clock loss detection timing diagram 1463
Figure 34-6. Channel pins redirection 1464
Figure 34-7. HPDF module external input data processing flow 1475
Figure 34-8. HPDF interrupt logic diagram 1477
Figure 35-1 FAC structure diagram 1499
Figure 35-2 Input buffer area 1500
Figure 35-3 Circular input buffer area 1501
Figure 35-4 Circular input buffer operation 1502
Figure 35-5 Circular output buffer 1502
Figure 35-6 Circular output buffer area 1503
Figure 35-7 The structure of FIR fliter function 1505
Figure 35-8 The structure of IIR fliter 1506

List of Tables

Table 1-1. Bus Interconnection Matrix 41
Table 1-2. Memory map of GD32G553 devices 44
Table 1-3. Boot modes 52
Table 2-1. Base address and size for 512KB-dual bank flash memory 92
Table 2-2. Base address and size for 512KB-single bank flash memory 93
Table 2-3. The relation between WSCNT and AHB clock frequency when LDO is 1.1V 95
Table 2-4. Option byte 102
Table 2-5 WP protection 109
Table 2-6 FMC interrupt requests 111
Table 3-1. Power domain summary 141
Table 4-1. Clock output source select 159
Table 4-2. Low-speed clock output source select 159
Table 5-1. NVIC exception types in Cortex®-M33 208
Table 5-2. Interrupt vector table 208
Table 6-1. Trigger input bit fields selection 221
Table 6-2. TRIGSEL input and output mapping 226
Table 7-1. GPIO configuration table 266
Table 8-1. DMA transfer operation 292
Table 8-2. interrupt events 295
Table 9-1. DMAMUX signals 305
Table 9-2. Interrupt events 309
Table 9-3. Request multiplexer input mapping 310
Table 9-4. Trigger input mapping 313
Table 9-5. Synchronization input mapping 314
Table 11-1. CLAxSIGS0 input selection 328
Table 11-2. CLAxSIGS1 input selection 328
Table 11-3. LCU control 329
Table 12-1. ALGO configurations 342
Table 14-1. Input data configuration 380
Table 14-2. Output data configuration 381
Table 14-3 IEEE754 32-bit single precision floating-point format 381
Table 14-4. TMU mode configuration 382
Table 14-5. Mode 0 description, when IFLTEN = 1 and OFLTEN = 1 383
Table 14-6. Mode 0 description, when IFLTEN = 0 or OFLTEN = 0 383
Table 14-7. Mode 1 description, when IFLTEN = 1 and OFLTEN = 1 384
Table 14-8. Mode 1 description, when IFLTEN = 0 or OFLTEN = 0 384
Table 14-9. Mode 2 description, when IFLTEN = 1 and OFLTEN = 1 386
Table 14-10. Mode 2 description, when IFLTEN = 0 or OFLTEN = 0 386
Table 14-11. Mode 3 description, when IFLTEN = 1 and OFLTEN = 1 387
Table 14-12. Mode 3 description, when IFLTEN = 0 or OFLTEN = 0 387

Table 14-13. Mode 4 description, when IFLTEN = 1 and OFLTEN = 1 388
Table 14-14. Mode 4 description, when IFLTEN = 0 or OFLTEN = 0 388
Table 14-15. Mode 5 description, when IFLTEN = 1 and OFLTEN = 1 389
Table 14-16. Mode 5 description, when IFLTEN = 0 or OFLTEN = 0 389
Table 14-17. Mode 6 description, when IFLTEN = 1 and OFLTEN = 1 390
Table 14-18. Mode 6 description, when IFLTEN = 0 or OFLTEN = 0 390
Table 14-19. Mode 7 description, when IFLTEN = 1 and OFLTEN = 1 391
Table 14-20. Mode 7 description, when IFLTEN = 0 or OFLTEN = 0 391
Table 14-21. Mode 8 description, when IFLTEN = 1 and OFLTEN = 1 392
Table 14-22. Mode 8 description, when IFLTEN = 0 or OFLTEN = 0 392
Table 14-23. Recommended scaling factors in mode 8 393
Table 14-24. Mode 9 description, when IFLTEN = 1 and OFLTEN = 1 393
Table 14-25. Mode 9 description, when IFLTEN = 0 or OFLTEN = 0 394
Table 14-26. Recommended scaling factors in mode 9 394
Table 15-1. IEEE 32-Bit Single Precision Floating-Point Format 402
Table 15-2. 8 points input sequence bits reverse operation 403
Table 16-1. Pin assignment 413
Table 16-2. OTP JTAG bytes 414
Table 17-1. ADC internal input signals 423
Table 17-2. ADC input pins definition 423
Table 17-3. ADC differential channel pin matching 425
Table 17-4. Trigger source for routine channels 431
Table 17-5. t_{CONV} timings depending on resolution for ADC 434
Table 17-6. ADC sync mode table 436
Table 18-1. DAC I/O description 465
Table 18-2. DAC triggers and outputs summary 465
Table 18-3 DAC data format (12-bit data) 466
Table 18-4. Triggers of DAC 467
Table 18-5. Reset triggers of DAC sawtooth wave generation 469
Table 18-6. Step triggers of DAC sawtooth wave generation 469
Table 18-7. Formula of sample and refresh time 474
Table 19-1. CMP inputs and outputs summary 494
Table 20-1. VREF modes 520
Table 21-1. Min/max FWDGT timeout period at 32KHz (IRC32K) 525
Table 21-2. Min-max timeout value at 216 MHz (f_{PCLK1}) 532
Table 22-1 RTC pin configuration and function 547
Table 22-2 RTC power saving mode management 548
Table 22-3 RTC interrupts control 548
Table 23-1. Timers (TIMERx) are divided into five sorts 569
Table 23-2. Advanced timer channel description 572
Table 23-3. CHxVAL and CARL bit-field change in edge-aligned 589
Table 23-4. CHxVAL and CARL bit-field changes in the center-aligned counting mode 590
Table 23-5.The Composite PWM pulse width 590
Table 23-6. Complementary outputs controlled by parameters (MCHxMSEL =2'b11) 597

Table 23-7. Output behavior of the channel in response to a BREAK0 and BREAK1 (the break input is high active) 602
Table 23-8. Break function input pins locked / released conditions 604
Table 23-9. Counting direction in different quadrature decoder signals 605
Table 23-10. the counter operation in decoder mode 1 608
Table 23-11. the counter operation in decoder mode 2 / 3 609
Table 23-12. Examples of slave mode 615
Table 23-13. CHxVAL and CARL bit-field change in edge-aligned 719
Table 23-14. CHxVAL bit-field changes in the center-aligned counting mode 720
Table 23-15.The Composite PWM pulse width 721
Table 23-16. Examples of slave mode 728
Table 23-17. CHxVAL and CARL bit-field change in edge-aligned 787
Table 23-18.The Composite PWM pulse width 788
Table 23-19. Complementary outputs controlled by parameters (MCHxMSEL =2'b11) 793
Table 23-20. Break function input pins locked / released conditions 798
Table 23-21. Slave mode example table 799
Table 23-22. Complementary outputs controlled by parameters (MCHxMSEL =2'b11) 854
Table 23-23. Break function input pins locked/ released conditions 859
Table 23-24. CARL bit-field change in edge-aligned 897
Table 24-1. Prescaler division factor 908
Table 24-2. External trigger mapping 911
Table 24-3. Counting direction versus decoder signals 917
Table 24-4. LPTIMER works in low-power modes 921
Table 24-5. LPTIMER interrupt events 922
Table 25-1. The limitations of auto-reload and compare y (y=0..3) register 940
Table 25-2. Resolution with $f_{HRTIMER_CK}$ = 216MHz 941
Table 25-3. Alternate mode selection 945
Table 25-4. Alternate mode selection 945
Table 25-5. Master_TIMER shadow registers and update event 946
Table 25-6. The limitations of counter and capture y(y=0,1) value registers 949
Table 25-7. Alternate mode selection 953
Table 25-8. Compare values for the two alternate modes 953
Table 25-9. Slave_TIMER interconnection event 961
Table 25-10. Blanking and windowing in up counting mode and center aligned mode 963
Table 25-11. Crossbar and IDLE control stage work together 973
Table 25-12. Request to enter in IDLE and exit IDLE state 973
Table 25-13. Output during IDEL state controlled by bunch mode 978
Table 25-14. Output stage status programming (x=0..7, y=0,1) 978
Table 25-15. Slave_TIMERx shadow registers and common registers and update event 982
Table 25-16. STxUPINy(y=0..2) and chip internal signal 982
Table 25-17. blanking signal remap in blanking mode 983
Table 25-18. Filtering signals mapping in windowing mode 985
Table 25-19. Chip internal signal in bunch mode 989
Table 25-20. External events mapping 995

Table 25-21. Fault channel mapping 997
Table 25-22. Fault channel blank 997
Table 25-23. Source of counter reset 999
Table 25-24. Interrupt mapping 1006
Table 25-25. DMA request mapping 1007
Table 27-1. Description of USART important pins 1213
Table 27-2. Configuration of stop bits 1214
Table 27-3. USART interrupt requests 1229
Table 28-1. Definition of I2C-bus terminology (refer to the I2C specification of Philips semiconductors) 1256
Table 28-2. Data setup time and data hold time 1262
Table 28-3. Communication modes to be shut down 1263
Table 28-4. I2C error flags 1279
Table 28-5. I2C interrupt events 1279
Table 29-1. Mailbox descriptor with 64 byte payload 1297
Table 29-2. Data bytes for DLC 1299
Table 29-3. Mailbox Rx CODE 1299
Table 29-4. Mailbox Tx CODE 1300
Table 29-5. Mailbox size 1302
Table 29-6. Rx FIFO descriptor 1303
Table 29-7. Mailbox arbitration value(32 bit) when local priority disabled 1312
Table 29-8. Mailbox arbitration value(35 bit) when local priority enabled 1313
Table 29-9. Rx mailbox matching 1319
Table 29-10. Rx FIFO matching 1320
Table 29-11. Interrupt events 1333
Table 29-12. Rx FIFO filter element number 1347
Table 30-1. SPI signal description 1366
Table 30-2. Quad-SPI signal description 1367
Table 30-3. NSS function in slave mode 1370
Table 30-4. NSS function in master mode 1371
Table 30-5. SPI operation modes 1371
Table 30-6. SPI interrupt requests 1382
Table 31-1. QSPI signal description 1392
Table 31-2. QSPI command description 1394
Table 31-3. QSPI singnal line modes 1395
Table 31-4. Correspondence between QSPI pins used and the highest supported communication clock 1395
Table 31-5. AHB write access mode and number of bytes add to FIFO 1398
Table 31-6. TERR and AHB error conditions 1401
Table 31-7. QSPI interrupt events 1401
Table 33-1. NOR Flash interface signals description 1424
Table 33-2. PSRAM non-muxed signal description 1425
Table 33-3. Supported 16-bit transactions 1425
Table 33-4. Supported 8-bit transactions 1426

Table 33-5. NOR / PSRAM controller timing parameters 1427
Table 33-6. EXMC_timing models 1428
Table 33-7. Mode 1 related registers configuration 1430
Table 33-8. Mode A related registers configuration 1431
Table 33-9. Mode 2/B related registers configuration 1434
Table 33-10. Mode C related registers configuration 1435
Table 33-11. Mode D related registers configuration 1437
Table 33-12. Multiplex mode related registers configuration 1439
Table 33-13. Timing configurations of synchronous multiplexed read mode 1443
Table 33-14. Timing configurations of synchronous multiplexed write mode 1445
Table 34-1. HPDF pins definition 1455
Table 34-2. HPDF internal signal 1456
Table 34-3. SPI interface clock configuration 1458
Table 34-4. Parallel data packed mode 1467
Table 34-5. Trigger signal of inserted group 1469
Table 34-6. The relationship between the maximum output resolution and oversampling filtering of SincX filtering 1470
Table 34-7. Relationship between the maximum output resolution and IOR, SFOR, SFO of the integrator 1471
Table 34-8. Features of threshold monitor working mode 1472
Table 34-9. Maximum output rate 1475
Table 34-10. HPDF interrupt event 1476
Table 35-1 IEEE 32-Bit Single Precision Floating-Point Format 1507
Table 36-1. List of abbreviations used in register 1518
Table 36-2. List of terms 1518
Table 37-1. Revision history 1520

1. System and memory architecture

The GD32G553 series are 32-bit general-purpose microcontrollers based on the Arm® Cortex®-M33 processor. The Arm® Cortex®-M33 processor includes two AHB buses known as Code and System buses. All memory accesses of the Arm® Cortex®-M33 processor are executed on these two buses according to the different purposes and the target memory spaces. The memory organization uses a Harvard architecture, pre-defined memory map and up to 4 GB of memory space, making the system flexible and extendable.

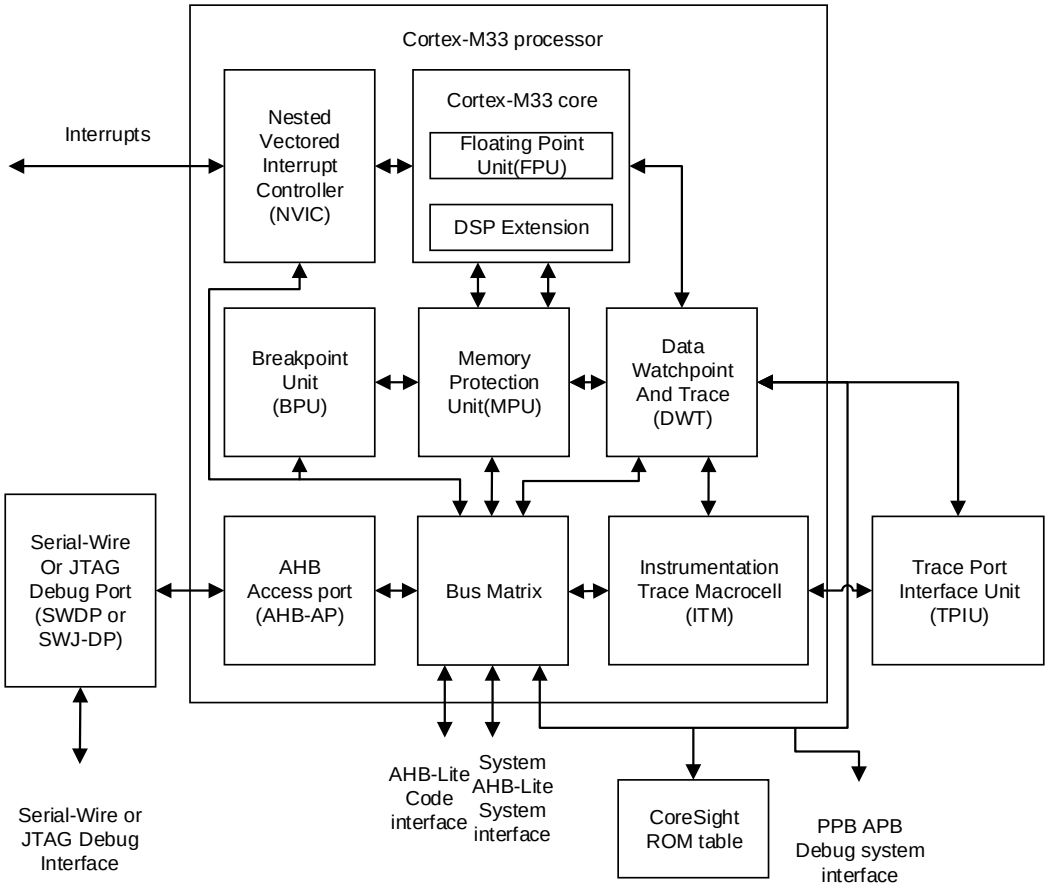
1.1. Arm® Cortex®-M33 processor

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

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

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

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



1.2. System architecture

A 32-bit multilayer bus is implemented in the GD32G553 devices, which enables parallel access paths between multiple masters and slaves in the system. The multilayer bus consists of an AHB interconnect matrix, one AHB bus and two APB buses. The interconnection relationship of the AHB interconnect matrix is shown below. In the following table, “1” indicates the corresponding master is able to access the corresponding slave through the AHB interconnect matrix, while the blank means the corresponding master cannot access the corresponding slave through the AHB interconnect matrix.

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

Table 1-1. Bus Interconnection Matrix

	CBUS	SBUS	DMA0	DMA1	FFT
FMC	1	0	1	1	1
SRAM0	1	1	1	1	1
SRAM1	0	1	1	1	1
TCMSRAM	1	1	1	1	1

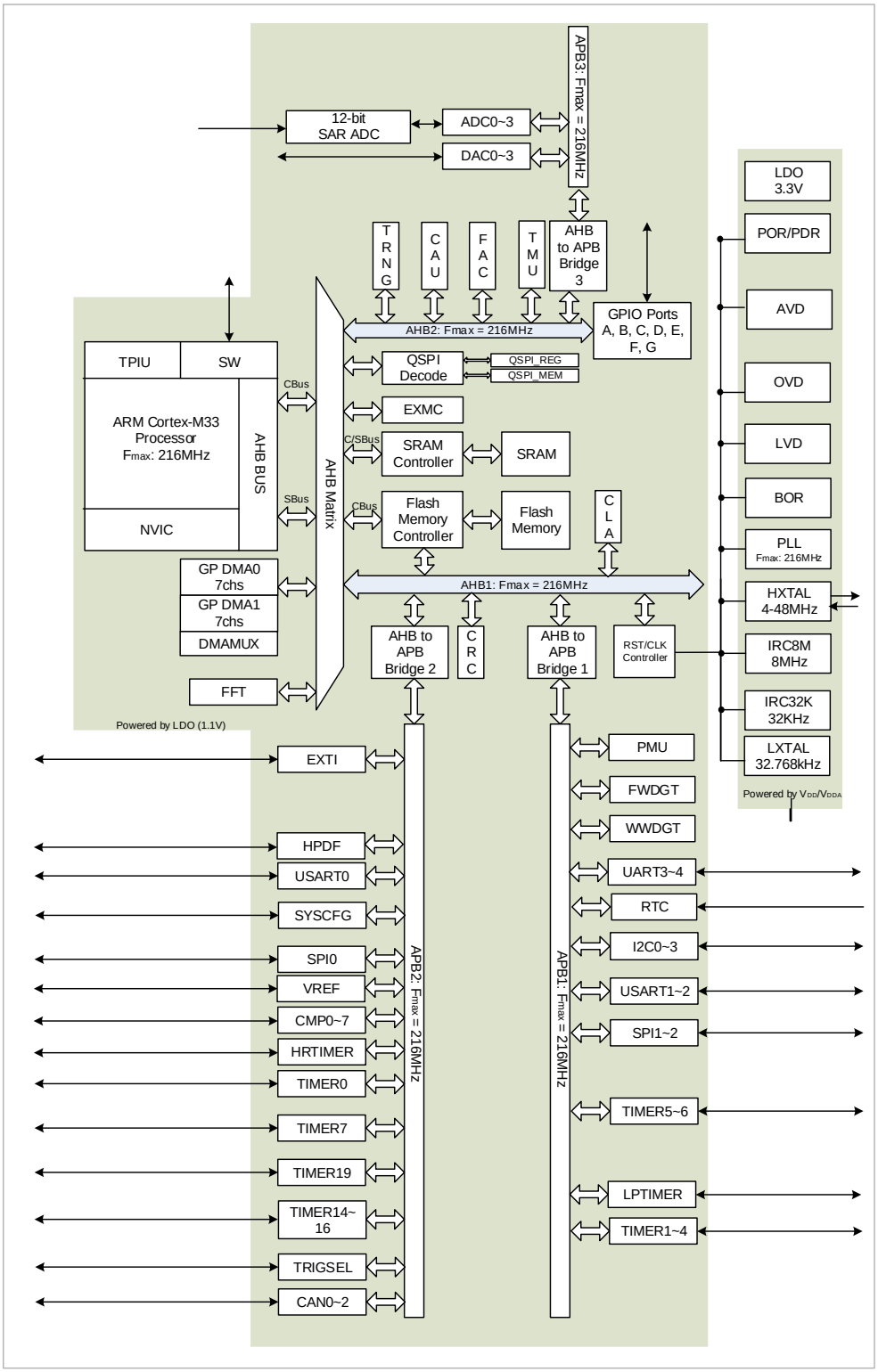
	CBUS	SBUS	DMA0	DMA1	FFT
AHB1	0	1	1	1	1
AHB2	0	1	1	1	1
EXMC	1	1	1	1	1
QSPI	1	1	1	1	1

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

There are also several slaves connected with the AHB interconnect matrix, including FMC, SRAM0, SRAM1, TCMSRAM, AHB1, AHB2, EXMC and QSPI. FMC is the bus interface of the flash memory controller. SRAM0 and SRAM1 are on-chip static random access memories. TCMSRAM is the tightly-coupled memory SRAM. AHB1 is the AHB bus connected with all of the AHB1 slaves. AHB2 is the AHB bus connected with AHB2 slaves. AHB1 peripherals including the APB1 and APB2 peripherals, the AHB2 peripherals including the APB3 peripherals and the external memories through the EXMC or the QSPI.

These are interconnected using a multilayer AHB bus architecture as shown in ***Figure 1-2. Series system architecture of GD32G553 series*** below:

Figure 1-2. Series system architecture of GD32G553 series



1.3. Memory map

The Arm® Cortex®-M33 processor is structured in Harvard architecture which can use

separate buses to fetch instructions and load/store data. Program memory, data memory, registers and I/O ports are organized within the same linear 4-Gbyte address space which is the maximum address range of the Cortex®-M33 since the bus address width is 32-bit. Additionally, a pre-defined memory map is provided by the Cortex®-M33 processor to reduce the software complexity of repeated implementation of different device vendors. In the map, some regions are used by the Arm® Cortex®-M33 system peripherals which can not be modified. However, the other regions are available to the vendors. ***Table 1-2. Memory map of GD32G553 devices*** shows the memory map of the GD32G553 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 GD32G553 devices

Pre-defined Regions	Bus	Address	Peripherals
External RAM		0xD000 0000 - 0xDFFF FFFF	Reserved
		0xC000 0000 - 0xCFFF FFFF	Reserved
		0xA000 1400 - 0xBFFF FFFF	Reserved
		0xA000 1000 - 0xA000 13FF	QSPI- REG
		0xA000 0400 - 0xA000 0FFF	EXMC - SWREG
		0xA000 0000 - 0xA000 03FF	
		0x9000 0000 - 0x9FFF FFFF	QSPI- MEM
		0x8000 0000 - 0x8FFF FFFF	Reserved
		0x7000 0000 - 0x7FFF FFFF	Reserved
		0x6000 0000 - 0x6FFF FFFF	EXMC - NOR/PSRAM/SRAM
Peripheral	AHB1	0x5001 0000 - 0x5FFF FFFF	Reserved
		0x5000 4000 - 0x5000 FFFF	Reserved
		0x5000 3000 - 0x5000 3FFF	Reserved
		0x5000 2000 - 0x5000 2FFF	Reserved
	APB3	0x5000 1C00 - 0x5000 1FFF	DAC3
		0x5000 1800 - 0x5000 1BFF	DAC2
		0x5000 1400 - 0x5000 17FF	DAC1
		0x5000 1000 - 0x5000 13FF	DAC0
		0x5000 0C00 - 0x5000 0FFF	ADC3
		0x5000 0800 - 0x5000 0BFF	ADC2
		0x5000 0400 - 0x5000 07FF	ADC1
		0x5000 0000 - 0x5000 03FF	ADC0
	AHB2	0x4802 5000 - 0x4FFF FFFF	Reserved
		0x4802 4800 - 0x4802 4FFF	FAC
		0x4802 4400 - 0x4802 47FF	TMU
		0x4802 4000 - 0x4802 43FF	Reserved
		0x4802 3000 - 0x4802 3FFF	Reserved
		0x4802 2C00 - 0x4802 2FFF	Reserved

Pre-defined Regions	Bus	Address	Peripherals
		0x4802 2800 - 0x4802 2BFF	CPDM
		0x4802 2400 - 0x4802 27FF	Reserved
		0x4802 1C00 - 0x4802 23FF	Reserved
		0x4802 1800 - 0x4802 1BFF	TRNG
		0x4802 1400 - 0x4802 17FF	Reserved
		0x4802 1000 - 0x4802 13FF	CAU
		0x4802 0400 - 0x4802 0FFF	Reserved
		0x4800 1C00 - 0x4802 03FF	Reserved
		0x4800 1800 - 0x4800 1BFF	GPIOG
		0x4800 1400 - 0x4800 17FF	GPIOF
		0x4800 1000 - 0x4800 13FF	GPIOE
		0x4800 0C00 - 0x4800 0FFF	GPIOD
		0x4800 0800 - 0x4800 0BFF	GPIOC
		0x4800 0400 - 0x4800 07FF	GPIOB
		0x4800 0000 - 0x4800 03FF	GPIOA
	AHB1	0x4003 8400 - 0x47FF FFFF	Reserved
		0x4003 8000 - 0x4003 83FF	CLA
		0x4002 7800 - 0x4003 7FFF	Reserved
		0x4002 5000 - 0x4002 77FF	FFT
		0x4002 3400 - 0x4002 4FFF	Reserved
		0x4002 3000 - 0x4002 33FF	CRC
		0x4002 2400 - 0x4002 2FFF	Reserved
		0x4002 2000 - 0x4002 23FF	FMC
		0x4002 1C00 - 0x4002 1FFF	Reserved
		0x4002 1800 - 0x4002 1BFF	Reserved
		0x4002 1400 - 0x4002 17FF	Reserved
		0x4002 1000 - 0x4002 13FF	RCU
		0x4002 0C00 - 0x4002 0FFF	Reserved
		0x4002 0800 - 0x4002 0BFF	DMAMUX
		0x4002 0400 - 0x4002 07FF	DMA1
		0x4002 0000 - 0x4002 03FF	DMA0
	APB2	0x4001 D000 - 0x4001 FFFF	Reserved
		0x4001 C000 - 0x4001 CFFF	CAN2
		0x4001 B000 - 0x4001 BFFF	CAN1
		0x4001 A000 - 0x4001 AFFF	CAN0
		0x4001 8800 - 0x4001 9FFF	Reserved
		0x4001 8400 - 0x4001 87FF	TRIGSEL
		0x4001 8000 - 0x4001 83FF	Reserved
		0x4001 7C00 - 0x4001 7FFF	CMP
		0x4001 7800 - 0x4001 7BFF	VREF

Pre-defined Regions	Bus	Address	Peripherals
		0x4001 7400 - 0x4001 77FF	Reserved
		0x4001 7000 - 0x4001 73FF	HPDF
		0x4001 6800 - 0x4001 6FFF	Reserved
		0x4001 5800 - 0x4001 67FF	HRTIMER
		0x4001 5400 - 0x4001 57FF	Reserved
		0x4001 5000 - 0x4001 53FF	TIMER19
		0x4001 4C00 - 0x4001 4FFF	Reserved
		0x4001 4800 - 0x4001 4BFF	TIMER16
		0x4001 4400 - 0x4001 47FF	TIMER15
		0x4001 4000 - 0x4001 43FF	TIMER14
		0x4001 3C00 - 0x4001 3FFF	Reserved
		0x4001 3800 - 0x4001 3BFF	USART0
		0x4001 3400 - 0x4001 37FF	TIMER7
		0x4001 3000 - 0x4001 33FF	SPI0
		0x4001 2C00 - 0x4001 2FFF	TIMER0
		0x4001 2800 - 0x4001 2BFF	Reserved
		0x4001 2400 - 0x4001 27FF	Reserved
		0x4001 2000 - 0x4001 23FF	Reserved
		0x4001 1C00 - 0x4001 1FFF	Reserved
		0x4001 1800 - 0x4001 1BFF	Reserved
		0x4001 1400 - 0x4001 17FF	Reserved
		0x4001 1000 - 0x4001 13FF	Reserved
		0x4001 0C00 - 0x4001 0FFF	Reserved
		0x4001 0800 - 0x4001 0BFF	Reserved
		0x4001 0400 - 0x4001 07FF	EXTI
		0x4001 0000 - 0x4001 03FF	SYSCFG
	APB1	0x4000 DC00 - 0x4000 FFFF	Reserved
		0x4000 D800 - 0x4000 DBFF	Reserved
		0x4000 D400 - 0x4000 D7FF	Reserved
		0x4000 D000 - 0x4000 D3FF	Reserved
		0x4000 CC00 - 0x4000 CFFF	Reserved
		0x4000 C800 - 0x4000 CBFF	Reserved
		0x4000 C400 - 0x4000 C7FF	Reserved
		0x4000 C000 - 0x4000 C3FF	I2C2
		0x4000 9800 - 0x4000 BFFF	Reserved
		0x4000 9400 - 0x4000 97FF	LPTIMER
		0x4000 8400 - 0x4000 93FF	Reserved
		0x4000 8000 - 0x4000 83FF	Reserved
		0x4000 7C00 - 0x4000 7FFF	Reserved
		0x4000 7800 - 0x4000 7BFF	Reserved

Pre-defined Regions	Bus	Address	Peripherals
		0x4000 7400 - 0x4000 77FF	Reserved
		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	I2C3
		0x4000 5800 - 0x4000 5BFF	I2C1
		0x4000 5400 - 0x4000 57FF	I2C0
		0x4000 5000 - 0x4000 53FF	UART4
		0x4000 4C00 - 0x4000 4FFF	UART3
		0x4000 4800 - 0x4000 4BFF	USART2
		0x4000 4400 - 0x4000 47FF	USART1
		0x4000 4000 - 0x4000 43FF	Reserved
		0x4000 3C00 - 0x4000 3FFF	SPI2
		0x4000 3800 - 0x4000 3BFF	SPI1
		0x4000 3400 - 0x4000 37FF	Reserved
		0x4000 3000 - 0x4000 33FF	FWDGT
		0x4000 2C00 - 0x4000 2FFF	WWDGT
		0x4000 2800 - 0x4000 2BFF	RTC
		0x4000 2400 - 0x4000 27FF	Reserved
		0x4000 2000 - 0x4000 23FF	Reserved
		0x4000 1C00 - 0x4000 1FFF	Reserved
		0x4000 1800 - 0x4000 1BFF	Reserved
		0x4000 1400 - 0x4000 17FF	TIMER6
		0x4000 1000 - 0x4000 13FF	TIMER5
		0x4000 0C00 - 0x4000 0FFF	TIMER4
		0x4000 0800 - 0x4000 0BFF	TIMER3
		0x4000 0400 - 0x4000 07FF	TIMER2
		0x4000 0000 - 0x4000 03FF	TIMER1
SRAM		0x2002 0000 - 0x3FFF FFFF	Reserved
		0x2001 C000 - 0x2001 FFFF	TCMSRAM aliased(32KB)
		0x2001 8000 - 0x2001 BFFF	
		0x2001 4000 - 0x2001 7FFF	SRAM1(16KB)
		0x2000 D000 - 0x2001 3FFF	SRAM0(80KB)
		0x2000 C000 - 0x2000 CFFF	
		0x2000 8000 - 0x2000 BFFF	
		0x2000 5000 - 0x2000 7FFF	
		0x2000 2000 - 0x2000 4FFF	
		0x2000 1000 - 0x2000 1FFF	

Pre-defined Regions	Bus	Address	Peripherals
		0x2000 0000 - 0x2000 0FFF	
Code		0x1FFF FC10 - 0x1FFF FFFF	Reserved
		0x1FFF FC00 - 0x1FFF FC0F	Reserved
		0x1FFF F830 - 0x1FFF BFFF	Reserved
		0x1FFF F800 - 0x1FFF F82F	Option Bytes
		0x1FFF C000 - 0x1FFF F7FF	Reserved
		0x1FFF 8000 - 0x1FFF BFFF	System memory 1
		0x1FFF 7830 - 0x1FFF 7FFF	Reserved
		0x1FFF 7800 - 0x1FFF 782F	Option Bytes
		0x1FFF 7000 - 0x1FFF 77FF	OTP
		0x1FFF 3400 - 0x1FFF 6FFF	Reserved
		0x1FFF 0000 - 0x1FFF 33FF	System memory 0
		0x1000 8000 - 0x1FFE FFFF	Reserved
		0x1000 0000 - 0x1000 7FFF	TCMSRAM
		0x0A00 8000 - 0x0FFF FFFF	Reserved
		0x0A00 6000 - 0x0A00 7FFF	Reserved
		0x0A00 4000 - 0x0A00 5FFF	Reserved
		0x0A00 0000 - 0x0A00 3FFF	Reserved
		0x08C0 1000 - 0x09FF FFFF	Reserved
		0x08C0 0000 - 0x08C0 0FFF	Reserved
		0x0881 0000 - 0x08BF FFFF	Reserved
		0x0880 0000 - 0x0880 FFFF	Reserved
		0x0808 0000 - 0x0871 FFFF	Reserved
		0x0806 0000 - 0x0807 FFFF	Main Flash memory
		0x0802 0000 - 0x0805 FFFF	
		0x0801 0000 - 0x0801 FFFF	
		0x0800 0000 - 0x0800 FFFF	
		0x0006 0000 - 0x07FF FFFF	Reserved
		0x0002 0000 - 0x0007 FFFF	Aliased to Flash or system memory
		0x0001 0000 - 0x0001 FFFF	
		0x0000 0000 - 0x0000 FFFF	

1.3.1. On-chip SRAM memory

The GD32G553 series contain up to 80KB of SRAM0, 16KB of SRAM1 and 32KB of TCMSRAM. It supports byte, half-word (16 bits), and word (32 bits) accesses.

ECC

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

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

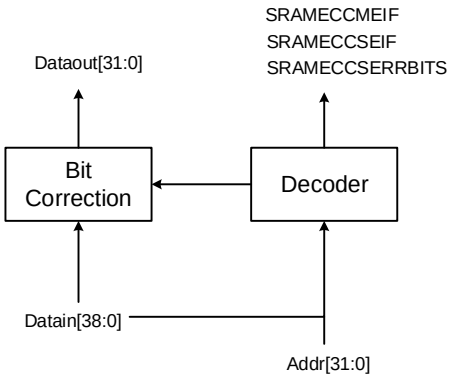
The ECC module is composed of an encoder and a decoder.

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

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

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

Figure 1-3. ECC decoder



EEIC

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

Single bit correction error event

When a single-bit correction error event is detected in SRAM0, EEIC:

(1) The SRAM0ECCSEIF bit int SYSCFG_STAT register will be set. Software can clear it by writing 1.
(2) The SYSCFG_CFG3 records the address where the single-bit correction error event occurred.

When a single-bit correction error event is detected in SRAM1, EEIC:

(1) The SRAM1ECCSEIF bit int SYSCFG_STAT register will be set. Software can clear it by writing 1.

(2) The SYSCFG_CFG4 records the address where the single-bit correction error event occurred.

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

(1) The TCMSRAMECCSEIF bit int SYSCFG_STAT register will be set. Software can clear it by writing 1.
(2) The SYSCFG_CFG5 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 in SRAM0, EEIC:

(1) The SRAM0ECCMEIF bit int SYSCFG_STAT register will be set. Software can clear it by writing 1.
(2) The SYSCFG_CFG3 records the address where the multi-bits non-correction error event occurred.

When a multi-bits non-correction error event is detected in SRAM1, EEIC:

(1) The SRAM1ECCMEIF bit int SYSCFG_STAT register will be set. Software can clear it by writing 1.
(2) The SYSCFG_CFG4 records the address where the multi-bits non-correction error event occurred.

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

(1) The TCMSRAMECCMEIF bit int SYSCFG_STAT register will be set. Software can clear it by writing 1.
(2) The SYSCFG_CFG5 records the address where the multi-bits non-correction error event occurred.

Single bit correction error interrupt

Set the SRAM0ECCSEIE bit in SYSCFG_CFG3 register. When a single-bit error correctable event is detected, a corresponding interrupt will be generated.

Set the SRAM1ECCSEIE bit in SYSCFG_CFG4 register. When a single-bit error correctable event is detected, a corresponding interrupt will be generated.

Set the TCMSRAMECCSEIE bit in SYSCFG_CFG5 register. When a single-bit error correctable event is detected, a corresponding interrupt will be generated.

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

Set the SRAM0ECCMEIE bit in SYSCFG_CFG3 register. When a multi-bits error non-correction event is detected, a NMI interrupt will be generated.

Set the SRAM1ECCMEIE bit in SYSCFG_CFG4 register. When a multi-bits error non-correction event is detected, a NMI interrupt will be generated.

Set the TCMSRAMECCMEIE bit in SYSCFG_CFG5 register. When a multi-bits error non-correction event is detected, a NMI interrupt will be generated.

TCMSRAM write protection

The TCMSRAM can be write protected with a page granularity of 1KB. Write protection function of TCMSRAM page can be enabled by setting PxWPEN(x=0..31) bit in SYSCFG TCMSRAM write protection register (SYSCFG_TCMSRAMWP) and write protection function can be removed/cleared by a system reset only. When TCMSRAM is set to write protection, operations on it will generate a hardfault.

TCMSRAM erase

The TCMRAM erase can also be requested by software by setting TCMSRAMERS bit in SYSCFG TCMSRAM control and status register (SYSCFG_TCMSRAMCS) and TCMSRAMBSYF bit reflects whether TCMSRAM erase operation is on going.

The TCMSRAM can be erased with a system reset using the option bit TCMSRAM_ERS in the user option byte.

1.3.2. On-chip Flash memory

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

- Up to 512KB of main Flash memory.
- Option bytes to configure the device.

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

1.4. Boot configuration

The GD32G553 series provide three kinds of boot sources which can be selected by the BOOT0 pin and boot configuration bits nBOOT1, nSWBT0 and nBOOT0 in the user option byte. The details are shown in ***Table 1-3. Boot modes***. 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. Once the pin has been sampled, it is free and can be used for other purposes.

Table 1-3. Boot modes

Selected boot area	Boot mode configuration				
	BOOTLK	nBOOT1 bit	BOOT0 pin	nSWBT0 bit	nBOOT0 bit
Main Flash memory	1	X	X	X	X
Main Flash memory	0	X	0	1	X
Main Flash memory	0	X	X	0	1
System memory	0	1	1	1	X
System memory	0	1	X	0	0
Embedded SRAM0	0	0	1	1	X
Embedded SRAM0	0	0	X	0	0

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 D000) is aliased in the boot memory space which begins at the address 0x0000 0000. When the on-chip SRAM whose memory space is beginning at 0x2000 0000 is selected as the boot source, in the application initialization code, you have to relocate the vector table in SRAM using the NVIC exception table and offset register.

The embedded boot loader is located in the System memory, which is used to reprogram the Flash memory. The boot loader can be activated through certain interfaces refer to the boot mode chapter of datasheet.

1.5. Remap configuration

When the boot mode configuration is selected, the software can configure the memory remap function through BOOT_MODE bit in system configuration register 0 (SYSCFG_CFG0). Memories including main flash memory, system memory, embedded SRAM, EXMC BANK0 and QSPI memory can be remapped.

Flash bank remap function is configured by FMC_SWP bit in system configuration register 0

(SYSCFG_CFG0).

1.6. System configuration controller

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

- Remapping memory areas.
- Configuring FPU interrupts.
- I2C Fm+ configuration and voltage booster for I/Os anolog switches.
- Configuring TCMSRAM write protection and software erase.
- Managing timer break input lock.
- Managing the external interrupt line connection to the GPIOs.

1.7. System configuration registers

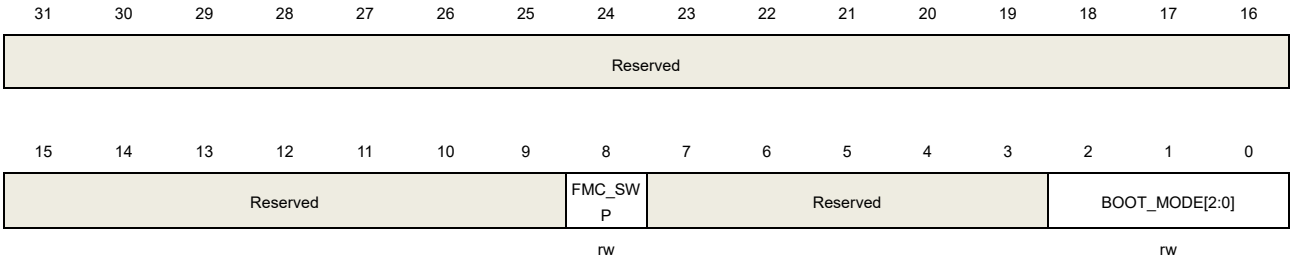
SYSCFG base address: 0x4001 0000

1.7.1. System configuration register 0 (SYSCFG_CFG0)

Address offset: 0x00

Reset value: 0x0000 010X (X indicates the memory mode may be any value according to the BOOT0 pin and boot configuration bits nBOOT1, SWBT0 and nBOOT0 in the user option byte after reset)

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



Bits	Fields	Descriptions
31:9	Reserved	Must be kept at reset value.
8	FMC_SWP	FMC memory mapping swap This bit controls the address mapping swap between bank 0 and bank 1 of the Main Flash. 0: Flash bank 1 is mapped at 0x0800 0000 and Flash bank 0 is mapped at 0x0804 0000 1: Flash bank 0 is mapped at 0x0800 0000 and Flash bank 1 is mapped at 0x0804 0000 **Note:** Depend on the specific series of bank size.
7:3	Reserved	Must be kept at reset value.
2:0	BOOT_MODE[2:0]	Boot mode These bits select the device accessible at address 0x00000000. 000: Main Flash memory is mapped at address 0x00000000 001: System Flash memory is mapped at address 0x00000000 010: EXMC memory is mapped at address 0x00000000 011: SRAM0 is mapped at address 0x00000000 100: QSPI memory is mapped at address 0x00000000 Others: Reserved

1.7.2. System configuration register 1 (SYSCFG_CFG1)

Address offset: 0x04

Reset value: 0x7C00 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
IXIE	IDIE	OVFIE	UFIE	DZIE	IOPIE	Reserved		I2C3FMP EN	I2C2FMP EN	I2C1FMP EN	I2C0FMP EN	PB9FMP EN	PB8FMP EN	PB7FMP EN	PB6FMP EN
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
Reserved															

Bits	Fields	Descriptions
31	IXIE	FPU inexact interrupt enable bit 0: Disable inexact interrupt 1: Enable inexact interrupt
30	IDIE	FPU input denormal interrupt enable bit 0: Disable input denormal interrupt 1: Enable input denormal interrupt
29	OVFIE	FPU overflow interrupt enable bit 0: Disable overflow interrupt 1: Enable overflow interrupt
28	UFIE	FPU underflow interrupt enable bit 0: Disable underflow interrupt 1: Enable underflow interrupt
27	DZIE	FPU divide by 0 interrupt enable bit 0: Disable divide by 0 interrupt 1: Enable divide by 0 interrupt
26	IOPIE	FPU invalid operation interrupt enable bit 0: Disable invalid operation interrupt 1: Enable invalid operation interrupt
25:24	Reserved	Must be kept at reset value
23	I2C3FMPEN	I2C3 Fm+ mode enable This bit controls I2C0 Fm+ mode. 0: Disable Fm+ mode 1: Enable Fm+ mode
22	I2C2FMPEN	I2C2 Fm+ mode enable This bit controls I2C2 Fm+ mode.

		0: Disable Fm+ mode 1: Enable Fm+ mode
21	I2C1FMPEN	I2C1 Fm+ mode enable This bit controls I2C1 Fm+ mode. 0: Disable Fm+ mode 1: Enable Fm+ mode
20	I2C0FMPEN	I2C0 Fm+ mode enable This bit controls I2C0 Fm+ mode. 0: Disable Fm+ mode 1: Enable Fm+ mode
19	PB9FMPEN	I2C Fm+ mode on PB9 pin enable This bit controls I2C Fm+ mode, the speed control of the pin is bypassed. 0: Disable Fm+ mode on the PB9 pin 1: Enable Fm+ mode on the PB9 pin
18	PB8FMPEN	I2C Fm+ mode on PB8 pin enable This bit controls I2C Fm+ mode, the speed control of the pin is bypassed. 0: Disable Fm+ mode on the PB8 pin 1: Enable Fm+ mode on the PB8 pin
17	PB7FMPEN	I2C Fm+ mode on PB7 pin enable This bit controls I2C Fm+ mode, the speed control of the pin is bypassed. 0: Disable Fm+ mode on the PB7 pin 1: Enable Fm+ mode on the PB7 pin
16	PB6FMPEN	I2C Fm+ mode on PB6 pin enable This bit controls I2C Fm+ mode, the speed control of the pin is bypassed. 0: Disable Fm+ mode on the PB6 pin 1: Enable Fm+ mode on the PB6 pin
15:0	Reserved	Must be kept at reset value.

1.7.3. EXTI sources selection register 0 (SYSCFG_EXTISS0)

Address offset: 0x08

Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI3_SS [3:0]				EXTI2_SS [3:0]				EXTI1_SS [3:0]				EXTI0_SS [3:0]			
rw				rw				rw				rw			

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:12	EXTI3_SS[3:0]	EXTI 3 sources selection 0000: PA3 pin 0001: PB3 pin 0010: PC3 pin 0011: PD3 pin 0100: PE3 pin 0101: PF3 pin 0110: PG3 pin
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 0101: PF2 pin 0110: PG2 pin
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 0101: PF1 pin 0110: PG1 pin
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 0101: PF0 pin 0110: PG0 pin

1.7.4. EXTI sources selection register 1 (SYSCFG_EXTISS1)

Address offset: 0x0C

Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:12	EXTI7_SS[3:0]	EXTI 7 sources selection 0000: PA7 pin 0001: PB7 pin 0010: PC7 pin 0011: PD7 pin 0100: PE7 pin 0101: PF7 pin 0110: PG7 pin
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 0101: PF6 pin 0110: PG6 pin
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 0101: PF5 pin 0110: PG5 pin
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 0101: PF4 pin 0110: PG4 pin

1.7.5. EXTI sources selection register 2 (SYSCFG_EXTISS2)

Address offset: 0x10

Reset value: 0x0000 0000

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

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

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

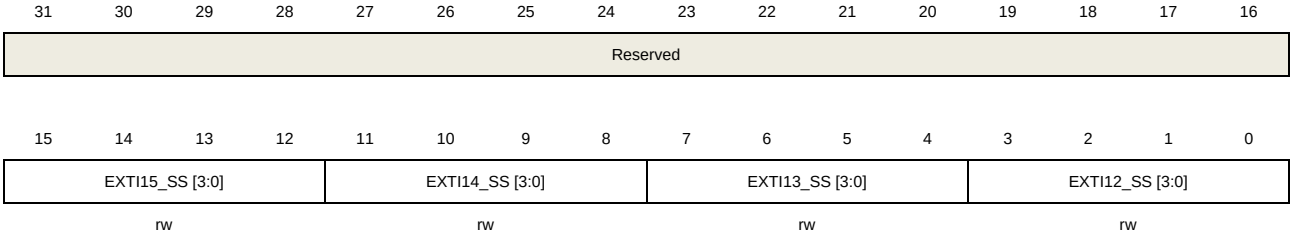
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:12	EXTI11_SS[3:0]	EXTI 11 sources selection 0000: PA11 pin 0001: PB11 pin 0010: PC11 pin 0011: PD11 pin 0100: PE11 pin 0101: PF11 pin
11:8	EXTI10_SS[3:0]	EXTI 10 sources selection 0000: PA10 pin 0001: PB10 pin 0010: PC10 pin 0011: PD10 pin 0100: PE10 pin 0101: PF10 pin 0110: PG10 pin
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 0101: PF9 pin 0110: PG9 pin
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
0101: PF8 pin
0110: PG8 pin

1.7.6. EXTI sources selection register 3 (SYSCFG_EXTISS3)

Address offset: 0x14
Reset value: 0x0000 0000

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



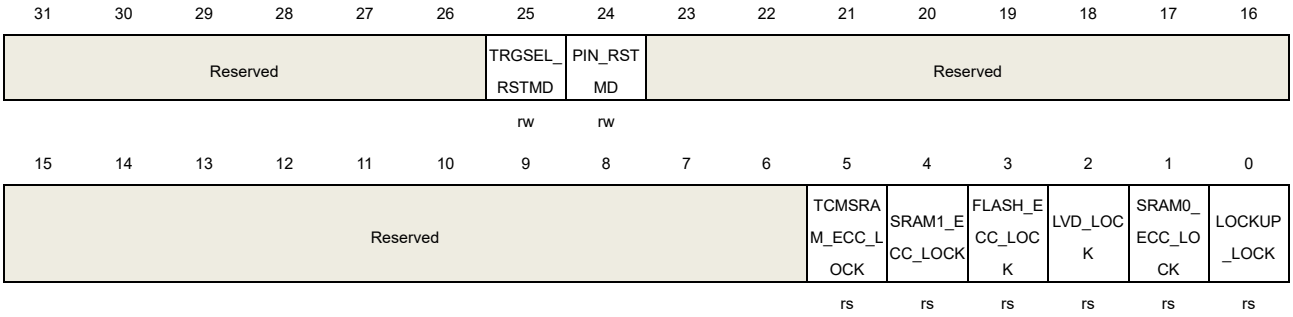
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:12	EXTI15_SS[3:0]	EXTI 15 sources selection 0000: PA15 pin 0001: PB15 pin 0010: PC15 pin 0011: PD15 pin 0100: PE15 pin 0101: PF15 pin
11:8	EXTI14_SS[3:0]	EXTI 14 sources selection 0000: PA14 pin 0001: PB14 pin 0010: PC14 pin 0011: PD14 pin 0100: PE14 pin 0101: PF14 pin
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 0101: PF13 pin
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
0101: PF12 pin

1.7.7. System configuration register 2 (SYSCFG_CFG2)

Address offset: 0x18
Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:26	Reserved	Must be kept at reset value
25	TRGSEL_RSTMD	Trigsel reset mode This bit can only be reset by power on reset 0: Normal mode. Trigger input source selection for CLA trigger input is reset when any reset event occurs. 1: Reset persist mode. Trigger input source selection for CLA trigger input will retain state across any reset event except for the POR event.
24	PIN_RSTMD	Pin reset mode This bit controls whether the GPIO pin configurations (GPIOx_CTL, GPIOx_OMODE, GPIOx_OSPD, GPIOx_PUD, GPIOx_OCTL, are reset when any reset event occurs). This bit will be retained across all reset events except POR. After a POR event, this bit is reset to 0. 0: GPIO pin configuration is reset when any reset event occurs. 1: GPIO pin configuration will retain state across any reset event except for the POR event.
23:6	Reserved	Must be kept at reset value
5	TCMSRAM_ECC_LOCK	TCMSRAM ECC lock enable bit This bit is set by software and cleared by system reset only. This is bit is used to enable and lock the TCMSRAM ECC error connection to

TIMER0/7/14/15/16/19 break input.
0: TCMSRAM ECC error disconnected from TIMER0/7/14/15/16/19 break input and system fault input of HRTIMER0.
1: TCMSRAM ECC error connected to TIMER0/7/14/15/16/19 break input and system fault input of HRTIMER0.

4	SRAM1_ECC_LOCK	SRAM1 ECC lock enable bit This bit is set by software and cleared by reset system only. This is bit is used to enable and lock the SRAM1 ECC error connection to TIMER0/7/14/15/16/19 break input. 0: SRAM1 ECC error disconnected from TIMER0/7/14/15/16/19 break input and system fault input of HRTIMER0. 1: SRAM1 ECC error connected to TIMER0/7/14/15/16/19 break input and system fault input of HRTIMER0.
3	FLASH_ECC_LOCK	Flash ECC lockup enable bit This bit is set by software and cleared by reset system only. 0: ECC error disconnected from TIMER0/7/14/15/16/19 break input and system fault input of HRTIMER0. 1: ECC error connected to TIMER0/7/14/15/16/19 break input and system fault input of HRTIMER0.
2	LVD_LOCK	LVD lcokup enable bit This bit is set by software and cleared by reset system only. 0: LVD interrupt disconnected from TIMER0/7/14/15/16/19 break input and system fault input of HRTIMER0. 1: LVD interrupt connected to TIMER0/7/14/15/16/19 break input and system fault input of HRTIMER0.
1	SRAM0_ECC_LOCK	SRAM0 ECC lockup enable bit This bit is set by software and cleared by reset system only. 0: SRAM0 ECC error disconnected from TIMER0/7/14/15/16/19 break input and system fault input of HRTIMER0. 1: SRAM0 ECC error connected to TIMER0/7/14/15/16/19 break input and system fault input of HRTIMER0.
0	LOCKUP_LOCK	Cortex®-M33 lockup (Hardfault) output enable bit This bit is set by software and cleared by reset system only. 0: Cortex®-M33 LOCKUP output disconnected from TIMER0/7/14/15/16/19 break input and system fault input of HRTIMER0 1: Cortex®-M33 LOCKUP output connected to TIMER0/7/14/15/16/19 break input and system fault input of HRTIMER0.

1.7.8. System status register (SYSCFG_STAT)

Address offset: 0x1C

Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							TCMSRAMECCSEIF	TCMSRAMECCMEIF	SRAM1ECCSEIF	SRAM1ECCMEIF	NMIPINIF	CKMNMIIF	FLASHECCIF	SRAM0ECCSEIF	SRAM0ECCMEIF
							rc_w1	rc_w1	rc_w1	rc_w1	r	rc_w1	rc_w1	rc_w1	rc_w1

Bits	Fields	Descriptions
31:9	Reserved	Must be kept at reset value
8	TCMSRAMECCSEIF	TCMSRAM single bit correction interrupt flag The software can clear it by writing 1. 0: no TCMSRAM single bit correction event is detected 1: TCMSRAM single bit correction event is detected
7	TCMSRAMECCMEIF	TCMSRAM multi-bits (two bits) non-correction interrupt flag The software can clear it by writing 1. 0: no TCMSRAM non-correction event is detected 1: TCMSRAM non-correction event is detected **Note:** TCMSRAM multi-bits non-correction ECC error will cause an NMI interrupt when the TCMSRAMECCMEIE bit is set.
6	SRAM1ECCSEIF	SRAM1 single bit correction interrupt flag The software can clear it by writing 1. 0: no SRAM1 single bit correction event is detected 1: SRAM1 single bit correction event is detected
5	SRAM1ECCMEIF	SRAM1 multi-bits (two bits) non-correction interrupt flag The software can clear it by writing 1. 0: no SRAM1 non-correction event is detected 1: SRAM1 non-correction event is detected **Note:** SRAM multi-bits non-correction ECC error will cause an NMI interrupt when the SRAM1ECCMEIE bit is set.
4	NMIPINIF	Interrupt flag from NMI pin 0: no input 1: interrupt input from nmi pin
3	CKNMIIF	HXTAL clock moniotor NMI interrupt flag The software can clear it by writing 1. 0: no HXTAL clock moniotor error 1: HXTAL clock moniotor is detected.

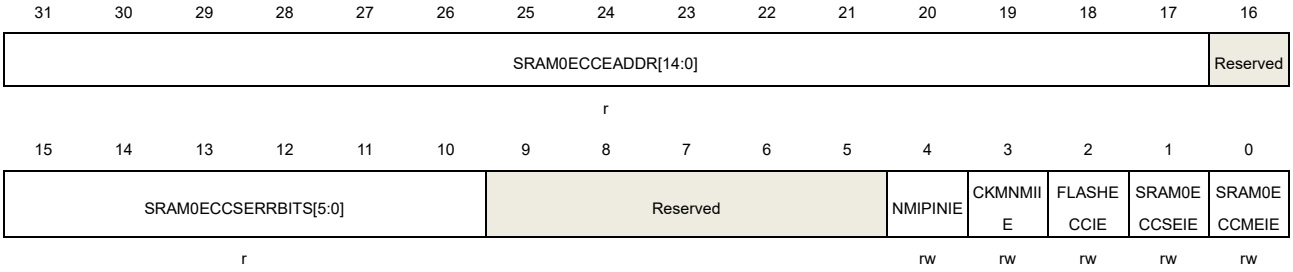
2	FLASHECCIF	Flash ECC NMI interrupt flag The software can clear it by writing 1. 0: no Flash ECC error 1: Flash ECC error is detected
1	SRAM0ECCSEIF	SRAM0 single bit correction interrupt flag The software can clear it by writing 1. 0: no SRAM0 single bit correction event is detected 1: SRAM0 single bit correction event is detected
0	SRAM0ECCMEIF	SRAM0 multi-bits (two bits) non-correction interrupt flag The software can clear it by writing 1. 0: no SRAM0 non-correction event is detected 1: SRAM0 non-correction event is detected **Note:** SRAM multi-bits non-correction ECC error will cause an NMI interrupt when the SRAM0ECCMEIE bit is set.

1.7.9. System configuration register 3 (SYSCFG_CFG3)

Address offset: 0x20

Reset value: 0xXXXX X00F

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



Bits	Fields	Descriptions
31:17	SRAM0ECCEADDR[14:0]	Record the faulting system address (SRAM0ECCEADDR = SRAM0 address[16:0] >> 2) where the last SRAM0 ECC event on SRAM occurred.
16	Reserved	Must be kept at reset value.
15:10	SRAM0ECCSERRBITS[5:0]	Which one bit has SRAM0 ECC single-bit correctable error 0: no error 1: bit 0 … 32: bit 31
9:5	Reserved	Must be kept at reset value.
4	NMIPINIE	NMI pin interrupt enable This bit can be used to enable the interrupt flag from NMI pin connection to NMI

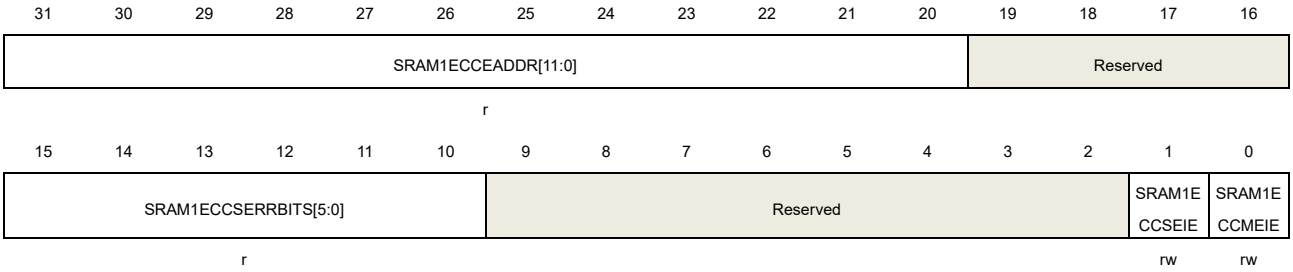
		interrupt. 0: Disable 1: Enable
3	CKNMIIF	HXTAL clock moniotor NMI interrupt enable This bit can be used to enable the HXTAL clock moniotor NMI interrupt flag connection to NMI interrupt. 0: Disable 1: Enable
2	FLASHECCIE	Flash ECC NMI interrupt flag This bit can be used to enable Flash ECC NMI interrupt flag connection to NMI interrupt. 0: Disable 1: Enable
1	SRAM0ECCSEIE	SRAM0 single bit correction error interrupt enable This bit can be used to enable SRAM0 single bit correction event flag connection to NVIC. 0: Disable 1: Enable
0	SRAM0ECCMEIE	SRAM0 multi-bits (two bits) non-correction error NMI interrupt enable This bit can be used to enable SRAM0 two bits non-correction event flag 0: Disable 1: Enable

1.7.10. System configuration register 4 (SYSCFG_CFG4)

Address offset: 0x24

Reset value: 0xXXX0 XX03

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



Bits	Fields	Descriptions
31:20	SRAM1ECCEADDR[11:0]	Record the faulting system address (SRAM1ECCEADDR = SRAM1 address[13:0] >> 2) where the last SRAM1 ECC event on SRAM occurred.
19:16	Reserved	Must be kept at reset value.

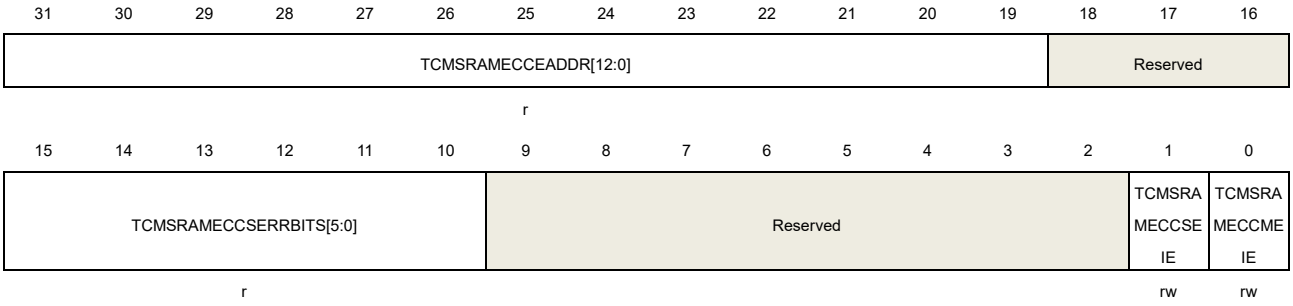
15:10	SRAM1ECCSERRBITS[5:0]	Which one bit has SRAM1 ECC single-bit correctable error 0: no error 1: bit 0 … 32: bit 31
9:2	Reserved	Must be kept at reset value.
1	SRAM1ECCSEIE	SRAM1 single bit correction error event flag connection to NVIC This bit can be used to enable SRAM1 single bit correction event flag connection to NVIC. 0: Disable 1: Enable
0	SRAM1ECCMEIE	SRAM1 multi-bits (two bits) non-correction error event flag connection to NMI This bit can be used to enable SRAM1 two bits non-correction event flag connection to NMI interrupt. 0: Disable 1: Enable

1.7.11. System configuration register 5 (SYSCFG_CFG5)

Address offset: 0x28

Reset value: 0xXXXX XX03

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



Bits	Fields	Descriptions
31:19	TCMSRAMECCEADDR[12:0]	Record the faulting system address (TCMSRAMECCEADDR = TCM SRAM address[14:0] >> 2) where the last TCMSRAM ECC event on SRAM occurred.
18:16	Reserved	Must be kept at reset value.
15:10	TCMSRAMECCSERRBITS[5:0]	Which one bit has TCMSRAM ECC single-bit correctable error 0: no error 1: bit 0 … 32: bit 31

9:2	Reserved	Must be kept at reset value.
1	TCMSRAMECCSEIE	TCMSRAM single bit correction error event flag This bit can be used to enable TCMSRAM single bit correction event flag connection to NVIC. 0: Disable 1: Enable
0	TCMSRAMECCMEIE	TCMSRAM multi-bits (two bits) non-correction error event flag This bit can be used to enable TCMSRAM two bits non-correction event flag connection to NMI interrupt. 0: Disable 1: Enable

1.7.12. SYSCFG TCMSRAM control and status register (SYSCFG_TCMSRAMCS)

Address offset: 0x30
Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value
1	TCMSRAMBSYF	TCMSRAM busy flag by erase operation 0: No TCMSRAM erase operation is on going. 1: TCMSRAM erase operation is on going.
0	TCMSRAMERS	TCMSRAM erase Hardware TCMSRAM erase operation will start when this bit is set to 1 and be automatically cleared when TCMSRAM erase operation is end.Before erasing the TCMSRAM, it is necessary to unlock the TCMSRAM first.

1.7.13. SYSCFG TCMSRAM key register (SYSCFG_TCMSRAMKEY)

Address offset: 0x34
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								WP_KEY[7:0]							
								w							

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value
7:0	WP_KEY[7:0]	TCMSRAMERS write protection key Unlock the write protection of the TCMSRAMERS bit in the SYSCFG_TCMSRAMCS register according to the following steps. Step 1. Write "0xCA" into WP_KEY[7:0] Step 2. Write "0x53" into WP_KEY[7:0] **Note:** Incorrect key will reactivate TCMSRAMERS write protection.

1.7.14. SYSCFG TCMSRAM write protection register (SYSCFG_TCMSRAMWP)

Address offset: 0x38

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
P31WPEN	P30WPEN	P29WPEN	P28WPEN	P27WPEN	P26WPEN	P25WPEN	P24WPEN	P23WPEN	P22WPEN	P21WPEN	P20WPEN	P19WPEN	P18WPEN	P17WPEN	P16WPEN
rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P15WPEN	P14WPEN	P13WPEN	P12WPEN	P11WPEN	P10WPEN	P9WPEN	P8WPEN	P7WPEN	P6WPEN	P5WPEN	P4WPEN	P3WPEN	P2WPEN	P1WPEN	P0WPEN
rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs

Bits	Fields	Descriptions
31:0	PxWPEN(x=0..31)	TCMSRAM page x write protection enable bit These bits are set by software and cleared only by system reset. 0: Disabled 1: Enabled

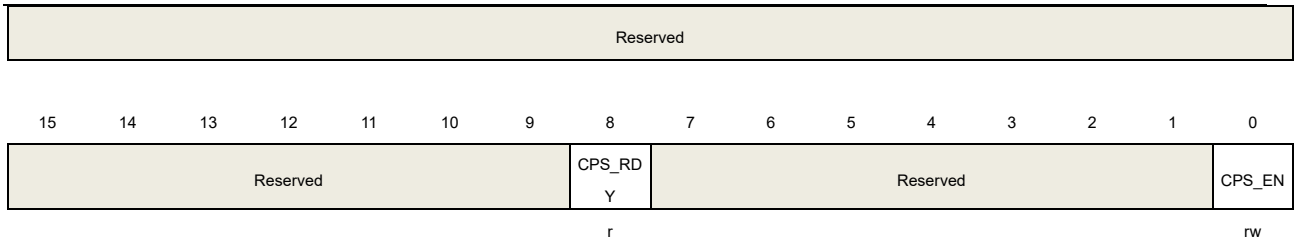
1.7.15. I/O Compensation cell control/status register (SYSCFG_CPSCTL)

Address offset: 0x3C

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



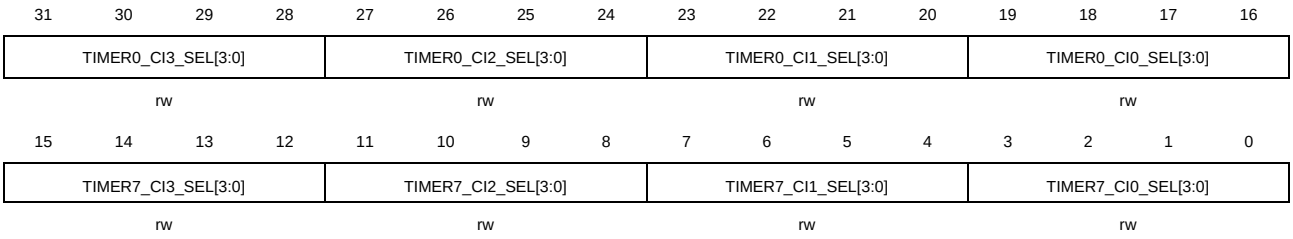
Bits	Fields	Descriptions
31:9	Reserved	Must be kept at reset value
8	CPS_RDY	Compensation cell ready flag This bit provides the status of the compensation cell. 0: I/O compensation cell not ready 1: I/O compensation cell ready
7:1	Reserved	Must be kept at reset value.
0	CPS_EN	I/O compensation cell enable This bit enables the I/O compensation cell. 0: I/O compensation cell disabled 1: I/O compensation cell enabled

1.7.16. Timer input selection register 0 (SYSCFG_TIMERCISEL0)

Address offset: 0x44

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:28	TIMER0_CI3_SEL[3:0]	Selects TIMER0_CI3 input 0000: TIMER0_CH3 0001: CLA3OUT Others: Reserved
27:24	TIMER0_CI2_SEL[3:0]	Selects TIMER0_CI2 input 0000: TIMER0_CH2 0001: CLA2OUT Others: Reserved

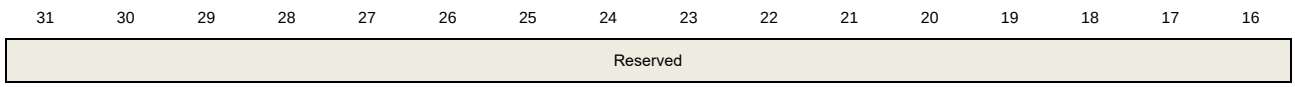
23:20	TIMER0_CI1_SEL[3:0]	Selects TIMER0_CI1 input 0000: TIMER0_CH1 0001: CLA1OUT Others: Reserved
19:16	TIMER0_CI0_SEL[3:0]	Selects TIMER0_CI0 input 0000: TIMER0_CH0 0001: CMP0 output 0010: CMP1 output 0011: CMP2 output 0100: CMP3 output 0101: CLA0OUT Others: Reserved
15:12	TIMER7_CI3_SEL[3:0]	Selects TIMER7_CI3 input 0000: TIMER7_CH3 0001: CLA3OUT Others: Reserved
11:8	TIMER7_CI2_SEL[3:0]	Selects TIMER7_CI2 input 0000: TIMER7_CH2 0001: CLA2OUT Others: Reserved
7:4	TIMER7_CI1_SEL[3:0]	Selects TIMER7_CI1 input 0000: TIMER7_CH1 0001: CLA1OUT Others: Reserved
3:0	TIMER7_CI0_SEL[3:0]	Selects TIMER7_CI0 input 0000: TIMER7_CH0 0001: CMP0 output 0010: CMP1 output 0011: CMP2 output 0100: CMP3 output 0101: CLA0OUT Others: Reserved

1.7.17. Timer input selection register 1 (SYSCFG_TIMERCISEL1)

Address offset: 0x48

Reset value: 0x0000 0000

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



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMER19_CI3_SEL[3:0]				TIMER19_CI2_SEL[3:0]				TIMER19_CI1_SEL[3:0]				TIMER19_CI0_SEL[3:0]			
rw				rw				rw				rw			

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:12	TIMER19_CI3_SEL[3:0]	Selects TIMER19_CI3 input 0000: TIMER19_CH3 0001: CLA3OUT Others: Reserved
11:8	TIMER19_CI2_SEL[3:0]	Selects TIMER19_CI2 input 0000: TIMER19_CH2 0001: CLA2OUT Others: Reserved
7:4	TIMER19_CI1_SEL[3:0]	Selects TIMER19_CI1 input 0000: TIMER19_CH1 0001: CLA1OUT Others: Reserved
3:0	TIMER19_CI0_SEL[3:0]	Selects TIMER19_CI0 input 0000: TIMER19_CH0 0001: CMP0 output 0010: CMP1 output 0011: CMP2 output 0100: CMP3 output 0101: CLA0OUT Others: Reserved

1.7.18. Timer input selection register 2 (SYSCFG_TIMERCISEL2)

Address offset: 0x4C

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
TIMER1_CI3_SEL[3:0]				TIMER1_CI2_SEL[3:0]				TIMER1_CI1_SEL[3:0]				TIMER1_CI0_SEL[3:0]			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMER2_CI3_SEL[3:0]				TIMER2_CI2_SEL[3:0]				TIMER2_CI1_SEL[3:0]				TIMER2_CI0_SEL[3:0]			
rw				rw				rw				rw			

Bits	Fields	Descriptions

31:28	TIMER1_CI3_SEL[3:0]	TIMER1_CI3 input selection These bits select the TIMER1_CI3 input source. 0000: TIMER1_CH3 0001: CMP0 output 0010: CMP1 output 0011: CLA3OUT Others: Reserved
27:24	TIMER1_CI2_SEL[3:0]	TIMER1_CI2 input selection These bits select the TIMER1_CI2 input source. 0000: TIMER1_CH2 0001: CMP3 output 0010: CLA2OUT Others: Reserved
23:20	TIMER1_CI1_SEL[3:0]	TIMER1_CI1 input selection These bits select the TIMER1_CI1 input source. 0000: TIMER1_CH1 0001: CMP0 output 0010: CMP1 output 0011: CMP2 output 0100: CMP3 output 0101: CMP5 output 0110: CLA1OUT Others: Reserved
19:16	TIMER1_CI0_SEL[3:0]	TIMER1_CI0 input selection These bits select the TIMER1_CI0 input source. 0000: TIMER1_CH0 0001: CMP0 output 0010: CMP1 output 0011: CMP2 output 0100: CMP3 output 0101: CMP4 output 0110: CLA0OUT Others: Reserved
15:12	TIMER2_CI3_SEL[3:0]	TIMER2_CI3 input selection These bits select the TIMER2_CI3 input source. 0000: TIMER2_CH3 0001: CLA3OUT Others: Reserved
11:8	TIMER2_CI2_SEL[3:0]	TIMER2_CI2 input selection These bits select the TIMER2_CI2 input source. 0000: TIMER2_CH2

		0001: CMP2 output 0010: CLA2OUT Others: Reserved
7:4	TIMER2_CI1_SEL[3:0]	TIMER2_CI1 input selection These bits select the TIMER2_CI1 input source. 0000: TIMER2_CH1 0001: CMP0 output 0010: CMP1 output 0011: CMP2 output 0100: CMP3 output 0101: CMP4 output 0110: CMP5 output 0111: CMP6 output 1000: CLA1OUT Others: Reserved
3:0	TIMER2_CI0_SEL[3:0]	TIMER2_CI0 input selection These bits select the TIMER2_CI0 input source. 0000: TIMER2_CH0 0001: CMP0 output 0010: CMP1 output 0011: CMP2 output 0100: CMP3 output 0101: CMP4 output 0110: CMP5 output 0111: CMP6 output 1000: CLA0OUT Others: Reserved

1.7.19. Timer input selection register 3 (SYSCFG_TIMERCISEL3)

Address offset: 0x50

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
TIMER3_CI3_SEL[3:0]				TIMER3_CI2_SEL[3:0]				TIMER3_CI1_SEL[3:0]				TIMER3_CI0_SEL[3:0]			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMER4_CI3_SEL[3:0]				TIMER4_CI2_SEL[3:0]				TIMER4_CI1_SEL[3:0]				TIMER4_CI0_SEL[3:0]			
rw				rw				rw				rw			

Bits	Fields	Descriptions
31:28	TIMER3_CI3_SEL[3:	TIMER3_CI3 input selection

	0]	These bits select the TIMER3_CI3 input source. 0000: TIMER3_CH3 0001: CMP5 0010: CLA3OUT Others: Reserved
27:24	TIMER3_CI2_SEL[3:0]	TIMER3_CI2 input selection These bits select the TIMER3_CI2 input source. 0000: TIMER3_CH2 0001: CMP4 output 0010: CLA2OUT Others: Reserved
23:20	TIMER3_CI1_SEL[3:0]	TIMER3_CI1 input selection These bits select the TIMER3_CI1 input source. 0000: TIMER3_CH1 0001: CMP0 output 0010: CMP1 output 0011: CMP2 output 0100: CMP3 output 0101: CMP4 output 0110: CMP5 output 0111: CMP6 output 1000: CLA1OUT Others: Reserved
19:16	TIMER3_CI0_SEL[3:0]	TIMER3_CI0 input selection These bits select the TIMER3_CI0 input source. 0000: TIMER3_CH0 0001: CMP0 output 0010: CMP1 output 0011: CMP2 output 0100: CMP3 output 0101: CMP4 output 0110: CMP5 output 0111: CMP6 output 1000: CLA0OUT Others: Reserved
15:12	TIMER4_CI3_SEL[3:0]	TIMER4_CI3 input selection These bits select the TIMER4_CI3 input source. 0000: TIMER4_CH3 0001: CLA3OUT Others: Reserved
11:8	TIMER4_CI2_SEL[3:	TIMER4_CI2 input selection

	0]	These bits select the TIMER4_CI2 input source. 0000: TIMER4_CH2 0001: CLA2OUT Others: Reserved
7:4	TIMER4_CI1_SEL[3:0]	TIMER4_CI1 input selection These bits select the TIMER4_CI1 input source. 0000: TIMER4_CH1 0001: CMP0 output 0010: CMP1 output 0011: CMP2 output 0100: CMP3 output 0101: CMP4 output 0110: CMP5 output 0111: CMP6 output 1000: CLA1OUT Others: Reserved
3:0	TIMER4_CI0_SEL[3:0]	TIMER4_CI0 input selection These bits select the TIMER4_CI0 input source. 0000: TIMER4_CH0 0001: IRC32K 0010: LXTAL 0011: RTC wake-up 0100: CMP0 output 0101: CMP1 output 0110: CMP2 output 0111: CMP3 output 1000: CMP4 output 1001: CMP5 output 1010: CMP6 output 1011: CLA0OUT Others: Reserved

1.7.20. Timer input selection register 4 (SYSCFG_TIMERCISEL4)

Address offset: 0x54

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
TIMER16_CI0_SEL[3:0]				TIMER15_CI0_SEL[3:0]				TIMER14_CI1_SEL[3:0]				TIMER14_CI0_SEL[3:0]			

rw rw rw rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:12	TIMER16_CI0_SEL[3:0]	Selects TIMER16_CI0 input 0000: TIMER16_CH0 0001: CMP4 output 0010: CKOUT 0011: HXTAL/32 0100: RTC Clock 0101: LXTAL 0110: IRC32K 0111: CLA0OUT 1000: CLA1OUT 1001: CLA2OUT 1010: CLA3OUT 1011: HXTAL Others: Reserved
11:8	TIMER15_CI0_SEL[3:0]	Selects TIMER15_CI0 input 0000: TIMER15_CH0 0001: CMP5 output 0010: CKOUT 0011: HXTAL/32 0100: RTC Clock 0101: LXTAL 0110: IRC32K 0111: CLA0OUT 1000: CLA1OUT 1001: CLA2OUT 1010: CLA3OUT 1011: HXTAL Others: Reserved
7:4	TIMER14_CI1_SEL[3:0]	Selects TIMER14_CI1 input 0000: TIMER14_CH1 0001: CMP1 output 0010: CMP2 output 0011: CMP5 output 0100: CMP6 output Others: Reserved
3:0	TIMER14_CI0_SEL[3:0]	Selects TIMER14_CI0 input 0000: TIMER14_CH0

0001: LXTAL
0010: CMP0 output
0011: CMP1 output
0100: CMP4 output
0101: CMP6 output
0110: CLA0OUT
0111: CLA1OUT
1000: CLA2OUT
1001: CLA3OUT
Others: Reserved

1.7.21. TIMERx configuration register 0 (SYSCFG_TIMERxCFG0, x=0, 1, 2, 3, 4, 7, 19)

Address offset: 0x100 for TIMER0
Address offset: 0x10C for TIMER1
Address offset: 0x118 for TIMER2
Address offset: 0x124 for TIMER3
Address offset: 0x130 for TIMER4
Address offset: 0x13C for TIMER7
Address offset: 0x154 for TIMER19
Reset value: 0x0000 0000

TSCFG0[4:0], TSCFG1[4:0]..TSCFG14[4:0] are mutually exclusive and cannot be configured at the same time.

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	TSCFG5[4:0]					TSCFG4[4:0]					TSCFG3[4:0]				
	rw					rw					rw				

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

Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30:26	TSCFG5[4:0]	Event mode configuration A rising edge of the trigger input enables the counter. 00000: Event mode disable 00001: Internal trigger input 0 (ITI0) 00010: Internal trigger input 1 (ITI1) 00011: Internal trigger input 2 (ITI2)

		00100: Internal trigger input 3 (ITI3) 00101: CI0 edge flag (CI0F_ED) 00110: The filtered output of channel 0 input (CI0FE0) 00111: The filtered output of channel 1 input (CI1FE1) 01000: The filtered output of external trigger input (ETIFP) 01001: Internal trigger input 4 (ITI4) 01010: Internal trigger input 5 (ITI5) 01011: Internal trigger input 6 (ITI6) 01100: Internal trigger input 7 (ITI7) 01101: Internal trigger input 8 (ITI8) 01110: Internal trigger input 9 (ITI9) 01111: Internal trigger input 10 (ITI10) 10000: Reserved 10001: Reserved 10010: Reserved 10011: Internal trigger input 14 (ITI14) Others: Reserved
25:21	TSCFG4[4:0]	Pause mode configuration The trigger input enables the counter clock when it is high and disables the counter when it is low when these bits are not 0. 00000: Pause mode disable 00001: Internal trigger input 0 (ITI0) 00010: Internal trigger input 1 (ITI1) 00011: Internal trigger input 2 (ITI2) 00100: Internal trigger input 3 (ITI3) 00101: Reserved 00110: The filtered output of channel 0 input (CI0FE0) 00111: The filtered output of channel 1 input (CI1FE1) 01000: The filtered output of external trigger input (ETIFP) 01001: Internal trigger input 4 (ITI4) 01010: Internal trigger input 5 (ITI5) 01011: Internal trigger input 6 (ITI6) 01100: Internal trigger input 7 (ITI7) 01101: Internal trigger input 8 (ITI8) 01110: Internal trigger input 9 (ITI9) 01111: Internal trigger input 10 (ITI10) 10000: Reserved 10001: Reserved 10010: Reserved 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) 01000: The filtered output of external trigger input (ETIFP) 01001: Internal trigger input 4 (ITI4) 01010: Internal trigger input 5 (ITI5) 01011: Internal trigger input 6 (ITI6) 01100: Internal trigger input 7 (ITI7) 01101: Internal trigger input 8 (ITI8) 01110: Internal trigger input 9 (ITI9) 01111: Internal trigger input 10 (ITI10) 10000: Reserved 10001: Reserved 10010: Reserved 10011: Internal trigger input 14 (ITI14) 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.

1.7.22. TIMERx configuration register 1 (SYSCFG_TIMERxCFG1, x=0, 1, 2, 3, 4, 7, 19)

Address offset: 0x104 for TIMER0

Address offset: 0x110 for TIMER1
Address offset: 0x11C for TIMER2
Address offset: 0x128 for TIMER3
Address offset: 0x134 for TIMER4
Address offset: 0x140 for TIMER7
Address offset: 0x158 for TIMER19
Reset value: 0x0000 0000

TSCFG0[4:0], TSCFG1[4:0]..TSCFG14[4:0] are mutually exclusive and cannot be configured at the same time.

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	TSCFG11[4:0]					TSCFG10[4:0]					TSCFG9[4:0]				
	rw					rw					rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	TSCFG8[4:0]					TSCFG7[4:0]					TSCFG6[4:0]				
	rw					rw					rw				

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	Reserved	Must be kept at reset value.
14:10	TSCFG8[4:0]	Pause + restart mode configuration The counter will be reset when a rising edge or falling edge (is configured by PRMRPSEL bit in TIMERx_SMCFG register) of trigger input comes. And the counter counts when the trigger input is high, and it will stop when the trigger input

		is low. In this mode, the start and stop of the counter can be controlled. 00000: Pause + 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: Internal trigger input 4 (ITI4) 01010: Internal trigger input 5 (ITI5) 01011: Internal trigger input 6 (ITI6) 01100: Internal trigger input 7 (ITI7) 01101: Internal trigger input 8 (ITI8) 01110: Internal trigger input 9 (ITI9) 01111: Internal trigger input 10 (ITI10) 10000: Reserved 10001: Reserved 10010: Reserved 10011: Internal trigger input 14 (ITI14) Others: Reserved
9:5	TSCFG7[4:0]	Restart + event mode configuration The counter is reinitialized and started, the shadow registers are updated on the rising edge of the selected trigger input when these bits are not 0. 00000: Restart + 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: Internal trigger input 4 (ITI4) 01010: Internal trigger input 5 (ITI5) 01011: Internal trigger input 6 (ITI6) 01100: Internal trigger input 7 (ITI7) 01101: Internal trigger input 8 (ITI8) 01110: Internal trigger input 9 (ITI9) 01111: Internal trigger input 10 (ITI10) 10000: Reserved 10001: Reserved

		10010: Reserved 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) 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: Internal trigger input 4 (ITI4) 01010: Internal trigger input 5 (ITI5) 01011: Internal trigger input 6 (ITI6) 01100: Internal trigger input 7 (ITI7) 01101: Internal trigger input 8 (ITI8) 01110: Internal trigger input 9 (ITI9) 01111: Internal trigger input 10 (ITI10) 10000: Reserved 10001: Reserved 10010: Reserved 10011: Internal trigger input 14 (ITI14) Others: Reserved

1.7.23. TIMERx configuration register 2 (SYSCFG_TIMERxCFG2, x=0, 1, 2, 3, 4, 7, 19)

Address offset: 0x108 for TIMER0
Address offset: 0x114 for TIMER1
Address offset: 0x120 for TIMER2
Address offset: 0x12C for TIMER3
Address offset: 0x138 for TIMER4
Address offset: 0x144 for TIMER7
Address offset: 0x15C for TIMER19
Reset value: 0x0000 0000

TSCFG0[4:0], TSCFG1[4:0]..TSCFG14[4:0] are mutually exclusive and cannot be configured at the same time.

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											TSCFG15[4:0]				
													rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	TSCFG14[4:0]					TSCFG13[4:0]					TSCFG12[4:0]				

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: Reserved 00111: Reserved 01000: Reserved 01001: Internal trigger input 4 (ITI4) 01010: Internal trigger input 5 (ITI5) 01011: Internal trigger input 6 (ITI6) 01100: Internal trigger input 7 (ITI7) 01101: Internal trigger input 8 (ITI8) 01110: Internal trigger input 9 (ITI9) 01111: Internal trigger input 10 (ITI10) 10000: Reserved 10001: Reserved 10010: Reserved 10011: Internal trigger input 14 (ITI14) Others: Reserved
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 CI1FE1 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

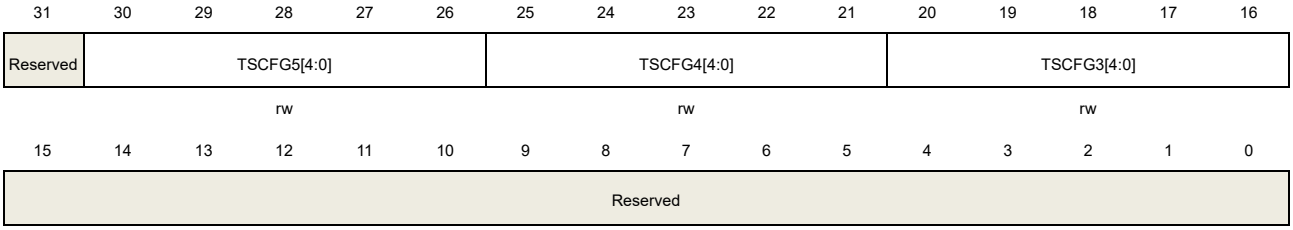
		CI0FE0 level.
4:0	TSCFG12[4:0]	Decoder mode 3 configuration 00000: disable Others: the counter will count on rising or falling edge of CI0FE0 and CI1FE1 signals. When CHxP=0, the counter will counter on the high level or the falling edge or the CIxFEx signal; When CHxP=1, the counter will counter on the low level or the rising edge or the CIxFEx signal.

1.7.24. TIMERx configuration register 0 (SYSCFG_TIMERxCFG0, x=14)

Address offset: 0x148 for TIMER14
Reset value: 0x0000 0000

TSCFG3[4:0], TSCFG4[4:0]..TSCFG7[4:0] are mutually exclusive and cannot be configured at the same time.

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



Bits	Fields	Descriptions
31	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: Reserved 01001: Internal trigger input 4 (ITI4) 01010: Internal trigger input 5 (ITI5) 01011: Internal trigger input 6 (ITI6) 01100: Internal trigger input 7 (ITI7) 01101: Internal trigger input 8 (ITI8) 01110: Internal trigger input 9 (ITI9)

		01111: Internal trigger input 10 (ITI10) 10000: Reserved 10001: Reserved 10010: Reserved 10011: Internal trigger input 14 (ITI14) Others: Reserved
25:21	TSCFG4[4:0]	Pause mode configuration The trigger input enables the counter clock when it is high and disables the counter when it is low when these bits are not 0. 00000: Pause mode disable 00001: Internal trigger input 0 (ITI0) 00010: Internal trigger input 1 (ITI1) 00011: Internal trigger input 2 (ITI2) 00100: Internal trigger input 3 (ITI3) 00101: Reserved 00110: The filtered output of channel 0 input (CI0FE0) 00111: The filtered output of channel 1 input (CI1FE1) 01000: Reserved 01001: Internal trigger input 4 (ITI4) 01010: Internal trigger input 5 (ITI5) 01011: Internal trigger input 6 (ITI6) 01100: Internal trigger input 7 (ITI7) 01101: Internal trigger input 8 (ITI8) 01110: Internal trigger input 9 (ITI9) 01111: Internal trigger input 10 (ITI10) 10000: Reserved 10001: Reserved 10010: Reserved 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) 01000: Reserved

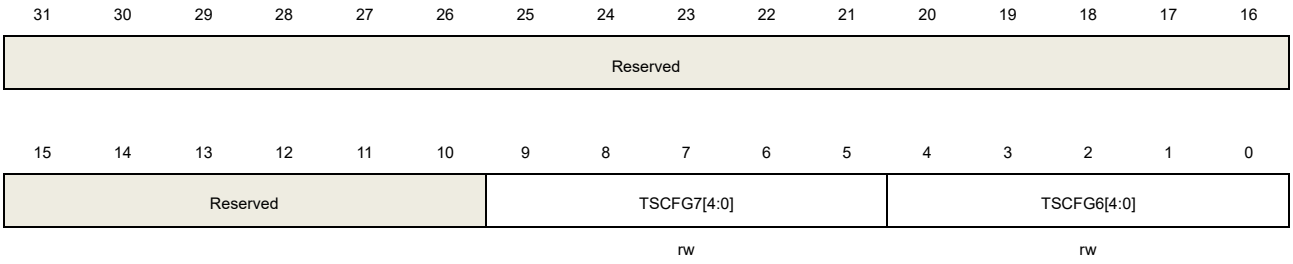
		01001: Internal trigger input 4 (ITI4) 01010: Internal trigger input 5 (ITI5) 01011: Internal trigger input 6 (ITI6) 01100: Internal trigger input 7 (ITI7) 01101: Internal trigger input 8 (ITI8) 01110: Internal trigger input 9 (ITI9) 01111: Internal trigger input 10 (ITI10) 10000: Reserved 10001: Reserved 10010: Reserved 10011: Internal trigger input 14 (ITI14) Others: Reserved
15:0	Reserved	Must be kept at reset value.

1.7.25. TIMERx configuration register 1 (SYSCFG_TIMERxCFG1, x=14)

Address offset: 0x14C for TIMER14
Reset value: 0x0000 0000

TSCFG3[4:0], TSCFG4[4:0]..TSCFG7[4:0] are mutually exclusive and cannot be configured at the same time.

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



Bits	Fields	Descriptions
31:10	Reserved	Must be kept at reset value.
9:5	TSCFG7[4:0]	Restart + event mode configuration The counter is reinitialized and started, the shadow registers are updated on the rising edge of the selected trigger input when these bits are not 0. 00000: Restart + 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: Reserved 01001: Internal trigger input 4 (ITI4) 01010: Internal trigger input 5 (ITI5) 01011: Internal trigger input 6 (ITI6) 01100: Internal trigger input 7 (ITI7) 01101: Internal trigger input 8 (ITI8) 01110: Internal trigger input 9 (ITI9) 01111: Internal trigger input 10 (ITI10) 10000: Reserved 10001: Reserved 10010: Reserved 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) 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: Reserved 01001: Internal trigger input 4 (ITI4) 01010: Internal trigger input 5 (ITI5) 01011: Internal trigger input 6 (ITI6) 01100: Internal trigger input 7 (ITI7) 01101: Internal trigger input 8 (ITI8) 01110: Internal trigger input 9 (ITI9) 01111: Internal trigger input 10 (ITI10) 10000: Reserved 10001: Reserved 10010: Reserved 10011: Internal trigger input 14 (ITI14) Others: Reserved

1.7.26. TIMERx configuration register 2 (SYSCFG_TIMERxCFG2, x=14)

Address offset: 0x150 for TIMER14

Reset value: 0x0000 0000

TSCFG3[4:0], TSCFG4[4:0]..TSCFG7[4:0] are mutually exclusive and cannot be configured at the same time.

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											TSCFG15[4:0]				
											rw				

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

Bits	Fields	Descriptions
31:21	Reserved	Must be kept at reset value.
20:16	TSCFG15[4:0]	Internal trigger input source configuration 00000: 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: Reserved 00111: Reserved 01000: Reserved 01001: Internal trigger input 4 (ITI4) 01010: Internal trigger input 5 (ITI5) 01011: Internal trigger input 6 (ITI6) 01100: Internal trigger input 7 (ITI7) 01101: Internal trigger input 8 (ITI8) 01110: Internal trigger input 9 (ITI9) 01111: Internal trigger input 10 (ITI10) 10000: Reserved 10001: Reserved 10010: Reserved 10011: Internal trigger input 14 (ITI14) Others: Reserved
15:0	Reserved	Must be kept at reset value.

1.8. Device electronic signature

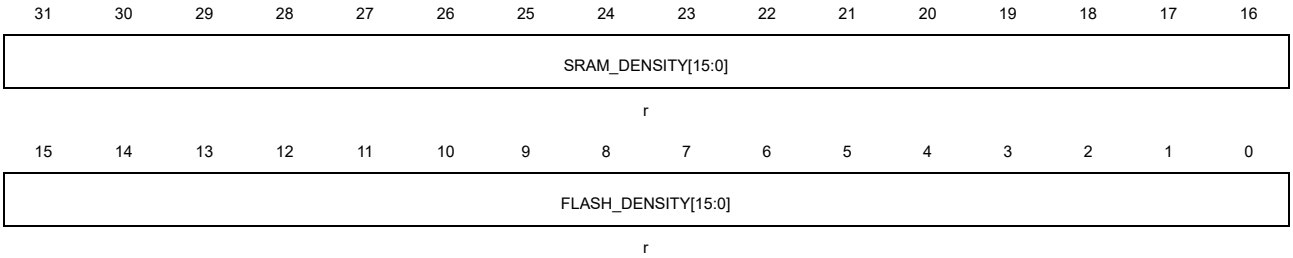
The device electronic signature contains memory density information and the 96-bit unique device ID. It is stored in the information block of the Flash memory. The 96-bit unique device ID is unique for any device. It can be used as serial numbers, or part of security keys, etc.

1.8.1. Memory density information

Base address: 0x1FFF B3E0

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

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



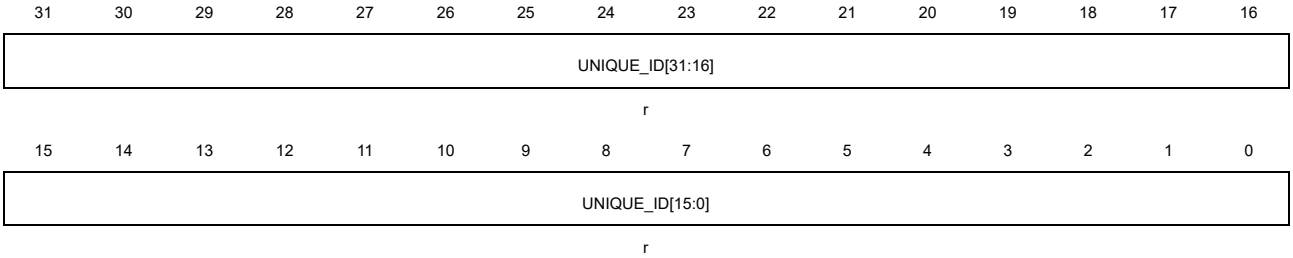
Bits	Fields	Descriptions
31:16	SRAM_DENSITY [15:0]	SRAM density The value indicates the on-chip SRAM density of the device in Kbytes. Example: 0x0008 indicates 8 Kbytes.
15:0	FLASH_DENSITY [15:0]	Flash memory density The value indicates the Flash memory density of the device in Kbytes. Example: 0x0020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

1.8.2. Unique device ID (96 bits)

Base address: 0x1FFF B3E8

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

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



Bits	Fields	Descriptions

31:0	UNIQUE_ID[31:0]	Unique device ID

Base address: 0x1FFF B3EC

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

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

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

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

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

Base address: 0x1FFF B3F0

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

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UNIQUE_ID[95:80]															
r															

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

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

2. Flash memory controller (FMC)

2.1. Overview

The flash memory controller, FMC, provides all the necessary functions for the on-chip flash memory. A little waiting time is needed while CPU executes instructions stored from the 512K bytes of the flash. It also provides page erase, mass erase, and program operations for flash memory.

2.2. Characteristics

- Up to 512KB of on-chip flash memory for instruction and data. Up to 2KB OTP.
  - bank0: 256KB.
  - bank1: 256KB.
  - OTP: 2KB.
  - Bootloader: 2 X 13KB.
  - Single bank selection DBS=0: read width of 128 bits, page size of 2 KB.
  - Dual bank selection DBS=1: read width of 64 bits, page size of 1 KB.
- ECC with single bit error correction and double bit errors detection.
- 0~7 waiting time within bank0 / bank1 when CPU executes instructions and read data.
- Pre-fetch buffer to speed read operations.
  - CBUS Instruction cache with 2K bytes which organized as 64 cache line of 4 X 64 bits or 2 X 128 bits.
  - CBUS data cache with 512 bytes which organized as 16 cache line of 4 X 64 or 2 X 128 bits.
- Dual bank architecture for read-while-write (RWW) capability.
- Double word programming, page erase and mass erase operation.
- 2KB OTP (one-time program) block used for user data storage.
- 2 X 48B option bytes block for user application requirements.
- Option bytes are uploaded to the option byte control registers when the system is reset.
- Flash security protection to prevent illegal code / data access.
- 4 erase / program protection areas (2 per bank when DBS=1 and 4 for full memory when DBS=0) to prevent unexpected operation.
  - Provides two execute-only dedicated code read protection (DCRP) area (1 per bank when DBS = 1, 2 for all memory when DBS = 0).
  - Provides two secure user area (1 per bank when DBS = 1, 1 for all memory when DBS = 0).
  - Low-power mode support.

2.3. Function overview

2.3.1. Flash memory architecture

The flash memory for dual bank consists of up to 512 KB main flash, which is organized into 2 X 256 pages with 1KB capacity, 2 X 13 KB information block for the boot loader. Each page of main flash memory can be erased individually. *Table 2-1. Base address and size for 512KB-dual bank flash memory* shows the base address and size for dual bank.

Table 2-1. Base address and size for 512KB-dual bank flash memory

Block		Name	Address	Size(bytes)
Main flash block	Bank0	Page 0	0x0800 0000 - 0x0800 03FF	1KB
		Page 1	0x0800 0400 - 0x0800 07FF	1KB
		Page 2	0x0800 0800 - 0x0800 0BFF	1KB
		. . .	. . .	. . .
		Page 255	0x0803 FC00 - 0x0803 FFFF	1KB
	Bank1	Page 0	0x0804 0000 - 0x0804 03FF	1KB
		Page 1	0x0804 0400 - 0x0804 07FF	1KB
		Page 2	0x0804 0800 - 0x0804 0BFF	1KB
		. . .	. . .	. . .
		Page 255	0x0807 FC00 - 0x0807 FFFF	1KB
Information block		Bank0 bootloader	0x1FFF 0000 - 0x1FFF 33FF	13KB
		Bank1 bootloader	0x1FFF 8000 - 0x1FFF B3FF	13KB
Option byte block		Option bytes 0	0x1FFF 7800~0x1FFF 782F	48B
		Option bytes 1	0x1FFF F800~0x1FFF F82F	48B
One-time program block		Bank 0 OTP bytes	0x1FFF 7000~0x1FFF 77FF	2KB
One-time program block for JTAG		Bank1 OTP JTAG bytes	0x1FFF F000~0x1FFF F00F	16B

Note: For 256KB products, from page 0 to page 127. For 128KB products, from page 0 to page 63.

For single-bank of 512 KB flash memory, the flash page is divided as shown in the following figure *Table 2-2. Base address and size for 512KB-single bank flash memory* after resetting DBS bit in the option byte.

Table 2-2. Base address and size for 512KB-single bank flash memory

Block	Name	Address	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 254	0x0807F000 - 0x0807 F7FF	2KB
	Page 255	0x0807 F800 - 0x0807 FFFF	2KB
Information block	Bank0 bootloader	0x1FFF 0000 - 0x1FFF 33FF	13KB
	Bank1 bootloader	0x1FFF 8000 - 0x1FFF B3FF	13KB
Option byte block	Option byte 0	0x1FFF 7800~0x1FFF 782F	48B
	Option byte 1	0x1FFF F800~0x1FFF F82F	48B
One-time program block	OTP bytes	0x1FFF 7000~0x1FFF 77FF	2KB
One-time program block for JTAG	Bank1 OTP JTAG bytes	0x1FFF F000~0x1FFF F00F	16B

Note: 1. For 256KB products, from page 0 to page 127. For 128KB products, from page 0 to page 63.

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

3. The one-time program block (OTP) has 2 KB (256 double word) to be written by user. This area is located in bank0. The OTP data can be written only once and cannot be erased. If any bit has been set 0, the whole double word cannot be written anymore.

2.3.2. Error Checking and Correcting (ECC)

The ECC mechanism supports:

- One error detection and correction
- Two errors detection

Dual bank selection(DBS = 1)

When one error is detected and corrected, the ECCCOR0 bit in FMC_ECCCS register will be set. If the ECCCORIE bit in FMC_ECCCS register is set, an interrupt is generated. The address is saved in ECCADDR [18:0] in FMC_ECCCS register. The ECC fail bank is saved in BK_ECC in FMC_ECCCS register.

When two errors are detected, the ECCDET0 bit in FMC_ECCCS register will be set and a NMI is generated. The address is saved in ECCADDR [18:0] in FMC_ECCCS register.

Note: If the bank is swapped, the ECC address will be reported according to the swapped address. For example, if access address 0x08000000 and in fact bank1 will be accessed, the ecc reports an error, and the error address stored is the address 0x08040000 accessed by user.

Single bank selection(DBS = 0)

The SYS_ECC flag bit must be checked firstly if DBS is reset. If SYS_ECC is 1, BK_ECC indicates that the ECC error of bootloader occurred in which bank.

When one error is detected and corrected in LSB bits (bits 63:0) ,the ECCCOR0 bit in FMC_ECCCS register will be set. When one error is detected and corrected in MSB bits (bits 127:64) ,the ECCCOR1 bit in FMC_ECCCS register will be set. If the ECCCORIE bit in FMC_ECCCS register is set, an interrupt is generated. The address is saved in ECCADDR [18:0] in FMC_ECCCS register.

When two errors are detected in LSB bits (bits 63:0), the ECCDET0 bit in FMC_ECCCS register will be set and a NMI is generated. When two errors are detected in MSB bits (bits 127:64), the ECCDET1 bit in FMC_ECCCS register will be set and a NMI is generated. The address is saved in ECCADDR [18:0] in FMC_ECCCS register.

Note: 1. Data in flash memory are saved as 72-bits words: 8 bits are added per double word (64 bits), but the added 8bits are calculated by hardware and cannot be accessed by user.

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

3. When a new ECC error occurs, ECCADDR and BK_ECC are not updated until ECCCOR0/ ECCCOR1 and ECCDET0/ ECCDET1 are cleared. After the bank swapped, the mapped address represented by the ECCADDR also changes.

4. When ECCDET0/ ECCDET1 is cleared, the FLASHECCIF bit in SYSCFG_STAT register will also be cleared. When the FLASHECCIF bit in SYSCFG_STAT register is cleared, the ECCDET0/ ECCDET1 will also be cleared.

5. Prefetching buffer or cache reading data or instructions can also generate ECC errors, even if the data or instructions are not being used by CPU.

6. When the original data programmed is all F, the ECC calculation will bypass this operation. Erasing is not required, and programming other values again can be written normally.

2.3.3. Read operations

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

Wait state added

The WSCNT bits in the FMC_WS register needs to be configured correctly depend on the

AHB clock frequency when reading the flash memory. The relation between WSCNT and AHB clock frequency is shown as the ***Table 2-3. The relation between WSCNT and AHB clock frequency when LDO is 1.1V***.

Table 2-3. The relation between WSCNT and AHB clock frequency when LDO is 1.1V

AHB clock frequency	WSCNT configured
<= 10MHz	0 (0 wait state added)
<= 20MHz	1 (1 wait state added)
<= 50MHz	2 (2 wait state added)
<= 70MHz	3 (3 wait state added)
<= 90MHz	4 (4 wait state added)
<= 120MHz	5 (5 wait state added)
<= 150MHz	6 (6 wait state added)
<= 170MHz	7 (7 wait state added)
<= 216MHz	7 (7 wait state added)

If system reset occurs, the AHB clock frequency is 8MHz and the WSCNT is 0. When the AHB clock frequency is greater than or equal to 170MHz, the WSCNT is 7 (7 wait states added).

Note:

1. If it is necessary to increase the AHB clock frequency, firstly, configure the WSCNT bits according to the target AHB clock frequency. Then, increase the AHB clock frequency to the target frequency. It is forbidden to increase the AHB clock frequency before configuring the WSCNT.

2. If it is necessary to decrease the AHB clock frequency, firstly, decrease the target AHB clock frequency. Then configure the WSCNT bits according the target AHB clock frequency. It is forbidden to configure the WSCNT bits before decrease the AHB clock frequency.

Considering that the wait state is added, the read efficiency is very low (such as add 7 wait state when 170MHz). In order to speed up the read access, there are some functions performed as below.

Current buffer

The current buffer is always enabled. Each time read from flash memory, 64-bits or 128-bits data will be got and stored in current buffer. The CPU only need 32-bits or 16-bits buffer in each read operation. So in the case of sequential code, the next data can be got from current buffer without repeat fetch from flash memory.

Pre-fetch buffer

The pre-fetch buffer is enabled by setting the PFEN bit in the FMC_WS register.

In the case of sequential code, 32-bits needs at least 2 clocks and 16-bit needs at least 4 clocks when the CPU executes the current buffer data (64-bits). In this case, the data of the

next double-word address is prefetched from the flash memory and stored in the pre-fetch buffer. So when the CPU finishes the current buffer and needs to execute the next instruction, the pre-fetch buffer hits.

ICODE cache

Instruction cache is enabled by set the ICEN bit in the FMC_WS register. The cache has 2K bytes, and it is organized by 64 cache lines. Each cache line has 4 x 64 bits or 2 x 128 bits.

If the data is in the instruction cache (cache hit), the CPU read data from cache without any wait state. If the data is not in the instruction cache (cache miss), nor in current buffer / pre-fetch buffer, the cache line fetch data from the flash memory and copied it into the instruction cache. If all cache line filled, LRU (least recently used) policy is used to replace the cache line.

DCODE cache

Data cache is enabled by set the DCEN bit in the FMC_WS register. The cache has 512 bytes, and it is organized by 16 cache lines. Each cache line has 4 x 64 bits or 2 x 128 bits.

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

Note: If modify the FMC_SWP bit in SYSCFG_CFG0 register while data cache is enabled, CPU may read incorrect flash data in the first cycle, which is related to accessing the flash sequence. It is recommended to delay reading the flash for two AHB periods to avoid this case. If dcache is disabled, read flash data is normal.

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

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

2.3.5. 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 FMC_CTL register is 1. An unlocking sequence consists of two write operations to the FMC_KEY register can open the access to the FMC_CTL register. The two write operations are writing 0x45670123 and 0xCDEF89AB to the FMC_KEY register. After the two write operations, the LK bit in FMC_CTL register is cleared by hardware. The software can lock the FMC_CTL again by setting the LK bit in FMC_CTL register. If there are some wrong operations on the

FMC_KEY register, the LK bit in FMC_CTL register will be set, and the FMC_CTL register will be locked, then it will generate a bus error.

The FMC_OBCTL register, OBRLD bit and OBSTART bit in FMC_CTL are protected by FMC_OBKEY register. The unlocking sequence includes two write operations, which are orderly writing 0x0819 2A3B and 0x4C5D 6E7F 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, OBRLD bit and OBSTART bit in FMC_CTL register again.

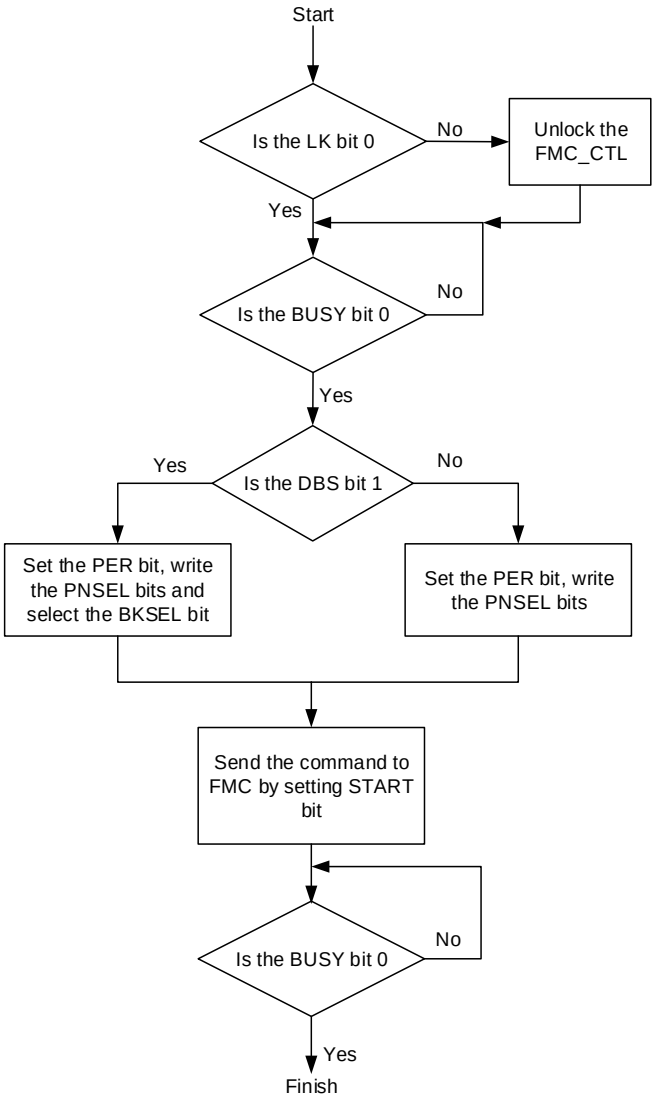
2.3.6. Page erase

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

- Unlock the FMC_CTL register if necessary.
- Check the BUSY bit in FMC_STAT register to confirm that no flash memory operation is in progress (BUSY equal to 0). Otherwise, wait until the operation has been finished.
- In dual bank case (DBS bit is set), the PER bit should be set firstly. Then and select the page to erase (PNSEL) and the bank where the page located (BKSEL) in the control register (FMC_CTL). In single bank case (DBS bit is reset), set the PER bit and select the page to erase (PNSEL). The BKSEL bit in the control register (FMC_CTL) must be kept cleared.
- Send the START command to the FMC by setting the START bit in FMC_CTL register.
- Wait until all the operations have been completed by checking the value of the BUSY bit in FMC_STAT register.
- Read and verify the page if required using a CBUS access.

When the operation is executed successfully and the operation end interrupt is enabled (ENDIE = 1), the ENDF in FMC_STAT register will be set. 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 for fetching codes or accessing data. The FMC will not provide any notification when it occurs. Additionally, the page erase operation will be ignored on protected pages. Flash operation error interrupt will be triggered by the FMC if the ERRIE bit in the FMC_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 ***Figure 2-1. Process of page erase operation*** shows the page erase operation flow.

Figure 2-1. Process of page erase operation



2.3.7. Mass erase

The FMC provides a complete erase function which is used to initialize the main flash block contents. This erase can affect only on Bank0 by setting MER0 bit to 1, or only on Bank1 by setting MER1 bit to 1, or on entire flash by setting MER0 and MER1 bits to 1. The following steps show the mass erase register access sequence.

- Unlock the FMC_CTL register if necessary.
- Check the BUSY bit in FMC_STAT register to confirm that no flash memory operation is in progress (BUSY equals to 0). Otherwise, wait until the operation has finished.
- Set MER0 bit in FMC_CTL register if erase Bank0 only. Set MER1 bit in FMC_CTL register if erase Bank1 only. Set MER0 and MER1 bits in FMC_CTL register if erase entire flash.
- Send the mass erase command to the FMC by setting the START bit in FMC_CTL

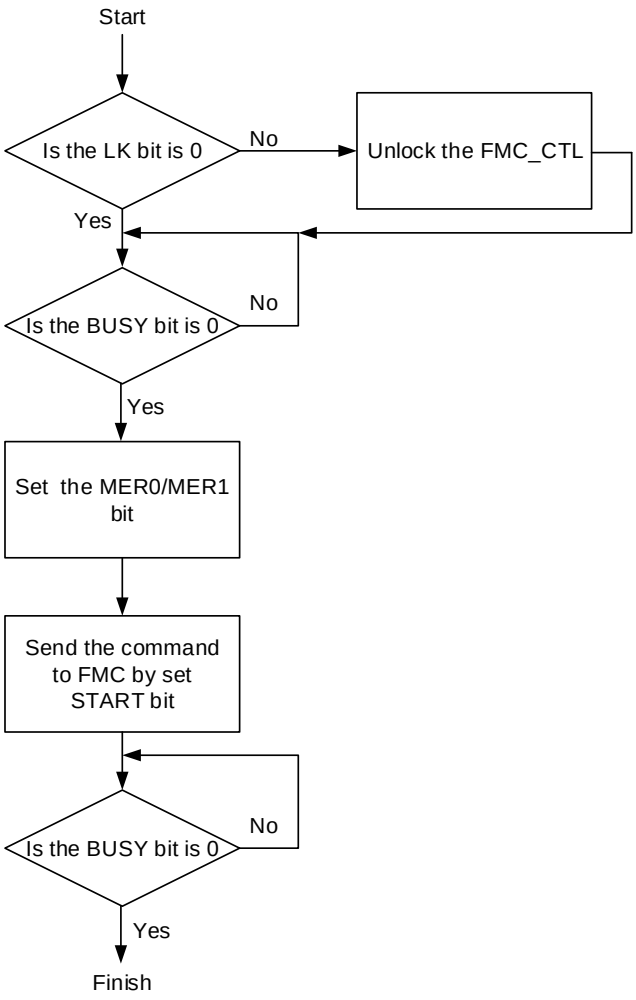
register.

- Wait until all the operations have been finished by checking the value of the BUSY bit in FMC_STAT register.
- Read and verify the flash memory if required using a DBUS access.

When the operation is executed successfully and the operation end interrupt is enabled (ENDIE = 1), the END in 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 by using the debugging tool that accesses the FMC registers directly.

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

Figure 2-2. Process of mass erase operation



2.3.8. Main flash programming

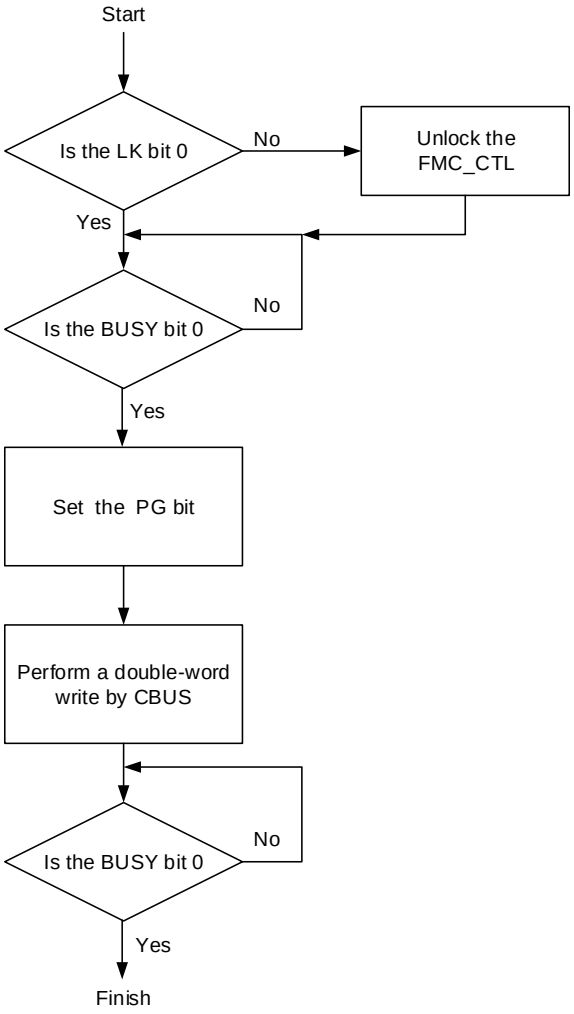
The FMC provides a double-word programming (2 x 32 bits + 8 bits ECC) function which is used to modify the main flash memory. The following steps show the register access sequence of the programming operation.

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

When the operation is executed successfully and the operation end interrupt is enabled (ENDIE = 1), the ENDF in 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 double-word programming operation you should check the address that it has been erased. If the address has not been erased, PGERR bit will set when programming the address, except if programming 0x0. Additionally, the program operation will be ignored on protected pages. The program operation will be ignored on protected pages and WPERR bit in FMC_STAT is set. 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, PGSERR, PGMERR, PGAERR and RPERR bits in the FMC_STAT register to detect which condition occurred in the interrupt handler.

The ***Figure 2-3. Process of double-word program operation*** displays the programming operation flow.

Figure 2-3. Process of double-word program operation



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

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

If the program / erase operation is interrupted by a power down, reset, 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.

2.3.9. OTP programming

The OTP programming operation flow is as same as the main flash programming. The OTP block can only be programed 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 very

little probability of corrupting the data stored in flash memory.

2.3.10. Option bytes

Option bytes description

The option bytes registers are reloaded to relevant block of flash memory after each system reset or OBRLD bit set in FMC_CTL register, and the option bytes work. The option complement bytes are the opposite of option bytes. When option bytes reload, if the option complement bytes do not match the option bytes, the OBERR bit in FMC_STAT register is set. The ***Table 2-4. Option byte*** is the detail of option bytes.

Table 2-4. Option byte

Address	Name	Description
0x1fff 7800	SPC	option byte Security Protection Code 0xAA: No protection 0xCC: Protection level high Any value except 0xAA or 0xCC: Protection level low.
0x1fff 7801	OB_USER[7:0]	[7]: FMC_SWP FMC memory mapping swap 0: Flash bank0 is mapped at 0x0804 0000 and flash bank1 is mapped at 0x0800 0000 1: Flash bank0 is mapped at 0x0800 0000 and flash bank1 is mapped at 0x0804 0000 [6]: reserved [5]: nRST_STDBY 0: Generates a reset if entering standby mode 1: No reset if entering standby mode [4]: nRST_DPSLP 0: Generates a reset if entering Deep-sleep mode 1: No reset if entering Deep-sleep mode [3:2]: reserved [1:0]: BOR_TH 00: No BOR function 01: BOR value 1. Reset value threshold is around 2.2 V 10: BOR value 2. Reset value threshold is around 2.5 V 11: BOR value 3. Reset value threshold is around 2.8 V
0x1fff 7802	OB_USER[15:8]	[7]: nBOOT1 0: BOOT1 is 1 1: BOOT1 is 0 Together with the BOOT0 pin, this bit selects boot mode [6]: DBS 0: Single bank mode with 128 bits width 1: Dual bank mode with 64 bits width

Address	Name	Description
		This bit can only be written when DCRP0/1 is disabled. [5]: reserved [4]: BB 0: Boot from bank0, when configured boot from main memory 1: Boot from bank1 or bank0 if bank1 is void, (or Bootloader continues executing if bank1 and bank0 are both void, and the chip is not under security protection level high,) when configured boot from main memory. [3]: reserved [2]: FWDGSPD_STDBY 0: FWDGT is suspend in system standby mode 1: FWDGT is running in system standby mode [1]: FWDGSPD_DPSLP 0: FWDGT is suspend in system deepsleep mode 1: FWDGT is running in system deepsleep mode. [0]: nFWDG_HW 0: Hardware free watchdog 1: Software free watchdog
0x1fff 7803	OB_USER[23:16]	[7:6]: reserved [5:4]: NRST_MDSEL 00: NRST pin configure as input/output mode. 01: A low level on the NRST pin can reset system, internal reset cannot drive NRST pin. 10: NRST pin function as normal GPIO, and only internal RESET. 11: NRST pin configure as input/output mode. [3]: nBOOT0 0: BOOT0 is 1 1: BOOT0 is 0 [2]: nSWBT0 0: BOOT0 depends on the option bit nBOOT0 1: BOOT0 depends on PB8/BOOT0 pin [1]: TCMSRAM_ERS 0: TCM SRAM erased if a system reset occurs 1: TCM SRAM is not erased if a system reset occurs [0]: SRAM_ECCEN 0: SRAM and TCM SRAM ECC enable 1: SRAM and TCM SRAM ECC disable
0x1fff 7804	SPC_N	SPC complement value
0x1fff 7805	OB_USER_N[7:0]	OB_USER complement value bit 7 to 0
0x1fff 7806	OB_USER_N[15:8]	OB_USER complement value bit 15 to 8
0x1fff 7807	OB_USER_N[23:16]	OB_USER complement value bit 23 to 16

Address	Name	Description
0x1fff 7808	DCRP_SADDR0[7:0]	DCRP area start address for bank0
0x1fff 7809	DCRP_SADDR0[14:8]	[7]: reserved [6:0]: DCRP area start address for bank0
0x1fff 780c	DCRP_SADDR0_N[7:0]	DCRP_SADDR0 complement value bit 7 to 0
0x1fff 780d	DCRP_SADDR0_N[14:8]	[7]: reserved [6:0]: DCRP_SADDR0 complement value bit 14 to 8
0x1fff 7810	DCRP_EADDR0[7:0]	DCRP area end address for bank0
0x1fff 7811	DCRP_EADDR0[14:8]	[7]: reserved [6:0]: DCRP area end address for bank0
0x1fff 7813	DCRP0_EREN	[7]: DCRP0_EREN 0: DCRP is not erased when a SPC value is decreased from low protection to no protection 1: DCRP is erased when a SPC value is decreased from low protection to no protection [6:0]: reserved
0x1fff 7814	DCRP_EADDR0_N[7:0]	DCRP_EADDR0 complement value bit 7 to 0
0x1fff 7815	DCRP_EADDR0_N[14:8]	[7]: reserved [6:0]: DCRP_EADDR0 complement value bit 14 to 8
0x1fff 7817	DCRP0_EREN_N	DCRP0_EREN complement value bit 7
0x1fff 7818	BK0WP0_SADDR[7:0]	DBS=1 BK0WP0_SADDR[7:0] contains the first page of WP first area in bank0. DBS=0 BK0WP0_SADDR[7:0] contains the first page of WP first area for all memory
0x1fff 781a	BK0WP0_EADDR[7:0]	DBS=1 BK0WP0_EADDR[7:0] contains the last page of WP first area in bank0. DBS=0 BK0WP0_EADDR[7:0] contains the last page of WP first area for all memory
0x1fff 781c	BK0WP0_SADDR_N[7:0]	BK0WP0_SADDR complement value bit 7 to 0
0x1fff 781e	BK0WP0_EADDR_N[7:0]	BK0WP0_EADDR complement value bit 7 to 0
0x1fff 7820	BK0WP1_SADDR[7:0]	DBS=1 BK0WP1_SADDR[7:0] contains the first page of WP second area in bank0. DBS=0 BK0WP1_SADDR[7:0] contains the first page of WP second area for all memory
0x1fff 7822	BK0WP1_EADDR[7:0]	DBS=1

Address	Name	Description
		BK0WP1_EADDR[7:0] contains the last page of WP second area in bank0. DBS=0 BK0WP1_EADDR[7:0] contains the last page of WP second area for all memory
0x1fff 7824	BK0WP1_SADDR_N[7:0]	BK0WP1_SADDR complement value bit 7 to 0
0x1fff 7826	BK0WP1_EADDR_N[7:0]	BK0WP1_EADDR complement value bit 7 to 0
0x1fff 7828	SCR_PAGE_CNT0[7:0]	[7:0]: Configure the number of pages in the bank 0 secure-user area.
0x1fff 7829	SCR_PAGE_CNT0[8]	[0]: SCR_PAGE_CNT0[8]
0x1fff 782a	BOOTLK	[0]: This bit is set to force boot from user flash area. 0: Support flash, ram and system boot 1: Only boot from main flash
0x1fff 782c	SCR_PAGE_CNT0_N[7:0]	SCR_PAGE_CNT0 complement value bit 7 to 0
0x1fff 782d	SCR_PAGE_CNT0_N[8]	SCR_PAGE_CNT0 complement value bit 8
0x1fff 783e	BOOTLK_N	BOOTLK complement value bit 0
0x1fff f808	DCRP_SADDR1[7:0]	DCRP area start address for bank1
0x1fff f809	DCRP_SADDR1[14:8]	[7]: reserved [6:0]: DCRP area start address for bank1
0x1fff f80c	DCRP_SADDR1_N[7:0]	DCRP_SADDR1 complement value bit 7 to 0
0x1fff f80d	DCRP_SADDR1_N[14:8]	[7]: reserved [6:0]: DCRP_SADDR1 complement value bit 14 to 8
0x1fff f810	DCRP_EADDR1[7:0]	DCRP area end address for bank1
0x1fff f811	DCRP_EADDR1[14:8]	[7]: reserved [6:0]: DCRP area end address for bank1
0x1fff f814	DCRP_EADDR1_N[7:0]	DCRP_EADDR1 complement value bit 7 to 0
0x1fff f815	DCRP_EADDR1_N[14:8]	[7]: reserved [6:0]: DCRP_EADDR1 complement value bit 14 to 8
0x1fff f818	BK1WP0_SADDR[7:0]	DBS=1 BK1WP0_SADDR[7:0] contains the first page of WP first area in bank1. DBS=0 BK1WP0_SADDR[7:0] contains the first page of WP third area for all memory
0x1fff f81a	BK1WP0_EADDR[7:0]	DBS=1 BK1WP0_EADDR[7:0] contains the last page of WP first area in bank1. DBS=0

Address	Name	Description
		BK1WP0_EADDR[7:0] contains the last page of WP third area for all memory
0x1fff f81c	BK1WP0_SADDR_N[7:0]	BK1WP0_SADDR complement value bit 7 to 0
0x1fff f81e	BK1WP0_EADDR_N[7:0]	BK1WP0_EADDR complement value bit 7 to 0
0x1fff f820	BK1WP1_SADDR[7:0]	DBS=1 BK1WP1_SADDR[7:0] contains the first page of WP second area in bank1. DBS=0 BK1WP1_SADDR[7:0] contains the first page of WP fourth area for all memory
0x1fff f822	BK1WP1_EADDR[7:0]	DBS=1 BK1WP1_EADDR[7:0] contains the last page of WP second area in bank1. DBS=0 BK1WP1_EADDR[7:0] contains the last page of WP fourth area for all memory
0x1fff f824	BK1WP1_SADDR_N[7:0]	BK1WP1_SADDR complement value bit 7 to 0
0x1fff f826	BK1WP1_EADDR_N[7:0]	BK1WP1_EADDR complement value bit 7 to 0
0x1fff f828	SCR_PAGE_CNT1[7:0]	[7:0]: Configure the number of pages in the bank 1 secure-user area bit 7 to 0.
0x1fff f829	SCR_PAGE_CNT1[8]	[0]: SCR_PAGE_CNT1[8]
0x1fff f82c	SCR_PAGE_CNT1_N[7:0]	SCR_PAGE_CNT1 complement value bit 7 to 0
0x1fff f82d	SCR_PAGE_CNT1_N[8]	SCR_PAGE_CNT1 complement value bit 8

Option bytes modify

The following steps show the modify 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 to 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.
  - Launch a system power on / down reset (or exit from Standby mode) or set the OBRLD bit in FMC_CTL register to load the option bytes.

Note:

1. Once a modification of one option byte is performed, the user options bytes of the two banks will be automatically erased first. When the operation is executed successfully, the ENDF in FMC_STAT register is set, and an interrupt will be triggered by FMC if the ENDIE bit in the FMC_CTL register is set.

2. Bits 63:32 is the complement byte of Bits 31:0. Every option bit has its complement bits in the same word. When option byte loading, the option bytes are checked by ECC. The option byte can be written into the corresponding register only when the option byte matches with its complement.

3. If the option byte does not match with its complement, the OBERR bit in FMC_STAT register is set. All the OB_USER register bytes are forced at 0xFF, except for BOR_TH which is at 0b000. The level of security protection is low. The status of WP pages is “No protection”. The status of DCRP is “All area protected”.

Switching dual / single bank mode

It is strongly recommended to execute the code from the SRAM when switching from one flash mode to the other. The data in the flash must be reprogrammed because it is corrupted after changing the bank mode. The following steps show the switch sequence.

- If the instruction cache, the data cache and prefetch are enabled, reset the ICEN, DCEN and PFEN bits in the FMC_WS register.
- Set the DCRST bit and the ICRST bit in the FMC_WS register to flush the instruction and data cache.
- Disable all the WP pages and set or reset the DBS bit. Set the OBSTART bit in the FMC_CTL register to send the option byte change command. Then set the OBRLD bit to load the option bytes.
- Erase the whole mass after reloading the option bytes. Reprogram the code and set the desired WP pages. Then set the ICEN, DCEN and PFEN bits in the FMC_WS register if necessary.

2.3.11. Dedicated code read protection area (DCRP)

In the main flash block, FMC can define executable-only areas, allowing only instruction transactions from the system, but not data access. It allows to specify one DCRP area per bank when the DBS bit is set or to specify two different DCRP areas when the DBS bit is reset.

Note: Users need to compile their native code accordingly, using the execution-only option when apply execute-only area function.

The DCRP area are defined by a start address offset and an end address offset. If single bank mode is selected, the minimum DCRP area is 32 bytes. The DCRP areax (x= 0,1) is defined by:

- From bank base address + [FMC_DCRP_SADDRx x 16] (include) to the address: bank base address + [(FMC_DCRP_EADDRx + 1) x 16] (exclude).

The DCRP area are defined by a start address offset and an end address offset. If dual bank mode is selected, the minimum DCRP area is 16 bytes. The DCRP areax (x= 0,1) is defined by:

- From bank base address + [FMC_DCRP_SADDRx x 8] (include) to the address: bank base address + [(FMC_DCRP_EADDRx + 1) x 8] (exclude).

For example, to protect by DCRP from the address 0x0802 6B40 (included) to the address 0x0803 2004 (included) while dual bank mode is selected (DBS = 1), the option bytes must be set as follows:

- FMC_DCRP_SADDR0 = 0x4D68.
- FMC_DCRP_EADDR0 = 0x6400.

If bank0 and bank1 are swapped, the option bytes must be set as follows:

- FMC_DCRP_SADDR1 = 0x4D68.
- FMC_DCRP_EADDR1 = 0x6400.

When executing code in this area, the debug events will be ignored. Only CPU can access DCRP area, using only instruction fetch transactions. In all other cases, access to the DCRP area is illegal. For example, read operations will trigger a RPERR flag in FMC_STAT register, and write operations will be ignored and a WPERR in FMC_STAT register is set.

A valid DCRP area is erase-protected. Pages which located in this area cannot be erased. If a valid DCRP area is setting, mass erase cannot be performed unless erase is performed during the SPC level low to no protection.

Only CPU can modify the DCRP area definition bits and DCRP_EREN bit. If DCRP area is valid (not empty), during SPC level low to no protection demotion, the DCRP_EREN bit is set to 0, the DCRP area will not be erased, otherwise the area will be erased.

NOTE: The only way to clear DCRP area is changing SPC level from low to no protection while the DCRP_EREN bit is set. Modifying the DCRP option bytes directly to reduce the DCRP region does not work, but increasing the DCRP region does.

2.3.12. Erase/program protection (WP)

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. 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_BK0WPx (x = 0,1) and FMC_BK1WPx (x = 0,1).

If single bank mode is selected (DBS = 0), four WP areas can be defined in the bank with a granularity of 2 Kbytes. The WP area is defined by:

- From bank base address + [(BKxWPy_SADDR [7:0]) x 0x800] (include) to the address: bank base address + [(BKxWPy_EADDR [7:0] + 1) x 0x800] (exclude).

If dual bank mode is selected (DBS = 1), two WP areas can be defined in each bank with a granularity of 1 Kbytes. The WP area is defined by:

- From bank base address + [(BKxWPy_SADDR [7:0]) x 0x400] (include) to the address: bank base address + [(BKxWPy_EADDR [7:0] + 1) x 0x400] (exclude).

For example, to protect the address from 0x0802 2800 (included) to 0x0803 07FF (included) while dual bank mode is selected (DBS = 1), the option bytes must be set as follows:

- BK0WP0_SADDR [7:0] = 0x8A.
- BK0WP0_EADDR [7:0] = 0xC1.

If bank0 and bank1 are swapped, the option bytes must be set as follows:

- BK1WP0_SADDR [7:0] = 0x8A.
- BK1WP0_SADDR [7:0] = 0xC1.

Erase/program protected pages cannot either be deleted or programmed. Therefore, if a page is erase/program protected, mass erase cannot be performed.

If the security protection (SPC) level is set to high, the WP area cannot be modified, else WP area can be modified without any restrictions

Note: DCRP or secure user area is erase/program protected.

Table 2-5 WP protection

WP registers value （x = 0,1）	WP area
BKxWPy_SADDR = BKxWPy_EADDR	Page BKxWPy_SADDR is protected.
BKxWPy_SADDR > BKxWPy_EADDR	No WP area.
BKxWPy_SADDR < BKxWPy_EADDR	The pages from BKxWPy_SADDR to BKxWPy_EADDR are protected.

2.3.13. Security protection (SPC)

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

There are 3 levels for protection:

No protection: when setting SPC byte and its complement value to 0xAA55, no protection performed. The main flash and option bytes block are accessible by all operations.

Low level protection: when setting SPC byte value to any value except 0xAA or 0xCC, the low security protection is performed. Note that a power reset should be followed instead of a system reset if the SPC modification has been performed while the debug module is still connected to JTAG/SWD device. Under the low security protection, the main flash can only be accessed by user code. In debug mode, boot from SRAM or boot loader mode, all operations to main flash is forbidden. If a read operation, program/erase operation to main flash in debug mode, boot from SRAM or boot loader mode, a bus error will be generated. Option bytes block is accessible by all operations, which can be used to disable the security protection. Back to no protection level by setting SPC byte and its complement value to 0x5AA5, then a mass erase for main flash will be performed. If some TCMSRAM pages are protected, the whole TCMSRAM will be also erased. Note if low level protection is configured and no DCRP area is defined, it is mandatory to set DCRP_EREN bit.

High level protection: when set SPC byte and its complement value to 0xCC33, high level protection performed. When this level is programmed, debug mode, boot from SRAM or boot from boot loader mode are disabled. The main flash block is accessible by all operations from user code. The user option bits can be read but cannot be modified. And accesses to the other secured areas are also allowed. The SPC byte cannot be reprogrammed. So if high level protection is programmed, it cannot move back to protection level low or no protection level.

2.3.14. 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 secured (the SCR0 (or SCR1) bit of the FMC_CTL register set), read operations will set the RDERR bit and write operations will set the WPERR bit in FMC_STAT register.

The size of the secure user area is defined by the SCR_PAGE_CNT0[8:0] (or SCR_PAGE_CNT1[8:0]) bits of the FMC_BK0SCR (or FMC_BK1SCR) register. It can be modified only in SPC level no protection. Its content is erased upon changing from SPC low level protection to no protection, even if it overlaps with DCRP pages.

The secure user area is defined:

- If single bank mode is selected (DBS = 0), from bank base address to bank base address + [(SCR_PAGE_CNT0[8:0]) x 0x800] (excluded).
- If dual bank mode is selected (DBS = 1), from bank0 base address(included) to bank0

base address + [(SCR_PAGE_CNT0[8:0]) x 0x400] (excluded) and from bank1 base address(included) to bank1 base address + [(SCR_PAGE_CNT1[8:0]) x 0x400] (excluded).

2.3.15. Disabling core debug access

The debug access to the core can be disabled temporarily when executing sensitive code or accessing sensitive data in securable user area.

While the SPC level is no protection or low level protection, the debugger can be disabled by software set the bit DBGEN in the FMC_WS register.

2.3.16. Forcing boot from Flash memory

The BOOTLK bit in the FMC_BK0SCR register can be configured to force the system to boot from the Main Flash memory. This bit can be reset only when:

1. SPC level is no protection
2. SPC level is low, while no protection level is requested and a full mass-erase is performed.

2.3.17. FMC interrupts

The FMC interrupt events and flags are listed in ***Table 2-6 FMC interrupt requests***.

Table 2-6 FMC interrupt requests

Flag	Description	Clear method	Interrupt enable bit
ENDF	End of operation	Write 1 to corresponding bit in FMC_STAT register	ENDIE
OPRERR	Operation failure error		ERRIE
RPERR	Read protection error		RPERRIE
ECCCOR0/ ECCCOR1	ECC correction	Write 1 to corresponding bit in FMC_ECCCS register	ECCCORIE

2.4. Register definition

FMC base address: 0x4002 2000

2.4.1. Wait state register (FMC_WS)

Address offset: 0x00

Reset value: 0x0004 0600

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	SLP_MDSEL	RUN_MDSEL	DCRST	ICRST	DCEN	ICEN	PFEN	Reserved					WSCNT[3:0]		
	rw	rw	rw	rw	rw	rw	rw						rw		

Bits	Fields	Descriptions
31:19	Reserved	Must be kept at reset value
18	DBGEN	This bit is used to enable/disable the debugger by software 0: Debugger disabled 1: Debugger enabled
17:15	Reserved	Must be kept at reset value
14	SLP_MDSEL	Flash power-down mode during sleep mode This bit is used to determines that the flash is in power-down mode or idle mode when system enter sleep mode. 0: Flash enter idle mode during sleep mode 1: Flash enter power-down mode during sleep mode
13	RUN_MDSEL	Flash power-down mode during run mode This bit is used to determines that the flash is in power-down mode or idle mode when system enter run mode. The flash can be put in power-down mode only when the code is executed from RAM. The flash must not be accessed when RUN_MDSEL is set. This bit is write-protected with FMC_RUNKEY. 0: Flash enter Idle mode during run mode 1: Flash enter Power-down mode during run mode
12	DCRST	Data cache reset. This bit can be write only when DCEN is set to 0. 0: No effect 1: Data cache reset

11	ICRST	Instruction cache reset. This bit can be write only when ICEN is set to 0. 0: No effect 1: Instruction cache reset
10	DCEN	Data cache enable 0: Data cache disable 1: Data cache enable
9	ICEN	Instruction cache enable 0: Instruction cache disable 1: Instruction cache enable
8	PFEN	Pre-fetch enable 0: Pre-fetch disable 1: Pre-fetch enable
7:4	Reserved	Must be kept at reset value
3:0	WSCNT[3:0]	Wait state counter register These bits set and reset by software. 0000: 0 wait state added 0001: 1 wait state added 0010: 2 wait state added 0011: 3 wait state added 0100: 4 wait state added … 1111: 15 wait state added

2.4.2. Unlock flash mode during run mode key register (FMC_RUNKEY)

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

Bits	Fields	Descriptions
31:0	RUN_KEY[31:0]	RUN_MDSEL unlock register These bits can only be written by software. Write RUN_KEY[31:0] with keys to unlock RUN_ MDSEL bit in FMC_WS register. RUN_KEY1: 0x04152637

RUN_KEY2: 0xFAFBFCFD

2.4.3. Unlock key register (FMC_KEY)

Address offset: 0x08

Reset value: 0x0000 0000

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

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

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

2.4.4. Option byte unlock key register (FMC_OBKEY)

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

Bits	Fields	Descriptions
31:0	OBKEY[31:0]	FMC_CTL option bytes operation unlock register. These bits can only be written by software. Write OBKEY[31:0] with keys to unlock option bytes command in the FMC_CTL register. OBKEY1: 0x0819 2A3B OBKEY2: 0x4C5D 6E7F

2.4.5. Status register (FMC_STAT)

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															BUSY
															r

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

Bits	Fields	Descriptions
31:17	Reserved	Must be kept at reset value
16	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 generated, this bit is clear to 0.
15	OBERR	Option byte read error bit. This bit is set by hardware when the option byte and its complement byte do not match, and the option byte set 0xFF.
14	RPERR	Read protection error flag bit. When access address through the Cbus protected by DCRP or SCR, this bit is set by hardware. This bit can be cleared it by writing 1. 0: No read protection error occurs 1: Read protection error occurs
13:8	Reserved	Must be kept at reset value
7	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, PGMERR, PGAERR or WPERR because a previous programming error occurred. The bit can be cleared by writing 1. 0: No program sequence error occurs 1: Program sequence error occurs
6	PGMERR	Program size not match error flag bit. This bit is set by hardware when program size is a half-word/word access. The only correct programming size is double word. The software can clear it by writing 1.
5	PGAERR	Program alignment error flag bit This bit is set by hardware when CBUS write data is not alignment. If the data to program cannot be contained in the same 64-bit Flash memory row This bit is set by hardware. This bit can be cleared it by writing 1.
4	WPERR	Erase / program protection error flag bit.

		When an erase/program protection error occurs, this bit is set by hardware. The software can clear it by writing 1. 0: No write protection error occurs 1: Write protection error occurs
3	PGERR	Program error flag bit When programming to the flash while it is not 0xFFFF FFFF FFFF FFFF, this bit is set by hardware. This bit can be cleared it by writing 1.
2	Reserved	Must be kept at reset value
1	OPRERR	Operation error flag bit If a flash memory programming or erase operation completes unsuccessfully and error interrupts are enabled (ERRIE = 1), it is set by hardware. This bit can be cleared it by writing 1.
0	ENDF	End of operation flag bit When the operation executed successful and the operation end interrupt is enabled (ENDIE = 1), this bit is set by hardware. This bit can be cleared it by writing 1.

2.4.6. Control register (FMC_CTL)

Address offset: 0x14

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	SCR1	SCR0	OBRLD	RPERRIE	ERRIE	ENDIE	Reserved						OBSTART	START
rs	rs	rs	rs	rc_w1	rw	rw	rw							rs	rs
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MER1	Reserved		BKSEL	Reserved	PNSEL[7:0]								MER0	PER	PG
rw			rw		rw								rw	rw	rw

Bits	Fields	Descriptions
31	LK	FMC_CTL lock bit When set, the FMC_CTL register is locked 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	SCR1	Bank1 secure user area enable bit This bit is set to lock the secure user area for bank1. It is set by software when

		exiting the secure user area, and can only be written once. In case DBS=0, this bit is useless. 0: Disable secure user area for bank1 1: Enable secure user area for bank1
28	SCR0	Bank0 secure user area enable bit. This bit is set to lock the secure user area for bank 0 (or when DBS=0). It is set by software when exiting the secure user area, and can only be written once. 0: Disable secure user area for bank0 1: Enable secure user area for bank0
27	OBRLD	Option byte reload bit This bit is set by software. 0: Option byte reload complete 1: Force option byte reload **NOTE:**1. It cannot be written if OBLK is set. 2. When the OBSTART bit is set, it cannot be written to. If both OBSTART and OBRLD are written at the same time, neither bit will take effect.
26	RPERRIE	Read protection error interrupt enable bit This bit is set or cleared by software only when LK is set to 0 0: Disable read protection error interrupt 1: Enable read protection error interrupt
25	ERRIE	OPRERR error interrupt enable bit This bit is set or cleared by software. 0: Disable OPRERR error interrupt 1: Enable OPRERR error interrupt
24	ENDIE	End of operation interrupt enable bit This bit is set or cleared by software. 0: Disable end of operation interrupt 1: Enable end of operation interrupt
23:18	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 FMC bit This bit is set by software to send erase command to FMC. This bit is cleared by hardware when the BUSY bit is cleared. If MER0, MER1 and PER bits are reset and the START bit is set, an unpredictable behavior may occur without generating any error flag. This condition should be forbidden.

15	MER1	Main flash mass erase for bank1 command bit This bit is set or cleared by software. 0: No effect 1: Main flash mass erase command for bank1
14:13	Reserved	Must be kept at reset value
12	BKSEL	Bank number selection for page erase. DBS=1 0: Bank0 is selected for page erase. 1: Bank1 is selected for page erase. DBS=0 Reserved, must be at reset value.
11	Reserved	Must be kept at reset value
10:3	PNSEL[7:0]	Page number selection These bits select the page to erase: 00000000: page 0 00000001: page 1 ... 11111111: page 255
2	MER0	Main flash mass erase for bank0 command bit This bit is set or cleared by software. 0: No effect 1: Main flash mass erase command for bank0
1	PER	Main flash page erase command bit This bit is set or cleared by software. 0: No effect 1: Main flash page erase command
0	PG	Main flash page program command bit This bit is set or cleared by software. 0: Disabled flash programming 1: Enabled flash programming

2.4.7. ECC control and status register (FMC_ECCCS)

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
ECCDET0	ECCCOR0	ECCDET1	ECCCOR1	Reserved			ECCCORIE	Reserved	SYS_ECC	BK_ECC	Reserved		ECCADDR[18:16]		

rc_w1	rc_w1	rc_w1	rc_w1				rw		r	r					r	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
ECCADDR[15:0]																
r																

Bits	Fields	Descriptions
31	ECCDET0	Two bit errors detected flag. When DBS = 0: This bit set when two ECC errors have been detected in LSB bits (bits 63:0). The software can clear it by writing 1. When DBS = 1: This bit set when two ECC errors have been detected. The software can clear it by writing 1. 0: Two ECC errors are not detected. 1: Two ECC errors are detected.
30	ECCCOR0	One-bit error detected and corrected flag. When DBS = 0: This bit set when one ECC errors have been detected and corrected in LSB bits (bits 63:0). The software can clear it by writing 1. When DBS = 1: This bit set when one ECC errors have been detected and corrected. The software can clear it by writing 1. 0: No ECC error is detected and corrected. 1: An ECC error is detected and corrected.
29	ECCDET1	Two bit errors detected flag. When DBS = 0: This bit set when two ECC errors have been detected in MSB bits (bits 127:64). The software can clear it by writing 1. When DBS = 1: Reserved, must be kept at reset value. 0: Two ECC errors are not detected. 1: Two ECC errors are detected.
28	ECCCOR1	One-bit error detected and corrected flag. When DBS = 0: This bit set when one ECC errors have been detected and corrected in MSB bits (bits 127:64). The software can clear it by writing 1. When DBS = 1: Reserved, must be kept at reset value. 0: No ECC error is detected and corrected. 1: An ECC error is detected and corrected.
27:25	Reserved	Must be kept at reset value

24	ECCCORIE	One-bit error correction interrupt enable. 0: Disable one-bit error correction interrupt. 1: Enable one-bit error correction interrupt. This bit enables the interrupt generation when the ECCCOR bit is set.
23	Reserved	Must be kept at reset value
22	SYS_ECC	If an ECC error correction or double ECC error is detected in bootloader, this bit will be set. And the ECCADDR records the offset address of bootloader. 0: No ECC error correction or double ECC error is detected in bootloader. 1: An ECC error correction or double ECC error is detected in bootloader.
21	BK_ECC	ECC fail bank DBS=1 It indicates that the ECC error correction or the double ECC error detection occurred in which bank. 0: Bank 0 1: Bank 1 DBS=0 If SYS_ECC is 1, it indicates that the ECC error of bootloader occurred in which bank. If SYS_ECC is 0, reserved, must be kept cleared.
20:19	Reserved	Must be kept at reset value
18:0	ECCADDR[18:0]	ECC fail address DBS=0 It indicates that the ECC error correction or the double ECC error detection occurred at which address in the flash memory. DBS=1 It indicates that the ECC error correction or the double ECC error detection occurred at which address in the bank.

2.4.8. Option byte control register (FMC_OBCTL)

Address offset: 0x20

Reset value: 0xXXXX XXXX

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		NRST_MDSEL[1:0]		nBOOT0	nSWBT0	TCMSRAM_ERS	SRAM_ECCEN	nBOOT1	DBS	Reserved	BB	Reserved	FWDGSPD_STDBY	FWDGSPD_DPSLP	nFWDG_HW
		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
FMC_SWP	Reserved	nRST_STDBY	nRST_DPSLP	Reserved		BOR_TH[1:0]		SPC[7:0]							
rw		rw	rw			rw		rw							

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value
29:28	NRST_MDSEL[1:0]	NRST pin mode selection. 00: NRST pin configure as input/output mode. 01: A low level on the NRST pin can reset system, internal reset cannot drive NRST pin. 10: NRST pin function as normal GPIO, and only internal RESET. 11: NRST pin configure as input/output mode.
27	nBOOT0	BOOT0 option bit 0: BOOT0 is 1 1: BOOT0 is 0
26	nSWBT0	Software BOOT0 disable 0: BOOT0 depends on the option bit nBOOT0 1: BOOT0 depends on PB8/BOOT0 pin
25	TCMSRAM_ERS	TCM SRAM erase if system reset 0: TCM SRAM erased if a system reset occurs 1: TCM SRAM is not erased if a system reset occurs
24	SRAM_ECCEN	SRAM and TCM SRAM ECC disable 0: SRAM and TCM SRAM ECC enable 1: SRAM and TCM SRAM ECC disable
23	nBOOT1	Boot1 configuration bit 0: BOOT1 is 1 1: BOOT1 is 0 Together with the BOOT0 pin, this bit selects boot mode
22	DBS	Double banks or single bank selection. 0: Single bank mode with 128 bits width 1: Dual bank mode with 64 bits width This bit can only be written when DCRP0/1 is disabled.
21	Reserved	Must be kept at reset value
20	BB	Option byte boot bank value 0: Boot from bank0, when configured boot from main memory 1: Boot from bank1 or bank0 if bank1 is void, (or Bootloader continues executing if bank1 and bank0 are both void, and the chip is not under security protection level high,) when configured boot from main memory. **Note:** To set this bit, FMC_SWP must be set firstly.
19	Reserved	Must be kept at reset value
18	FWDGSPD_STDBY	FWDGT suspend option in standby mode configuration bit

		0: FWDGT is suspend in system standby mode 1: FWDGT is running in system standby mode
17	FWDGSPD_DPSLP	FWDGT suspend option in deepsleep mode configuration bit 0: FWDGT is suspend in system deepsleep mode 1: FWDGT is running in system deepsleep mode.
16	nFWDG_HW	Free watchdog configuration bit 0: Hardware free watchdog 1: Software free watchdog
15	FMC_SWP	FMC memory mapping swap This bit controls the address mapping swap between bank0 and bank1 of the main Flash. 0: Flash bank0 is mapped at 0x0804 0000 and Flash bank1 is mapped at 0x0800 0000 1: Flash bank0 is mapped at 0x0800 0000 and Flash bank1 is mapped at 0x0804 0000 **Note:** Depend on the specific series of bank size. If the BB bit is set, this bit cannot be reset.
14	Reserved	Must be kept at reset value
13	nRST_STDBY	Option byte standby reset value 0: Generates a reset if entering standby mode 1: No reset if entering standby mode
12	nRST_DPSLP	Option byte deepsleep reset value 0: Generates a reset if entering Deep-sleep mode 1: No reset if entering Deep-sleep mode
11:10	Reserved	Must be kept at reset value
9:8	BOR_TH[1:0]	BOR threshold status bits. 00: No BOR function 01: BOR value 1. Reset value threshold is around 2.2 V 10: BOR value 2. Reset value threshold is around 2.5 V 11: BOR value 3. Reset value threshold is around 2.8 V
7:0	SPC[7:0]	Security protection level option byte status bits 0xAA: No protection 0xCC: Protection level high Any value except 0xAA or 0xCC: Protection level low.

2.4.9. DCRP start address register 0(FMC_DCRP_SADDR0)

Address offset: 0x24

Reset value: 0xFFFF XXXX

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	DCRP0_SADDR [14:0]														
	rw														

Bits	Fields	Descriptions
31:15	Reserved	Must be kept at reset value
14:0	DCRP0_SADDR [14:0]	DCRP area start address offset configuration bits for bank0 DBS=1: DCRP0_SADDR contains the start address of the bank0 DCRP area. DBS=0: DCRP0_SADDR contains the start address of the first DCRP area for all memory.

2.4.10. DCRP end address register 0(FMC_DCRP_EADDR0)

Address offset: 0x28

Reset value: 0xX0FF 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
DCRP_EREN	Reserved														
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	DCRP0_EADDR [14:0]														
	rw														

Bits	Fields	Descriptions
31	DCRP_EREN	DCRP area erase enable configuration bit. This bit is set only. This bi can be reset only when SPC level is changing from low to no protection. 0: DCRP is not erased when SPC value is decreased from low to no protection. 1: DCRP is erased when SPC value is decreased from low to no protection.
30:15	Reserved	Must be kept at reset value
14:0	DCRP0_EADDR [14:0]	DCRP area end address offset configuration bits for bank0 DBS=1: DCRP0_EADDR contains the end address of the bank0 DCRP area. DBS=0: DCRP0_EADDR contains the end address of the first DCRP area for all memory.

2.4.11. Bank0 erase/program protection area 0 register (FMC_BK0WP0)

Address offset: 0x2C

Reset value: 0xFEXX FEXX

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								BK0WP0_EADDR[7:0]							
								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								BK0WP0_SADDR[7:0]							
								rw							

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value
23:16	BK0WP0_EADDR[7:0]	WP first area end offset DBS=1: BK0WP0_EADDR[7:0] contains the last page of the WP first area in bank0. DBS=0: BK0WP0_EADDR[7:0] contains the last page of the WP first area for all memory.
15:8	Reserved	Must be kept at reset value
7:0	BK0WP0_SADDR[7:0]	WP first area start offset DBS=1: BK0WP0_SADDR[7:0] contains the first page of the WP first area for bank0. DBS=0: BK0WP0_SADDR[7:0] contains the first page of the WP first area for all memory.

2.4.12. Bank0 erase/program protection area 1 register (FMC_BK0WP1)

Address offset: 0x30

Reset value: 0xFEXX FEXX

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								BK0WP1_EADDR[7:0]							
								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								BK0WP1_SADDR[7:0]							
								rw							

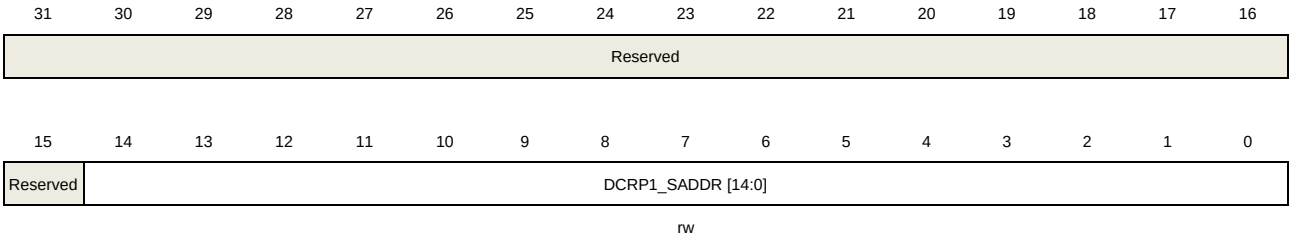
Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value
23:16	BK0WP1_EADDR[7:0]	WP second area end offset DBS=1: BK0WP1_EADDR[7:0] contains the last page of the WP second area for bank0. DBS=0: BK0WP1_EADDR[7:0] contains the last page of the WP second area for all memory.
15:8	Reserved	Must be kept at reset value
7:0	BK0WP1_SADDR[7:0]	WP second area start offset DBS=1: BK0WP1_SADDR[7:0] contains the first page of the WP second area for bank0. DBS=0: BK0WP1_SADDR[7:0] contains the first page of the WP second area for all memory

2.4.13. DCRP start address register 1(FMC_DCRP_SADDR1)

Address offset: 0x44

Reset value: 0xFFFF XXXX

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



Bits	Fields	Descriptions
31:15	Reserved	Must be kept at reset value
14:0	DCRP1_SADDR[14:0]	DCRP area start address offset configuration bits for bank1 DBS=1: DCRP1_SADDR contains the start address of the bank1 DCRP area. DBS=0: DCRP1_SADDR contains the start address of the second DCRP area for all memory.

2.4.14. DCRP end address register 1(FMC_DCRP_EADDR1)

Address offset: 0x48

Reset value: 0x00FF XXXX

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	DCRP1_EAREA [14:0]														
	rw														

Bits	Fields	Descriptions
31:15	Reserved	Must be kept at reset value
14:0	DCRP1_EADDR [14:0]	DCRP area end address offset configuration bits for bank1 DBS=1: DCRP1_EADDR contains the end address of the bank1 DCRP area. DBS=0: DCRP1_EADDR contains the end address of the second DCRP area for all memory.

2.4.15. Bank1 erase/program protection area 0 register (FMC_BK1WP0)

Address offset: 0x4C

Reset value: 0xFEXX FEXX

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								BK1WP0_EADDR[7:0]							
								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								BK1WP0_SADDR[7:0]							
								rw							

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value
23:16	BK1WP0_EADDR[7:0]	WP third area end offset DBS=1 BK1WP0_EADDR[7:0] contains the last page of the WP first area for bank1. DBS=0 BK1WP0_EADDR[7:0] contains the last page of the WP third area for all memory
15:8	Reserved	Must be kept at reset value
7:0	BK1WP0_SADDR[7:0]	WP third area start offset DBS=1 BK1WP0_SADDR[7:0] contains the first page of the WP first area for bank1. DBS=0

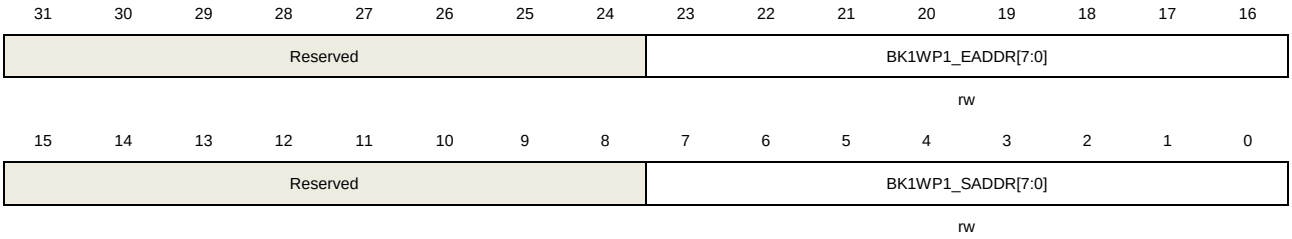
BK1WP0_SADDR[7:0] contains the first page of the WP third area for all memory.

2.4.16. Bank1 erase/program protection area 1 register (FMC_BK1WP1)

Address offset: 0x50

Reset value: 0xFEXX FEXX

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



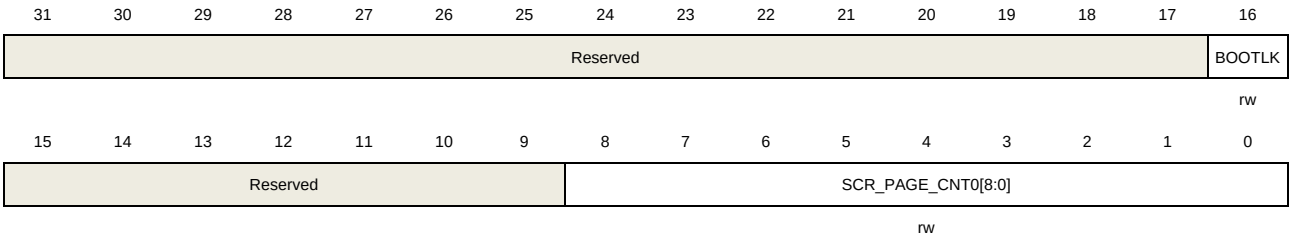
Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value
23:16	BK1WP1_EADDR[7:0]	WP fourth area end offset DBS=1: BK1WP1_EADDR[7:0] contains the last page of the WP second area for bank1. DBS=0: BK1WP1_EADDR[7:0] contains the last page of the WP fourth area for all memory
15:8	Reserved	Must be kept at reset value
7:0	BK1WP1_SADDR[7:0]	WP fourth area start offset DBS=1: BK1WP1_SADDR[7:0] contains the first page of the WP second area for bank1. DBS=0: BK1WP1_SADDR[7:0] contains the first page of the WP fourth area for all memory.

2.4.17. Bank0 secure user area register (FMC_BK0SCR)

Address offset: 0x70

Reset value: 0xFF0X FXXX

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



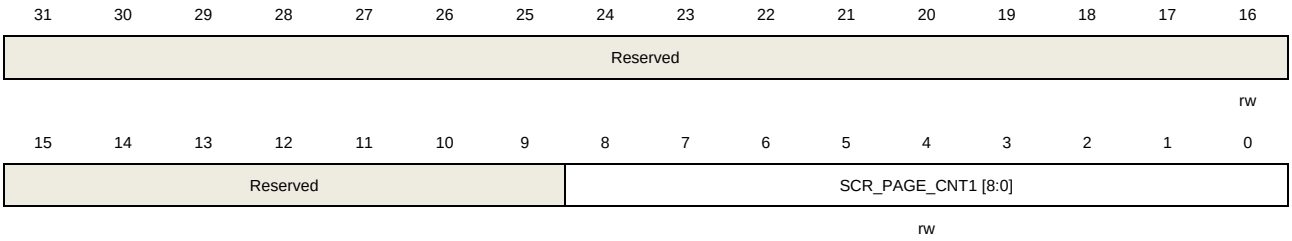
Bits	Fields	Descriptions
31:17	Reserved	Must be kept at reset value
16	BOOTLK	This bit is set to force boot from user flash area. 0: Support flash, RAM and system boot 1: Only boot from main flash
15:9	Reserved	Must be kept at reset value
8:0	SCR_PAGE_CNT0[8:0]	Configure the number of pages in the bank0 secure user area. Secure user area starts at bank0 base address.The memory size is SCR_PAGE_CNT0 * page size. This field can be changed when SPC level is no protection only.

2.4.18. Bank1 secure user area register (FMC_BK1SCR)

Address offset: 0x74

Reset value: 0xFF00 FXXX

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



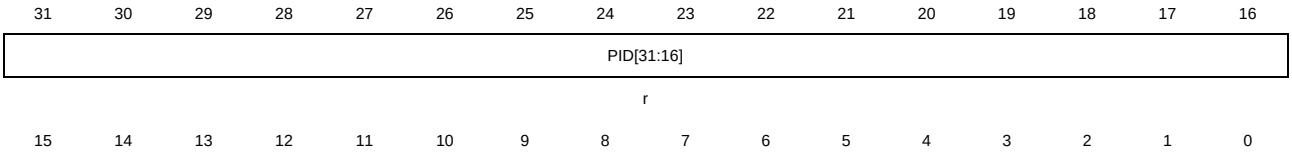
Bits	Fields	Descriptions
31:9	Reserved	Must be kept at reset value
8:0	SCR_PAGE_CNT1[8:0]	Configure the number of pages in the bank1 secure user area. Secure user area starts at at bank1 base address.The memory size is SCR_PAGE_CNT1 * page size. This field can be changed when SPC level is no protection only. When DBS = 0, this field is useless.

2.4.19. Product ID register (FMC_PID)

Address offset: 0x100

Reset value: 0xXXXX XXXX

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



PID[15:0]
r

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

3. Power management unit (PMU)

3.1. Overview

The power consumption is regarded as one of the most important issues for the devices of GD32G553 series. The Power management unit (PMU), provides three types of power saving modes, including Sleep, Deep-sleep, and Standby mode. These modes reduce the power consumption and allow the application to achieve the best tradeoff among the conflicting demands of CPU operating time, speed and power consumption. For GD32G553 series, there are three power domains, including V_{DD} / V_{DDA} domain, 1.1V domain, and Backup domain, as is shown in ***Figure 3-1. Power supply overview***. The power of the V_{DD} domain is supplied directly by V_{DD}. An embedded LDO in the V_{DD} / V_{DDA} domain is used to supply the 1.1V domain power. 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.

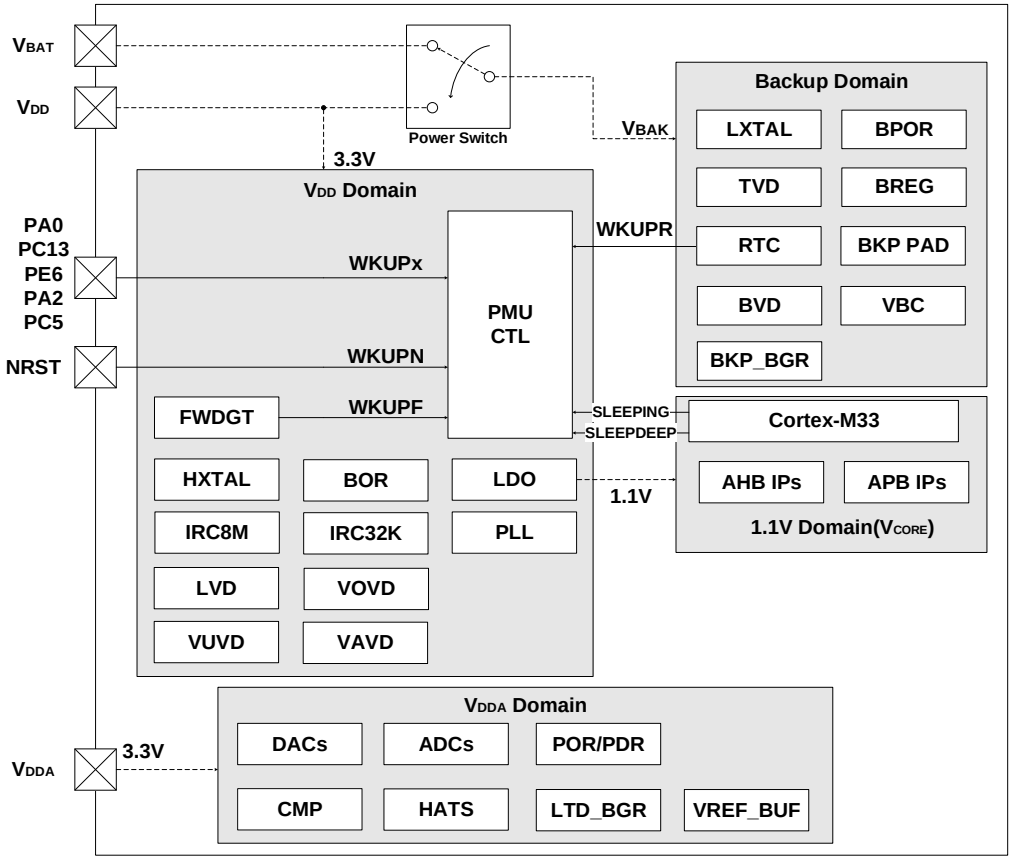
3.2. Characteristics

- Three Voltage area: V_{BAK}, V_{DD}/V_{DDA} and 1.1V(V_{CORE}) power domains
- three power saving modes: Sleep, Deep-sleep, and Standby modes
- Internal Voltage regulator(LDO) supplies around 1.1V voltage source for core domain(V_{CORE})
- Low Voltage Detector(LVD) 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.
- Power supply supervision: POR / PDR monitor / BOR monitor / LVD monitor /VOVD monitor / VAVD monitor / VUVD monitor / V_{BAK} thresholds / Temperature thresholds.

3.3. Function overview

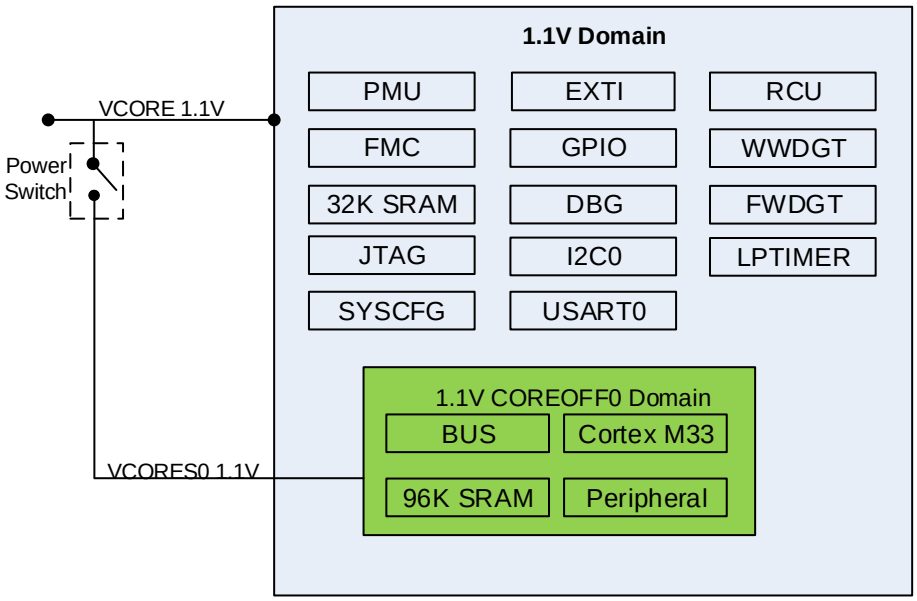
Figure 3-1. Power supply overview provides details on the internal configuration of the PMU and the relevant power domains.

Figure 3-1. Power supply overview



LVD: Low Voltage Detector
LDO: Voltage Regulator
BPOR: V_{BAK} Power On Reset
VOVD: $V_{1.1V}$ Over Voltage Detector
POR: Power On Reset
PDR: Power Down Reset
VAVD: Analog Voltage Detector
VUVD: $V_{1.1V}$ Under Voltage Detector
BVD: V_{BAK} Voltage Detector
BREG:Backup Register
TVD: Temperature Voltage Detector
BOR: Brown Out Reset
VBC: V_{BAK} Battery Charge
BKP_BGR: Backup Domain Bandgap
HATS: High Accuracy Temperature Sensor
LTD_BGR: Low Temperature Drift Bandgap
VREF_BUF: Voltage Reference Buffer

Figure 3-2. 1.1V domain supply overview



As shown in ***Figure 3-2. 1.1V domain supply overview***, the 1.1V domain includes two parts:

1. 1.1V CORE_OFF0 domain includes: BUS, Cortex®-M33 CPU, Last 96K SRAM (except the first 32K), Peripheral
2. Deep-sleep power on modules: FMC, PMU, RCU, EXTI, GPIO, First 32K SRAM, DBG, FWDGT, JTAG, I2C0, LPTIMER, SYSCFG, USART0.

The power switch S0 is used for power supply control of power saving modes.

3.3.1. 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 and BREG (Backup Register), three BKP PADs, including PC13 to PC15, BVD (V_{BAK} Voltage Detector), VBC (V_{BAK} Battery Charge) and BKP_BGP (Backup Domain Bandgap). 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 32KHz

RC oscillator (IRC32K) or the Low Speed Crystal oscillator (LXTAL), or HXTAL clock divided by 32. When V_{DD} is shut down, only LXTAL is valid for RTC. Before entering the power saving mode by executing the WFI/WFE instruction, the Cortex®-M33 can setup the RTC register with an expected 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} (V_{BAK} pin is connected to V_{DD}), the following functions are available:

- PC13 can be used as GPIO or RTC function pin described in ***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} (V_{BAK} pin is connected to V_{BAT}), the following functions are available:

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

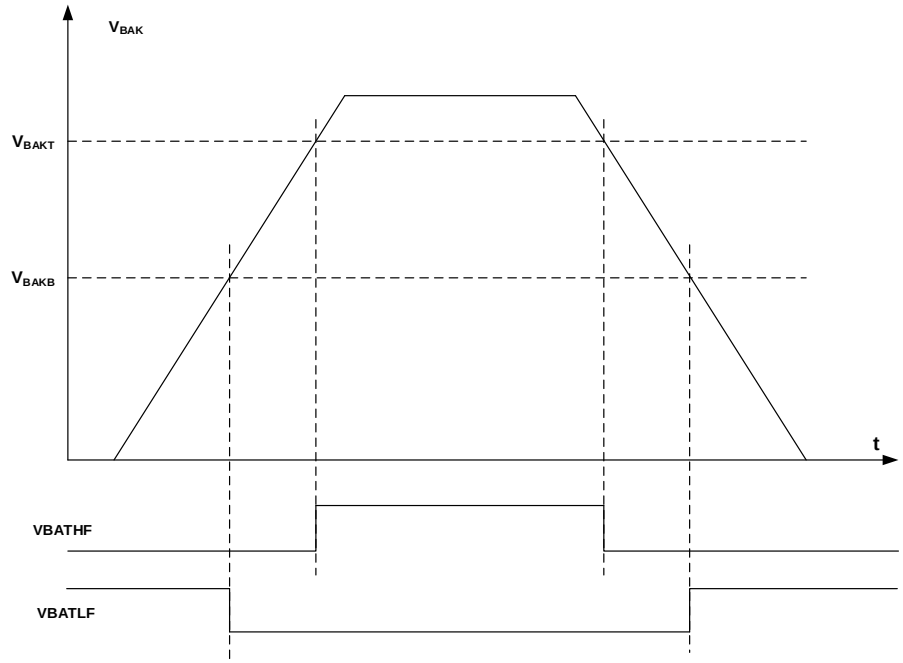
Note: Since PC13, PC14, PC15 are supplied through the Power Switch, which can only be obtained by a small current, the speed of GPIOs PC13 to PC15 should not exceed 2MHz when they are in output mode(maximum load: 30pF)

The external V_{BAT} battery can be charged by the V_{DD} through an internal resistor. The charging resistor can be selected by configuring the VCRSEL bit in PMU_CTL2 register. A 5 kOhms resistor or a 1.5 kOhms resistor can be selected for external V_{BAT} battery charing. The external V_{BAT} battery charing is enabled by setting the VCEN bit in PMU_CTL2 register. When in BKP only mode, the V_{BAT} battery charing is disabled by hardware.

Backup domain voltage thresholds

Thers is an internal power switch, which can select the voltage source of Backup domain V_{BAT} or V_{DD} The supply voltage for Backup domain (V_{BAK}) can be monitored with a top voltage and a bottom voltage (V_{BAKT} and V_{BAKB}), when VBTMEN bit is set, if V_{BAK} is over V_{BAKT}, the falg bit VBATHF will set, if V_{BAK} is lower than V_{BAKB}, the falg bit VBATLF will set. As is shown in ***Figure 3-3. Waveform of the Backup domain voltage thresholds***

Figure 3-3. Waveform of the Backup domain voltage thresholds

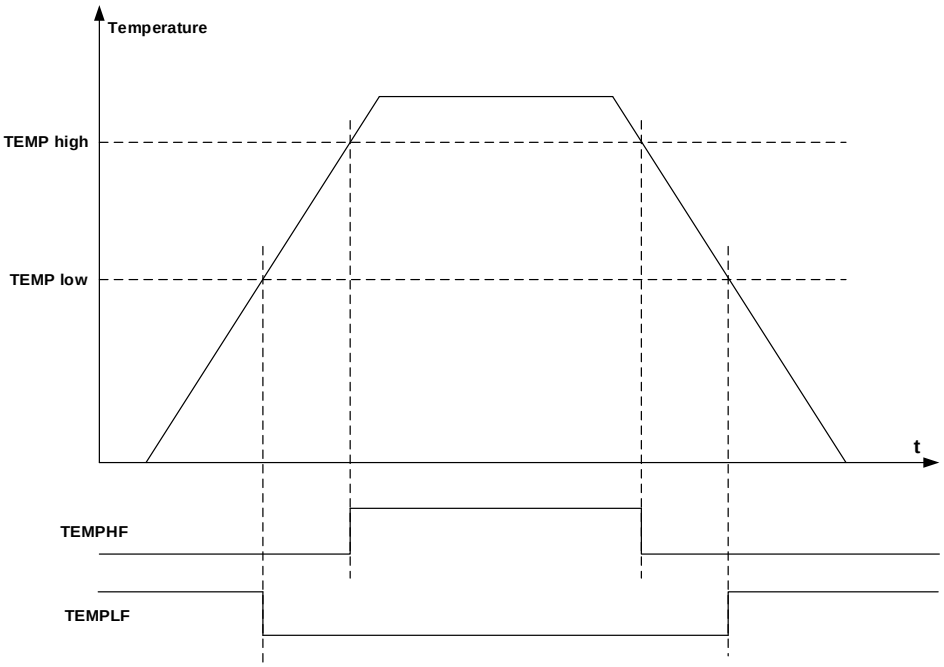


Temperature voltage thresholds

The junction temperature can be monitored by comparing it with two threshold levels, TEMP high and TEMP low. TEMPHF and TEMPLF flags, in the PMU_CTL1, indicate whether the device temperature is higher or lower than the threshold. The temperature monitoring can be enabled / disabled via VBTMEN bit in PMU_CTL1.When enabled, the temperature thresholds increase power consumption. As an example the levels could be used to trigger a routine to perform temperature control tasks. The temperature thresholds are available only when the Bandgap Voltage Reference is enabled (BKPVSEN bit set in the PMU_CTL1 register).

TEMPHF and TEMPLF wakeup interrupts are available on the RTC tamper signals .

Figure 3-4. Temperature thresholds



3.3.2. VDD / VDDA power domain

V_{DD}/V_{DDA} domain includes two parts: V_{DD} domain and V_{DDA} domain. V_{DD} domain includes HXTAL (High Speed Crystal oscillator), IRC8M (Internal 8MHz RC oscillator), IRC32K (Internal 32KHz RC oscillator), LDO (Voltage Regulator), BOR (Brown Out Reset), FWDGT (Free Watchdog Timer), PLL (Phase Locking Loop), LVD (Low Voltage Detector), VOVD ($V_{1.1V}$ over voltage detector), VAVD (V_{DDA} voltage detector), VUVD ($V_{1.1v}$ Under Voltage Detector), all pads except PC13/PC14/PC15, etc. V_{DDA} domain includes ADC/DAC (AD/DA Converter), POR/PDR (Power On/Down Reset), CMP (Comparator) , HATS (High Accuracy Temperature Sensor), LTD_BGR (Low Temperature Drift Bandgap), VREF_BUF (Voltage Reference Buffer), etc.

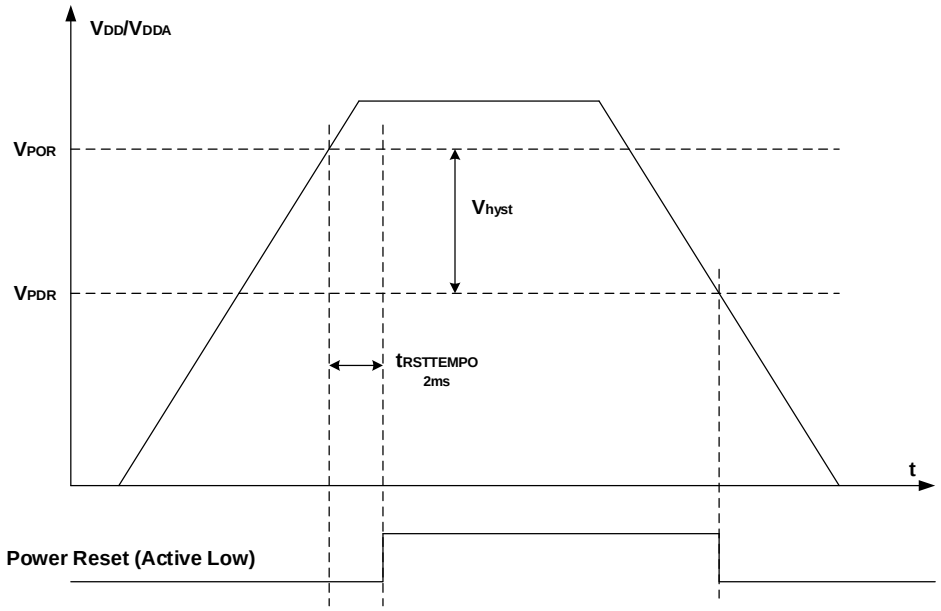
VDD domain

The LDO, which is implemented to supply power for the 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(on or low power), and in the Standby mode (power off).

The POR/PDR circuit is implemented to detect V_{DD}/V_{DDA} and generate the power reset signal which resets the whole chip except the Backup domain when the supply voltage is lower than the specified threshold. ***Figure 3-5. 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

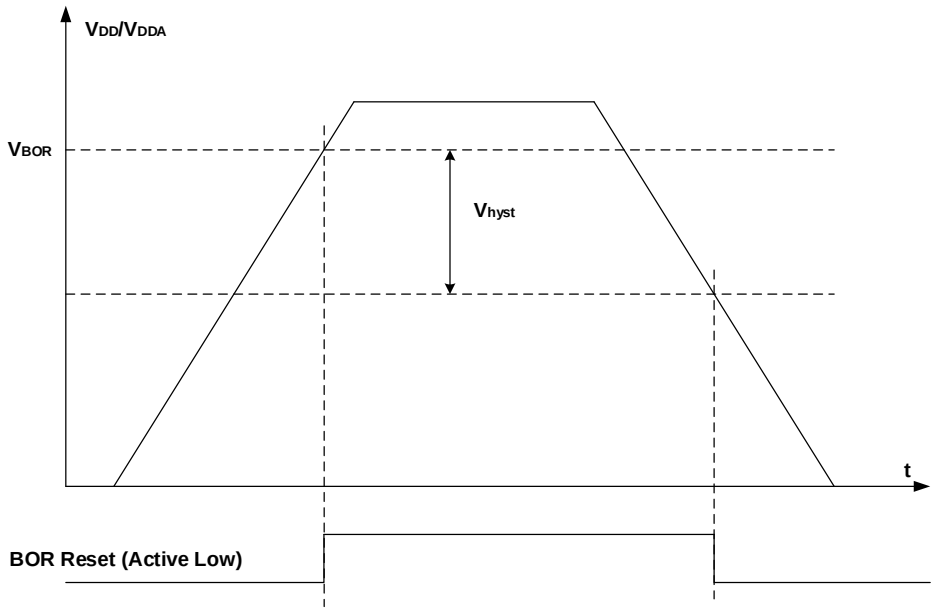
(V_{hyst}) is refer to device datasheet.

Figure 3-5. Waveform of the POR/PDR



The BOR circuit is used to detect V_{DD}/V_{DDA} and generate the power reset signal which resets the whole chip except the Backup domain when the BOR_TH bits in option bytes is not 0b11 and the supply voltage is lower than the specified threshold which defined in the BOR_TH bits in option bytes. Notice that the POR/PDR circuit is always implemented regardless of BOR_TH bits in option bytes is 0b11 or not. ***Figure 3-6. 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 hysteresis voltage (V_{hyst}) is refer to device datasheet.

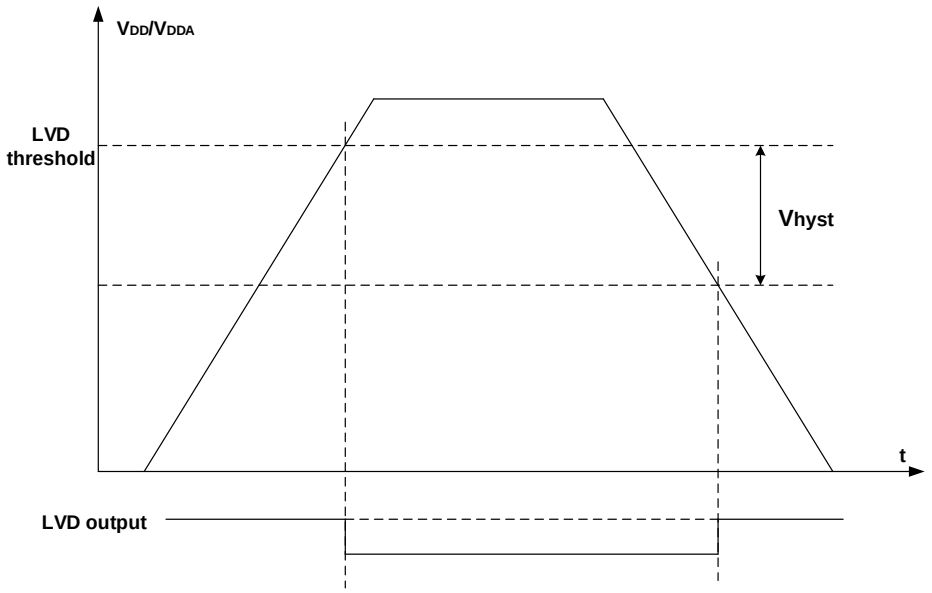
Figure 3-6. Waveform of the BOR



V_{DDA} domain

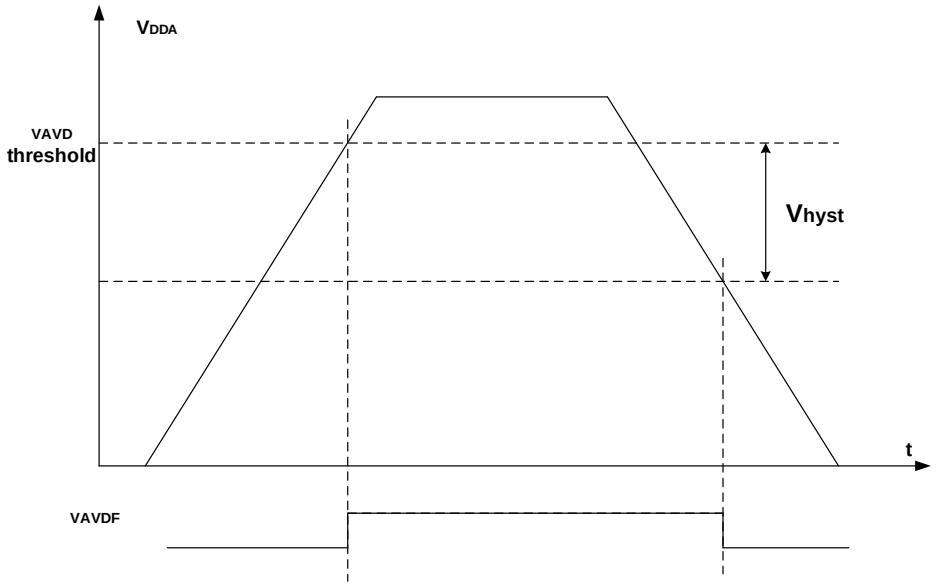
The LVD is used to detect whether the V_{DD}/V_{DDA} supply voltage is lower than a programmed threshold selected by the LVDT[2:0] bits in the Power control register(PMU_CTL0). The LVD is enabled by setting the LVDEN bit, and LVDF bit, which in PMU_CS, indicates if V_{DD}/V_{DDA} is higher or lower than the LVD threshold. This event is internally connected to the EXTI line 16 and can generate an interrupt if it is enabled through the EXTI registers. ***Figure 3-7. 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. The hysteresis voltage (V_{hyst}) please refer to device datasheet.

Figure 3-7. Waveform of the LVD threshold



The VDDA analog voltage detector is used to detect whether the VDDA supply voltage is lower than a programmed threshold selected by the VAVDVC[1:0] bits in the power control register(PMU_CTL0). The VAVD is enabled by setting the VAVDEN (AVDEN)bit, and VAVDF bit, which in PMU_CS, indicates if VDDA is higher or lower than the specified VAVD threshold. This event is internally connected to the EXTI line 16 and can generate an interrupt if it is enabled through the EXTI registers.*Figure 3-8. Waveform of the VAVD threshold* shows the relationship between the VAVD threshold and the VAVDF The hysteresis voltage (Vhyst) please refer to device datasheet.

Figure 3-8. Waveform of the VAVD threshold



Generally, digital circuits are powered by V_{DD}, while most of analog circuits are powered by

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

To ensure a high accuracy on ADC and DAC, the ADC/DAC independent external reference voltage should be connected to VREFP/VREFN pins. According to the different packages, VREFP pin can be connected to VDDA pin, or external reference voltage which refers to ***Table 17-2. ADC input pins definition*** and ***Table 18-1. 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 VSSA.

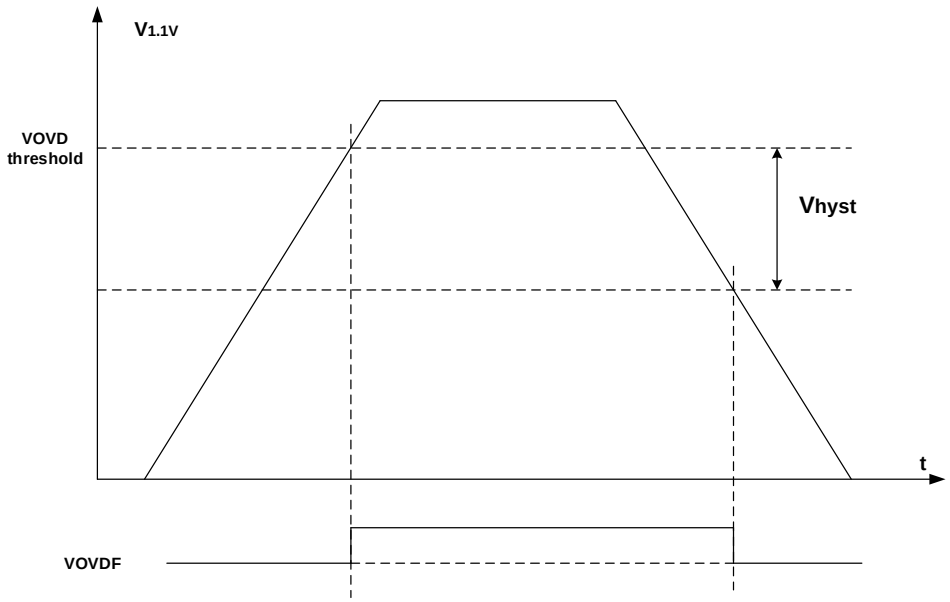
3.3.3. 1.1V power domain

The main functions that include Cortex®-M33 logic, AHB / APB peripherals, the APB interfaces for the Backup domain and the VDD/VDDA domain, etc, are located in this power domain. Once the core is powered up, the POR will generate a reset sequence on the 1.1V(V_{core}) domain. To enter the expected power saving mode, the associated control bits must be configured. Then, once a WFI (Wait for Interrupt) or WFE (Wait for Event) instruction is executed, the device will enter an expected power saving mode which will be discussed in the following section. The voltage of this power domain can be configured by LDOVS[4:0] in the PMU CTL0 register.

1.1V voltage thresholds detector

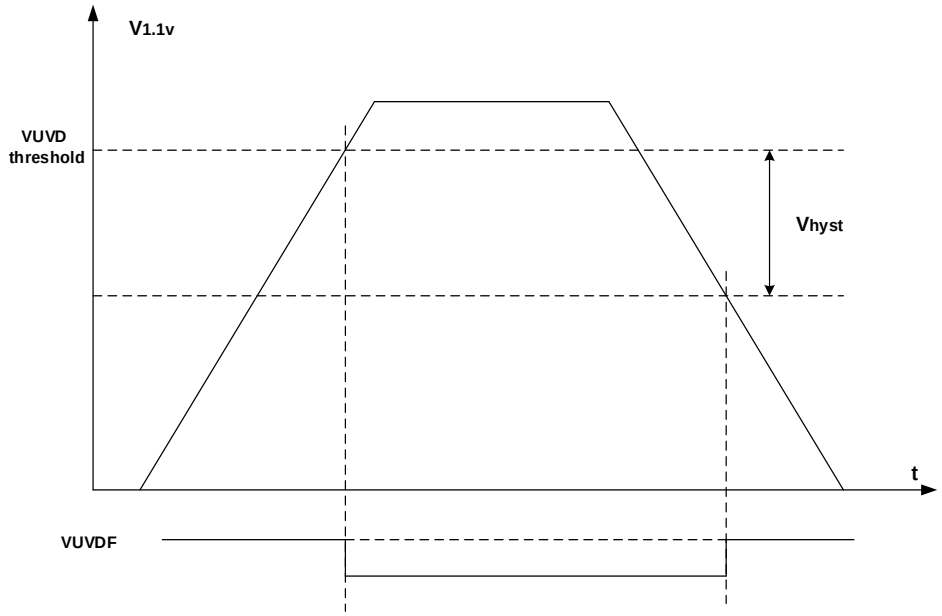
There is an internal 1.1V power over voltage detector, when VOVDEN is 0b1, it means 1.1V power over voltage detector is enabled. Once V1.1V power domain is over than a programmed threshold selected by the VOVDVC[1:0] bits in the power control register(PMU_CTL0), VOVDF0 will be set immediately after two flip-flops synchronization of the analog output. This event is internally connected to the EXTI line 16 and can generate an interrupt if it is enabled through the EXTI registers. VOVDF1 will be set after digital filter which can be used by configuring the VOVDO_DNF[7:0]bit in PMU_CTL3 register.This allows to suppress spikes with a programmable length of 1 to 255 of 1024* T_{pclk}. VOVDF1 interrupt is internally connected to the IRQ63. The hysteresis voltage (Vhyst) please refer to device datasheet.

Figure 3-9. waveform of VOVD



There is an internal 1.1V power under voltage detector, when VUVDEN is 0b1, it means 1.1V power under voltage detector is enabled. Once $V_{1.1V}$ 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, This event is internally connected to the EXTI line 16 and can generate an interrupt if it is enabled through the EXTI registers. VUVDF1 will be set after digital filter which can be used by configuring the VUVDO_DNF[7:0] bits in PMU_CTL3 register.This allows to suppress spikes with a programmable length of 1 to 255 of 1024* T_{pclk}. VUVDF1 interrupt is internally connected to the IRQ63. The hysteresis voltage (Vhyst) please refer to device datasheet.

Figure 3-10. waveform of VUVD



3.3.4. Power saving modes

After a system reset or a power reset, the GD32G553 MCU operates at full function and all power domains are active. Users can achieve lower power consumption through slowing down the system clocks (HCLK, PCLK1, PCLK2, PCLK3) or gating the clocks of the unused peripherals. Besides, three power saving modes are provided to achieve even lower power consumption, they are Sleep mode, Deep-sleep mode, and Standby mode.

Table 3-1. Power domain summary

Mode	Sleep	Deep-sleep	Standby
Description	Only CPU clock is off	1. All clocks in the 1.1V domain are off 2. Disable IRC8M / HXTAL and PLL	1. The 1.1V domain is power off 2. Disable IRC8M / HXTAL and PLL
LDO Status	On	On or in low power mode	Off
Configuration	SLEEPDEEP = 0	SLEEPDEEP = 1, STBMOD = 0	SLEEPDEEP = 1 STBMOD = 1, WURST=1
Entry	WFI or WFE	WFI or WFE	WFI or WFE

Mode	Sleep	Deep-sleep	Standby
Wakeup	Any interrupt for WFI Any event (or interrupt when SEVONPEND is 1) for WFE	Any interrupt from EXTI lines for WFI Any event(or interrupt when SEVONPEND is 1) from EXTI for WFE	1. NRST pin 2. WKUP pins 3. FWDGT reset 4. RTC 5. LCKMD
Wakeup Latency	None	IRC8M wakeup time,LDO wakeup time added if LDO is in low power mode	Power on sequence

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 from EXTI lines 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 1.1V domain are off, and all of IRC8M, HXTALand PLL 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. If not, the program will skip the entry process of Deep-sleep mode to continue to execute the following procedure.

Standby mode

The Standby mode is based on the SLEEPDEEP mode of the Cortex®-M33, too. In Standby mode, the whole 1.1V domain is power off, the LDO is shut down, and all of IRC8M, HXTAL and PLL are disabled. Before entering the Standby mode, it is necessary to set the SLEEPDEEP bit in the Cortex®-M33 System Control Register, and set the STBMOD bit in the PMU_CTL0 register, and clear WUF bit in the PMU_CS register. Then, the device enters the Standby mode after a WFI or WFE instruction is executed, and the STBF status flag in the PMU_CS register indicates that the MCU has been in Standby mode. There are five wakeup sources for the Standby mode, including the external reset from NRST pin, the RTC alarm/time stamp / tamper events, the FWDGT reset, LXTAL clock failure detection and the rising edge on WKUP pins. The Standby mode achieves the lowest power consumption, but spends longest time to wake up. Besides, the contents of SRAM and registers in 1.1V power domain are lost in Standby mode. When exiting from the Standby mode, a power-on reset occurs and the Cortex®-M33 will execute instruction code from the 0x00000000 address.

BKP only mode

The BKP only mode is enter when VDD pin cut off by external Power switch. While the VBAT pin supply. The BKP domain include RTC / LXTAL / BKP POR is on. In this mode, the customer can use RTC. This mode exit when VDD pin supply.

3.4. Register definition

PMU base address: 0x40007000

3.4.1. Control register 0 (PMU_CTL0)

Address offset: 0x00

Reset value: 0x0002 6000

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			VUVDVC		VOVDVC		VUVDEN	VOVDEN	VAVDVC		VAVDEN	Reserved		DSLPVS	
			rw		rw		rw	rw	rw		rw			rw	rw

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

Bits	Fields	Descriptions
31:29	Reserved	Must be kept at reset value.
28:27	VUVDVC[1:0]	$V_{1.1V}$ under voltage detector voltage level configure bits These bits are set and cleared by software 00: Configure $V_{1.1V}$ under voltage detector voltage level to 0.95V 01: Configure $V_{1.1V}$ under voltage detector voltage level to 0.85V. 10: Configure $V_{1.1V}$ under voltage detector voltage level to 0.75V. 11: Configure $V_{1.1V}$ under voltage detector voltage level to 0.65V.
26:25	VOVDVC[1:0]	$V_{1.1V}$ over voltage detector voltage level configure bits These bits are set and cleared by software 00: Configure $V_{1.1V}$ over voltage detector voltage level to 1.25V. 01: Configure $V_{1.1V}$ over voltage detector voltage level to 1.35V. 10: Configure $V_{1.1V}$ over voltage detector voltage level to 1.45V. 11: Configure $V_{1.1V}$ over voltage detector voltage level to 1.55V.
24	VUVDEN	$V_{1.1V}$ under Voltage detector enable 0: Disable $V_{1.1V}$ under Voltage Detector. 1: Enable $V_{1.1V}$ under Voltage Detector.
23	VOVDEN	$V_{1.1V}$ over Voltage detector enable This bit is set and cleared by software. 0: Peripheral voltage on $V_{1.1V}$ detector disabled. 1: Peripheral voltage on $V_{1.1V}$ detector enabled
22:21	VAVDVC[1:0]	VDDA analog voltage detector voltage level configure bits These bits are set and cleared by software 00: Configure V_{DDA} analog voltage detector voltage level to 1.8V

		01: Configure V_{DDA} analog voltage detector voltage level to 2.2V 10: Configure V_{DDA} analog voltage detector voltage level to 2.6V 11: Configure V_{DDA} analog voltage detector voltage level to 2.9V
20	VAVDEN	V_{DDA} analog voltage detector voltage enable bit This bit is set and cleared by software. 0: V_{DDA} analog voltage detector voltage disabled. 1: V_{DDA} analog voltage detector voltage enabled.
19:18	Reserved	Must be kept at reset value.
17:16	DSLPVS	Deep-sleep mode voltage scaling selection These bits control the V_{CORE} voltage level in system Deep-sleep, to obtain the best trade-off between power consumption and performance. 00: 0.8V 01: 0.9V 10: 1.0V(default) 11: 1.1V
15:11	LDOVS[4:0]	LDO output voltage select These bits are set by software when the main PLL closed. And the LDO output voltage selected by LDOVS bits takes effect when the main PLL enabled. If the main PLL closed, the LDO output voltage low mode selected (value of this bit filed not changed). 00000~01011:reserved 01100: V_{CORE} voltage is 1.1V 01101:reserved 01110: V_{CORE} voltage is 1.15V 01111: reserved 10000: reserved 10001~11111:reserved **Note:** This bit can only be configured as 1.1V or 1.15V,Configured this bit as reserved is prohibited.
10:9	Reserved	Must be kept at reset value.
8	BKPWEN	Backup Domain Write Enable 0: Disable write access to the registers in Backup domain 1: Enable write access to the registers in Backup domain After reset, any write access to the registers in Backup domain is ignored. This bit has to be set to enable write access to these registers.
7:5	LVDT[2:0]	Low Voltage Detector Threshold 000: 2.15V 001: 2.3V 010: 2.45V 011: 2.6V 100: 2.75V

		101: 2.9V 110: 3.0V 111: PA10 input analog voltage LVD_IN (compared with 1.2V)
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 mode when the Cortex®-M33 enters SLEEPDEEP mode 1: Enter the Standby mode when the Cortex®-M33 enters SLEEPDEEP mode
0	LDOLP	LDO Low Power Mode 0: The LDO operates normally during the Deep-sleep mode 1: The LDO is in low power mode during the Deep-sleep mode

3.4.2. Control and status register 0 (PMU_CS)

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			WUPEN4	WUPEN3	WUPEN2	WUPEN1	WUPEN0	VUVDF1	VOVDF1	VUVDF0	VOVDF0	VAVDF	LVDF	STBF	WUF
			rw	rw	rw	rw	rw	r	r	r	r	r	r	r	r

Bits	Fields	Descriptions
31:13	Reserved	Must be kept at reset value
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 edge on the WKUP pin4 wakes up the system from the power saving mode. As the WKUP pin4 is active high, the WKUP pin4 is internally configured to input pull down mode. And set this bit will trigger a wakeup event when the input is already high.

11	WUPEN3	WKUP Pin3 (PA2) Enable 0: Disable WKUP pin1 function 1: Enable WKUP pin1 function If WUPEN3 is set before entering the power saving mode, a rising edge on the WKUP pin3 wakes up the system from the power saving mode. As the WKUP pin3 is active high, the WKUP pin3 is internally configured to input pull down mode. And set this bit will trigger a wakeup event when the input is already high.
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 edge on the WKUP pin2 wakes up the system from the power saving mode. As the WKUP pin2 is active high, the WKUP pin2 is internally configured to input pull down mode. And set this bit will trigger a wakeup event when the input is already high.
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 edge on the WKUP pin1 wakes up the system from the power saving mode. As the WKUP pin1 is active high, the WKUP pin1 is internally configured to input pull down mode. And set this bit will trigger a wakeup event when the input is already high.
8	WUPEN0	WKUP Pin0 (PA0) Enable (EWUP) 0: Disable WKUP pin0 function 1: Enable WKUP pin0 function If WUPEN0 is set before entering the power saving mode, a rising edge on the WKUP pin0 wakes up the system from the power saving mode. As the WKUP pin0 is active high, the WKUP pin0 is internally configured to input pull down mode. And set this bit will trigger a wakeup event when the input is already high.
7	VUVDF1	$V_{1.1v}$ 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: $V_{1.1v}$ is higher than VUVD threshold. 1: $V_{1.1v}$ is equal or lower than VUVD threshold.
6	VOVDF1	V1.1v 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: $V_{1.1v}$ is lower than VOVD threshold 1: $V_{1.1v}$ is equal or higher than VOVD threshold

5	VUVDF0	$V_{1.1}$v under voltage detector flag bit this bit is set and cleared by hardware. It si valid only if VUVDEN is enabled. 0: $V_{1.1}$v is higher than VUVD threshold. 1: $V_{1.1}$v is equal or lower than VUVD threshold.
4	VOVDF0	$V_{1.1}$v over voltage detector flag bit. this bit is set and cleared by hardware. It si valid only if VOVDEN is enabled. 0: $V_{1.1}$v is lower than VOVD threshold. 1: $V_{1.1}$v is equal or higher than VOVD threshold.
3	VAVDF	V_{DDA} analog voltage detector voltage output on V_{DDA} flag bit. This bit is set and cleared by hardware. It is valid only if VAVDEN is enabled. 0: V_{DDA} is equal or higher than the VAVD threshold configured by VAVDVC bits. 1: V_{DDA} is lower than the VAVD threshold configured by VAVDVC bits.
2	LVDF	Low Voltage Detector Status Flag 0: Low Voltage event has not occurred (VDD is higher than the specified LVD threshold) 1: Low Voltage event occurred (VDD is equal to or lower than the specified LVD threshold) Note: The LVD function is stopped in Standby mode.
1	STBF	Standby Flag 0: The device has not entered the Standby mode 1: The device has been in the Standby mode This bit is cleared only by a POR/PDR or by setting the STBRST bit in the PMU_CTL0 register.
0	WUF	Wakeup Flag 0: No wakeup event has been received 1: Wakeup event occurred from the WKUP pin or the RTC wakeup event including RTC Tamper event, RTC alarm event，RTC Time Stamp event This bit is cleared only by a POR/PDR or by setting the WURST bit in the PMU_CTL0 register.

3.4.3. Control register 1 (PMU_CTL1)

Address offset: 0x08

Reset value: 0x0000 0000

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

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

rw rw

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23	TEMPHF	Temperature level monitoring versus high threshold 0: Temperature below high threshold level. 1: Temperature equal or above high threshold level.
22	TEMPLF	Temperature level monitoring versus low threshold 0: Temperature above low threshold level. 1: Temperature equal or below low threshold level.
21	VBATHF	VBAT level monitoring versus high threshold 0: V_{BAT} level below high threshold level. 1: V_{BAT} level equal or above high threshold level.
20	VBATLF	VBAT level monitoring versus low threshold 0: V_{BAT} level above low threshold level. 1: V_{BAT} level equal or below low threshold level.
19:17	Reserved	Must be kept at reset value.
16	BKPVSRF	Bandgap Voltage Reference ready BRRDY This bit is set by hardware to indicate that the Bandgap Voltage Reference is ready. 0: Bandgap Voltage Reference not ready. 1: Bandgap Voltage Reference ready.
15:5	Reserved	Must be kept at reset value.
4	VBTMEN	VBAT and temperature monitoring enable. When set, the VBAT supply and temperature monitoring is enabled. 0: VBAT and temperature monitoring disabled. 1: VBAT and temperature monitoring enabled.
3:1	Reserved	Must be kept at reset value.
0	BKPVSEN	Bandgap Voltage Reference(BGR) enable 0: Bandgap Voltage Reference disabled. 1: Bandgap Voltage Reference enabled.

3.4.4. Control register 2 (PMU_CTL2)

Address offset: 0x0C

Reset value: 0x0000 0000 (reset only by power-on reset(POR))

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						VCRSEL	VCEN	Reserved							
						rw	rw								

Bits	Fields	Descriptions
31:10	Reserved	Must be kept at reset value.
9	VCRSEL	VBAT battery charging resistor selection 0: 5 kOhms resistor is selected for charing VBAT battery. 1:1.5 kOhms resistor is selected for charing VBAT battery.
8	VCEN	VBAT battery charging enable 0: Disable VBAT battery charging. 1: Enable VBAT battery charging.
7:0	Reserved	Must be kept at reset value.

3.4.5. Control register 3(PMU_CTL3)

Address offset: 0x18
Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:8	VUVDO_DNF	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_{PCLK} 0: Digital filter is disabled 1: Digital filter is enabled and filter spikes with a length of up to 1024 * T_{PCLK} ... 255: Digital filter is enabled and filter spikes with a length of up to 255*1024 * T_{PCLK}
7:0	VOVDO_DNF	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_{PCLK}

0: Digital filter is disabled

1: Digital filter is enabled and filter spikes with a length of up to 1024 * T_{PCLK}

...

255: Digital filter is enabled and filter spikes with a length of up to 255*1024 * T_{PCLK}

4. Reset and clock unit (RCU)

4.1. Reset control unit (RCTL)

4.1.1. Overview

GD32G553 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.

4.1.2. Function overview

Power reset

The power reset is generated by either an external reset as power on and power down reset (POR / PDR reset) or by the internal reset generator when exiting Standby mode. The power reset sets all registers to their reset values except the backup domain. The power reset whose active signal is low, it will be de-asserted when the internal LDO voltage regulator is ready to provide 1.1V 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).
- Option byte loader reset (OBL_RSTn)
- Reset generated when entering Standby mode when resetting nRST_STDBY bit in user option bytes (OB_STDBY_RSTn).
- Reset generated when entering Deep-sleep mode when resetting nRST_DPSLP bit in user option bytes (OB_DPSLP_RSTn).

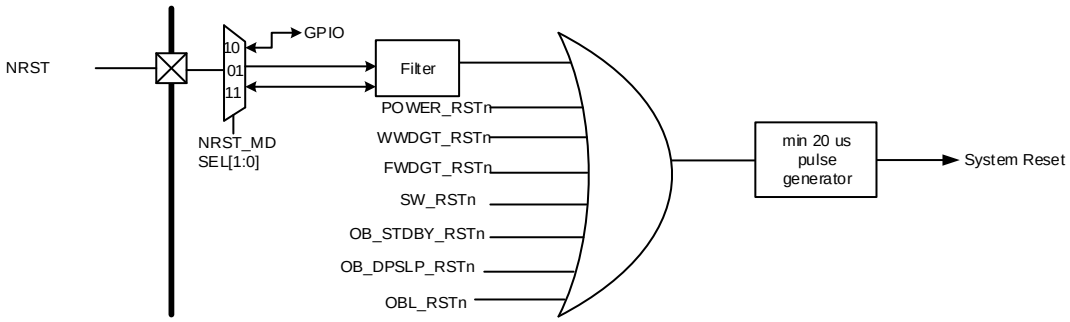
Note: The NRST pin can be configured with the option byte NRST_MDSEL[1:0] in the following three modes:

1. Input/output mode (default mode): the GPIO function of the NRST pin is not available in this mode. The reset signal can be transferred from the NRST pin to the device, causing the device to reset, the reset pulse signal can be reflected through the NRST pin, and the minimum reset pulse duration is 20 μs.
2. Input mode: the GPIO function of the NRST pin is not available in this mode, the reset signal can be transferred from the NRST pin to the device, causing the device to reset, but the internal reset of the device is not visible at the NRST pin.
3. GPIO mode: NRST pin can only function as standard GPIO, reset function is not available, reset signal is only inside the device, not reflected in NRST pin.

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

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

Figure 4-1. The system reset circuit



Backup domain reset

A backup domain reset is generated by setting the BKPRST bit in the backup domain control register or backup domain power on reset (V_{DD} or V_{BAT} power on, if both supplies have previously been powered off).

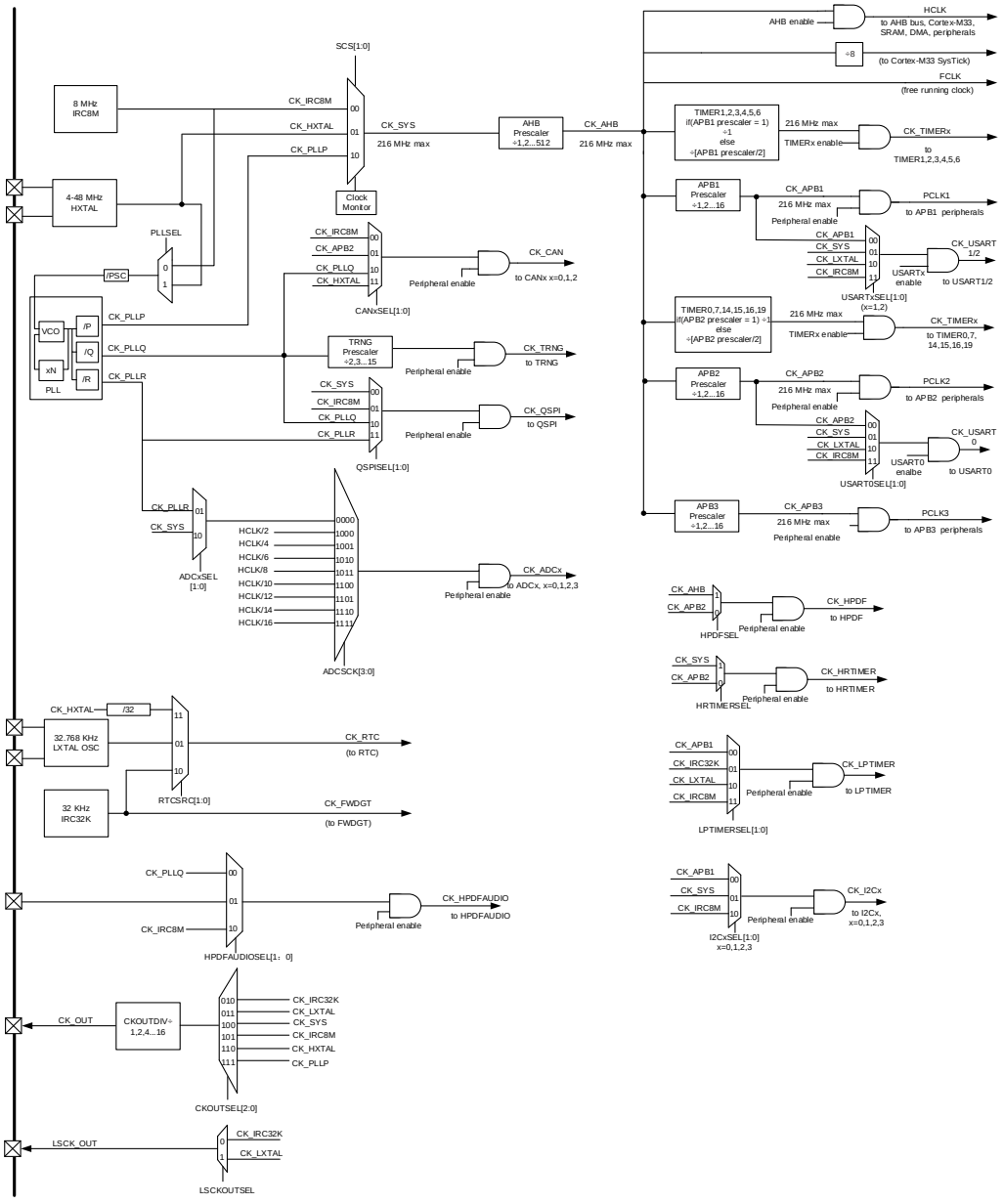
4.2. Clock control unit (CCTL)

4.2.1. Overview

The clock control unit provides a range of frequencies and clock functions. These include a Internal 8M RC oscillator (IRC8M), a High Speed crystal oscillator (HXTAL), a Low Speed Internal 32K RC oscillator (IRC32K), a Low Speed crystal oscillator (LXTAL), a Phase Lock Loop (PLL), a HXTAL clock monitor, a LXTAL 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 216 MHz.

Figure 4-2. Clock tree



The frequency of AHB, APB3, APB2 and the APB1 domains can be configured by each prescaler. The maximum frequency of the AHB, APB3, APB2 and APB1 domains is 216 / 216 / 216 / 216MHz. 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 ADCx（x = 0, 1, 2, 3）is clocked by the CK_PLLR or CK_SYS or HCLK divided by 2, 4, 6, 8，10, 12, 14, 16, which defined by ADCxSEL in RCU_CFG2 register and ADCSCK in

ADC_SYNCCTL register.

The CANx (x=0, 1, 2) is clocked by IRC8M or HXTAL clock or PLLQ or APB2 clock, which selected by CANxSEL (x=0, 1, 2) bits in configuration register 1 (RCU_CFG1).

The HPDF_AUDIO is clocked by the clock of CK_PLLQ or CK_IRC8M or External HPDF_CKIN PIN which defined by HPDFAUDIOSEL bits in RCU_CFG1 register.

The TRNG is clocked by the clock of CK_PLLQ divided by 2 to 15, which defined by TRNGPSC in RCU_CFG2 register.

The QSPI is clocked by IRC8M clock or PLLQ or PLLR or system clock, which selected by QSPISEL bits in configuration register 2 (RCU_CFG2).

The USART0 is clocked by IRC8M clock or LXTAL clock or system clock or APB2 clock, which selected by USART0SEL bits in configuration register 1 (RCU_CFG1). The USARTx (x = 1,2) is clocked by IRC8M clock or HXTAL clock or system clock or APB1 clock, which selected by USARTxSEL (x = 1,2) bits in configuration register 1 (RCU_CFG1).

The HPDF is clocked by AHB clock or APB2 clock, which selected by HPDFSEL bit in clock configuration register 1 (RCU_CFG1).

The HRTIMER is clocked by CK_APB2 or CK_SYS clock, which selected by HRTIMERSEL bit in clock configuration register 2 (RCU_CFG2).

If the HRTIMER high-resolution mode is not required, the HRTIMERSEL bit in the RCU_CFG2 register can remain cleared. In this case, the CNTCKDIV[2:0] value in the HRTIMER_MTCTL0 register must be at least 5 (prescaler ratio of 32 or greater).

When high-resolution mode is required for the HRTIMER, the source of system clock source must be selected as PLL, and the HRTIMERSEL bit in the RCU_CFG2 register must be set to 1 to select CK_SYS as the clock source. In this scenario, any value can be used for CNTCKDIV[2:0] in the HRTIMER_MTCTL0 register.

Note: For high-resolution configuration, the AHB and APB2 prescalers (AHBPSC[3:0] and APB2PSC[2:0] bits in the RCU_CFG0 register) must be set to maintain a proportional relationship of 1, 2, or 4 between the system clock CK_SYS and the APB2 clock PCLK2.

The LPTIMER is clocked by CK_APB1 or CK_IRC32K or CK_LXTAL or CK_IRC8M clock, which selected by LPTIMERSEL bit in clock configuration register 2 (RCU_CFG2).

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 RTC is clocked by LXTAL clock or IRC32K clock or HXTAL clock divided by 32 (defined which select by RTCSRC bit in backup domain control register (RCU_BDCTL). After the RTC select HXTAL clock divided by 32, the clock disappeared when the 1.1V core domain power off. After the RTC select IRC32K, 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 IRC32K clock, which is forced on when FWDGT started.

4.2.2. Characteristics

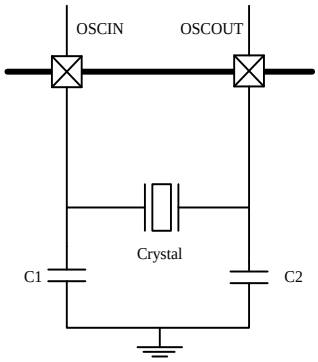
- 4 to 48 MHz high speed crystal oscillator (HXTAL).
- Internal 8 MHz RC oscillator (IRC8M).
- 32,768 Hz low speed crystal oscillator (LXTAL).
- Internal 32KHz RC oscillator (IRC32K).
- PLL clock source can be HXTAL, IRC8M.
- HXTAL clock monitor.
- LXTAL clock monitor

4.2.3. Function overview

High speed crystal oscillator (HXTAL)

The high speed external crystal oscillator (HXTAL), which has a frequency from 4 to 48 MHz, produces a highly accurate clock source for use as the system clock. A crystal with a specific frequency must be connected and located close to the two HXTAL pins. The external resistor and capacitor components connected to the crystal are necessary for proper oscillation.

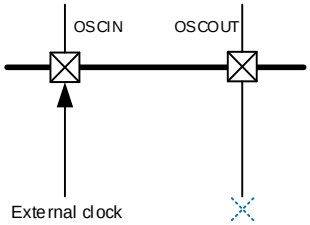
Figure 4-3. HXTAL clock source



The HXTAL crystal oscillator can be switched on or off using the HXTALEN bit in the control register 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 4-4. HXTAL clock source in bypass mode***. The CK_HXTAL is equal to the external clock which drives the OSCIN pin.

Figure 4-4. 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.

Phase locked loop (PLL)

The internal Phase Locked Loop, PLL, can provide 16 ~ 216 MHz clock output which is 2 ~31 multiples of a fundamental reference frequency of 2 ~ 40 MHz.

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

Enable the desired PLL outputs by configuring PLLPEN, PLLQEN, PLLREN.The PLLP could be used to generator system clock (no more than 216MHz) and PLLQ clock which used to TRNG/QSPI/CAN. The PLLR is used to generator the clock to ADC. The enable bit of each PLL output clock (PLLPEN, PLLQEN, PLLREN) can be modified at any time without stopping the corresponding PLL. PLLPEN cannot be cleared if CK_PLLP is used as system clock.

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

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 32K RC oscillator (IRC32K)

The internal RC oscillator has a frequency of about 32 kHz and is a low power clock source for the real time clock circuit or the free watchdog timer. The IRC32K offers a low cost clock source as no external components are required. The IRC32K RC oscillator can be switched on or off by using the IRC32KEN bit in the reset source/clock register (RCU_RSTSCK). The IRC32KSTB flag in the reset source/clock register RCU_RSTSCK will indicate if the IRC32K clock is stable. An interrupt can be generated if the related interrupt enable bit IRC32KSTBIE in the clock interrupt register (RCU_INT) is set when the IRC32K 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_PLL 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.

LXTAL clock monitor (LCKM)

A clock monitor on LXTAL can be activated by software writing the LCKMEN, in the control register, RCU_CTL. LCKMEN can not be enabled before LXTAL and IRC32K are enabled and ready.

A 4-bits plus one counter will work at IRC32K domain when LCKMEN enable. If the LXTAL clock has stuck at 0 / 1 error or slow down about 20KHz, the counter will overflow. The LXTAL clock failure will been found.

Clock output capability

The clock output capability is ranging from 32KHz to 216MHz. 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 4-1. Clock output source select

Clock source selection bits	Clock source
000	NO CLK
010	CK_IRC32K
011	CK_LXTAL
100	CK_SYS
101	CK_IRC8M
110	CK_HXTAL
111	CK_PLLP

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

The CK_LXTAL and CK_IRC32K clock signals also can be output on LSCK_OUT pin, even in Deepsleep mode and Standby mode，which seleted by LSCKOUTSEL in the backup domain control register (RCU_BDCTL).

Table 4-2. Low-speed clock output source select

Clock source selection bits	Clock source
0	CK_IRC32K
1	LXTAL

Deep-sleep mode clock control

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

If the USART0 / 1 / 2 clock is selected IRC8M clock in Deep-sleep mode, they have capable of open IRC8M clock or close IRC8M clock, which used to the USART0 / 1 / 2 to wake up the Deep-sleep mode.

4.3. Register definition

RCU base address: 0x4002 1000

4.3.1. Control register (RCU_CTL)

Address offset: 0x00

Reset value: 0x0000 XX83 where X is undefined.

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

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

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

Bits	Fields	Descriptions
31:26	Reserved	Must be kept at reset value.
25	PLLSTB	PLL clock stabilization flag Set by hardware to indicate if the PLL output clock is stable and ready for use. 0: PLL is not stable 1: PLL is stable
24	PLLEN	PLL enable Set and reset by software. This bit cannot be reset if the PLL clock is used as the system clock. Reset by hardware when entering Deep-sleep or Standby mode. 0: PLL is switched off 1: PLL is switched on
23:21	Reserved	Must be kept at reset value.
20	HXTALSTBRST	HXTAL stabilization reset Set and reset by software. 0: HXTAL stabilization no reset 1: HXTAL stabilization reset
19	CKMEN	HXTAL clock monitor enable 0: Disable the High speed 4 ~ 48 MHz crystal oscillator (HXTAL) clock monitor 1: Enable the High speed 4 ~ 48 MHz crystal oscillator (HXTAL) clock monitor When the hardware detects that the HXTAL clock is stuck at a low or high state, the internal hardware will switch the system clock to be the internal high speed IRC8M RC clock. The way to recover the original system clock is by either an external reset,

		power on reset or clearing CKMIF by software. **Note:** When the HXTAL clock monitor is enabled, the hardware will automatically enable the IRC8M internal RC oscillator regardless of the control bit, IRC8MEN, state.
18	HXTALBPS	High speed crystal oscillator (HXTAL) clock bypass mode enable The HXTALBPS bit can be written only if the HXTALEN is 0. 0: Disable the HXTAL Bypass mode 1: Enable the HXTAL Bypass mode in which the HXTAL output clock is equal to the input clock.
17	HXTALSTB	High speed crystal oscillator (HXTAL) clock stabilization flag Set by hardware to indicate if the HXTAL oscillator is stable and ready for use. 0: HXTAL oscillator is not stable 1: HXTAL oscillator is stable
16	HXTALEN	High speed crystal oscillator (HXTAL) Enable Set and reset by software. This bit cannot be reset if the HXTAL clock is used as the system clock or the PLL input clock when PLL clock is selected to the system clock. Reset by hardware when entering Deep-sleep or Standby mode. 0: High speed 4 ~ 48 MHz crystal oscillator disabled 1: High speed 4 ~ 48 MHz crystal oscillator enabled
15:8	IRC8MCALIB[7:0]	Internal 8MHz RC Oscillator calibration value register These bits are load automatically at power on.
7:3	IRC8MADJ[4:0]	Internal 8MHz RC Oscillator clock trim adjust value These bits are set by software. The trimming value is these bits (IRC8MADJ) added to the IRC8MCALIB[7:0] bits. The trimming value should trim the IRC8M to 8 MHz ± 1%.
2	Reserved	Must be kept at reset value.
1	IRC8MSTB	IRC8M internal 8MHz RC oscillator stabilization flag Set by hardware to indicate if the IRC8M oscillator is stable and ready for use. 0: IRC8M oscillator is not stable 1: IRC8M oscillator is stable
0	IRC8MEN	Internal 8MHz RC oscillator enable Set and reset by software. This bit cannot be reset if the IRC8M clock is used as the system clock. Set by hardware when leaving Deep-sleep or Standby mode or the HXTAL clock is stuck at a low or high state when CKMEN is set. 0: Internal 8 MHz RC oscillator disabled 1: Internal 8 MHz RC oscillator enabled

4.3.2. PLL register (RCU_PLL)

Address offset: 0x04

Reset value: 0x0000 0400

To configure the PLL clock, refer to the following formula:

CK_PLLVCOSRC = CK_PLLSRC / PLLPSC

CK_PLLVCO = CK_PLLVCOSRC × PLLN

CK_PLLP = CK_PLLVCO / PLLP

CK_PLLQ = CK_PLLVCO / PLLQ

CK_PLLR = CK_PLLVCO / PLLR

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
PLLR[4:0]					PLLQ[3:0]				PLLSEL	PLLREN	PLLQEN	PLLPEN	Reserved	PLLP[1:0]	
rw					rw				rw	rw	rw	rw		rw	

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

Bits	Fields	Descriptions
31:27	PLLR[4:0]	The PLLR output frequency division factor from PLL VCO clock Set and reset by software when the PLL is disable. These bits used to generator PLLR output clock (CK_PLLR) from PLL VCO clock (CK_PLLVCO). The CK_PLLVCO is described in PLLN bits in RCU_PLL register. 00000: Reserved 00001: Reserved 00010: CK_PLLR = CK_PLLVCO / 2 00011: CK_PLLR = CK_PLLVCO / 3 00100: CK_PLLR = CK_PLLVCO / 4 … 11111: CK_PLLR = CK_PLLVCO / 31
26:23	PLLQ[3:0]	The PLLQ output frequency division factor from PLL VCO clock Set and reset by software when the PLL is disable. These bits used to generator PLL Q output clock (CK_PLLQ) from PLL VCO clock (CK_PLLVCO). The CK_PLLQ is used to TRNG (48MHz). The CK_PLLVCO is described in PLLN bits in RCU_PLL register. 0000: Reserved 0001: Reserved 0010: CK_PLLQ = CK_PLLVCO / 2 0011: CK_PLLQ = CK_PLLVCO / 3 0100: CK_PLLQ = CK_PLLVCO / 4 …

		1111: CK_PLLQ = CK_PLLVCO / 15
22	PLLSEL	PLL clock source selection Set and reset by software to control the PLL clock source. 0: IRC8M clock selected as source clock of PLL 1: HXTAL selected as source clock of PLL
21	PLLREN	PLLR divider output enable This bit is set and reset by software. The PLLREN bit can be written only if the PLLEN is 0. 0: Disable the CK_PLLR output 1: Enable the CK_PLLR output
20	PLLQEN	PLLQ divider output enable This bit is set and reset by software. The PLLQEN bit can be written only if the PLLEN is 0. 0: Disable the CK_PLLQ output 1: Enable the CK_PLLQ output
19	PLLPEN	PLLP divider output enable This bit is set and reset by software. The PLLPEN bit can be written only if the PLLEN is 0. 0: Disable the CK_PLLP output 1: Enable the CK_PLLP output
18	Reserved	Must be kept at reset value.
17:16	PLLP[1:0]	The PLLP output frequency division factor from PLL VCO clock Set and reset by software when the PLL is disable. These bits used to generator PLLP output clock (CK_PLLP) from PLL VCO clock (CK_PLLVCO). The CK_PLLP is used to system clock (no more than 216MHz). The CK_PLLVCO is described in PLLN bits in RCU_PLL register. 00: CK_PLLP = CK_PLLVCO / 2 01: CK_PLLP = CK_PLLVCO / 4 10: CK_PLLP = CK_PLLVCO / 6 11: CK_PLLP = CK_PLLVCO / 8
15:14	Reserved	Must be kept at reset value.
13:6	PLLN[7:0]	The PLL VCO clock multiplication factor Set and reset by software (only use word / half-word write) when the PLL is disable. These bits used to generator PLL VCO clock (CK_PLLVCO) from PLL VCO source clock (CK_PLLVCOSRC). The CK_PLLVCOSRC is described in PLLPSC bits in RCU_PLL register. **Note:** The frequency of CK_PLLVCO is between 96MHz to 480MHz The value of PLLN must : 8 ≤ PLLN ≤ 180 0000000: Reserved

		… 0000111: Reserved 0001000: PLLN = 8 0001001: PLLN = 9 0001010: PLLN = 10 … 10110100: PLLN = 180
5:4	Reserved	Must be kept at reset value.
3:0	PLLPSC[3:0]	The PLL VCO source clock prescaler Set and reset by software when the PLL is disable. These bits used to generate the clock of PLL VCO source clock (CK_PLLVCOSRC) from PLL source clock (CK_PLLSRC) which described in PLLSEL in RCU_PLL register. The VCO source clock is between 2M to 16MHz 0000: CK_PLLSRC 0001: CK_PLLSRC / 2 0010: CK_PLLSRC / 3 0011: CK_PLLSRC / 4 … 1111: CK_PLLSRC / 16

4.3.3. Clock configuration register 0 (RCU_CFG0)

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		APB3PSC[2:0]			CKOUTDIV[2:0]			CKOUTSEL[2:0]			Reserved				
		rw			rw			rw							

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

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29:27	APB3PSC[2:0]	APB3 prescaler selection Set and reset by software to control the APB3 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

26:24	CKOUTDIV[2:0]	The CK_OUT divider which the CK_OUT frequency can be reduced see bits 23:21 of RCU_CFG0 for CK_OUT 000: The CK_OUT is divided by 1 001: The CK_OUT is divided by 2 010: The CK_OUT is divided by 4 011: The CK_OUT is divided by 8 100: The CK_OUT is divided by 16 Others: Reserved
23:21	CKOUTSEL[2:0]	CK_OUT clock source selection Set and reset by software. 000: No clock selected 001: Reserved 010: Internal 32K RC oscillator clock selected 011: External low speed oscillator clock selected 100: System clock selected 101: Internal 8MHz RC oscillator clock selected 110: External high speed oscillator clock selected 111: CK_PLLP selected
20:16	Reserved	Must be kept at reset value.
15:13	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
12:10	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
9:8	Reserved	Must be kept at reset value.
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: Reserved 11: Select CK_PLLP as the CK_SYS source
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: Reserved 11: Select CK_PLLP as the CK_SYS source

4.3.4. Clock interrupt register (RCU_INT)

Address offset: 0x0C

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			LCKMIC	LCKMIF	Reserved			CKMIC	Reserved		PLL STBIC	HXTAL STBIC	IRC8M STBIC	LXTAL STBIC	IRC32K STBIC
			w	r				w			w	w	w	w	w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			PLL STBIE	HXTAL STBIE	IRC8M STBIE	LXTAL STBIE	IRC32K STBIE	CKMIF	Reserved		PLL STBIF	HXTAL STBIF	IRC8M STBIF	LXTAL STBIF	IRC32K STBIF
			rw	rw	rw	rw	rw	r			r	r	r	r	r

Bits	Fields	Descriptions
31:29	Reserved	Must be kept at reset value.
28	LCKMIC	LXTAL clock monitor interrupt clear Write 1 by software to reset the LCKMIF flag. 0: Not reset LCKMIF flag

		1: Reset LCKMIF flag
27	LCKMIF	LXTAL clock monitor interrupt flag Set by hardware when LXTAL clock is stuck. Reset by software when setting the LCKMIC bit. 0: LXTAL clock operating normally 1: LXTAL clock stuck
26:24	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
22:21	Reserved	Must be kept at reset value.
20	PLLSTBIC	PLL stabilization interrupt clear Write 1 by software to reset the PLLSTBIF flag. 0: Not reset PLLSTBIF flag 1: Reset PLLSTBIF flag
19	HXTALSTBIC	HXTAL stabilization interrupt clear Write 1 by software to reset the HXTALSTBIF flag. 0: Not reset HXTALSTBIF flag 1: Reset HXTALSTBIF flag
18	IRC8MSTBIC	IRC8M stabilization interrupt clear Write 1 by software to reset the IRC8MSTBIF flag. 0: Not reset IRC8MSTBIF flag 1: Reset IRC8MSTBIF flag
17	LXTALSTBIC	LXTAL stabilization interrupt clear Write 1 by software to reset the LXTALSTBIF flag. 0: Not reset LXTALSTBIF flag 1: Reset LXTALSTBIF flag
16	IRC32KSTBIC	IRC32K stabilization interrupt clear Write 1 by software to reset the IRC32KSTBIF flag. 0: Not reset IRC32KSTBIF flag 1: Reset IRC32KSTBIF flag
15:13	Reserved	Must be kept at reset value.
12	PLLSTBIE	PLL stabilization interrupt enable Set and reset by software to enable / disable the PLL stabilization interrupt. 0: Disable the PLL stabilization interrupt 1: Enable the PLL stabilization interrupt

11	HXTALSTBIE	HXTAL stabilization interrupt enable Set and reset by software to enable/disable the HXTAL stabilization interrupt 0: Disable the HXTAL stabilization interrupt 1: Enable the HXTAL stabilization interrupt
10	IRC8MSTBIE	IRC8M stabilization interrupt enable Set and reset by software to enable/disable the IRC8M stabilization interrupt 0: Disable the IRC8M stabilization interrupt 1: Enable the IRC8M stabilization interrupt
9	LXTALSTBIE	LXTAL stabilization interrupt enable Set and reset by software to enable/disable LXTAL stabilization interrupt 0: Disable the LXTAL stabilization interrupt 1: Enable the LXTAL stabilization interrupt
8	IRC32KSTBIE	IRC32K stabilization interrupt enable Set and reset by software to enable/disable IRC32K stabilization interrupt 0: Disable the IRC32K stabilization interrupt 1: Enable the IRC32K stabilization interrupt
7	CKMIF	HXTAL clock stuck interrupt flag Set by hardware when the HXTAL clock is stuck. Reset when setting the CKMIC bit by software. 0: Clock operating normally 1: HXTAL clock stuck
6:5	Reserved	Must be kept at reset value.
4	PLLSTBIF	PLL stabilization interrupt flag Set by hardware when the PLL is stable and the PLLSTBIE bit is set. Reset when setting the PLLSTBIC bit by software. 0: No PLL stabilization interrupt generated 1: PLL stabilization interrupt generated
3	HXTALSTBIF	HXTAL stabilization interrupt flag Set by hardware when the high speed 4 ~ 48 MHz crystal oscillator clock is stable and the HXTALSTBIE bit is set. Reset when setting the HXTALSTBIC bit by software. 0: No HXTAL stabilization interrupt generated 1: HXTAL stabilization interrupt generated
2	IRC8MSTBIF	IRC8M stabilization interrupt flag Set by hardware when the Internal 8 MHz RC oscillator clock is stable and the IRC8MSTBIE bit is set. Reset when setting the IRC8MSTBIC bit by software. 0: No IRC8M stabilization interrupt generated 1: IRC8M stabilization interrupt generated

1	LXTALSTBIF	LXTAL stabilization interrupt flag Set by hardware when the Low speed 32,768 Hz crystal oscillator clock is stable and the LXTALSTBIE bit is set. Reset when setting the LXTALSTBIC bit by software. 0: No LXTAL stabilization interrupt generated 1: LXTAL stabilization interrupt generated
0	IRC32KSTBIF	IRC32K stabilization interrupt flag Set by hardware when the Internal 32kHz RC oscillator clock is stable and the IRC32KSTBIE bit is set. Reset when setting the IRC32KSTBIC bit by software. 0: No IRC32K stabilization clock ready interrupt generated 1: IRC32K stabilization interrupt generated

4.3.5. AHB1 reset register (RCU_AHB1RST)

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
FFTRST	Reserved							DMAMUXRST	DMA1RST	DMA0RST	Reserved				
rw								rw	rw	rw					

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

Bits	Fields	Descriptions
31	FFTRST	FFT reset This bit is set and reset by software. 0: No reset 1: Reset the FFT
30:24	Reserved	Must be kept at reset value.
23	DMAMUXRST	DMAMUX reset This bit is set and reset by software. 0: No reset 1: Reset the DMAMUX
22	DMA1RST	DMA1 reset This bit is set and reset by software. 0: No reset 1: Reset the DMA1

21	DMA0RST	DMA0 reset This bit is set and reset by software. 0: No reset 1: Reset the DMA0
20:16	Reserved	Must be kept at reset value.
15	CLARST	CLA reset This bit is set and reset by software. 0: No reset 1: Reset the CLA
14:13	Reserved	Must be kept at reset value.
12	CRCRST	CRC reset This bit is set and reset by software. 0: No reset 1: Reset the CRC
11:0	Reserved	Must be kept at reset value.

4.3.6. AHB2 reset register (RCU_AHB2RST)

Address offset: 0x14

Reset value: 0x0000 0000

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

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

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23	PGRST	GPIO port G reset This bit is set and reset by software. 0: No reset 1: Reset the GPIO port G
22	PFRST	GPIO portF reset This bit is set and reset by software. 0: No reset 1: Reset the GPIO port F

21	PERST	GPIO port E reset This bit is set and reset by software. 0: No reset 1: Reset the GPIO port E
20	PDRST	GPIO port D reset This bit is set and reset by software. 0: No reset 1: Reset the GPIO port D
19	PCRST	GPIO port C reset This bit is set and reset by software. 0: No reset 1: Reset the GPIO port C
18	PBRST	GPIO port B reset This bit is set and reset by software. 0: No reset 1: Reset the GPIO port B
17	PARST	GPIO port A reset This bit is set and reset by software. 0: No reset 1: Reset the GPIO port A
16:8	Reserved	Must be kept at reset value.
7	TMURST	TMU reset This bit is set and reset by software. 0: No reset 1: Reset the TMU
6	TRNGRST	TRNG reset This bit is set and reset by software. 0: No reset 1: Reset the TRNG
5:4	Reserved	Must be kept at reset value.
3	CAURST	CAU reset This bit is set and reset by software. 0: No reset 1: Reset the CAU
2	Reserved	Must be kept at reset value.
1	FACRST	FAC reset This bit is set and reset by software. 0: No reset

		1: Reset the FAC
0	Reserved	Must be kept at reset value.

4.3.7. AHB3 reset register (RCU_AHB3RST)

Address offset: 0x18

Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	QSPIRST	QSPI reset This bit is set and reset by software. 0: No reset 1: Reset the QSPI
0	EXMCRST	EXMC reset This bit is set and reset by software. 0: No reset 1: Reset the EXMC

4.3.8. APB1 reset register (RCU_APB1RST)

Address offset: 0x20

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			PMURST	Reserved			I2C3RST	I2C2RST	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	Reserved		WWDGTRST	Reserved	LPTIMERRST	Reserved			TIMER6RST	TIMER5RST	TIMER4RST	TIMER3RST	TIMER2RST	TIMER1RST
rw	rw			rw		rw				rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:29	Reserved	Must be kept at reset value.
28	PMURST	Power control reset This bit is set and reset by software. 0: No reset 1: Reset power control unit
27:25	Reserved	Must be kept at reset value.
24	I2C3RST	I2C3 reset This bit is set and reset by software. 0: No reset 1: Reset the I2C3
23	I2C2RST	I2C2 reset This bit is set and reset by software. 0: No reset 1: Reset the I2C2
22	I2C1RST	I2C1 reset This bit is set and reset by software. 0: No reset 1: Reset the I2C1
21	I2C0RST	I2C0 reset This bit is set and reset by software. 0: No reset 1: Reset the I2C0
20	UART4RST	UART4 reset This bit is set and reset by software. 0: No reset 1: Reset the UART4
19	UART3RST	UART3 reset This bit is set and reset by software. 0: No reset 1: Reset the UART3
18	USART2RST	USART2 reset This bit is set and reset by software. 0: No reset 1: Reset the USART2
17	USART1RST	USART1 reset This bit is set and reset by software.

		0: No reset 1: Reset the USART1
16	Reserved	Must be kept at reset value.
15	SPI2RST	SPI2 reset This bit is set and reset by software. 0: No reset 1: Reset the SPI2
14	SPI1RST	SPI1 reset This bit is set and reset by software. 0: No reset 1: Reset the SPI1
13:12	Reserved	Must be kept at reset value.
11	WWDGTRST	WWDGT reset This bit is set and reset by software. 0: No reset 1: Reset the WWDGT
10	Reserved	Must be kept at reset value.
9	LPTIMERRST	LPTIMER reset This bit is set and reset by software. 0: No reset 1: Reset the LPTIMER
8:6	Reserved	Must be kept at reset value.
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

4.3.9. APB2 reset register (RCU_APB2RST)

Address offset: 0x24

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TRIGSELRST	Reserved	HRTIMERRST	Reserved									HPDFRST	TIMER16RST	TIMER15RST	TIMER14RST
rw		rw										rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMER19RST	SYSCFGRST	Reserved	SPI0RST	Reserved	CAN2RST	CAN1RST	CAN0RST	Reserved			USART0RST	CMPRST	VREFRST	TIMER7RST	TIMER0RST
rw	rw		rw		rw	rw	rw				rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31	TRIGSELRST	TRIGSELreset This bit is set and reset by software. 0: No reset 1: Reset TRIGSEL
30	Reserved	Must be kept at reset value.
29	HRTIMERRST	HRTIMER reset This bit is set and reset by software. 0: No reset 1: Reset HRTIMER
28:20	Reserved	Must be kept at reset value.
19	HPDFRST	HPDF reset This bit is set and reset by software. 0: No reset 1: Reset the HPDF

18	TIMER16RST	TIMER16 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER16
17	TIMER15RST	TIMER15 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER15
16	TIMER14RST	TIMER14 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER14
15	TIMER19RST	TIMER19 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER19
14	SYSCFGRST	SYSCFG reset This bit is set and reset by software. 0: No reset 1: Reset the SYSCFG
13	Reserved	Must be kept at reset value.
12	SPI0RST	SPI0 reset This bit is set and reset by software. 0: No reset 1: Reset the SPI0
11	Reserved	Must be kept at reset value.
10	CAN2RST	CAN2 reset This bit is set and reset by software. 0: No reset 1: Reset the CAN2
9	CAN1RST	CAN1 reset This bit is set and reset by software. 0: No reset 1: Reset the CAN1
8	CAN0RST	CAN0 reset This bit is set and reset by software. 0: No reset 1: Reset the CAN0

7:5	Reserved	Must be kept at reset value.
4	USART0RST	USART0 reset This bit is set and reset by software. 0: No reset 1: Reset the USART0
3	CMPRST	CMP reset This bit is set and reset by software. 0: No reset 1: Reset the CMP
2	VREFRST	VREF reset This bit is set and reset by software. 0: No reset 1: Reset the VREF
1	TIMER7RST	TIMER7 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER7
0	TIMER0RST	TIMER0 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER0

4.3.10. APB3 reset register (RCU_APB3RST)

Address offset: 0x28

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											DAC3RST	DAC2RST	DAC1RST	DAC0RST	DACHOLDRST
											rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				ADC3RST	ADC2RST	ADC1RST	ADC0RST	Reserved							
				rw	rw	rw	rw								

Bits	Fields	Descriptions
31:21	Reserved	Must be kept at reset value.
20	DAC3RST	DAC3 reset This bit is set and reset by software.

		0: No reset 1: Reset DAC3
19	DAC2RST	DAC2 reset This bit is set and reset by software. 0: No reset 1: Reset DAC2
18	DAC1RST	DAC1 reset This bit is set and reset by software. 0: No reset 1: Reset DAC1
17	DAC0RST	DAC0 reset This bit is set and reset by software. 0: No reset 1: Reset DAC0
16	DACHOLDRST	DAC hold clock reset This bit is set and reset by software. The hold clock source is IRC32K. 0: No reset 1: Reset the hold clock
15:12	Reserved	Must be kept at reset value.
11	ADC3RST	ADC3 reset This bit is set and reset by software. 0: No reset 1: Reset the ADC3
10	ADC2RST	ADC2 reset This bit is set and reset by software. 0: No reset 1: Reset the ADC2
9	ADC1RST	ADC1 reset This bit is set and reset by software. 0: No reset 1: Reset the ADC1
8	ADC0RST	ADC0 reset This bit is set and reset by software. 0: No reset 1: Reset the ADC0
7:0	Reserved	Must be kept at reset value.

4.3.11. AHB1 enable register (RCU_AHB1EN)

Address offset: 0x30

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

Bits	Fields	Descriptions
31	FFTEN	FFT clock enable This bit is set and reset by software. 0: Disabled FFT clock 1: Enabled FFT clock
30:24	Reserved	Must be kept at reset value.
23	DMAMUXEN	DMAMUX clock enable This bit is set and reset by software. 0: Disabled DMAMUX clock 1: Enabled DMAMUX clock
22	DMA1EN	DMA1 clock enable This bit is set and reset by software. 0: Disabled DMA1 clock 1: Enabled DMA1 clock
21	DMA0EN	DMA0 clock enable This bit is set and reset by software. 0: Disabled DMA0 clock 1: Enabled DMA0 clock
20:16	Reserved	Must be kept at reset value.
15	CLAEN	CLA clock enable This bit is set and reset by software. 0: Disabled CLA clock 1: Enabled CLA clock
14:13	Reserved	Must be kept at reset value.
12	CRCEN	CRC clock enable This bit is set and reset by software.

0: Disabled CRC clock
1: Enabled CRC clock

11:0	Reserved	Must be kept at reset value.

4.3.12. AHB2 enable register (RCU_AHB2EN)

Address offset: 0x34
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								PGEN	PFEN	PEEN	PDEN	PCEN	PBEN	PAEN	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								TMUEN	TRNGEN	Reserved		CAUEN	Reserved	FACEN	Reserved
								rw	rw			rw		rw	

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23	PGEN	GPIO port G clock enable This bit is set and reset by software. 0: Disabled GPIO port G clock 1: Enabled GPIO port G clock
22	PFEN	GPIO port F clock enable This bit is set and reset by software. 0: Disabled GPIO port F clock 1: Enabled GPIO port F clock
21	PEEN	GPIO port E clock enable This bit is set and reset by software. 0: Disabled GPIO port E clock 1: Enabled GPIO port E clock
20	PDEN	GPIO port D clock enable This bit is set and reset by software. 0: Disabled GPIO port D clock 1: Enabled GPIO port D clock
19	PCEN	GPIO port C clock enable This bit is set and reset by software. 0: Disabled GPIO port C clock 1: Enabled GPIO port C clock
18	PBEN	GPIO port B clock enable

		This bit is set and reset by software. 0: Disabled GPIO port B clock 1: Enabled GPIO port B clock
17	PAEN	GPIO port A clock enable This bit is set and reset by software. 0: Disabled GPIO port A clock 1: Enabled GPIO port A clock
16:8	Reserved	Must be kept at reset value.
7	TMUEN	TMU clock enable This bit is set and reset by software. 0: Disabled TMU clock 1: Enabled TMU clock
6	TRNGEN	TRNG clock enable This bit is set and reset by software. 0: Disabled TRNG clock 1: Enabled TRNG clock
5:4	Reserved	Must be kept at reset value.
3	CAUEN	CRY clock enable This bit is set and reset by software. 0: Disabled CAU clock 1: Enabled CAU clock
2	Reserved	Must be kept at reset value.
1	FACEN	FAC clock enable This bit is set and reset by software. 0: Disabled FAC clock 1: Enabled FAC clock

4.3.13. AHB3 enable register (RCU_AHB3EN)

Address offset: 0x38

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														QSPIEN	EXMCEN
														rw	rw

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	QSPIEN	QSPI clock enable This bit is set and reset by software. 0: Disabled QSPI clock 1: Enabled QSPI clock
0	EXMCEN	EXMC clock enable This bit is set and reset by software. 0: Disabled EXMC clock 1: Enabled EXMC clock

4.3.14. APB1 enable register (RCU_APB1EN)

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			PMUEN	Reserved			I2C3EN	I2C2EN	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	Reserved		WWDGTEN	Reserved	LPTIMEREN	Reserved			TIMER6EN	TIMER5EN	TIMER4EN	TIMER3EN	TIMER2EN	TIMER1EN
rw	rw			rw		rw				rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:29	Reserved	Must be kept at reset value.
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	I2C3EN	I2C3 clock enable This bit is set and reset by software. 0: Disabled I2C3 clock 1: Enabled I2C3 clock
23	I2C2EN	I2C2 clock enable This bit is set and reset by software. 0: Disabled I2C2 clock

		1: Enabled I2C2 clock
22	I2C1EN	I2C1 clock enable This bit is set and reset by software. 0: Disabled I2C1 clock 1: Enabled I2C1 clock
21	I2C0EN	I2C0 clock enable This bit is set and reset by software. 0: Disabled I2C0 clock 1: Enabled I2C0 clock
20	UART4EN	UART4 clock enable This bit is set and reset by software. 0: Disabled UART4 clock 1: Enabled UART4 clock
19	UART3EN	UART3 clock enable This bit is set and reset by software. 0: Disabled UART3 clock 1: Enabled UART3 clock
18	USART2EN	USART2 clock enable This bit is set and reset by software. 0: Disabled USART2 clock 1: Enabled USART2 clock
17	USART1EN	USART1 clock enable This bit is set and reset by software. 0: Disabled USART1 clock 1: Enabled USART1 clock
16	Reserved	Must be kept at reset value.
15	SPI2EN	SPI2 clock enable This bit is set and reset by software. 0: Disabled SPI2 clock 1: Enabled SPI2 clock
14	SPI1EN	SPI1 clock enable This bit is set and reset by software. 0: Disabled SPI1 clock 1: Enabled SPI1 clock
13:12	Reserved	Must be kept at reset value.
11	WWDGTEN	WWDGT clock enable This bit is set and reset by software. 0: Disabled WWDGT clock

		1: Enabled WWDGT clock
10	Reserved	Must be kept at reset value.
9	LPTIMEREN	LPTIMER clock enable This bit is set and reset by software. 0: Disabled LPTIMER clock 1: Enabled LPTIMER clock
8:6	Reserved	Must be kept at reset value.
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

4.3.15. APB2 enable register (RCU_APB2EN)

Address offset: 0x44

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

TRIGSELEN	Reserved	HRTIMEREN	Reserved	HPDFEN	TIMER16EN	TIMER15EN	TIMER14EN
rw		rw		rw	rw	rw	rw
15	14	13	12 11 10 9 8 7 6 5 4	3	2	1	0

TIMER19EN	SYSCFGEN	Reserved	SPI0EN	Reserved	CAN2EN	CAN1EN	CAN0EN	Reserved	USART0EN	CMPEN	VREFEN	TIMER7EN	TIMER0EN
rw	rw		rw		rw	rw	rw		rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31	TRIGSELEN	TRIGSEL clock enable This bit is set and reset by software. 0: Disabled TRIGSEL clock 1: Enabled TRIGSEL clock
30	Reserved	Must be kept at reset value.
29	HRTIMEREN	HRTIMER clock enable This bit is set and reset by software. 0: Disabled HRTIMER clock 1: Enabled HRTIMER clock
28:20	Reserved	Must be kept at reset value.
19	HPDFEN	HPDF clock enable This bit is set and reset by software. 0: Disabled HPDF clock 1: Enabled HPDF clock
18	TIMER16EN	TIMER16 clock enable This bit is set and reset by software. 0: Disabled TIMER16 clock 1: Enabled TIMER16 clock
17	TIMER15EN	TIMER15 clock enable This bit is set and reset by software. 0: Disabled TIMER15 clock 1: Enabled TIMER15 clock
16	TIMER14EN	TIMER14 clock enable This bit is set and reset by software. 0: Disabled TIMER14 clock 1: Enabled TIMER14 clock
15	TIMER19EN	TIMER19 clock enable This bit is set and reset by software. 0: Disabled TIMER19 clock 1: Enabled TIMER19 clock

14	SYSCFGEN	SYSCFG clock enable This bit is set and reset by software. 0: Disabled SYSCFG clock 1: Enabled SYSCFG clock
13	Reserved	Must be kept at reset value.
12	SPI0EN	SPI0 clock enable This bit is set and reset by software. 0: Disabled SPI0 clock 1: Enabled SPI0 clock
11	Reserved	Must be kept at reset value.
10	CAN2EN	CAN2 clock enable This bit is set and reset by software. 0: Disabled CAN2 clock 1: Enabled CAN2 clock
9	CAN1EN	CAN1 clock enable This bit is set and reset by software. 0: Disabled CAN1 clock 1: Enabled CAN1 clock
8	CAN0EN	CAN0 clock enable This bit is set and reset by software. 0: Disabled CAN0 clock 1: Enabled CAN0 clock
7:5	Reserved	Must be kept at reset value.
4	USART0EN	USART0 clock enable This bit is set and reset by software. 0: Disabled USART0 clock 1: Enabled USART0 clock
3	CMPEN	CMP clock enable This bit is set and reset by software. 0: Disabled CMP clock 1: Enabled CMP clock
2	VREFEN	VREF clock enable This bit is set and reset by software. 0: Disabled VREF clock 1: Enabled VREF clock
1	TIMER7EN	TIMER7 clock enable This bit is set and reset by software. 0: Disabled TIMER7 clock

		1: Enabled TIMER7 clock
0	TIMER0EN	TIMER0 clock enable This bit is set and reset by software. 0: Disabled TIMER0 clock 1: Enabled TIMER0 clock

4.3.16. APB3 enable register (RCU_APB3EN)

Address offset: 0x48

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											DAC3EN	DAC2EN	DAC1EN	DAC0EN	DACHOLDEN
											rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				ADC3EN	ADC2EN	ADC1EN	ADC0EN	Reserved							
				rw	rw	rw	rw								

Bits	Fields	Descriptions
31:21	Reserved	Must be kept at reset value.
20	DAC3EN	DAC3 clock enable This bit is set and reset by software. 0: Disabled DAC3 clock 1: Enabled DAC3 clock
19	DAC2EN	DAC2 clock enable This bit is set and reset by software. 0: Disabled DAC2 clock 1: Enabled DAC2 clock
18	DAC1EN	DAC1 clock enable This bit is set and reset by software. 0: Disabled DAC1 clock 1: Enabled DAC1 clock
17	DAC0EN	DAC0 clock enable This bit is set and reset by software. 0: Disabled DAC0 clock 1: Enabled DAC0 clock
16	DACHOLDEN	DAC hold clock clock enable This bit is set and reset by software. The hold clock source is IRC32K. 0: Disabled DAC hold clock

		1: Enabled DAC hold clock
15:12	Reserved	Must be kept at reset value.
11	ADC3EN	ADC3 clock enable This bit is set and reset by software. 0: Disabled ADC3 clock 1: Enabled ADC3 clock
10	ADC2EN	ADC2 clock enable This bit is set and reset by software. 0: Disabled ADC2 clock 1: Enabled ADC2 clock
9	ADC1EN	ADC1 clock enable This bit is set and reset by software. 0: Disabled ADC1 clock 1: Enabled ADC1 clock
8	ADC0EN	ADC0 clock enable This bit is set and reset by software. 0: Disabled ADC0 clock 1: Enabled ADC0 clock
7:0	Reserved	Must be kept at reset value.

4.3.17. AHB1 sleep and deep-sleep mode enable register (RCU_AHB1SPDPEN)

Address offset: 0x50

Reset value: 0x80EB 9100

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
FFTSPDPEN	Reserved							DMAMUXSPDPEN	DMA1SPDPEN	DMA0SPDPEN	Reserved	TCMSRAMSPDPEN	Reserved	SRAM1SPDPEN	SRAM0SPDPEN
rw								rw	rw	rw		rw		rw	rw

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

Bits	Fields	Descriptions
31	FFTSPDPEN	FFT clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled FFT clock when sleep and deep sleep mode

		1: Enabled FFT clock when sleep and deep sleep mode
30:24	Reserved	Must be kept at reset value.
23	DMAMUXSPDPEN	DMAMUX clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled DMAMUX clock when sleep and deep sleep mode 1: Enabled DMAMUX clock when sleep and deep sleep mode
22	DMA1SPDPEN	DMA1 clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled DMA1 clock when sleep and deep sleep mode 1: Enabled DMA1 clock when sleep and deep sleep mode
21	DMA0SPDPEN	DMA0 clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled DMA0 clock when sleep and deep sleep mode 1: Enabled DMA0 clock when sleep and deep sleep mode
20	Reserved	Must be kept at reset value.
19	TCMSRAMSPDPEN	TCMSRAM enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled TCMSRAM clock when sleep and deep sleep mode 1: Enabled TCMSRAM clock when sleep and deep sleep mode
18	Reserved	Must be kept at reset value.
17	SRAM1SPDPEN	SRAM1 enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled SRAM1 clock when sleep and deep sleep mode 1: Enabled SRAM1 clock when sleep and deep sleep mode
16	SRAM0SPDPEN	SRAM0 enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled SRAM0 clock when sleep and deep sleep mode 1: Enabled SRAM0 clock when sleep and deep sleep mode
15	CLASPDPEN	CLA clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled CLA clock when sleep and deep sleep mode 1: Enabled CLA clock when sleep and deep sleep mode
14:13	Reserved	Must be kept at reset value.
12	CRCSPDPEN	CRC clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled CRC clock when sleep and deep sleep mode 1: Enabled CRC clock when sleep and deep sleep mode

11:9	Reserved	Must be kept at reset value.
8	FMCSPDPEN	FMC clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled FMC clock when sleep and deep sleep mode 1: Enabled FMC clock when sleep and deep sleep mode
7:0	Reserved	Must be kept at reset value.

4.3.18. AHB2 sleep and deep-sleep mode enable register (RCU_AHB2SPDPEN)

Address offset: 0x54

Reset value: 0x00FE 00CA

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								PGSPDPEN	PFSPDPEN	PESPDPEN	PDSPDPEN	PCSPDPEN	PBSPDPEN	PASPDPEN	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								TMUSPDPEN	TRNGSPDPEN	Reserved		CAUSPDPEN	Reserved	FACSPDPEN	Reserved
								rw	rw			rw		rw	

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23	PGSPDPEN	GPIO port G clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled GPIO port G clock when sleep and deep sleep mode 1: Enabled GPIO port G clock when sleep and deep sleep mode
22	PFSPDPEN	GPIO port F clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled GPIO port F clock when sleep and deep sleep mode 1: Enabled GPIO port F clock when sleep and deep sleep mode
21	PESPDPEN	GPIO port E clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled GPIO port E clock when sleep and deep sleep mode 1: Enabled GPIO port E clock when sleep and deep sleep mode
20	PDSPDPEN	GPIO port D clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled GPIO port D clock when sleep and deep sleep mode 1: Enabled GPIO port D clock when sleep and deep sleep mode

19	PCSPDPEN	GPIO port C clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled GPIO port C clock when sleep and deep sleep mode 1: Enabled GPIO port C clock when sleep and deep sleep mode
18	PBSPDPEN	GPIO port B clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled GPIO port B clock when sleep and deep sleep mode 1: Enabled GPIO port B clock when sleep and deep sleep mode
17	PASPDPEN	GPIO port A clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled GPIO port A clock when sleep and deep sleep mode 1: Enabled GPIO port A clock when sleep and deep sleep mode
16:8	Reserved	Must be kept at reset value.
7	TMUSPDPEN	TMU clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled TMU clock when sleep and deep sleep mode 1: Enabled TMU clock when sleep and deep sleep mode
6	TRNGSPDPEN	TRNG clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled TRNG clock when sleep and deep sleep mode 1: Enabled TRNG clock when sleep and deep sleep mode
5:4	Reserved	Must be kept at reset value.
3	CAUSPDPEN	CAU clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled CAU clock when sleep and deep sleep mode 1: Enabled CAU clock when sleep and deep sleep mode
2	Reserved	Must be kept at reset value.
1	FACSPDPEN	FAC clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled FAC clock when sleep and deep sleep mode 1: Enabled FAC clock when sleep and deep sleep mode
0	Reserved	Must be kept at reset value.

4.3.19. AHB3 sleep and deep-sleep mode enable register (RCU_AHB3SPDPEN)

Address offset: 0x58

Reset value: 0x0000 0003

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

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

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

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	QSPISPDPEN	QSPI clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled QSPI clock when sleep and deep sleep mode 1: Enabled QSPI clock when sleep and deep sleep mode
0	EXMCSPDPEN	EXMC clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled EXMC clock when sleep and deep sleep mode 1: Enabled EXMC clock when sleep and deep sleep mode

4.3.20. APB1 sleep and deep-sleep mode enable register (RCU_APB1SPDPEN)

Address offset: 0x60

Reset value: 0x11FE CA3F

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			PMUSPDPEN	Reserved			I2C3SPDPEN	I2C2SPDPEN	I2C1SPDPEN	I2C0SPDPEN	UART4SPDPEN	UART3SPDPEN	USART2SPDPEN	USART1SPDPEN	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
SPI2SPDPEN	SPI1SPDPEN	Reserved		WWDGTSPDPEN	Reserved	LPTIMERSPDPEN	Reserved			TIMER6SPDPEN	TIMER5SPDPEN	TIMER4SPDPEN	TIMER3SPDPEN	TIMER2SPDPEN	TIMER1SPDPEN
rw	rw			rw		rw				rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:29	Reserved	Must be kept at reset value.
28	PMUSPDPEN	PMU clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled PMU clock when sleep and deep sleep mode 1: Enabled PMU clock when sleep and deep sleep mode
27:25	Reserved	Must be kept at reset value.

24	I2C3SPDPEN	I2C3 clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled I2C3 clock when sleep and deep sleep mode 1: Enabled I2C3 clock when sleep and deep sleep mode
23	I2C2SPDPEN	I2C2 clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled I2C2 clock when sleep and deep sleep mode 1: Enabled I2C2 clock when sleep and deep sleep mode
22	I2C1SPDPEN	I2C1 clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled I2C1 clock when sleep and deep sleep mode 1: Enabled I2C1 clock when sleep and deep sleep mode
21	I2C0SPDPEN	I2C0 clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled I2C0 clock when sleep and deep sleep mode 1: Enabled I2C0 clock when sleep and deep sleep mode
20	UART4SPDPEN	UART4 clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled UART4 clock when sleep and deep sleep mode 1: Enabled UART4 clock when sleep and deep sleep mode
19	UART3SPDPEN	UART3 clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled UART3 clock when sleep and deep sleep mode 1: Enabled UART3 clock when sleep and deep sleep mode
18	USART2SPDPEN	USART2 clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled USART2 clock when sleep and deep sleep mode 1: Enabled USART2 clock when sleep and deep sleep mode
17	USART1SPDPEN	USART1 clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled USART1 clock when sleep and deep sleep mode 1: Enabled USART1 clock when sleep and deep sleep mode
16	Reserved	Must be kept at reset value.
15	SPI2SPDPEN	SPI2 clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled SPI2 clock when sleep and deep sleep mode 1: Enabled SPI2 clock when sleep and deep sleep mode
14	SPI1SPDPEN	SPI1 clock enable when sleep and deep sleep mode

		This bit is set and reset by software. 0: Disabled SPI1 clock when sleep and deep sleep mode 1: Enabled SPI1 clock when sleep and deep sleep mode
13:12	Reserved	Must be kept at reset value.
11	WWDGTSPDPEN	WWDGT clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled WWDGT clock when sleep and deep sleep mode 1: Enabled WWDGT clock when sleep and deep sleep mode
10	Reserved	Must be kept at reset value.
9	LPTIMERSPDPEN	LPTIMER clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled LPTIMER clock when sleep and deep sleep mode 1: Enabled LPTIMER clock when sleep and deep sleep mode
8:6	Reserved	Must be kept at reset value.
5	TIMER6SPDPEN	TIMER6 clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled TIMER6 clock when sleep and deep sleep mode 1: Enabled TIMER6 clock when sleep and deep sleep mode
4	TIMER5SPDPEN	TIMER5 clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled TIMER5 clock when sleep and deep sleep mode 1: Enabled TIMER5 clock when sleep and deep sleep mode
3	TIMER4SPDPEN	TIMER4 clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled TIMER4 clock when sleep and deep sleep mode 1: Enabled TIMER4 clock when sleep and deep sleep mode
2	TIMER3SPDPEN	TIMER3 clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled TIMER3 clock when sleep and deep sleep mode 1: Enabled TIMER3 clock when sleep and deep sleep mode
1	TIMER2SPDPEN	TIMER2 clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled TIMER2 clock when sleep and deep sleep mode 1: Enabled TIMER2 clock when sleep and deep sleep mode
0	TIMER1SPDPEN	TIMER1 clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled TIMER1 clock when sleep and deep sleep mode

1: Enabled TIMER1 clock when sleep and deep sleep mode

4.3.21. APB2 sleep and deep-sleep mode enable register (RCU_APB2SPDPEN)

Address offset: 0x64

Reset value: 0xA00F D71F

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
TRIGSELSPDPEN	Reserved	HRTIMERSPDPEN	Reserved									HPDFSPDPEN	TIMER16SPDPEN	TIMER15SPDPEN	TIMER14SPDPEN
rw		rw										rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMER19SPDPEN	SYSCFGSPDPEN	Reserved	SPI0SPDPEN	Reserved	CAN2SPDPEN	CAN1SPDPEN	CAN0SPDPEN	Reserved			USART0SPDPEN	CMPSPDPEN	VREFSPDPEN	TIMER7SPDPEN	TIMER0SPDPEN
rw	rw		rw		rw	rw	rw				rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31	TRIGSELSPDPEN	TRIGSEL clock enable when sleep mode This bit is set and reset by software. 0: Disabled TRIGSEL clock when sleep and deep sleep mode 1: Enabled TRIGSEL clock when sleep and deep sleep mode
30	Reserved	Must be kept at reset value.
29	HRTIMERSPDPEN	HRTIMER clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled HRTIMER clock when sleep and deep sleep mode 1: Enabled HRTIMER clock when sleep and deep sleep mode
28:20	Reserved	Must be kept at reset value.
19	HPDFSPDPEN	HPDF clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled HPDF clock when sleep and deep sleep mode 1: Enabled HPDF clock when sleep and deep sleep mode
18	TIMER16SPDPEN	TIMER16 clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled TIMER16 clock when sleep and deep sleep mode 1: Enabled TIMER16 clock when sleep and deep sleep mode
17	TIMER15SPDPEN	TIMER15 clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled TIMER15 clock when sleep and deep sleep mode

		1: Enabled TIMER15 clock when sleep and deep sleep mode
16	TIMER14SPDPEN	TIMER14 clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled TIMER14 clock when sleep and deep sleep mode 1: Enabled TIMER14 clock when sleep and deep sleep mode
15	TIMER19SPDPEN	TIMER19 clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled TIMER19 clock when sleep and deep sleep mode 1: Enabled TIMER19 clock when sleep and deep sleep mode
14	SYSCFGSPDPEN	SYSCFG clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled SYSCFG clock when sleep and deep sleep mode 1: Enabled SYSCFG clock when sleep and deep sleep mode
13	Reserved	Must be kept at reset value.
12	SPI0SPDPEN	SPI0 clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled SPI0 clock when sleep and deep sleep mode 1: Enabled SPI0 clock when sleep and deep sleep mode
11	Reserved	Must be kept at reset value.
10	CAN2SPDPEN	CAN2 clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled CAN2 clock when sleep and deep sleep mode 1: Enabled CAN2 clock when sleep and deep sleep mode
9	CAN1SPDPEN	CAN1 clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled CAN1 clock when sleep and deep sleep mode 1: Enabled CAN1 clock when sleep and deep sleep mode
8	CAN0SPDPEN	CAN0 clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled CAN0 clock when sleep and deep sleep mode 1: Enabled CAN0 clock when sleep and deep sleep mode
7:5	Reserved	Must be kept at reset value.
4	USART0SPDPEN	USART0 clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled USART0 clock when sleep and deep sleep mode 1: Enabled USART0 clock when sleep and deep sleep mode
3	CMPSPDPEN	CMP clock enable when sleep and deep sleep mode

		This bit is set and reset by software. 0: Disabled CMP clock when sleep and deep sleep mode 1: Enabled CMP clock when sleep and deep sleep mode
2	VREFSPDPEN	VREF clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled VREF clock when sleep and deep sleep mode 1: Enabled VREF clock when sleep and deep sleep mode
1	TIMER7SPDPEN	TIMER7 clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled TIMER7 clock when sleep and deep sleep mode 1: Enabled TIMER7 clock when sleep and deep sleep mode
0	TIMER0SPDPEN	TIMER0 clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled TIMER0 clock when sleep and deep sleep mode 1: Enabled TIMER0 clock when sleep and deep sleep mode

4.3.22. APB3 sleep and deep-sleep mode enable register (RCU_APB3SPDPEN)

Address offset: 0x68

Reset value: 0x001F 0F00

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											DAC3SPDPEN	DAC2SPDPEN	DAC1SPDPEN	DAC0SPDPEN	DACHOLDSPDPEN
											rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				ADC3SPDPEN	ADC2SPDPEN	ADC1SPDPEN	ADC0SPDPEN	Reserved							
				rw	rw	rw	rw								

Bits	Fields	Descriptions
31:21	Reserved	Must be kept at reset value.
20	DAC3SPDPEN	DAC3 clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled DAC3 clock when sleep and deep sleep mode 1: Enabled DAC3 clock when sleep and deep sleep mode
19	DAC2SPDPEN	DAC2 clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled DAC2 clock when sleep and deep sleep mode

		1: Enabled DAC2 clock when sleep and deep sleep mode
18	DAC1SPDPEN	DAC1 clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled DAC1 clock when sleep and deep sleep mode 1: Enabled DAC1 clock when sleep and deep sleep mode
17	DAC0SPDPEN	DAC0 clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled DAC0 clock when sleep and deep sleep mode 1: Enabled DAC0 clock when sleep and deep sleep mode
16	DACHOLDSPDPEN	DAC hold clock clock enable when sleep and deep sleep mode This bit is set and reset by software. The hold clock source is IRC32K. 0: Disabled DAC hold clock when sleep and deep sleep mode 1: Enabled DAC hold clock when sleep and deep sleep mode
15:12	Reserved	Must be kept at reset value.
11	ADC3SPDPEN	ADC3 clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled ADC3 clock when sleep and deep sleep mode 1: Enabled ADC3 clock when sleep and deep sleep mode
10	ADC2SPDPEN	ADC2 clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled ADC2 clock when sleep and deep sleep mode 1: Enabled ADC2 clock when sleep and deep sleep mode
9	ADC1SPDPEN	ADC1 clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled ADC1 clock when sleep and deep sleep mode 1: Enabled ADC1 clock when sleep and deep sleep mode
8	ADC0SPDPEN	ADC0 clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled ADC0 clock when sleep and deep sleep mode 1: Enabled ADC0 clock when sleep and deep sleep mode
7:0	Reserved	Must be kept at reset value.

4.3.23. Backup domain control register (RCU_BDCTL)

Address offset: 0x70

Reset value: 0x0000 0018, reset by backup domain reset except LSCKOUTSEL, LSCKOUTEN and BKPRST which are reset only by RTC domain power-on reset.

Note: The LXTALEN, LXTALBPS, RTCSRC and RTCEN bits of the backup domain control

register (RCU_BDCTL) are only reset after a backup domain reset. These bits can be modified only when the BKPWEN bit in the power control register (PMU_CTL) is set.

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						LSCKOUTSEL	LSCKOUTEN	Reserved							BKPRST
						rw	rw								rw

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

Bits	Fields	Descriptions
31:26	Reserved	Must be kept at reset value.
25	LSCKOUTSEL	Low speed clock output selection 0: IRC32K selected 1: LXTAL selected
24	LSCKOUTEN	Low speed clock output enalbe 0: Low speed clock output(LSCKOUT) disable 1: Low speed clock output(LSCKOUT) enable
23:17	Reserved	Must be kept at reset value.
16	BKPRST	Backup domain reset This bit is set and reset by software. 0: No reset 1: Resets backup domain
15	RTCEN	RTC clock enable This bit is set and reset by software. 0: Disabled RTC clock 1: Enabled RTC clock
14:10	Reserved	Must be kept at reset value.
9:8	RTCSRC[1:0]	RTC clock entry selection Set and reset by software to control the RTC clock source. Once the RTC clock source has been selected, it cannot be changed anymore unless the Backup domain is reset. 00: No clock selected 01: CK_LXTAL selected as RTC source clock 10: CK_IRC32K selected as RTC source clock 11: (CK_HXTAL / 32) selected as RTC source clock
7	LXTALSTBRST	LXTAL stabilization reset

		Set and reset by software. 0: LXTAL stabilization no reset 1: LXTAL stabilization reset
6	LCKMD	LXTALclock 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 or LXTAL gets right. 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 system 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: Prohibit 01: Low driving capability 10: Medium driving capability 11: High 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

4.3.24. Reset source/clock register (RCU_RSTSCK)

Address offset: 0x74

Reset value: 0x0E00 0000, all reset flags reset by power reset only, RSTFC / IRC32KEN 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	OBLRST	Reserved						
r	r	r	r	r	r	r	rw	r							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														IRC32KSTB	IRC32KEN
														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

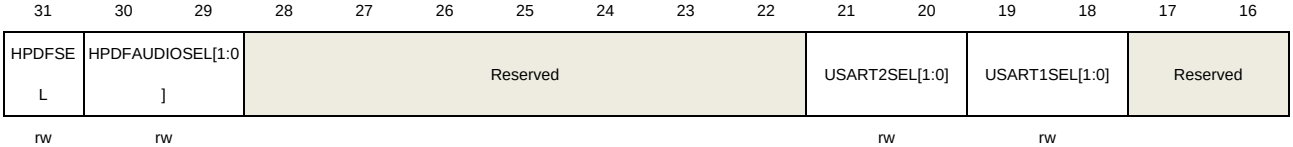
		1: Power reset generated
26	EPRSTF	External PIN reset flag Set by hardware when an External PIN reset generated. Reset by writing 1 to the RSTFC bit. 0: No External PIN reset generated 1: External PIN reset generated
25	BORRSTF	BOR reset flag Set by hardware when a BOR reset generated. Reset by writing 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	OBLRSTF	Option byte loader reset flag Set by hardware when an option byte loader generated. Reset by writing 1 to the RSTFC bit. 0: No option byte loader reset generated 1: Option byte loader reset generated
22:2	Reserved	Must be kept at reset value.
1	IRC32KSTB	IRC32K stabilization flag Set by hardware to indicate if the IRC32K output clock is stable and ready for use. 0: IRC32K is not stable 1: IRC32K is stable
0	IRC32KEN	IRC32K enable Set and reset by software. 0: Disable IRC32K 1: Enable IRC32K

4.3.25. Clock configuration register 1 (RCU_CFG1)

Address offset: 0x8C

Reset value: 0x0000 0000

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



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		CAN2SEL[1:0]		CAN1SEL[1:0]		CAN0SEL[1:0]		Reserved						USART0SEL[1:0]	
		rw		rw		rw								rw	

Bits	Fields	Descriptions
31	HPDFSEL	HPDF clock source selection Set and reset by software to control the HPDF clock source 0: CK_APB2 selected as HPDF source clock 1: CK_AHB selected as HPDF source clock
30:29	HPDFAUDIOSEL[1:0]	HPDF AUDIO clock source selection Set and reset by software to control the HPDF AUDIO clock source. 00: CK_PLLQ output clock selected as HPDF AUDIO source clock 01: External HPDF_CKIN pin selected as HPDF AUDIO source clock 10: IRC8M selected as HPDF AUDIO source clock 11: Reserved
28:22	Reserved	Must be kept at reset value.
21:20	USART2SEL[1:0]	USART2 clock source selection Set and reset by software to control the USART2 clock source 00: CK_APB1 selected as USART2 source clock 01: CK_SYS selected as USART2 source clock 10: CK_LXTAL selected as USART2 source clock 11: CK_IRC8M selected as USART2 source clock
19:18	USART1SEL[1:0]	USART1 clock source selection Set and reset by software to control the USART1 clock source 00: CK_APB1 selected as USART1 source clock 01: CK_SYS selected as USART1 source clock 10: CK_LXTAL selected as USART1 source clock 11: CK_IRC8M selected as USART1 source clock
17:14	Reserved	Must be kept at reset value.
13:12	CAN2SEL[1:0]	CAN2 clock source selection Set and reset by software to control the CAN2 clock source 00: CK_IRC8M selected as CAN2 source clock 01: CK_APB2 selected as CAN2 source clock 10: CK_PLLQ selected as CAN2 source clock 11: CK_HXTAL selected as CAN2 source clock
11:10	CAN1SEL[1:0]	CAN1 clock selection Set and reset by software to control the CAN1 clock source 00: CK_IRC8M selected as CAN1 source clock 01: CK_APB2 selected as CAN1 source clock 10: CK_PLLQ selected as CAN1 source clock

		11: CK_HXTAL selected as CAN1 source clock
9:8	CAN0SEL[1:0]	CAN0 clock source selection Set and reset by software to control the CAN0 clock source 00: CK_IRC8M selected as CAN0 source clock 01: CK_APB2 selected as CAN0 source clock 10: CK_PLLQ selected as CAN0 source clock 11: CK_HXTAL selected as CAN0 source clock
7:2	Reserved	Must be kept at reset value.
1:0	USART0SEL[1:0]	USART0 clock source selection Set and reset by software to control the USART0 clock source 00: CK_APB2 selected as USART0 source clock 01: CK_SYS selected as USART0 source clock 10: CK_LXTAL selected as USART0 source clock 11: CK_IRC8M selected as USART0 source clock

4.3.26. Clock configuration register 2 (RCU_CFG2)

Address offset: 0x90

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		ADC3SEL[1:0]		ADC0_1_2SEL[1:0]		Reserved						HRTIME RSEL	Reserved	QSPISEL[1:0]	
		rw		rw								rw		rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRNGPSC[3:0]				Reserved	LPTIMERSEL[1:0]		Reserved	I2C3SEL[1:0]		I2C2SEL[1:0]		I2C1SEL[1:0]		I2C0SEL[1:0]	
rw				rw				rw		rw		rw		rw	

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29:28	ADC3SEL[1:0]	ADC3 clock source selection Set and reset by software to control the ADC3 clock source 00: Reserved 01: CK_PLLR selected as ADC3 source clock 10: CK_SYS selected as ADC3 source clock 11: Reserved
27:26	ADC0_1_2SEL[1:0]	ADC0 / 1 / 2 clock source selection Set and reset by software to control the ADC0 / 1 / 2 clock source 00: Reserved 01: CK_PLLR selected as ADC0 / 1 / 2 source clock

		10: CK_SYS selected as ADC0 / 1 / 2 source clock 11: Reserved
25:20	Reserved	Must be kept at reset value.
19	HRTIMERSEL	HRTIMER clock source selection Set and reset by software to control the HRTIMER clock source. 0: CK_APB2 selected as HRTIMER source clock 1: CK_SYS selected as HRTIMER source clock
18	Reserved	Must be kept at reset value.
17:16	QSPISEL[1:0]	QSPI clock source selection Set and reset by software to control the QSPI clock source. 00: CK_SYS selected as QSPI source clock 01: CK_IRC8M selected as QSPI source clock 10: CK_PLLQ selected as QSPI source clock 11: CK_PLLR selected as QSPI source clock
15:12	TRNGPSC[3:0]	TRNG prescaler selection Set and reset by software to control the TRNG clock division ratio. 0000: Reserved 0001: Reserved 0010: CK_PLLQ / 2 0011: CK_PLLQ / 3 0100: CK_PLLQ / 4 … 1111: CK_PLLQ / 15
11	Reserved	Must be kept at reset value.
10:9	LPTIMERSEL[1:0]	LPTIMER clock source selection Set and reset by software to control the LPTIMER clock source. 00: CK_ APB1 selected as LPTIMER source clock 01: CK_ IRC32K selected as LPTIMER source clock 10: CK_ LXTAL selected as LPTIMER source clock 11: CK_ IRC8M selected as LPTIMER source clock
8	Reserved	Must be kept at reset value.
7:6	I2C3SEL[1:0]	I2C3 clock source selection Set and reset by software to control the I2C3 clock source. 00: CK_APB1 selected as I2C3 source clock 01: CK_SYS selected as I2C3 source clock 10: CK_IRC8M selected as I2C3 source clock 11: Reserved
5:4	I2C2SEL[1:0]	I2C2 clock source selection

		Set and reset by software to control the I2C2 clock source. 00: CK_APB1 selected as I2C2 source clock 01: CK_SYS selected as I2C2 source clock 10: CK_IRC8M selected as I2C2 source clock 11: Reserved
3:2	I2C1SEL[1:0]	I2C1 clock source selection Set and reset by software to control the I2C1 clock source. 00: CK_APB1 selected as I2C1 source clock 01: CK_SYS selected as I2C1 source clock 10: CK_IRC8M selected as I2C1 source clock 11: Reserved
1:0	I2C0SEL[1:0]	I2C0 clock source selection Set and reset by software to control the I2C0 clock source. 00: CK_APB1 selected as I2C0 source clock 01: CK_SYS selected as I2C0 source clock 10: CK_IRC8M selected as I2C0 source clock 11: Reserved

5. EXTI introduction Interrupt / event controller (EXTI)

5.1. 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 controls power management. It's tightly coupled to the processer core. More details about NVIC could refer to the technical reference manual of Cortex®-M33.

EXTI (interrupt / event controller) contains up to 39 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.

5.2. Characteristics

- Cortex®-M33 system exception.
- Up to 137 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 39 independent edge detectors in EXTI.
- Three trigger types: rising, falling and both edges.
- Software interrupt or event trigger.
- Trigger sources configurable.

5.3. 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 following tables list all exception types.

Table 5-1. 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

The SysTick calibration value is 25000 and SysTick clock frequency is fixed to CK_SYS*0.125. So this will give a 1ms SysTick interrupt if CK_SYS is configured to 200MHz.

Table 5-2. Interrupt vector table

Interrupt Number	Vector Number	Peripheral Interrupt Description	Vector Address
IRQ 0	16	WWDGT interrupt	0x0000_0040
IRQ 1	17	LVD / VAVD / VOVD / VUVD through EXTI line16 interrupt	0x0000_0044
IRQ 2	18	RTC Tamper and TimeStamp from EXTI line18 Interrupt, LXTAL clock stuck interrupt	0x0000_0048
IRQ 3	19	RTC wakeup from EXTI line19 interrupt	0x0000_004C
IRQ 4	20	FMC global interrupt	0x0000_0050
IRQ 5	21	RCU global interrupt	0x0000_0054
IRQ 6	22	EXTI line0 interrupt	0x0000_0058
IRQ 7	23	EXTI line1 interrupt	0x0000_005C
IRQ 8	24	EXTI line2 interrupt	0x0000_0060
IRQ 9	25	EXTI line3 interrupt	0x0000_0064
IRQ 10	26	EXTI line4 interrupt	0x0000_0068
IRQ 11	27	DMA0 channel0 global interrupt	0x0000_006C
IRQ 12	28	DMA0 channel1 global interrupt	0x0000_0070

Interrupt Number	Vector Number	Peripheral Interrupt Description	Vector Address
IRQ 13	29	DMA0 channel2 global interrupt	0x0000_0074
IRQ 14	30	DMA0 channel3 global interrupt	0x0000_0078
IRQ 15	31	DMA0 channel4 global interrupt	0x0000_007C
IRQ 16	32	DMA0 channel5 global interrupt	0x0000_0080
IRQ 17	33	DMA0 channel6 global interrupt	0x0000_0084
IRQ 18	34	ADC0 and ADC1 global interrupt	0x0000_0088
IRQ 19-22	35-38	Reserved	0x0000_008C-0x0000_0098
IRQ 23	39	EXTI line5-9 interrupt	0x0000_009C
IRQ 24	40	TIMER0 break interrupt	0x0000_00A0
IRQ 25	41	TIMER0 update interrupt	0x0000_00A4
IRQ 26	42	TIMER0 trigger and commutation interrupt / direction change interrupt / index interrupt	0x0000_00A8
IRQ 27	43	TIMER0 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 and wakeup through EXTI line31 interrupt	0x0000_00BC
IRQ 32	48	I2C0 error interrupt	0x0000_00C0
IRQ 33	49	I2C1 event interrupt and wakeup through EXTI line32 interrupt	0x0000_00C4
IRQ 34	50	I2C1 error interrupt	0x0000_00C8
IRQ 35	51	SPI0 global interrupt	0x0000_00CC
IRQ 36	52	SPI1 global interrupt	0x0000_00D0
IRQ 37	53	USART0 global interrupt and wakeup through EXTI line28 interrupt	0x0000_00D4
IRQ 38	54	USART1 global interrupt and wakeup through EXTI line29 interrupt	0x0000_00D8
IRQ 39	55	USART2 global interrupt and wakeup through EXTI line30 interrupt	0x0000_00DC
IRQ 40	56	EXTI Line10-15 interrupt	0x0000_00E0
IRQ 41	57	RTC alarm from EXTI line17 interrupt	0x0000_00E4
IRQ 42	58	Reserved	0x0000_00E8
IRQ 43	59	TIMER7 break interrupt / transition error interrupt / index error interrupt	0x0000_00EC
IRQ 44	60	TIMER7 update interrupt	0x0000_00F0
IRQ 45	61	TIMER7 trigger and commutation interrupt / direction change interrupt / index interrupt	0x0000_00F4
IRQ 46	62	TIMER7 capture compare interrupt	0x0000_00F8

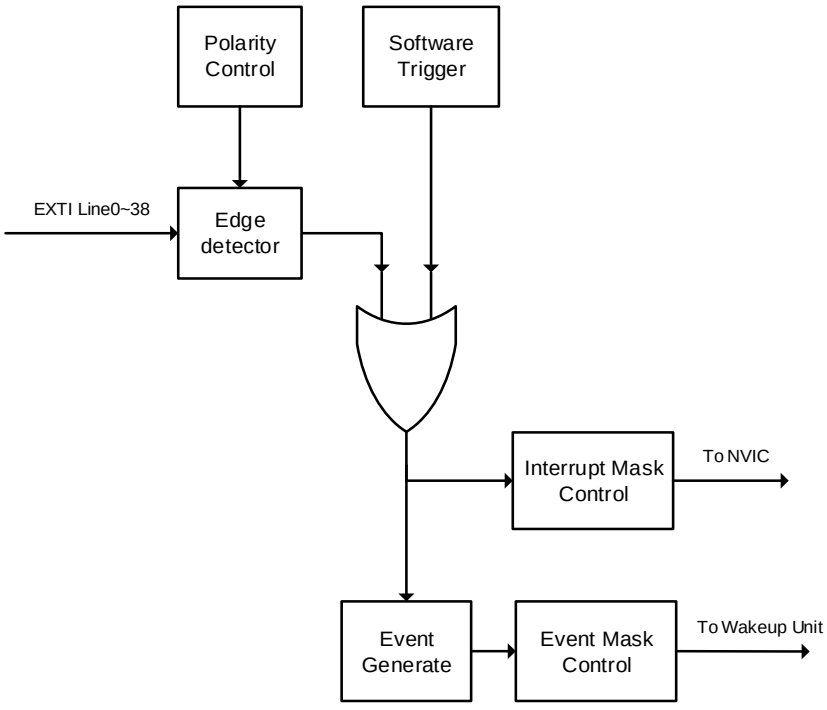
Interrupt Number	Vector Number	Peripheral Interrupt Description	Vector Address
IRQ 47	63	ADC2 global interrupt	0x0000_00FC
IRQ 48	64	SYSCFG global interrupt	0x0000_0100
IRQ 49	65	LPTIMER global interrupt and wakeup through EXTI line35 interrupt	0x0000_0104
IRQ 50	66	TIMER4 global interrupt	0x0000_0108
IRQ 51	67	SPI2 global interrupt	0x0000_010C
IRQ 52	68	UART3 global interrupt	0x0000_0110
IRQ 53	69	UART4 global interrupt	0x0000_0114
IRQ 54	70	TIMER5 global interrupt and DAC2 / DAC0 underrun error interrupt	0x0000_0118
IRQ 55	71	TIMER6 global interrupt and DAC3 / DAC1 underrun error interrupt	0x0000_011C
IRQ 56	72	DMA1 channel 0 global interrupt	0x0000_0120
IRQ 57	73	DMA1 channel 1 global interrupt	0x0000_0124
IRQ 58	74	DMA1 channel 2 global interrupt	0x0000_0128
IRQ 59	75	DMA1 channel 3 global interrupt	0x0000_012C
IRQ 60	76	DMA1 channel 4 global interrupt	0x0000_0130
IRQ 61	77	ADC3 global interrupt	0x0000_0134
IRQ 62	78	Reserved	0x0000_0138
IRQ 63	79	VUVD1 / VOVD1 interrupt	0x0000_013C
IRQ 64	80	CMP0 / CMP1 / CMP2 / CMP3 through EXTI lines 20 / 21 / 22 / 23 interrupts	0x0000_0140
IRQ 65	81	CMP4 / CMP5 / CMP6 / CMP7 through EXTI lines 24 / 36 / 37 / 38 interrupts	0x0000_0144
IRQ 66	82	CMP global interrupt	0x0000_0148
IRQ 67	83	HRTIMER interrupt0	0x0000_014C
IRQ 68	84	HRTIMER interrupt1	0x0000_0150
IRQ 69	85	HRTIMER interrupt2	0x0000_0154
IRQ 70	86	HRTIMER interrupt3	0x0000_0158
IRQ 71	87	HRTIMER interrupt4	0x0000_015C
IRQ 72	88	HRTIMER interrupt5	0x0000_0160
IRQ 73	89	HRTIMER interrupt6	0x0000_0164
IRQ 74	90	HRTIMER interrupt7	0x0000_0168
IRQ 75	91	HRTIMER interrupt8	0x0000_016C
IRQ 76	92	HRTIMER interrupt9	0x0000_0170
IRQ 77	93	TIMER19 break interrupt / transition error interrupt / Index error interrupt	0x0000_0174
IRQ 78	94	TIMER19 update interrupt	0x0000_0178
IRQ 79	95	TIMER19 trigger and commutation interrupt / direction change interrupt / Index interrupt	0x0000_017C

Interrupt Number	Vector Number	Peripheral Interrupt Description	Vector Address
IRQ 80	96	TIMER19 capture compare interrupt	0x0000_0180
IRQ 81	97	FPU global interrupt	0x0000_0184
IRQ 82	98	I2C2 event interrupt and I2C2 wakeup through EXTI line33 interrupt	0x0000_0188
IRQ 83	99	I2C2 error interrupt	0x0000_018C
IRQ 84	100	Reserved	0x0000_0190
IRQ 85	101	CAU global interrupt	0x0000_0194
IRQ 86-89	102-105	Reserved	0x0000_0198-0x0000_01A4
IRQ 90	106	TRNG global interrupt	0x0000_01A8
IRQ 91	107	Reserved	0x0000_01AC
IRQ 92	108	I2C3 event interrupt and I2C3 wakeup through EXTI line34 interrupt	0x0000_01B0
IRQ 93	109	I2C3 error interrupt	0x0000_01B4
IRQ 94	110	DMAMUX overrun interrupt	0x0000_01B8
IRQ 95	111	QSPI global interrupt	0x0000_01BC
IRQ 96	112	FFT global interrupt	0x0000_01C0
IRQ 97	113	DMA1 channel5 global interrupt	0x0000_01C4
IRQ 98	114	DMA1 channel6 global interrupt	0x0000_01C8
IRQ 99	115	CLA global interrupt	0x0000_01CC
IRQ 100	116	TMU global interrupt	0x0000_01D0
IRQ 101	117	FAC global interrupt	0x0000_01D4
IRQ 102	118	HPDF global interrupt 0	0x0000_01D8
IRQ 103	119	HPDF global interrupt 1	0x0000_01DC
IRQ 104	120	HPDF global interrupt 2	0x0000_01E0
IRQ 105	121	HPDF global interrupt 3	0x0000_01E4
IRQ 106	122	TIMER14 global interrupt	0x0000_01E8
IRQ 107	123	TIMER15 global interrupt	0x0000_01EC
IRQ 108	124	TIMER16 global interrupt	0x0000_01F0
IRQ 109	125	CAN0 wakeup through EXTI line25 interrupt	0x0000_01F4
IRQ 110	126	CAN0 interrupt for message buffer	0x0000_01F8
IRQ 111	127	CAN0 interrupt for bus off / bus off done	0x0000_01FC
IRQ 112	128	CAN0 interrupt for error	0x0000_0200
IRQ 113	129	CAN0 interrupt for error in fast transmission	0x0000_0204
IRQ 114	130	CAN0 interrupt for transmit warning	0x0000_0208
IRQ 115	131	CAN0 interrupt for receive warning	0x0000_020C
IRQ 116	132	CAN1 wakeup through EXTI line26 interrupt	0x0000_0210
IRQ 117	133	CAN1 interrupt for message buffer	0x0000_0214
IRQ 118	134	CAN1 interrupt for bus off / bus off done	0x0000_0218
IRQ 119	135	CAN1 interrupt for error	0x0000_021C

Interrupt Number	Vector Number	Peripheral Interrupt Description	Vector Address
IRQ 120	136	CAN1 interrupt for error in fast transmission	0x0000_0220
IRQ 121	137	CAN1 interrupt for transmit warning	0x0000_0224
IRQ 122	138	CAN1 interrupt for receive warning	0x0000_0228
IRQ 123	139	CAN2 wakeup through EXTI line27 interrupt	0x0000_022C
IRQ 124	140	CAN2 interrupt for message buffer	0x0000_0230
IRQ 125	141	CAN2 interrupt for bus off / bus off done	0x0000_0234
IRQ 126	142	CAN2 interrupt for error	0x0000_0238
IRQ 127	143	CAN2 interrupt for error in fast transmission	0x0000_023C
IRQ 128	144	CAN2 interrupt for transmit warning	0x0000_0240
IRQ 129	145	CAN2 interrupt for receive warning	0x0000_0244
IRQ 130	146	TIMER0 DEC interrupt	0x0000_0248
IRQ 131	147	TIMER1 DEC interrupt	0x0000_024C
IRQ 132	148	TIMER2 DEC interrupt	0x0000_0250
IRQ 133	149	TIMER3 DEC interrupt	0x0000_0254
IRQ 134	150	TIMER4 DEC interrupt	0x0000_0258
IRQ 135	151	TIMER7 DEC interrupt	0x0000_025C
IRQ 136	152	TIMER19 DEC interrupt	0x0000_0260

5.4. External interrupt and event (EXTI) block diagram

Figure 5-1. Block diagram of EXTI



5.5. External Interrupt and Event function overview

The EXTI contains up to 39 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 23 lines from internal modules (including LVD, RTC Alarm, RTC Tamper and TimeStamp event, LXTAL clock stuck, RTC Wakeup, CMP, CAN, USART Wakeup, I2C Wakeup, LPTIM Wakeup, OVD). All GPIO pins can be selected as an EXTI trigger source by configuring SYSCFG_EXTISSx registers in SYSCFG module (please refer to ***System configuration registers*** 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.

Table 5-3 EXTI source

EXTI Line Number	Source
0	PA0 / PB0 / PC0 / PD0 / PE0 / PF0 / PG0
1	PA1 / PB1 / PC1 / PD1 / PE1 / PF1 / PG1
2	PA2 / PB2 / PC2 / PD2 / PE2 / PF2 / PG2
3	PA3 / PB3 /PC3 / PD3 / PE3 / PF3 / PG3
4	PA4 / PB4 / PC4 / PD4 / PE4 / PF4 / PG4
5	PA5 / PB5 / PC5 / PD5 / PE5 / PF5 / PG5
6	PA6 / PB6 / PC6 / PD6 / PE6 / PF6 / PG6
7	PA7 / PB7 / PC7 / PD7 / PE7 / PF7 / PG7
8	PA8 / PB8 / PC8 / PD8 / PE8 / PF8 / PG8
9	PA9 / PB9 / PC9 / PD9 / PE9 / PF9 / PG9
10	PA10 / PB10 / PC10 / PD10 / PE10 / PF10 / PG10
11	PA11 / PB11 / PC11 / PD11 / PE11 / PF11
12	PA12 / PB12 / PC12 / PD12 / PE12 / PF12
13	PA13 / PB13 / PC13 / PD13 / PE13 / PF13
14	PA14 / PB14 / PC14 / PD14 / PE14 / PF14
15	PA15 / PB15 / PC15 / PD15 / PE15 / PF15
16	LVD/VAVD/VOVD/VUVD
17	RTC alarm
18	RTC tamper and timestamp event or LXTAL clock stuck
19	RTC wakeup timer
20	CMP0 output

EXTI Line Number	Source
21	CMP1 output
22	CMP2 output
23	CMP3 output
24	CMP4 output
25	CAN0 wakeup
26	CAN1 wakeup
27	CAN2 wakeup
28	USART0 wakeup
29	USART1 wakeup
30	USART2 wakeup
31	I2C0 wakeup
32	I2C1 wakeup
33	I2C2 wakeup
34	I2C3 wakeup
35	LPTIMER wakeup
36	CMP5 output
37	CMP6 output
38	CMP7 output

5.6. Register definition Register definition

EXTI base address: 0x4001 0400

5.6.1. Interrupt enable register 0 (EXTI_INTEN0)

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
INTEN31	INTEN30	INTEN29	INTEN28	INTEN27	INTEN26	INTEN25	INTEN24	INTEN23	INTEN22	INTEN21	INTEN20	INTEN19	INTEN18	INTEN17	INTEN16
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
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
31:0	INTENx	Interrupt enable bit x (x = 0…31) 0: Interrupt from linex is disabled 1: Interrupt from linex is enabled

5.6.2. Event enable register 0 (EXTI_EVEN0)

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
EVEN31	EVEN30	EVEN29	EVEN28	EVEN27	EVEN26	EVEN25	EVEN24	EVEN23	EVEN22	EVEN21	EVEN20	EVEN19	EVEN18	EVEN17	EVEN16
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
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
31:0	EVENx	Event enable bit x (x = 0…31) 0: Event from linex is disabled 1: Event from linex is enabled

5.6.3. Rising edge trigger enable register 0 (EXTI_RTEN0)

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
RTEN31	RTEN30	RTEN29	RTEN28	RTEN27	RTEN26	RTEN25	RTEN24	RTEN23	RTEN22	RTEN21	RTEN20	RTEN19	RTEN18	RTEN17	RTEN16
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
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
31:0	RTENx	Rising edge trigger enable bit x (x = 0…31) 0: Rising edge of linex is invalid 1: Rising edge of linex is valid as an interrupt/event request

5.6.4. Falling edge trigger enable register 0 (EXTI_FTEN0)

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
FTEN31	FTEN30	FTEN29	FTEN28	FTEN27	FTEN26	FTEN25	FTEN24	FTEN23	FTEN22	FTEN21	FTEN20	FTEN19	FTEN18	FTEN17	FTEN16

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
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
31:0	FTENx	Falling edge trigger enable bit x (x = 0…31) 0: Falling edge of linex is invalid 1: Falling edge of linex is valid as an interrupt/event request

5.6.5. Software interrupt event register 0 (EXTI_SWIEV0)

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
SWIEV31	SWIEV30	SWIEV29	SWIEV28	SWIEV27	SWIEV26	SWIEV25	SWIEV24	SWIEV23	SWIEV22	SWIEV21	SWIEV20	SWIEV19	SWIEV18	SWIEV17	SWIEV16
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
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
31:0	SWIEVx	Interrupt/Event software trigger bit x (x = 0…31) 0: Deactivate the EXTIx software interrupt/event request 1: Activate the EXTIx software interrupt/event request

5.6.6. Pending register 0 (EXTI_PD0)

Address offset: 0x14

Reset value: undefined

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PD31	PD30	PD29	PD28	PD27	PD26	PD25	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	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
31:0	PDx	Interrupt pending status bit x (x = 0…31) 0: EXTI linex is not triggered

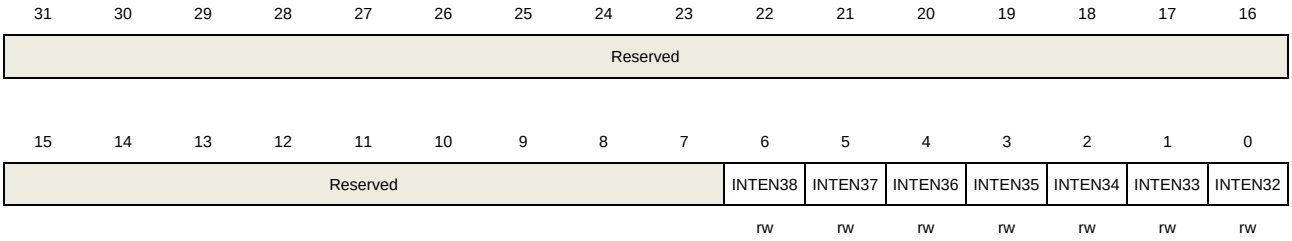
1: EXTI linex is triggered. This bit is cleared to 0 by writing 1 to it

5.6.7. Interrupt enable register 1 (EXTI_INTEN1)

Address offset: 0x18

Reset value: 0x0000 0000

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



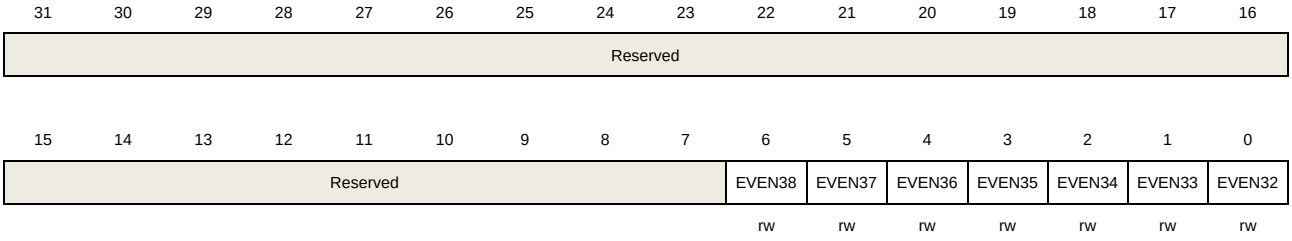
Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value.
6:0	INTENx	Interrupt enable bit x (x = 32…38) 0: Interrupt from linex is disabled 1: Interrupt from linex is enabled

5.6.8. Event enable register 1 (EXTI_EVEN1)

Address offset: 0x1C

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.
6:0	EVENx	Event enable bit x (x = 32…38) 0: Event from linex is disabled 1: Event from linex is enabled

5.6.9. Rising edge trigger enable register 1 (EXTI_RTEN1)

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									RTEN38	RTEN37	RTEN36	RTEN35	RTEN34	RTEN33	RTEN32
									rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value.
6:0	RTENx	Rising edge trigger enable x (x = 32…38) 0: Rising edge of linex is invalid 1: Rising edge of linex is valid as an interrupt/event request

5.6.10. Falling edge trigger enable register 1 (EXTI_FTEN1)

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									FTEN38	FTEN37	FTEN36	FTEN35	FTEN34	FTEN33	FTEN32
									rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value.
6:0	FTENx	Falling edge trigger enable x (x = 32…38) 0: Falling edge of linex is invalid 1: Falling edge of linex is valid as an interrupt/event request

5.6.11. Software interrupt event register 1 (EXTI_SWIEV1)

Address offset: 0x28

Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									SWIEV38	SWIEV37	SWIEV36	SWIEV35	SWIEV34	SWIEV33	SWIEV32
									rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value.
6:0	SWIEVx	Interrupt / Event software trigger x (x = 32…38) 0: Deactivate the EXTIx software interrupt/event request 1: Activate the EXTIx software interrupt/event request

5.6.12. Pending register 1 (EXTI_PD1)

Address offset: 0x2C
Reset value: undefined

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									PD38	PD37	PD36	PD35	PD34	PD33	PD32
									rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value.
6:0	PDx	Interrupt pending status x (x = 32…38) 0: EXTI linex is not triggered 1: EXTI linex is triggered. This bit is cleared to 0 by writing 1 to it

6. Trigger selection controller (TRIGSEL)

6.1. Overview

The trigger selection controller (TRIGSEL) allows software to select the trigger input signal for various peripherals. TRIGSEL provides a flexible mechanism, there are up to 243 trigger input signals could be selected. Each peripheral corresponding to the independent trigger selection controller. Configure the corresponding register to select the different trigger signal for the specified trigger input of each peripheral.

6.2. Characteristics

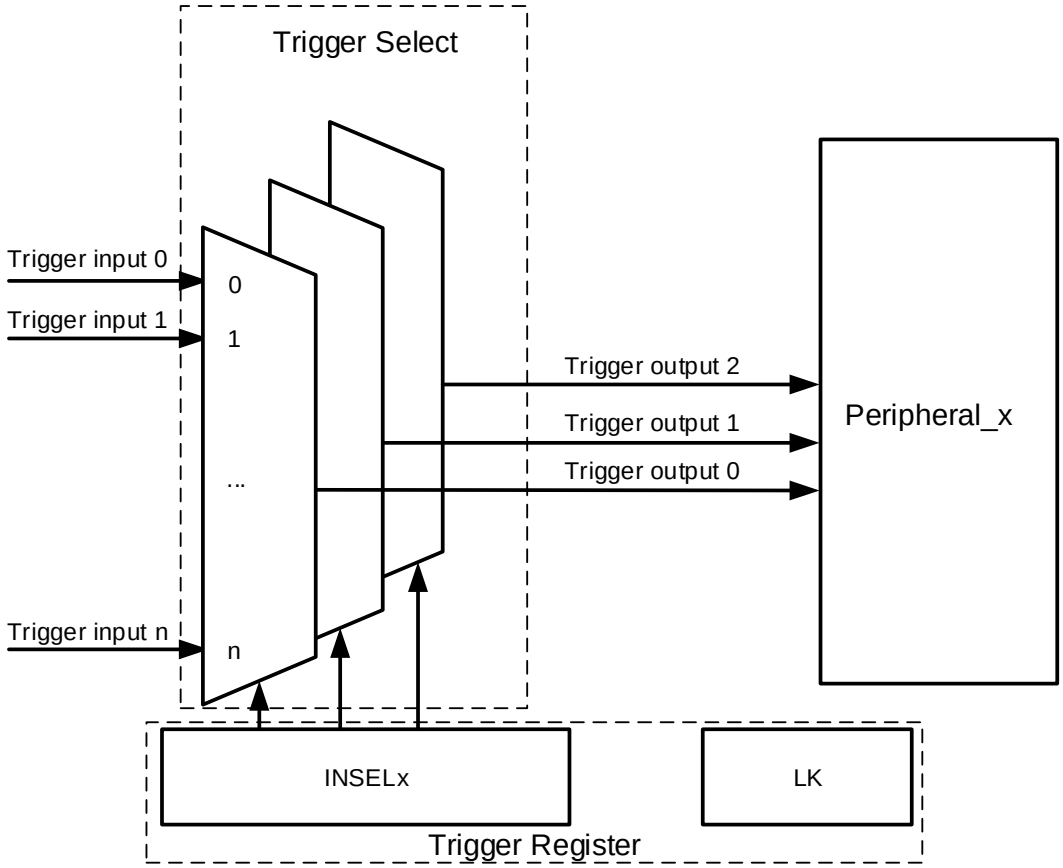
- Support different optional trigger inputs.
- Each peripheral has its corresponding register to select trigger input signal.
- Trigger input source could be external input signal or output of peripheral.
- Trigger selection output could be for external output or peripheral input.

6.3. Function overview

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

The ***Figure 6-1. TRIGSEL main composition example*** shows the main composition of TRIGSEL.

Figure 6-1. TRIGSEL main composition example



6.4. Internal connect

The TRIGSEL allows software to select the trigger input for peripherals. The ***Figure 6-1. TRIGSEL main composition example*** shows the trigger input register selection.

Table 6-1. Trigger input bit fields selection

fields	bits value	trigger input selection
INSELx	0x00	0
	0x01	1
	0x02	TRIGSEL_IN0
	0x03	TRIGSEL_IN1
	0x04	TRIGSEL_IN2
	0x05	TRIGSEL_IN3
	0x06	TRIGSEL_IN4
	0x07	TRIGSEL_IN5
	0x08	TRIGSEL_IN6
	0x09	TRIGSEL_IN7
	0x0a	TRIGSEL_IN8
	0x0b	TRIGSEL_IN9

fields	bits value	trigger input selection
	0x0c	TRIGSEL_IN10
	0x0d	TRIGSEL_IN11
	0x0e	TRIGSEL_IN12
	0x0f	TRIGSEL_IN13
	0x10	TIMER0_TRGO0
	0x11	TIMER0_TRGO1
	0x12	TIMER0_CH0
	0x13	TIMER0_CH1
	0x14	TIMER0_CH2
	0x15	TIMER0_CH3
	0x16	TIMER0_MCH0
	0x17	TIMER0_MCH1
	0x18	TIMER0_MCH2
	0x19	TIMER0_MCH3
	0x1a-0x1f	Reserved
	0x20	TIMER0_ETI
	0x21	TIMER1_TRGO0
	0x22	TIMER1_CH0
	0x23	TIMER1_CH1
	0x24	TIMER1_CH2
	0x25	TIMER1_CH3
	0x26	TIMER1_ETI
	0x27	TIMER2_TRGO0
	0x28	TIMER2_CH0
	0x29	TIMER2_CH1
	0x2a	TIMER2_CH2
	0x2b	TIMER2_CH3
	0x2c	TIMER2_ETI
	0x2d	TIMER3_TRGO0
	0x2e	TIMER3_CH0
	0x2f	TIMER3_CH1
	0x30	TIMER3_CH2
	0x31	TIMER3_CH3
	0x32	TIMER3_ETI
	0x33	TIMER4_TRGO0
	0x34	TIMER4_CH0
	0x35	TIMER4_CH1
	0x36	TIMER4_CH2
	0x37	TIMER4_CH3
	0x38	TIMER4_ETI
	0x39	TIMER5_TRGO0

fields	bits value	trigger input selection
	0x3a	TIMER6_TRGO0
	0x3b	TIMER7_TRGO0
	0x3c	TIMER7_TRGO1
	0x3d	TIMER7_CH0
	0x3e	TIMER7_CH1
	0x3f	TIMER7_CH2
	0x40	TIMER7_CH3
	0x41	TIMER7_MCH0
	0x42	TIMER7_MCH1
	0x43	TIMER7_MCH2
	0x44	TIMER7_MCH3
	0x45-0x4a	Reserved
	0x4b	TIMER7_ETI
	0x4c	TIMER14_TRGO0
	0x4d	TIMER14_CH0
	0x4e	TIMER14_CH1
	0x4f	TIMER14_MCH0
	0x50-0x52	Reserved
	0x53	TIMER15_CH0
	0x54	TIMER15_MCH0
	0x55-0x57	Reserved
	0x58	TIMER16_CH0
	0x59	TIMER16_MCH0
	0x5a-0x5b	Reserved
	0x5c	TIMER19_TRGO0
	0x5d	TIMER19_TRGO1
	0x5e	TIMER19_CH0
	0x5f	TIMER19_CH1
	0x60	TIMER19_CH2
	0x61	TIMER19_CH3
	0x62	TIMER19_MCH0
	0x63	TIMER19_MCH1
	0x64	TIMER19_MCH2
	0x65	TIMER19_MCH3
	0x66-0x6b	Reserved
	0x6c	TIMER19_ETI
	0x6d	TIMER0_BKIN0
	0x6e	TIMER0_BKIN1
	0x6f	TIMER0_BKIN2
	0x70	TIMER7_BKIN0
	0x71	TIMER7_BKIN1

fields	bits value	trigger input selection
	0x72	TIMER7_BKIN2
	0x73	TIMER14_BKIN0
	0x74	TIMER15_BKIN0
	0x75	TIMER16_BKIN0
	0x76	TIMER19_BKIN0
	0x77	TIMER19_BKIN1
	0x78	TIMER19_BKIN2
	0x79	LPTIMER_OUT
	0x7a	LPTIMER_ETI
	0x7b	HRTIMER_SCOUT
	0x7c	HRTIMER_SCIN
	0x7d	HRTIMER_ADCTRIG0
	0x7e	HRTIMER_ADCTRIG1
	0x7f	HRTIMER_ADCTRIG2
	0x80	HRTIMER_ADCTRIG3
	0x81	HRTIMER_ADCTRIG4
	0x82	HRTIMER_ADCTRIG5
	0x83	HRTIMER_ADCTRIG6
	0x84	HRTIMER_ADCTRIG7
	0x85	HRTIMER_ADCTRIG8
	0x86	HRTIMER_ADCTRIG9
	0x87	HRTIMER_DACTRIG0
	0x88	HRTIMER_DACTRIG1
	0x89	HRTIMER_DACTRIG2
	0x8a	HRTIMER_ST0_TRIG1
	0x8b	HRTIMER_ST1_TRIG1
	0x8c	HRTIMER_ST2_TRIG1
	0x8d	HRTIMER_ST3_TRIG1
	0x8e	HRTIMER_ST4_TRIG1
	0x8f	HRTIMER_ST5_TRIG1
	0x90	HRTIMER_ST6_TRIG1
	0x91	HRTIMER_ST7_TRIG1
	0x92~0x95	reserved
	0x96	HRTIMER_SYSFLT
	0x97	ADC0_WD0_OUT
	0x98	ADC0_WD1_OUT
	0x99	ADC0_WD2_OUT
	0x9a	ADC1_WD0_OUT
	0x9b	ADC1_WD1_OUT
	0x9c	ADC1_WD2_OUT
	0x9d	ADC2_WD0_OUT

fields	bits value	trigger input selection
	0x9e	ADC2_WD1_OUT
	0x9f	ADC2_WD2_OUT
	0xa0	ADC3_WD0_OUT
	0xa1	ADC3_WD1_OUT
	0xa2	ADC3_WD2_OUT
	0xa3	HXTAL_DIV32_TRIG
	0xa4	IRC32K_TRIG
	0xa5	LXTAL_TRIG
	0xa6	CKOUT_TRIG
	0xa7	EXTI2_TRIG
	0xa8	EXTI3_TRIG
	0xa9	EXTI9_TRIG
	0xaa	EXTI10_TRIG
	0xab	EXTI11_TRIG
	0xac	EXTI15_TRIG
	0xad	RTC_WAKEUP
	0xae	RTC_ALARM0
	0xaf	RTC_ALARM1
	0xb0	RTC_TAMP0
	0xb1	RTC_TAMP1
	0xb2	RTC_TAMP2
	0xb3	CMP0_OUT
	0xb4	CMP1_OUT
	0xb5	CMP2_OUT
	0xb6	CMP3_OUT
	0xb7	CMP4_OUT
	0xb8	CMP5_OUT
	0xb9	CMP6_OUT
	0xba	CMP7_OUT
	0xbb	CLA0_OUT
	0xbc	CLA1_OUT
	0xbd	CLA2_OUT
	0xbe	CLA3_OUT
	0xbf	HRTIMER_ST0_TRIG0
	0xc0	HRTIMER_ST1_TRIG0
	0xc1	HRTIMER_ST2_TRIG0
	0xc2	HRTIMER_ST3_TRIG0
	0xc3	HRTIMER_ST4_TRIG0
	0xc4	HRTIMER_ST5_TRIG0
	0xc5	HRTIMER_ST6_TRIG0
	0xc6	HRTIMER_ST7_TRIG0

fields	bits value	trigger input selection
	0xc7~0xde	reserved
	0xdf	HRTIMER_ST0_CH0
	0xe0	HRTIMER_ST0_CH1
	0xe1	HRTIMER_ST1_CH0
	0xe2	HRTIMER_ST1_CH1
	0xe3	HRTIMER_ST2_CH0
	0xe4	HRTIMER_ST2_CH1
	0xe5	HRTIMER_ST3_CH0
	0xe6	HRTIMER_ST3_CH1
	0xe7	HRTIMER_ST4_CH0
	0xe8	HRTIMER_ST4_CH1
	0xe9	HRTIMER_ST5_CH0
	0xea	HRTIMER_ST5_CH1
	0xeb	HRTIMER_ST6_CH0
	0xec	HRTIMER_ST6_CH1
	0xed	HRTIMER_ST7_CH0
	0xee	HRTIMER_ST7_CH1
	0xef	ADC0_CONV
	0xf0	ADC1_CONV
	0xf1	ADC2_CONV
	0xf2	ADC3_CONV
	0xf3	LCKM_OUT
	0xf4~0xff	reserved

Table 6-1. Trigger input bit fields selection shows the connection relationship between TRIGSEL input and output. Through the INSELx[7:0] bits of TRIGSEL register, an input trigger source can be selected for the output of TRIGSEL. Each TRIGSEL register can configure with up to 3 outputs, which are connected to the corresponding peripherals.

Table 6-2. TRIGSEL input and output mapping

Trigger Source	Trigger select	TRIGSEL Register	TRIGSEL output	Peripherals
0 1 TRIGSEL_IN0 TRIGSEL_IN1 TRIGSEL_IN2 TRIGSEL_IN3 TRIGSEL_IN4 TRIGSEL_IN5	INSELx[7:0]	TRIGSEL_EXTOUT_0	Output0 Output1	TRIGSEL_OUT0 TRIGSEL_OUT1
		TRIGSEL_EXTOUT_1	Output0 Output1	TRIGSEL_OUT2 TRIGSEL_OUT3

Trigger Source	Trigger select	TRIGSEL Register	TRIGSEL output	Peripherals
TRIGSEL_IN6 TRIGSEL_IN7 TRIGSEL_IN8 TRIGSEL_IN9		TRIGSEL_EXTOUT_2	Output0 Output1	TRIGSEL_OUT4 TRIGSEL_OUT5
TRIGSEL_IN10 TRIGSEL_IN11 TRIGSEL_IN12 TRIGSEL_IN13		TRIGSEL_EXTOUT_3	Output0 Output1	TRIGSEL_OUT6 TRIGSEL_OUT7
TIMER0_TRGO0 TIMER0_TRGO1 TIMER0_CH0 TIMER0_CH1		TRIGSEL_ADC0	Output0	ADC0_ROUTRG
TIMER0_CH2 TIMER0_CH3 TIMER0_MCH0 TIMER0_MCH1		TRIGSEL_ADC1	Output0	ADC1_ROUTRG
TIMER0_MCH2 TIMER0_MCH3 TIMER0_ETI TIMER1_TRGO0		TRIGSEL_ADC2	Output0	ADC2_ROUTRG
TIMER1_CH0 TIMER1_CH1 TIMER1_CH2 TIMER1_CH3		TRIGSEL_ADC3	Output0	ADC3_ROUTRG
TIMER1_ETI TIMER2_TRGO0 TIMER2_CH0 TIMER2_CH1		TRIGSEL_TIMER0BRKIN	Output0 Output1 Output2	TIMER0_BRKIN0 TIMER0_BRKIN1 TIMER0_BRKIN2
TIMER2_CH2 TIMER2_CH3 TIMER2_ETI TIMER3_TRGO0		TRIGSEL_TIMER7BRKIN	Output0 Output1 Output2	TIMER7_BRKIN0 TIMER7_BRKIN1 TIMER7_BRKIN2
TIMER3_CH0 TIMER3_CH1 TIMER3_CH2 TIMER3_CH3		TRIGSEL_TIMER14BRKIN	Output0	TIMER14_BRKIN0
TIMER3_ETI TIMER4_TRGO0 TIMER4_CH0 TIMER4_CH1		TRIGSEL_TIMER15BRKIN	Output0	TIMER15_BRKIN0

Trigger Source	Trigger select	TRIGSEL Register	TRIGSEL output	Peripherals
TIMER4_CH2 TIMER4_CH3 TIMER4_ETI TIMER5_TRGO0 TIMER6_TRGO0 TIMER7_TRGO0 TIMER7_TRGO1 TIMER7_CH0 TIMER7_CH1 TIMER7_CH2 TIMER7_CH3 TIMER7_MCH0 TIMER7_MCH1 TIMER7_MCH2 TIMER7_MCH3 TIMER7_ETI TIMER14_TRGO0 TIMER14_CH0 TIMER14_CH1 TIMER14_MCH0 TIMER15_CH0 TIMER15_MCH0 TIMER15_BRKIN TIMER16_CH0 TIMER16_MCH0 TIMER16_BRKIN TIMER19_TRGO0 TIMER19_TRGO1 TIMER19_CH0 TIMER19_CH1 TIMER19_CH2 TIMER19_CH3 TIMER19_MCH0 TIMER19_MCH1 TIMER19_MCH2 TIMER19_MCH3 TIMER0_BKIN0 TIMER0_BKIN1 TIMER0_BKIN2 TIMER7_BKIN0		TRIGSEL_TIMER16BRKIN	Output0	TIMER16_BRKIN0
		TRIGSEL_TIMER19BRKIN	Output0 Output1 Output2	TIMER19_BRKIN0 TIMER19_BRKIN1 TIMER19_BRKIN2
		TRIGSEL_CAN0	Output0	CAN0_EX_TIME_TICK
		TRIGSEL_CAN1	Output0	CAN1_EX_TIME_TICK
		TRIGSEL_CAN2	Output0	CAN2_EX_TIME_TICK
		TRIGSEL_TIMER0_ETI	Output0	TIMER0_ETI
		TRIGSEL_TIMER1_ETI	Output0	TIMER1_ETI
		TRIGSEL_TIMER2_ETI	Output0	TIMER2_ETI
		TRIGSEL_TIMER3_ETI	Output0	TIMER3_ETI
		TRIGSEL_TIMER4_ETI	Output0	TIMER4_ETI

Trigger Source	Trigger select	TRIGSEL Register	TRIGSEL output	Peripherals
TIMER7_BKIN1 TIMER7_BKIN2 TIMER14_BKIN0 TIMER15_BKIN0 TIMER16_BKIN0 TIMER19_BKIN0 TIMER19_BKIN1 TIMER19_BKIN2 LPTIMER_OUT LPTIMER_ETI HRTIMER_SCOUT HRTIMER_SCIN HRTIMER_ADCTRIG0 HRTIMER_ADCTRIG1 HRTIMER_ADCTRIG2 HRTIMER_ADCTRIG3 HRTIMER_ADCTRIG4 HRTIMER_ADCTRIG5 HRTIMER_ADCTRIG6 HRTIMER_ADCTRIG7 HRTIMER_ADCTRIG8 HRTIMER_ADCTRIG9 HRTIMER_DACTRIG0 HRTIMER_DACTRIG1 HRTIMER_DACTRIG2 HRTIMER_ST0_TRIG1 HRTIMER_ST1_TRIG1 HRTIMER_ST2_TRIG1 HRTIMER_ST3_TRIG1 HRTIMER_ST4_TRIG1 HRTIMER_ST5_TRIG1 HRTIMER_ST6_TRIG1 HRTIMER_ST7_TRIG1 HRTIMER_SYSFLT ADC0_WD0_OUT ADC0_WD1_OUT ADC0_WD2_OUT ADC1_WD0_OUT ADC1_WD1_OUT ADC1_WD2_OUT ADC2_WD0_OUT		TRIGSEL_TIMER7_ETI	Output0	TIMER7_ETI
		TRIGSEL_TIMER19_ETI	Output0	TIMER19_ETI
		TRIGSEL_HPDF	Output0	HPDF_ITRG
		TRIGSEL_TIMER0ITI14	Output0	TIMER0_ITI14
		TRIGSEL_TIMER1ITI14	Output0	TIMER1_ITI14
		TRIGSEL_TIMER2ITI14	Output0	TIMER2_ITI14
		TRIGSEL_TIMER3ITI14	Output0	TIMER3_ITI14
		TRIGSEL_TIMER4ITI14	Output0	TIMER4_ITI14
		TRIGSEL_TIMER7ITI14	Output0	TIMER7_ITI14
		TRIGSEL_TIMER14ITI14	Output0	TIMER14_ITI14

Trigger Source	Trigger select	TRIGSEL Register	TRIGSEL output	Peripherals
ADC2_WD1_OUT ADC2_WD2_OUT ADC3_WD0_OUT ADC3_WD1_OUT ADC3_WD2_OUT HXTAL_DIV32_TRIG IRC32K_TRIG LXTAL_TRIG CKOUT_TRIG EXTI2_TRIG EXTI3_TRIG EXTI9_TRIG EXTI10_TRIG EXTI11_TRIG EXTI15_TRIG RTC_WAKEUP RTC_ALARM0 RTC_ALARM1 RTC_TAMP0 RTC_TAMP1 RTC_TAMP2 CMP0_OUT CMP1_OUT CMP2_OUT CMP3_OUT CMP4_OUT CMP5_OUT CMP6_OUT CMP7_OUT CLA0_OUT CLA1_OUT CLA2_OUT CLA3_OUT HRTIMER_ST0_TRIG0 HRTIMER_ST1_TRIG0 HRTIMER_ST2_TRIG0 HRTIMER_ST3_TRIG0 HRTIMER_ST4_TRIG0 HRTIMER_ST5_TRIG0 HRTIMER_ST6_TRIG0 HRTIMER_ST7_TRIG0		TRIGSEL_TIMER19ITI14	Output0	TIMER19_ITI14
		TRIGSEL_DAC0	Output0 Output1	DAC0_EXTRIG DAC0_EXTRIG
		TRIGSEL_DAC1	Output0 Output1	DAC1_EXTRIG DAC1_EXTRIG
		TRIGSEL_DAC2	Output0 Output1	DAC2_EXTRIG DAC2_EXTRIG
		TRIGSEL_DAC3	Output0 Output1	DAC3_EXTRIG DAC3_EXTRIG
		TRIGSEL_EXTDAC0	Output0 Output1	DAC0_ST_EXTRIG DAC0_ST_EXTRIG
		TRIGSEL_EXTDAC1	Output0 Output1	DAC1_ST_EXTRIG DAC1_ST_EXTRIG
		TRIGSEL_EXTDAC2	Output0 Output1	DAC2_ST_EXTRIG DAC2_ST_EXTRIG
		TRIGSEL_EXTDAC3	Output0 Output1	DAC3_ST_EXTRIG DAC3_ST_EXTRIG
		TRIGSEL_CLA_0	Output0 Output1 Output2	TRIGSEL_CLA_IN0 TRIGSEL_CLA_IN1 TRIGSEL_CLA_IN2

Trigger Source	Trigger select	TRIGSEL Register	TRIGSEL output	Peripherals
HRTIMER_ST0_CH0 HRTIMER_ST0_CH1 HRTIMER_ST1_CH0 HRTIMER_ST1_CH1 HRTIMER_ST2_CH0 HRTIMER_ST2_CH1 HRTIMER_ST3_CH0 HRTIMER_ST3_CH1 HRTIMER_ST4_CH0 HRTIMER_ST4_CH1 HRTIMER_ST5_CH0 HRTIMER_ST5_CH1 HRTIMER_ST6_CH0 HRTIMER_ST6_CH1 HRTIMER_ST7_CH0 HRTIMER_ST7_CH1 ADC0_CONV ADC1_CONV ADC2_CONV ADC3_CONV LCKM_OUT		TRIGSEL_CLA_1	Output0 Output1 Output2	TRIGSEL_CLA_IN3 TRIGSEL_CLA_IN4 TRIGSEL_CLA_IN5
		TRIGSEL_CLA_2	Output0 Output1 Output2	TRIGSEL_CLA_IN6 TRIGSEL_CLA_IN7 TRIGSEL_CLA_IN8
		TRIGSEL_CLA_3	Output0 Output1	TRIGSEL_CLA_IN9 TRIGSEL_CLA_IN10
		TRIGSEL_CLA_4	Output0	TRIGSEL_CLA_IN11

Note: All output can select all input as trigger source except TIMERx_BRKINy, TIMERx_ETI, TIMERx_ITI14, CLA_IN_x, LCKM_OUT and ADCx_CONV.

1. TIMERx_BRKINy can only select TIMERx_BRKINy as trigger.
2. TIMERx_BRKINy can only be seclected by TIMERx_BRKINy and EXTOUT.
3. TIMERx_ETI can not select TIMERx_BRKINy and CKOUT_TRIG as trigger.
4. TIMERx_ITI14 can not select HRTIMER signals and their own signals as trigger.
5. CLA_IN_x (x = 0,..,5) can only select HRTIMER_STx_CHy, TIMERx_CHy and CMPx_OUT.
6. HRTIMER_STx_CHy can only be seclected by CLA_IN_x (x = 0,..,5) and EXTOUT.
7. CLA_IN_x (x = 6,..,10) can only select TIMERx_TRGOy as trigger.
8. CLA_IN_11 can only select ADCx_CONV as trigger.
9. LCKM_OUT can only be selected by TIMERx_ETI and EXTOUT.
10. ADCx_CONV can only be seclected by CLA_IN_11 and EXTOUT.

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.

6.5. Register definition

TRIGSEL base address: 0x4001 8400

6.5.1. Trigger selection for EXTOUT register 0 (TRIGSEL_EXTOUT_0)

Address offset: 0x00
Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_EXTOUT_0 register. 0: TRIGSEL_EXTOUT_0 register write is enabled. 1: TRIGSEL_EXTOUT_0 register write is disabled.
30:16	Reserved	Must be kept at reset value.
15:8	INSEL1[7:0]	Trigger input source selection for output1 These bits are used to select trigger input signal connected to output1. The output can be as the source of TRIGSEL_OUT1 (external output1 signal). For the detailed configuration, please refer to ***Table 6-1. 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 output is used as the source of TRIGSEL_OUT0 (external output0 signal). For the detailed configuration, please refer to ***Table 6-1. Trigger input bit fields selection***.

6.5.2. Trigger selection for EXTOUT register 1 (TRIGSEL_EXTOUT_1)

Address offset: 0x04
Reset value: 0x0000 0000

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

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

rs

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

Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_EXTOUT_1 register. 0: TRIGSEL_EXTOUT_1 register write is enabled. 1: TRIGSEL_EXTOUT_1 register write is disabled.
30:16	Reserved	Must be kept at reset value.
15:8	INSEL1[7:0]	Trigger input source selection for output1 These bits are used to select trigger input signal connected to output1. The output can be as the source of TRIGSEL_OUT3 (external output3 signal). For the detailed configuration, please refer to ***Table 6-1. 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 output is used as the source of TRIGSEL_OUT2 (external output2 signal). For the detailed configuration, please refer to ***Table 6-1. Trigger input bit fields selection***.

6.5.3. 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.

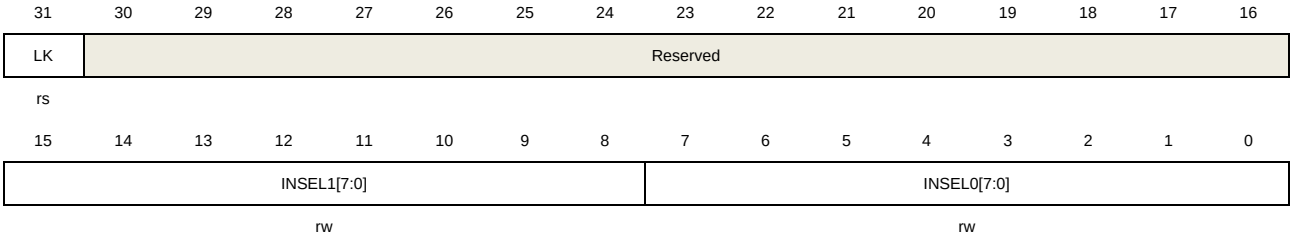
15:8	INSEL1[7:0]	Trigger input source selection for output1 These bits are used to select trigger input signal connected to output1. The output is used as the source of TRIGSEL_OUT5 (external output5 signal). For the detailed configuration, please refer to ***Table 6-1. Trigger input bit fields selection***.
7:0	INSEL0[7:0]	Trigger input source selection for output0 These bits are used to select trigger input signal connected to output0. The output is used as the source of TRIGSEL_OUT4 (external output4 signal). For the detailed configuration, please refer to ***Table 6-1. Trigger input bit fields selection***.

6.5.4. 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 output is used as the source of TRIGSEL_OUT7 (external output7 signal). For the detailed configuration, please refer to ***Table 6-1. 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 output is used as the source of TRIGSEL_OUT6 (external output6 signal). For the detailed configuration, please refer to ***Table 6-1. Trigger input bit fields selection***.

6.5.5. Trigger selection for ADC0 register (TRIGSEL_ADC0)

Address offset: 0x10

Reset value: 0x0000 1012

This register 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 register. 0: TRIGSEL_ADC0 register write is enabled. 1: TRIGSEL_ADC0 register write is disabled.
30:16	Reserved	Must be kept at reset value.
15:8	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 output is used as the source of ADC0_ROUTRG(ADC0 routine sequence) trigger input. For the detailed configuration, please refer to ***Table 6-1. Trigger input bit fields selection***.

6.5.6. Trigger selection for ADC1 register (TRIGSEL_ADC1)

Address offset: 0x14

Reset value: 0x0000 1012

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

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

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

Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_ADC1 register. 0: TRIGSEL_ADC1 register write is enabled.

		1: TRIGSEL_ADC1 register write is disabled.
30:16	Reserved	Must be kept at reset value.
15:8	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 output is used as the source of ADC1_ROUTRG (ADC1 routine sequence) trigger input. For the detailed configuration, please refer to ***Table 6-1. Trigger input bit fields selection***.

6.5.7. Trigger selection for ADC2 register (TRIGSEL_ADC2)

Address offset: 0x18
Reset value: 0x0000 1028

This register 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 register. 0: TRIGSEL_ADC2 register write is enabled. 1: TRIGSEL_ADC2 register write is disabled.
30:16	Reserved	Must be kept at reset value.
15:8	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 output is used as the source of ADC2_ROUTRG(ADC2 routine sequence) trigger input. For the detailed configuration, please refer to ***Table 6-1. Trigger input bit fields selection***.

6.5.8. Trigger selection for ADC3 register (TRIGSEL_ADC3)

Address offset: 0x1C
Reset value: 0x0000 1028

This register 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_ADC3 register. 0: TRIGSEL_ADC3 register write is enabled. 1: TRIGSEL_ADC3 register write is disabled.
30:16	Reserved	Must be kept at reset value.
15:8	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 output is used as the source of ADC3_ROUTRG(ADC3 routine sequence) trigger input. For the detailed configuration, please refer to ***Table 6-1. Trigger input bit fields selection***.

6.5.9. Trigger selection for TIMER0_BRKIN register (TRIGSEL_TIMER0BRKIN)

Address offset: 0x2C

Reset value: 0x006F 6E6D

This register 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_TIMER0BRKIN register. 0: TRIGSEL_TIMER0BRKIN register write is enabled. 1: TRIGSEL_TIMER0BRKIN register write is disabled.

30:24	Reserved	Must be kept at reset value.
23:16	INSEL2[7:0]	Trigger input source selection for output2 These bits are used to select trigger input signal connected to output2. The output is used as the source of TIMER0BRKIN2 trigger input. For the detailed configuration, please refer to ***Table 6-1. Trigger input bit fields selection***.
15:8	INSEL1[7:0]	Trigger input source selection for output1 These bits are used to select trigger input signal connected to output1. The output is used as the source of TIMER0BRKIN1 trigger input. For the detailed configuration, please refer to ***Table 6-1. 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 output is used as the source of TIMER0BRKIN0 trigger input. For the detailed configuration, please refer to ***Table 6-1. Trigger input bit fields selection***.

6.5.10. Trigger selection for TIMER7_BRKIN register (TRIGSEL_TIMER7BRKIN)

Address offset: 0x30

Reset value: 0x0072 7170

This register 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_TIMER7BRKIN register. 0: TRIGSEL_TIMER7BRKIN register write is enabled. 1: TRIGSEL_TIMER7BRKIN register write is disabled.
30:24	Reserved	Must be kept at reset value.
23:16	INSEL2[7:0]	Trigger input source selection for output2 These bits are used to select trigger input signal connected to output2. The output is used as the source of TIMER7BRKIN2 trigger input. For the detailed configuration, please refer to ***Table 6-1. Trigger input bit fields selection***.
15:8	INSEL1[7:0]	Trigger input source selection for output1 These bits are used to select trigger input signal connected to output1. The output

		is used as the source of TIMER7BRKIN1 trigger input. For the detailed configuration, please refer to ***Table 6-1. 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 output is used as the source of TIMER7BRKIN0 trigger input. For the detailed configuration, please refer to ***Table 6-1. Trigger input bit fields selection***.

6.5.11. Trigger selection for TIMER14_BRKIN register (TRIGSEL_TIMER14BRKIN)

Address offset: 0x34
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_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 output0 These bits are used to select trigger input signal connected to output0. The output is used as the source of TIMER14BRKIN0 trigger input. For the detailed configuration, please refer to ***Table 6-1. Trigger input bit fields selection***.

6.5.12. Trigger selection for TIMER15_BRKIN register (TRIGSEL_TIMER15BRKIN)

Address offset: 0x38
Reset value: 0x0000 0074

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

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

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

Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER15BRKIN register. 0: TRIGSEL_TIMER15BRKIN register write is enabled. 1: TRIGSEL_TIMER15BRKIN register write is disabled.
30:8	Reserved	Must be kept at reset value.
7:0	INSEL0[7:0]	Trigger input source selection for output0 These bits are used to select trigger input signal connected to output0. The output is used as the source of TIMER15BRKIN0 trigger input. For the detailed configuration, please refer to ***Table 6-1. Trigger input bit fields selection***.

6.5.13. Trigger selection for TIMER16_BRKIN register (TRIGSEL_TIMER16BRKIN)

Address offset: 0x3C

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_TIMER16BRKIN register. 0: TRIGSEL_TIMER16BRKIN register write is enabled. 1: TRIGSEL_TIMER16BRKIN register write is disabled.
30:8	Reserved	Must be kept at reset value.

7:0	INSEL0[7:0]	Trigger input source selection for output0 These bits are used to select trigger input signal connected to output0. The output is used as the source of TIMER16BRKIN0 trigger input. For the detailed configuration, please refer to ***Table 6-1. Trigger input bit fields selection***.

6.5.14. Trigger selection for TIMER19_BRKIN register (TRIGSEL_TIMER19BRKIN)

Address offset: 0x40

Reset value: 0x0078 7776

This register 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_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 output0 These bits are used to select trigger input signal connected to output0. The output is used as the source of TIMER19BRKIN0 trigger input. For the detailed configuration, please refer to ***Table 6-1. Trigger input bit fields selection***.

6.5.15. Trigger selection for CAN0 register (TRIGSEL_CAN0)

Address offset: 0x44

Reset value: 0x0000 0039

This register 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_CAN0 register. 0: TRIGSEL_CAN0 register write is enabled. 1: TRIGSEL_CAN0 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 output is used as the source of CAN0_EX_TIME_TICK trigger input. For the detailed configuration, please refer to ***Table 6-1. Trigger input bit fields selection***.

6.5.16. Trigger selection for CAN1 register (TRIGSEL_CAN1)

Address offset: 0x48

Reset value: 0x0000 0039

This register 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_CAN1 register. 0: TRIGSEL_CAN1 register write is enabled. 1: TRIGSEL_CAN1 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 output is used as the source of CAN1_EX_TIME_TICK trigger input. For the detailed configuration, please refer to ***Table 6-1. Trigger input bit fields selection***.

6.5.17. Trigger selection for CAN2 register (TRIGSEL_CAN2)

Address offset: 0x4C

Reset value: 0x0000 0039

This register 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_CAN2 register. 0: TRIGSEL_CAN2 register write is enabled. 1: TRIGSEL_CAN2 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 output is used as the source of CAN2_EX_TIME_TICK trigger input. For the detailed configuration, please refer to ***Table 6-1. Trigger input bit fields selection***.

6.5.18. Trigger selection for TIMER0_ETI register (TRIGSEL_TIMER0ETI)

Address offset: 0x50

Reset value: 0x0000 0020

This register 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_TIMER0ETI register.

		0: TRIGSEL_TIMER0ETI register write is enabled. 1: TRIGSEL_TIMER0ETI register write is disabled.
30:8	Reserved	Must be kept at reset value.
7:0	INSEL0[7:0]	Trigger input source selection for output0 These bits are used to select trigger input signal connected to output0. The output is used as the source of TIMER0_ETI trigger input. For the detailed configuration, please refer to ***Table 6-1. Trigger input bit fields selection***.

6.5.19. Trigger selection for TIMER1_ETI register (TRIGSEL_TIMER1ETI)

Address offset: 0x54
Reset value: 0x0000 0026

This register 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_TIMER1ETI register. 0: TRIGSEL_TIMER1ETI register write is enabled. 1: TRIGSEL_TIMER1ETI 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 output is used as the source of TIMER1_ETI trigger input. For the detailed configuration, please refer to ***Table 6-1. Trigger input bit fields selection***.

6.5.20. Trigger selection for TIMER2_ETI register (TRIGSEL_TIMER2ETI)

Address offset: 0x58
Reset value: 0x0000 002C

This register 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_TIMER2ETI register. 0: TRIGSEL_TIMER2ETI register write is enabled. 1: TRIGSEL_TIMER2ETI 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 output is used as the source of TIMER2_ETI trigger input. For the detailed configuration, please refer to ***Table 6-1. Trigger input bit fields selection***.

6.5.21. Trigger selection for TIMER3_ETI register (TRIGSEL_TIMER3ETI)

Address offset: 0x5C
Reset value: 0x0000 0032

This register 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_TIMER3ETI register. 0: TRIGSEL_TIMER3ETI register write is enabled. 1: TRIGSEL_TIMER3ETI 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 output is used as the source of TIMER3_ETI trigger input. For the detailed configuration, please refer to ***Table 6-1. Trigger input bit fields selection***.

6.5.22. Trigger selection for TIMER4_ETI register (TRIGSEL_TIMER4ETI)

Address offset: 0x60

Reset value: 0x0000 0038

This register 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_TIMER4ETI register. 0: TRIGSEL_TIMER4ETI register write is enabled. 1: TRIGSEL_TIMER4ETI 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 output is used as the source of TIMER4_ETI trigger input. For the detailed configuration, please refer to ***Table 6-1. Trigger input bit fields selection***.

6.5.23. Trigger selection for TIMER7_ETI register (TRIGSEL_TIMER7ETI)

Address offset: 0x64

Reset value: 0x0000 004B

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

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

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

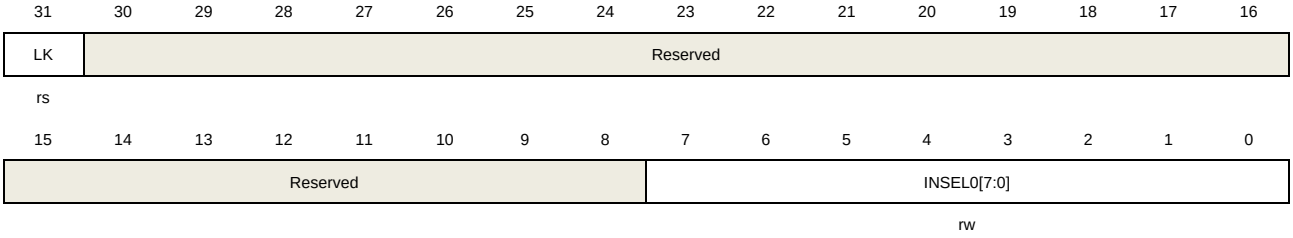
Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER7ETI register.

		0: TRIGSEL_TIMER7ETI register write is enabled. 1: TRIGSEL_TIMER7ETI 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 output is used as the source of TIMER7_ETI trigger input. For the detailed configuration, please refer to ***Table 6-1. Trigger input bit fields selection***.

6.5.24. Trigger selection for TIMER19_ETI register (TRIGSEL_TIMER19ETI)

Address offset: 0x68
Reset value: 0x0000 006C

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

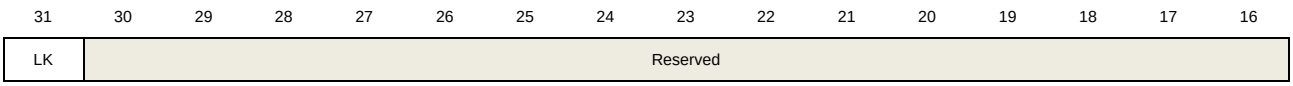


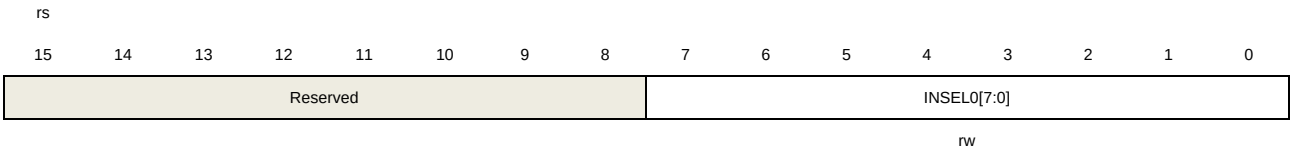
Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER19ETI register. 0: TRIGSEL_TIMER19ETI register write is enabled. 1: TRIGSEL_TIMER19ETI register write is disabled.
30:8	Reserved	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 output is used as the source of TIMER19_ETI trigger input. For the detailed configuration, please refer to ***Table 6-1. Trigger input bit fields selection***.

6.5.25. Trigger selection for HPDF_ITRG register (TRIGSEL_HPDF)

Address offset: 0x6C
Reset value: 0x0000 0000

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





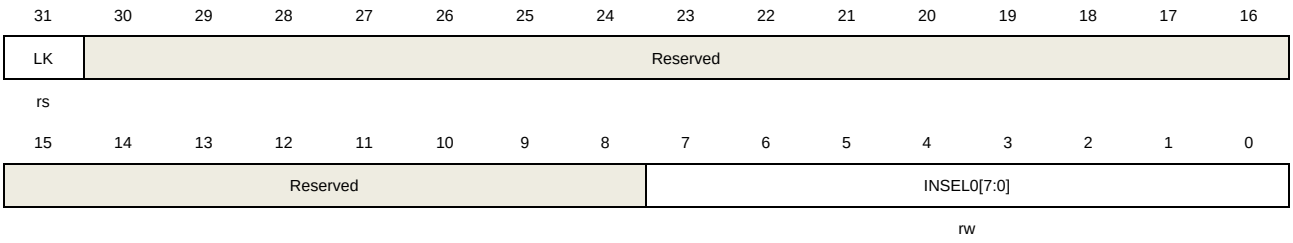
Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_HPDF register. 0: TRIGSEL_HPDF register write is enabled. 1: TRIGSEL_HPDF register write is disabled.
30:8	Reserved	Must be kept at 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 output is used as the source of HPDF_ITRG trigger input. For the detailed configuration, please refer to ***Table 6-1. Trigger input bit fields selection***.

6.5.26. Trigger selection for TIMER0_ITI14 register (TRIGSEL_TIMER0ITI14)

Address offset: 0x70

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_TIMER0ITI14 register. 0: TRIGSEL_TIMER0ITI14 register write is enabled. 1: TRIGSEL_TIMER0ITI14 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 output is used as the source of TIMER0_ITI14 trigger input. For the detailed configuration, please refer to ***Table 6-1. Trigger input bit fields selection***.

6.5.27. Trigger selection for TIMER1_ITI14 register (TRIGSEL_TIMER1ITI14)

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
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_TIMER1ITI14 register. 0: TRIGSEL_TIMER1ITI14 register write is enabled. 1: TRIGSEL_TIMER1ITI14 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 output is used as the source of TIMER1_ITI14 trigger input. For the detailed configuration, please refer to ***Table 6-1. Trigger input bit fields selection***.

6.5.28. Trigger selection for TIMER2_ITI14 register (TRIGSEL_TIMER2ITI14)

Address offset: 0x78

Reset value: 0x0000 0000

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

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

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

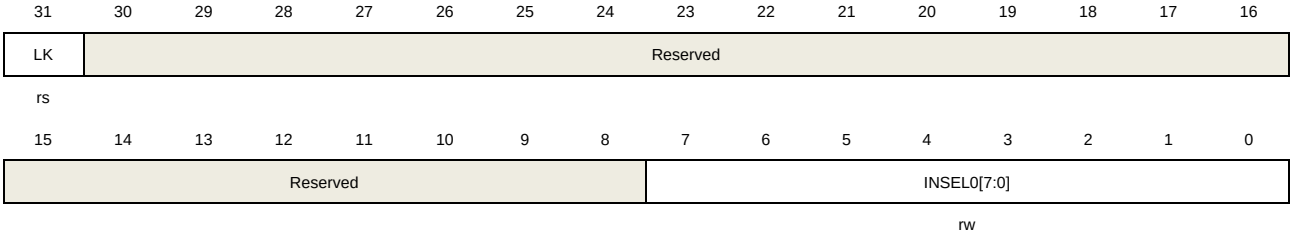
Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER2ITI14 register.

		0: TRIGSEL_TIMER2ITI14 register write is enabled. 1: TRIGSEL_TIMER2ITI14 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 output is used as the source of TIMER2_ITI14 trigger input. For the detailed configuration, please refer to ***Table 6-1. Trigger input bit fields selection***.

6.5.29. Trigger selection for TIMER3_ITI14 register (TRIGSEL_TIMER3ITI14)

Address offset: 0x7C
Reset value: 0x0000 0000

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

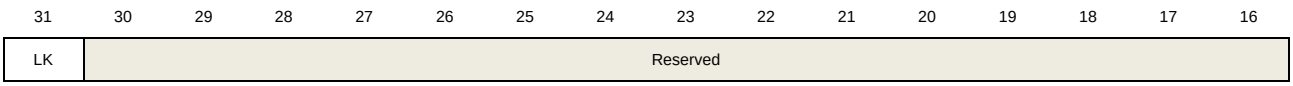


Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER3ITI14 register. 0: TRIGSEL_TIMER3ITI14 register write is enabled. 1: TRIGSEL_TIMER3ITI14 register write is disabled.
30:8	Reserved	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 output is used as the source of TIMER3_ITI14 trigger input. For the detailed configuration, please refer to ***Table 6-1. Trigger input bit fields selection***.

6.5.30. Trigger selection for TIMER4_ITI14 register (TRIGSEL_TIMER4ITI14)

Address offset: 0x80
Reset value: 0x0000 0000

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



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_TIMER4ITI14 register. 0: TRIGSEL_TIMER4ITI14 register write is enabled. 1: TRIGSEL_TIMER4ITI14 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 output is used as the source of TIMER4_ITI14 trigger input. For the detailed configuration, please refer to ***Table 6-1. Trigger input bit fields selection***.

6.5.31. Trigger selection for TIMER7_ITI14 register (TRIGSEL_TIMER7ITI14)

Address offset: 0x84

Reset value: 0x0000 0000

This register 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_TIMER7ITI14 register. 0: TRIGSEL_TIMER7ITI14 register write is enabled. 1: TRIGSEL_TIMER7ITI14 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 output is used as the source of TIMER7_ITI14 trigger input. For the detailed configuration, please refer to ***Table 6-1. Trigger input bit fields selection***.

6.5.32. Trigger selection for TIMER14_ITI14 register (TRIGSEL_TIMER14ITI14)

Address offset: 0x88

Reset value: 0x0000 0000

This register 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_TIMER14ITI14 register. 0: TRIGSEL_TIMER14ITI14 register write is enabled. 1: TRIGSEL_TIMER14ITI14 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 output is used as the source of TIMER14_ITI14 trigger input. For the detailed configuration, please refer to ***Table 6-1. Trigger input bit fields selection***.

6.5.33. Trigger selection for TIMER19_ITI14 register (TRIGSEL_TIMER19ITI14)

Address offset: 0x8C

Reset value: 0x0000 0000

This register 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							

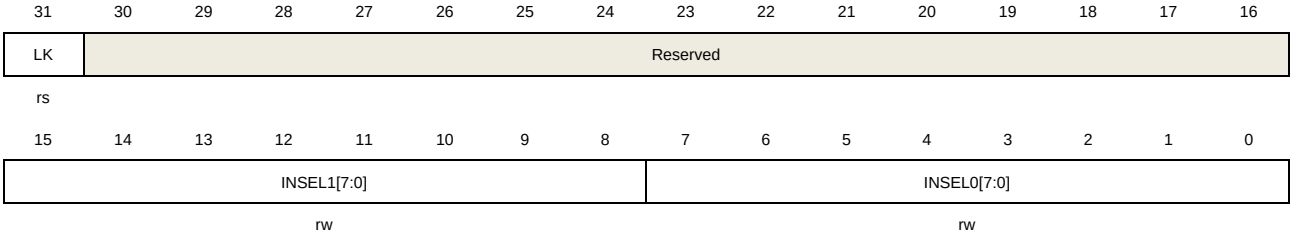
Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER19ITI14 register.

		0: TRIGSEL_TIMER19ITI14 register write is enabled. 1: TRIGSEL_TIMER19ITI14 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 output is used as the source of TIMER19_ITI14 trigger input. For the detailed configuration, please refer to ***Table 6-1. Trigger input bit fields selection***.

6.5.34. Trigger selection for DAC0 register (TRIGSEL_DAC0)

Address offset: 0x90
Reset value: 0x0000 3B3B

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



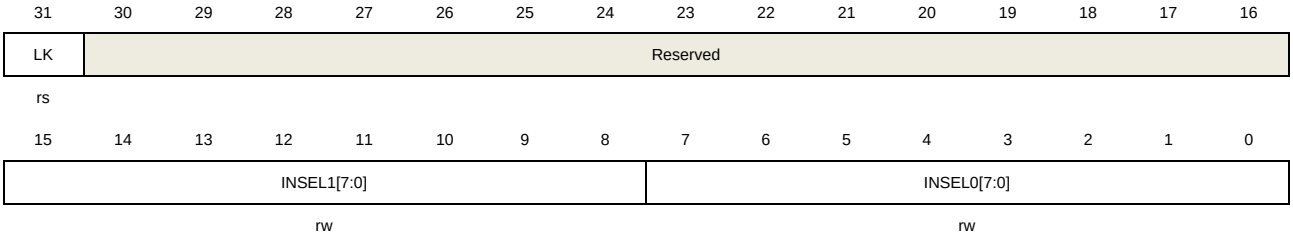
Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_DAC0 register. 0: TRIGSEL_DAC0 register write is enabled. 1: TRIGSEL_DAC0 register write is disabled.
30:16	Reserved	Must be kept at 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 output is used as the source of DAC0_OUT1_EXTRIG (DAC0_OUT1 external trigger) input. For the detailed configuration, please refer to ***Table 6-1. 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 output is used as the source of DAC0_OUT0_EXTRIG (DAC0_OUT0 external trigger) input. For the detailed configuration, please refer to ***Table 6-1. Trigger input bit fields selection***.

6.5.35. Trigger selection for DAC1 register (TRIGSEL_DAC1)

Address offset: 0x94

Reset value: 0x0000 3B3B

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



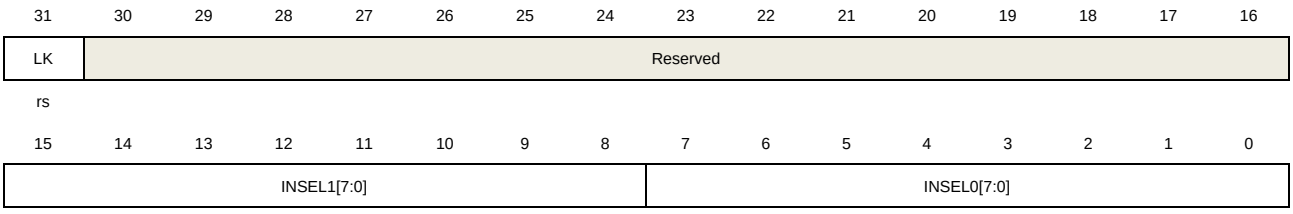
Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_DAC1 register. 0: TRIGSEL_DAC1 register write is enabled. 1: TRIGSEL_DAC1 register write is disabled.
30:16	Reserved	Must be kept at 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 output is used as the source of DAC1_OUT1_EXTRIG (DAC1_OUT1 external trigger) input. For the detailed configuration, please refer to ***Table 6-1. 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 output is used as the source of DAC1_OUT0_EXTRIG (DAC1_OUT0 external trigger) input. For the detailed configuration, please refer to ***Table 6-1. Trigger input bit fields selection***.

6.5.36. Trigger selection for DAC2 register (TRIGSEL_DAC2)

Address offset: 0x98

Reset value: 0x0000 1010

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



rw rw

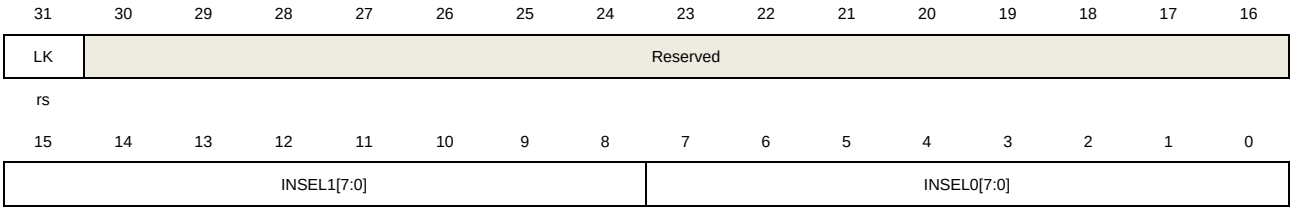
Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_DAC2 register. 0: TRIGSEL_DAC2 register write is enabled. 1: TRIGSEL_DAC2 register write is disabled.
30:16	Reserved	Must be kept at 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 output is used as the source of DAC2_OUT1_EXTRIG (DAC2_OUT1 external trigger) input. For the detailed configuration, please refer to ***Table 6-1. 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 output is used as the source of DAC2_OUT0_EXTRIG (DAC2_OUT0 external trigger) input. For the detailed configuration, please refer to ***Table 6-1. Trigger input bit fields selection***.

6.5.37. Trigger selection for DAC3 register (TRIGSEL_DAC3)

Address offset: 0x9C

Reset value: 0x0000 3B3B

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_DAC3 register. 0: TRIGSEL_DAC3 register write is enabled. 1: TRIGSEL_DAC3 register write is disabled.
30:16	Reserved	Must kept at reset value.
15:8	INSEL1[7:0]	Trigger input source selection for output1

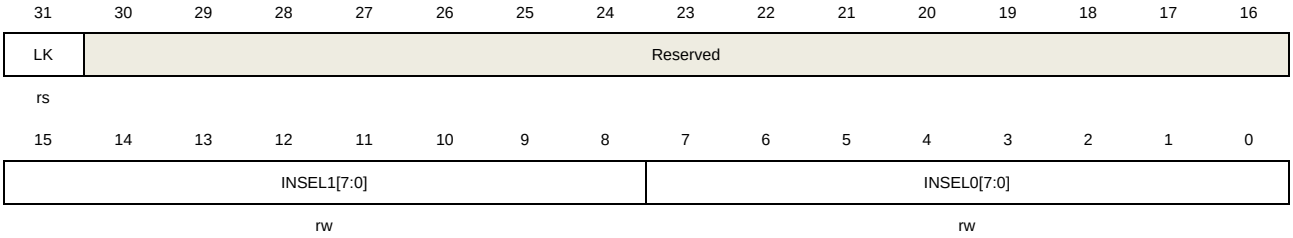
		These bits are used to select trigger input signal connected to output1. The output is used as the source of DAC3_OUT1_EXTRIG (DAC3_OUT1 external trigger) input. For the detailed configuration, please refer to ***Table 6-1. Trigger input bit fields selection***.
7: 0	INSEL0[7:0]	Trigger input source selection for output0 These bits are used to select trigger input signal connected to output0. The output is used as the source of DAC3_OUT0_EXTRIG (DAC3_OUT0 external trigger) input. For the detailed configuration, please refer to ***Table 6-1. Trigger input bit fields selection***.

6.5.38. Trigger selection for DAC0 extended register (TRIGSEL_EXTDAC0)

Address offset: 0xA0

Reset value: 0x0000 3B3B

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



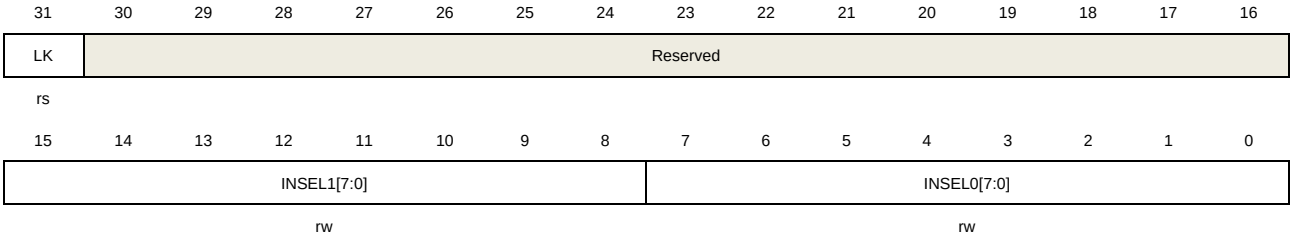
Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_EXTDAC0 register. 0: TRIGSEL_EXTDAC0 register write is enabled. 1: TRIGSEL_EXTDAC0 register write is disabled.
30:16	Reserved	Must 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 output is used as the souce of DAC0_OUT1_ST_EXTRIG (DAC0_OUT1 sawtooth step-up/down external trigger) trigger input. For the detail configuration, please see ***Table 6-1. 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 output is used as the souce of DAC0_OUT0_ST_EXTRIG (DAC0_OUT0 sawtooth step-up/down external trigger) trigger input. For the detail configuration, please see ***Table 6-1. Trigger input bit fields selection***.

6.5.39. Trigger selection for DAC1 extended register (TRIGSEL_EXTDAC1)

Address offset: 0xA4

Reset value: 0x0000 3B3B

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



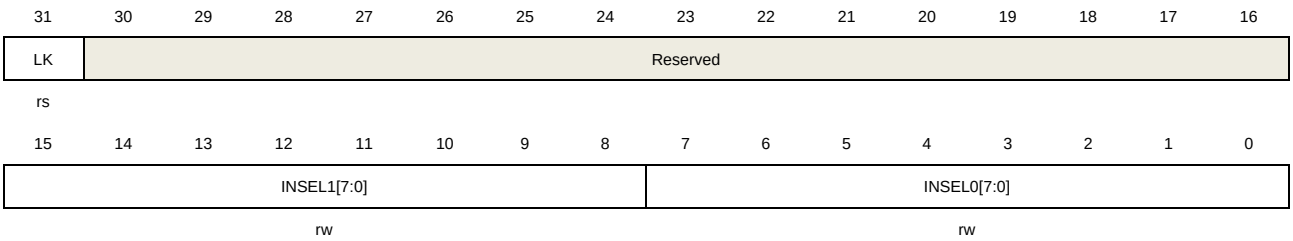
Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_EXTDAC1 register. 0: TRIGSEL_EXTDAC1 register write is enabled. 1: TRIGSEL_EXTDAC1 register write is disabled.
30:16	Reserved	Must 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 output is used as the souce of DAC1_OUT1_ST_EXTRIG (DAC1_OUT1 sawtooth step-up/down external trigger) trigger input. For the detail configuration, please see ***Table 6-1. 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 output is used as the souce of DAC1_OUT0_ST_EXTRIG (DAC1_OUT0 sawtooth step-up/down external trigger) trigger input. For the detail configuration, please see ***Table 6-1. Trigger input bit fields selection***.

6.5.40. Trigger selection for DAC2 extended register (TRIGSEL_EXTDAC2)

Address offset: 0xA8

Reset value: 0x0000 1010

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_EXTDAC2 register. 0: TRIGSEL_EXTDAC2 register write is enabled. 1: TRIGSEL_EXTDAC2 register write is disabled.
30:16	Reserved	Must 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 output is used as the souce of DAC2_OUT1_ST_EXTRIG (DAC2_OUT1 sawtooth step-up/down external trigger) trigger input. For the detail configuration, please see ***Table 6-1. 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 output is used as the souce of DAC2_OUT0_ST_EXTRIG (DAC2_OUT0 sawtooth step-up/down external trigger) trigger input. For the detail configuration, please see ***Table 6-1. Trigger input bit fields selection***.

6.5.41. Trigger selection for DAC3 extended register (TRIGSEL_EXTDAC3)

Address offset: 0xAC

Reset value: 0x0000 3B3B

This register 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_EXTDAC3 register. 0: TRIGSEL_EXTDAC3 register write is enabled. 1: TRIGSEL_EXTDAC3 register write is disabled.
30:16	Reserved	Must 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 output

		is used as the souce of DAC3_OUT1_ST_EXTRIG (DAC3_OUT1 sawtooth step-up/down external trigger) trigger input. For the detail configuration, please see ***Table 6-1. 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 output is used as the souce of DAC3_OUT0_ST_EXTRIG (DAC3_OUT0 sawtooth step-up/down external trigger) trigger input. For the detail configuration, please see ***Table 6-1. Trigger input bit fields selection***.

6.5.42. Trigger selection for CLA register 0 (TRIGSEL_CLA_0)

Address offset: 0xB0

Reset value: 0x00C1 C0BF

The reset of this register is depends on TRIGSEL_RSTMD in SYSCFG_CFG2. If TRIGSEL_RSTMD = 0, this register will reset when system reset event occurs. If TRIGSEL_RSTMD = 1, this register will reset only when POR reset event occurs.

This register 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 by a system or POR reset that depends on TRIGSEL_RSTMD. When it is set, it disables write access to TRIGSEL_CLA_0 register. 0: TRIGSEL_CLA_0 register write is enabled. 1: TRIGSEL_CLA_0 register write is disabled.
30:24	Reserved	Must kept at reset value.
23:16	INSEL2[7:0]	Trigger input source selection for output2 These bits are used to select trigger input signal connected to output2. The output is used as the souce of TRIGSEL_CLA_IN2 trigger input. For the detail configuration, please see ***Table 6-1. Trigger input bit fields selection***.
15:8	INSEL1[7:0]	Trigger input source selection for output1 These bits are used to select trigger input signal connected to output2. The output is used as the souce of TRIGSEL_CLA_IN1 trigger input. For the detail configuration, please see ***Table 6-1. Trigger input bit fields selection***.

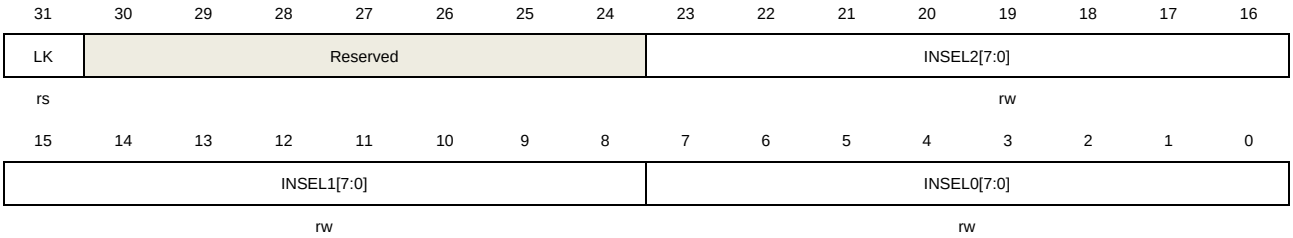
7:0	INSEL0[7:0]	Trigger input source selection for output0 These bits are used to select trigger input signal connected to output2. The output is used as the souce of TRIGSEL_CLA_IN0 trigger input. For the detail configuration, please see ***Table 6-1. Trigger input bit fields selection***.

6.5.43. Trigger selection for CLA register 1 (TRIGSEL_CLA_1)

Address offset: 0xB4
Reset value: 0x00C4 C3C2

The reset of this register is depends on TRIGSEL_RSTMD in SYSCFG_CFG2. If TRIGSEL_RSTMD = 0, this register will reset when system reset event occurs. If TRIGSEL_RSTMD = 1, this register will reset only when POR reset event occurs.

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 by a system or POR reset that depends on TRIGSEL_RSTMD. When it is set, it disables write access to TRIGSEL_CLA_1 register. 0: TRIGSEL_CLA_1 register write is enabled. 1: TRIGSEL_CLA_1 register write is disabled.
30:24	Reserved	Must kept at reset value.
23:16	INSEL2[7:0]	Trigger input source selection for output2 These bits are used to select trigger input signal connected to output2. The output is used as the souce of TRIGSEL_CLA_IN5 trigger input. For the detail configuration, please see ***Table 6-1. Trigger input bit fields selection***.
15:8	INSEL1[7:0]	Trigger input source selection for output1 These bits are used to select trigger input signal connected to output2. The output is used as the souce of TRIGSEL_CLA_IN4 trigger input. For the detail configuration, please see ***Table 6-1. 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 output2. The output is used as the souce of TRIGSEL_CLA_IN3 trigger input. For the detail configuration, please see ***Table 6-1. Trigger input bit fields selection***.

6.5.44. Trigger selection for CLA register 2 (TRIGSEL_CLA_2)

Address offset: 0xB8

Reset value: 0x0027 2110

This register 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_CLA_0 register. 0: TRIGSEL_CLA_0 register write is enabled. 1: TRIGSEL_CLA_0 register write is disabled.
30:24	Reserved	Must kept at reset value.
23:16	INSEL2[7:0]	Trigger input source selection for output2 These bits are used to select trigger input signal connected to output2. The output is used as the souce of TRIGSEL_CLA_IN8 trigger input. For the detail configuration, please see ***Table 6-1. Trigger input bit fields selection***.
15:	INSEL1[7:0]	Trigger input source selection for output1 These bits are used to select trigger input signal connected to output2. The output is used as the souce of TRIGSEL_CLA_IN7 trigger input. For the detail configuration, please see ***Table 6-1. 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 output2. The output is used as the souce of TRIGSEL_CLA_IN6 trigger input. For the detail configuration, please see ***Table 6-1. Trigger input bit fields selection***.

6.5.45. Trigger selection for CLA register 3 (TRIGSEL_CLA_3)

Address offset: 0xBC

Reset value: 0x0000 3A39

This register 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_CLA_3 register. 0: TRIGSEL_CLA_3 register write is enabled. 1: TRIGSEL_CLA_3 register write is disabled.
30: 16	Reserved	Must 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 output2. The output is used as the souce of TRIGSEL_CLA_IN10 trigger input. For the detail configuration, please see ***Table 6-1. 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 output2. The output is used as the souce of TRIGSEL_CLA_IN9 trigger input. For the detail configuration, please see ***Table 6-1. Trigger input bit fields selection***.

6.5.46. Trigger selection for CLA register 4 (TRIGSEL_CLA_4)

Address offset: 0xC0

Reset value: 0x0000 00EF

This register 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_CLA_4 register. 0: TRIGSEL_CLA_4 register write is enabled. 1: TRIGSEL_CLA_4 register write is disabled.
30:8	Reserved	Must 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 output2. The output is used as the souce of TRIGSEL_CLA_IN11 trigger input. For the detail configuration, please see ***Table 6-1. Trigger input bit fields selection***.

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

7.1. Overview

There are up to 107 general purpose I / O pins (GPIO), named PA0~PA15, PB0~PB15, PC0~PC15, PD0~PD15, PE0~PE15, PF0~PF15, PG0~PG10 for the device to implement logic input/output functions. Each GPIO port has related control and configuration registers to satisfy the requirements of specific applications. The external interrupts on the GPIO pins of the device have related control and configuration registers in the Interrupt / Event Controller Unit (EXTI).

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

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

7.2. Characteristics

- Input/output direction control.
- Schmitt trigger input function enable control.
- Each pin weak pull-up / pull-down function.
- Output push-pull / open drain enable control.
- Output set / reset control.
- External interrupt with programmable trigger edge – using EXTI configuration registers.
- Analog input / output configuration.
- Alternate function input / output configuration.
- Port configuration lock.
- Single cycle toggle output capability.
- The input signals can be qualified to remove unwanted noise.
- GPIO have the capable of state retention when system reset (exclude power reset).

7.3. 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

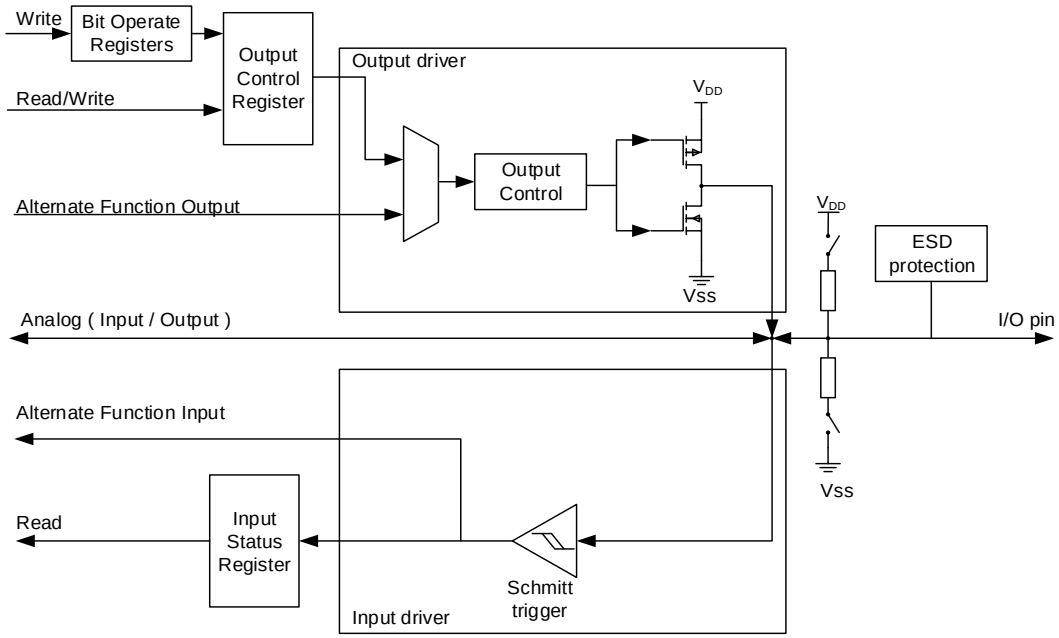
the port is output (GPIO output or AFIO output), it can be configured as push-pull or open drain mode by GPIO output mode registers (GPIOx_OMODE). And the port max speed can be configured by GPIO output speed registers (GPIOx_OSPD). Each port can be configured as floating (no pull-up and pull-down), pull-up or pull-down function by GPIO pull-up / pull-down registers (GPIOx_PUD).

Table 7-1. GPIO configuration table

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

Figure 7-1. Basic structure of a standard I/O port bit shows the basic structure of an I / O port bit.

Figure 7-1. Basic structure of a standard I/O port bit



7.3.1. 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.

PB8 / BOOT0 is in input mode during the reset until at least the end of the option byte loading phase.PB8 may be used as boot pin (BOOT0) or as a GPIO. Depending on the nSWBOOT0 bit in the user option byte, it switches from the input mode to the analog input mode:

- After the option byte loading phase if nSWBOOT0 = 1.
- After reset if nSWBOOT0 = 0.

Note: It is recommended to set PB8 in another mode than analog mode to limit consumption if the pin is left unconnected.

Using PG10 as GPIO: PG10 may be used as reset pin (NRST) or as a GPIO. Depending on the NRST_MODE bits in the user option byte, it switches to those mode:

- Reset input/output: default at power-on reset or after option bytes loading NRST_MODE = 3.

- Reset input only: after option bytes loading NRST_MODE = 1.
- GPIO PG10 mode: after option bytes loading NRST_MODE = 2.

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

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

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

7.3.2. External interrupt / event lines

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

7.3.3. Alternate functions (AF)

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

7.3.4. Additional functions

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

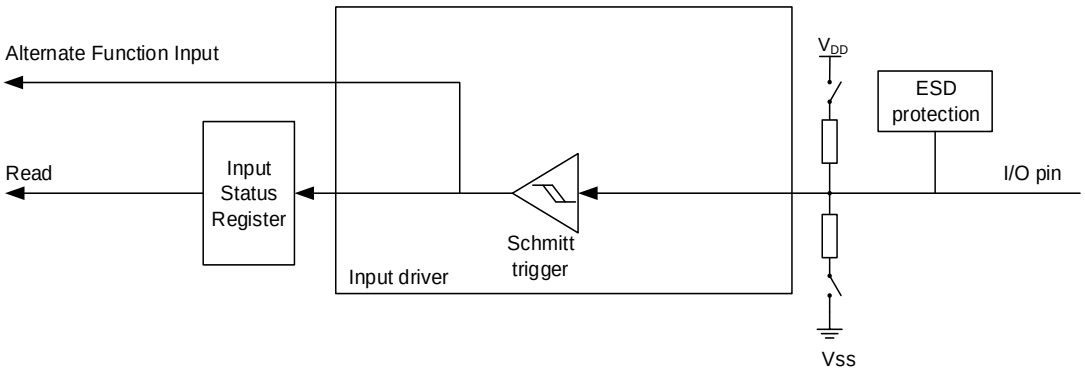
7.3.5. Input configuration

When GPIO pin is configured as input:

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

Figure 7-2. Input configuration shows the input configuration.

Figure 7-2. Input configuration



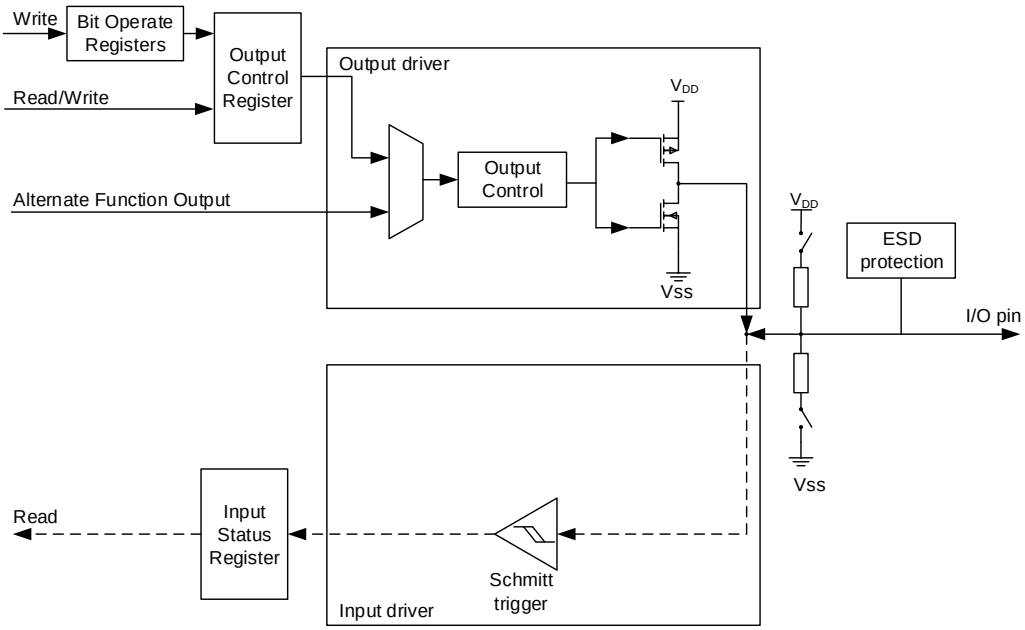
7.3.6. Output configuration

When GPIO pin is configured as output:

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

Figure 7-3. Output configuration shows the output configuration.

Figure 7-3. Output configuration



7.3.7. Analog configuration

When GPIO pin is used as analog configuration:

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

Figure 7-4. Analog configuration shows the analog configuration.

Figure 7-4. Analog configuration



7.3.8. Alternate function (AF) configuration

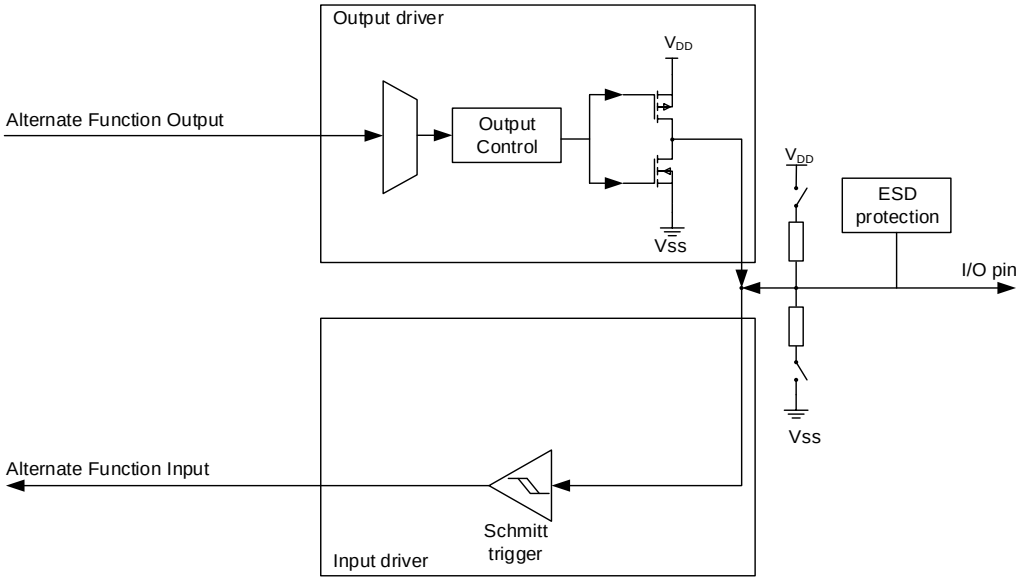
To suit for different device packages, the GPIO supports some alternate functions mapped to some other pins by software.

When be configured as alternate function:

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

Figure 7-5. Alternate function configuration shows the alternate function configuration.

Figure 7-5. Alternate function configuration



7.3.9. GPIO locking function

The locking mechanism allows the IO configuration to be protected.

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

7.3.10. GPIO single cycle toggle function

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

7.3.11. I/O compensation unit

The compensation unit is used to control the commutation slew rate (t_{fall} / t_{rise}) to reduce I / O noise on the power supply.

The unit is divided into two areas. The first area provides the best compensation code for the current PVT. When the CPS_RDY bit of SYSCFG_CPSCTL is set, the compensation code stored in this area can be read. The second area controls the I / O slew rate. The user selects the compensation code to be applied through software programming.

The I / O compensation unit has two voltage ranges: 1.62 to 2.0 V and 2.7 to 3.6 v.

7.3.12. 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_AHB or through the sampling window. For pins configured as peripheral input, in addition to synchronization to CK_AHB or through the sampling window, the input can also be asynchronous.

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_AHB 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.

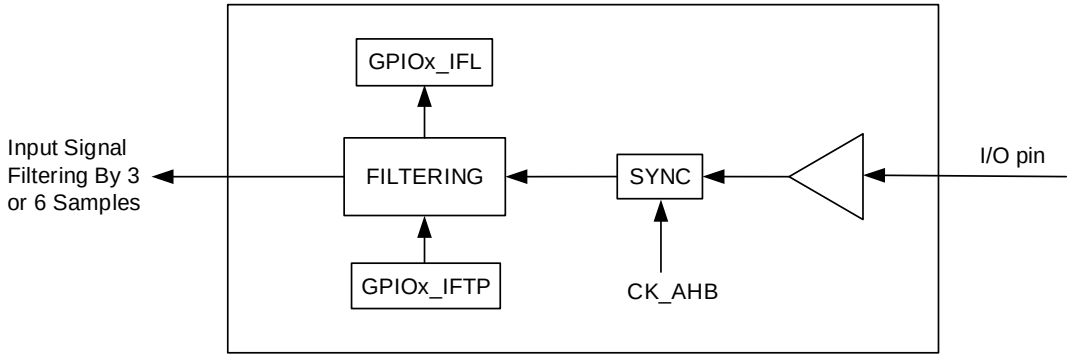
Synchronization to CK_AHB only

This is the default filter mode for all GPIO by default. In this mode, the input signal is only associated with CK_AHB synchronization. Since the input signal is asynchronous, it may need a delay period of CK_AHB 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_AHB), 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 7-6. 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_AHB.

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_AHB}. For example, if f_{CK_AHB}=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_AHB}×1÷(2×FLPRDx). For example, if f_{CK_AHB}=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 7-7. 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_AHB 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_AHB}。

Example of sampling window

As shown in ***Figure 7-7. 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×FLPRD0×T_{CK_AHB}=2×T_{CK_AHB};

The configuration results are as follows:

- The sampling width is: T_{SW}=6×T_{SP}=6×2×FLPRD0×T_{CK_AHB}=6×2×T_{CK_AHB};
- If T_{CK_AHB}=10ns, the duration of the sampling window is:

$T_{SP}=2\times T_{CK_AHB}=2\times 10ns=20ns$

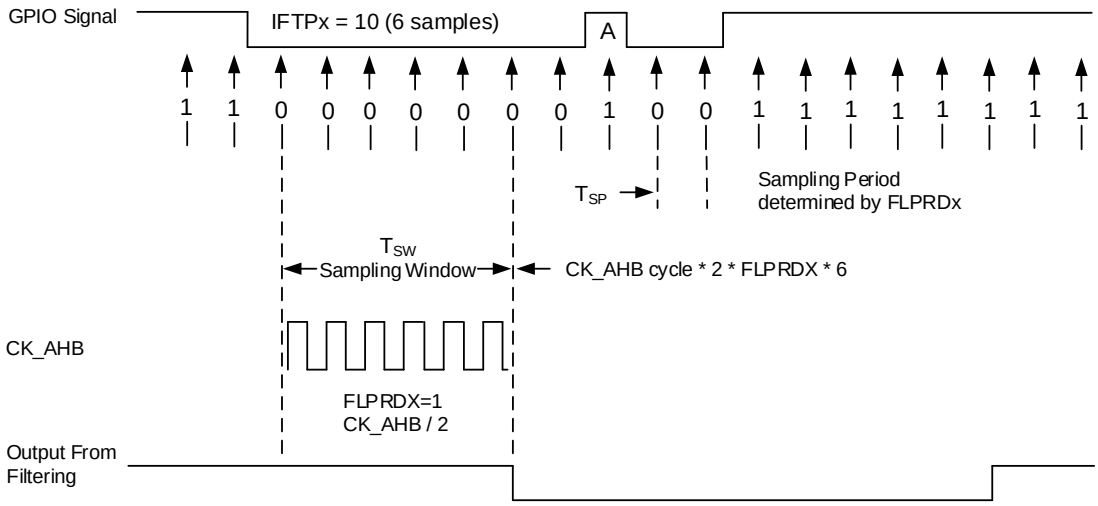
$T_{SW}=6\times 2\times T_{CK_AHB}=6\times 20ns=120ns$

- To illustrate the asynchronous nature of the input relative to the sampling period and the system clock, an additional smpling period and CK_AHB cycle may be required to detect the change of input signal.

  $T_{SW}+T_{CK_AHB}=120+10=130ns$

- In ***Figure 7-7. Input filtering clock cycle***, the interference (A) is shorter than the total sampling window, so it will be filtered.

Figure 7-7. Input filtering clock cycle



7.4. Register definition

GPIOA base address: 0x4800 0000

GPIOB base address: 0x4800 0400

GPIOC base address: 0x4800 0800

GPIOD base address: 0x4800 0C00

GPIOE base address: 0x4800 1000

GPIOF base address: 0x4800 1400

GPIOG base address: 0x4800 1800

7.4.1. Port control register (GPIOx_CTL, x=A..G)

Address offset: 0x00

Reset value: 0xABFF FFFF for port A; 0xFFFF 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: Input mode 01: GPIO output mode 10: Alternate function mode 11: Analog mode (reset value)

7.4.2. Port output mode register (GPIOx_OMODE, x=A..G)

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

7.4.3. Port output speed register (GPIOx_OSPD, x=A..G)

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 max speed 12M (reset value) 01: Output max speed 60M 10: Output max speed 85M 11: Output max speed 100/220M

7.4.4. Port pull-up/down register (GPIOx_PUD, x=A..G)

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

7.4.5. Port input status register (GPIOx_ISTAT, x=A..G)

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

7.4.6. Port output control register (GPIOx_OCTL, x=A..G)

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

1: Pin output high

7.4.7. Port bit operate register (GPIOx_BOP, x=A..G)

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

7.4.8. Port configuration lock register (GPIOx_LOCK, x=A..G)

Address offset: 0x1C

Reset value: 0x0000 0000.

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															LKK
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LK15	LK14	LK13	LK12	LK11	LK10	LK9	LK8	LK7	LK6	LK5	LK4	LK3	LK2	LK1	LK0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:17	Reserved	Must be kept at reset value
16	LKK	Lock key It can only be set by using the lock key writing sequence. And is always readable.

		0: GPIOx_LOCK register and the port configuration are not locked 1: GPIOx_LOCK register is locked until an MCU reset LOCK key writing sequence: Write 1→Write 0→Write 1→ Read 0→ Read 1 **Note:** The value of LKy(y=0..15) must be held during the LOCK Key writing sequence.
15:0	LKy	Port lock bit y(y=0..15) These bits are set and cleared by software. 0: Port configuration not locked 1: Port configuration locked

7.4.9. Alternate function selected register 0 (GPIOx_AFSEL0, x=A..G)

Address offset: 0x20

Reset value: 0x0000 0000.

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

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

Bits	Fields	Descriptions
31:28	SEL7[3:0]	Pin 7 alternate function selected These bits are set and cleared by software. Refer to SEL0[3:0] description
27:24	SEL6[3:0]	Pin 6 alternate function selected These bits are set and cleared by software. Refer to SEL0[3:0] description
23:20	SEL5[3:0]	Pin 5 alternate function selected These bits are set and cleared by software. Refer to SEL0[3:0] description
19:16	SEL4[3:0]	Pin 4 alternate function selected These bits are set and cleared by software. Refer to SEL0[3:0] description
15:12	SEL3[3:0]	Pin 3 alternate function selected These bits are set and cleared by software. Refer to SEL0[3:0] description
11:8	SEL2[3:0]	Pin 2 alternate function selected

		These bits are set and cleared by software. Refer to SEL0[3:0] description
7:4	SEL1[3:0]	Pin 1 alternate function selected These bits are set and cleared by software. Refer to SEL0[3:0] description
3:0	SEL0[3:0]	Pin 0 alternate function selected These bits are set and cleared by software. 0000: AF0 selected (reset value) 0001: AF1 selected 0010: AF2 selected 0011: AF3 selected 0100: AF4 selected 0101: AF5 selected 0110: AF6 selected 0111: AF7 selected 1000: AF8 selected 1001: AF9 selected 1010 ~ 1111: Reserved

7.4.10. Alternate function selected register 1 (GPIOx_AFSEL1, x=A..G)

Address offset: 0x24

Reset value: 0x0000 0000.

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

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

Bits	Fields	Descriptions
31:28	SEL15[3:0]	Pin 15 alternate function selected These bits are set and cleared by software. Refer to SEL8[3:0] description
27:24	SEL14[3:0]	Pin 14 alternate function selected These bits are set and cleared by software. Refer to SEL8[3:0] description
23:20	SEL13[3:0]	Pin 13 alternate function selected These bits are set and cleared by software.

		Refer to SEL8[3:0] description
19:16	SEL12[3:0]	Pin 12 alternate function selected These bits are set and cleared by software. Refer to SEL8[3:0] description
15:12	SEL11[3:0]	Pin 1 alternate function selected These bits are set and cleared by software. Refer to SEL8[3:0] description
11:8	SEL10[3:0]	Pin 10 alternate function selected These bits are set and cleared by software. Refer to SEL8[3:0] description
7:4	SEL9[3:0]	Pin 9 alternate function selected These bits are set and cleared by software. Refer to SEL8[3:0] description
3:0	SEL8[3:0]	Pin 8 alternate function selected These bits are set and cleared by software. 0000: AF0 selected (reset value) 0001: AF1 selected 0010: AF2 selected 0011: AF3 selected 0100: AF4 selected 0101: AF5 selected 0110: AF6 selected 0111: AF7 selected 1000: AF8 selected 1001: AF9 selected 1010 ~ 1111: Reserved

7.4.11. Bit clear register (GPIOx_BC, x=A..G)

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	Port clear bit y(y=0..15) These bits are set and cleared by software. 0: No action on the corresponding OCTLy bit 1: Clear the corresponding OCTLy bit

7.4.12. Port bit toggle register (GPIOx_TG, x=A..G)

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
TG15	TG14	TG13	TG12	TG11	TG10	TG9	TG8	TG7	TG6	TG5	TG4	TG3	TG2	TG1	TG0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

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

7.4.13. Input filtering register (GPIOx_IFL, x=A..G)

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_AHB 01: FLPRDx = CK_AHB / 2 02: FLPRDx = CK_AHB / 4 .... FF: FLPRDx = CK_AHB / 510
7:0	FLPRD0	Filter sampling period for GPIO1 to GPIO7: 00: FLPRDx = CK_AHB 01: FLPRDx = CK_AHB / 2 02: FLPRDx = CK_AHB / 4 .... FF: FLPRDx = CK_AHB / 510

7.4.14. Input filtering type register (GPIOx_IFTP, x=A…G)

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)

8. Direct memory access controller (DMA)

8.1. Overview

The direct memory access (DMA) controller provides a hardware method of transferring data between peripherals and/or memory without intervention from the CPU, thereby freeing up bandwidth for other system functions. Data can be quickly moved by DMA between peripherals and memory as well as memory and memory without any CPU actions. There are 14 channels in the DMA controller(7 for DMA0 and 7 for DMA1). Each channel is dedicated to manage memory access requests from one or more peripherals. An arbiter is implemented inside to handle the priority among DMA requests.

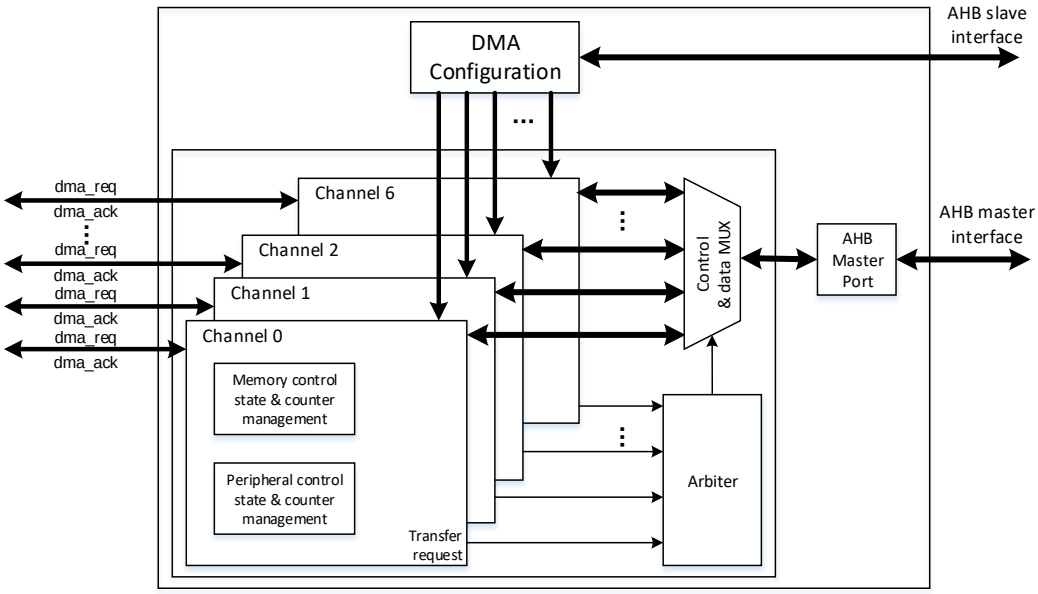
The system bus is shared by the DMA controller and the Cortex®-M33 core. When the DMA and the CPU are targeting the same destination, the DMA access may stop the CPU access to the system bus for some bus cycles. Round-robin scheduling is implemented in the bus matrix to ensure at least half of the system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

8.2. Characteristics

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

8.3. Block diagram

Figure 8-1. Block diagram of DMA



As shown in ***Figure 8-1. Block diagram of DMA***, a DMA controller consists of four main parts:

- DMA configuration through AHB slave interface.
- Data transmission through two AHB master interfaces for memory access and peripheral access.
- An arbiter inside to manage multiple peripheral requests coming at the same time.
- Channel management to control address/data selection and data counting.

8.4. Function overview

8.4.1. DMA operation

Each DMA transfer consists of two operations, including the loading of data from the source and the storage of the loaded data to the destination. The source and destination addresses are computed by the DMA controller based on the programmed values in the DMA_CHxPADDR, DMA_CHxMADDR, and DMA_CHxCTL registers. The DMA_CHxCNT register controls how many transfers to be transmitted on the channel. The PWIDTH and MWIDTH bits in the DMA_CHxCTL register determine how many bytes to be transmitted in a transfer.

Suppose DMA_CHxCNT is 4, and both PNAGA and MNAGA are set. The DMA transfer operations for each combination of PWIDTH and MWIDTH are shown in the following ***Table 8-1. DMA transfer operation***.

Table 8-1. DMA transfer operation

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

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

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

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

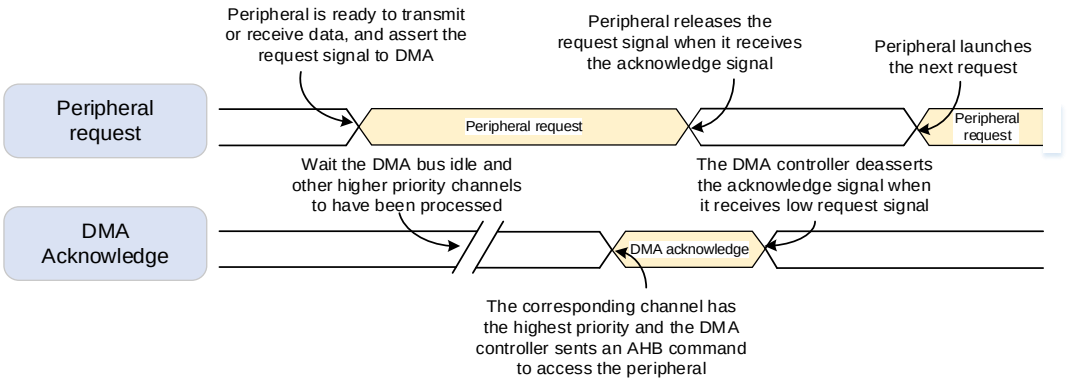
8.4.2. Peripheral handshake

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

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

Figure 8-2. Handshake mechanism shows how the handshake mechanism works between the DMA controller and peripherals.

Figure 8-2. Handshake mechanism



8.4.3. Arbitration

When two or more requests are received at the same time, the arbiter determines which request is served based on the priorities of channels. There are two-stage priorities, including

the software priority and the hardware priority. The arbiter determines which channel is selected to respond according to the following priority rules:

- Software priority: Four levels, including low, medium, high and ultra-high by configuring the PRIO bits in the DMA_CHxCTL register.
- For channels with equal software priority level, priority is given to the channel with lower channel number.

8.4.4. Address generation

Two kinds of address generation algorithm are implemented independently for memory and peripheral, including the fixed mode and the increased mode. The PNAGA and MNAGA bit in the DMA_CHxCTL register are used to configure the next address generation algorithm of peripheral and memory.

In the fixed mode, the next address is always equal to the base address configured in the base address registers (DMA_CHxPADDR, DMA_CHxMADDR).

In the increasing mode, the next address is equal to the current address plus 1 or 2 or 4, depending on the transfer data width.

8.4.5. Circular mode

Circular mode is implemented to handle continue peripheral requests (for example, ADC scan mode). The circular mode is enabled by setting the CMEN bit in the DMA_CHxCTL register.

In circular mode, the CNT bits are automatically reloaded with the pre-programmed value and the full transfer finish flag is asserted at the end of every DMA transfer. DMA can always responds the peripheral request until the CHEN bit in the DMA_CHxCTL register is cleared.

8.4.6. Memory to memory mode

The memory to memory mode is enabled by setting the M2M bit in the DMA_CHxCTL register. In this mode, the DMA channel can also work without being triggered by a request from a peripheral. The DMA channel starts transferring as soon as it is enabled by setting the CHEN bit in the DMA_CHxCTL register, and completed when the DMA_CHxCNT register reaches zero.

8.4.7. Channel configuration

When starting a new DMA transfer, it is recommended to respect the following steps:

1. Read the CHEN bit and judge whether the channel is enabled or not. If the channel is enabled, clear the CHEN bit by software. When the CHEN bit is read as ‘0’, configuring and starting a new DMA transfer is allowed.
2. Configure the M2M bit and DIR bit in the DMA_CHxCTL register to set the transfer mode.

3. Configure the CMEN bit in the DMA_CHxCTL register to enable/disable the circular mode.
4. Configure the PRIO bits in the DMA_CHxCTL register to set the channel software priority.
5. Configure the memory and peripheral transfer width, memory and peripheral address generation algorithm in the DMA_CHxCTL register.
6. Configure the enable bit for full transfer finish interrupt, half transfer finish interrupt, transfer error interrupt in the DMA_CHxCTL register.
7. Configure the DMA_CHxPADDR register for setting the peripheral base address.
8. Configure the DMA_CHxMADDR register for setting the memory base address.
9. Configure the DMA_CHxCNT register to set the total transfer data number.
10. Configure the CHEN bit with '1' in the DMA_CHxCTL register to enable the channel.

8.4.8. Interrupt

Each DMA channel has a dedicated interrupt. There are three types of interrupt event, including full transfer finish, half transfer finish, and transfer error.

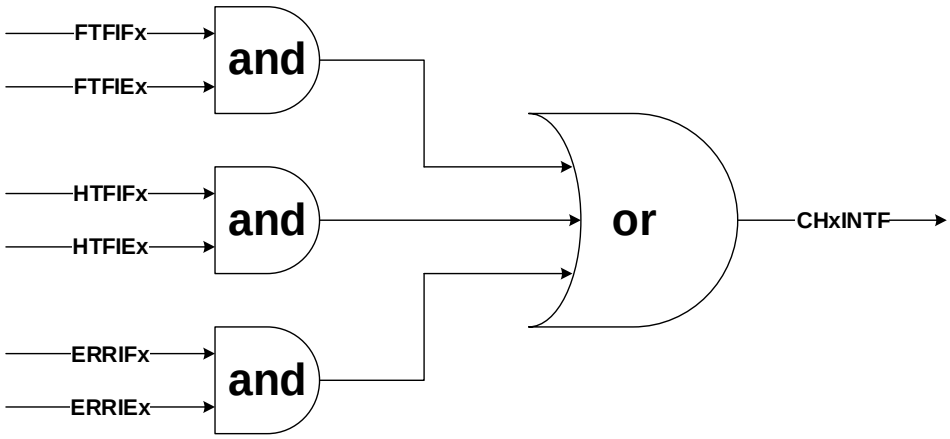
Each interrupt event has a dedicated flag bit in the DMA_INTF register, a dedicated clear bit in the DMA_INTC register, and a dedicated enable bit in the DMA_CHxCTL register. The relationship is described in the following ***Table 8-2. interrupt events***.

Table 8-2. interrupt events

Interrupt event	Flag bit	Clear bit	Enable bit
	DMA_INTF	DMA_INTC	DMA_CHxCTL
Full transfer finish	FTFIF	FTFIFC	FTFIE
Half transfer finish	HTFIF	HTFIFC	HTFIE
Transfer error	ERRIF	ERRIFC	ERRIE

The DMA interrupt logic is shown in the ***Figure 8-3. DMA interrupt logic***, an interrupt can be produced when any type of interrupt event occurs and enabled on the channel.

Figure 8-3. DMA interrupt logic



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

8.4.9. DMA request mapping

The DMA requests of a channel are coming from the AHB/APB peripherals through the corresponding channel output of DMAMUX request multiplexer, refer to ***Table 9-3. Request multiplexer input mapping***.

8.5. Register definition

DMA base address: 0x4002 0000

DMA1 base address: 0x4002 0400

8.5.1. Interrupt flag register (DMA_INTF)

Address offset: 0x00

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				ERRIF6	HTFIF6	FTFIF6	GIF6	ERRIF5	HTFIF5	FTFIF5	GIF5	ERRIF4	HTFIF4	FTFIF4	GIF4
				r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERRIF3	HTFIF3	FTFIF3	GIF3	ERRIF2	HTFIF2	FTFIF2	GIF2	ERRIF1	HTFIF1	FTFIF1	GIF1	ERRIF0	HTFIF0	FTFIF0	GIF0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

Bits	Fields	Descriptions
31:28	Reserved	Must be kept at reset value.
27/23/19/15 /11/7/3	ERRIFx	Error flag of channel x (x=0…6) Hardware set and software cleared by configuring DMA_INTC register. 0: Transfer error has not occurred on channel x 1: Transfer error has occurred on channel x
26/22/18/14 /10/6/2	HTFIFx	Half transfer finish flag of channel x (x=0…6) Hardware set and software cleared by configuring DMA_INTC register. 0: Half number of transfer has not finished on channel x 1: Half number of transfer has finished on channel x
25/21/17/13 /9/5/1	FTFIFx	Full Transfer finish flag of channel x (x=0…6) Hardware set and software cleared by configuring DMA_INTC register. 0: Transfer has not finished on channel x 1: Transfer has finished on channel x
24/20/16/12 /8/4/0	GIFx	Global interrupt flag of channel x (x=0…6) Hardware set and software cleared by configuring DMA_INTC register. 0: None of ERRIF, HTFIF or FTFIF occurs on channel x 1: At least one of ERRIF, HTFIF or FTFIF occurs on channel x

8.5.2. Interrupt flag clear register (DMA_INTC)

Address offset: 0x04

Reset value: 0x0000 0000

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

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

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

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

8.5.3. Channel x control register (DMA_CHxCTL)

x = 0...6, where x is a channel number

Address offset: 0x08 + 0x14 × x

Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	M2M	PRIO[1:0]		MWIDTH[1:0]		PWIDTH[1:0]		MNAGA	PNAGA	CMEN	DIR	ERRIE	HTFIE	FTFIE	CHEN
	rw	rw		rw		rw		rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:15	Reserved	Must be kept at reset value.
14	M2M	Memory to Memory mode Software set and cleared

		0: Disable Memory to Memory mode 1: Enable Memory to Memory mode This bit can not be written when CHEN is ‘1’.
13:12	PRIO[1:0]	Priority level Software set and cleared 00: Low 01: Medium 10: High 11: Ultra high These bits can not be written when CHEN is ‘1’.
11:10	MWIDTH[1:0]	Transfer data size of memory Software set and cleared 00: 8-bit 01: 16-bit 10: 32-bit 11: Reserved These bits can not be written when CHEN is ‘1’.
9:8	PWIDTH[1:0]	Transfer data size of peripheral Software set and cleared 00: 8-bit 01: 16-bit 10: 32-bit 11: Reserved These bits can not be written when CHEN is ‘1’.
7	MNAGA	Next address generation algorithm of memory Software set and cleared 0: Fixed address mode 1: Increasing address mode This bit can not be written when CHEN is ‘1’.
6	PNAGA	Next address generation algorithm of peripheral Software set and cleared 0: Fixed address mode 1: Increasing address mode This bit can not be written when CHEN is ‘1’.
5	CMEN	Circular mode enable Software set and cleared 0: Disable circular mode 1: Enable circular mode This bit can not be written when CHEN is ‘1’.
4	DIR	Transfer direction

		Software set and cleared 0: Read from peripheral and write to memory 1: Read from memory and write to peripheral This bit can not be written when CHEN is ‘1’.
3	ERRIE	Enable bit for channel error interrupt Software set and cleared 0: Disable the channel error interrupt 1: Enable the channel error interrupt
2	HTFIE	Enable bit for channel half transfer finish interrupt Software set and cleared 0: Disable channel half transfer finish interrupt 1: Enable channel half transfer finish interrupt
1	FTFIE	Enable bit for channel full transfer finish interrupt Software set and cleared 0: Disable channel full transfer finish interrupt 1: Enable channel full transfer finish interrupt
0	CHEN	Channel enable Software set and cleared 0: Disable channel 1: Enable channel

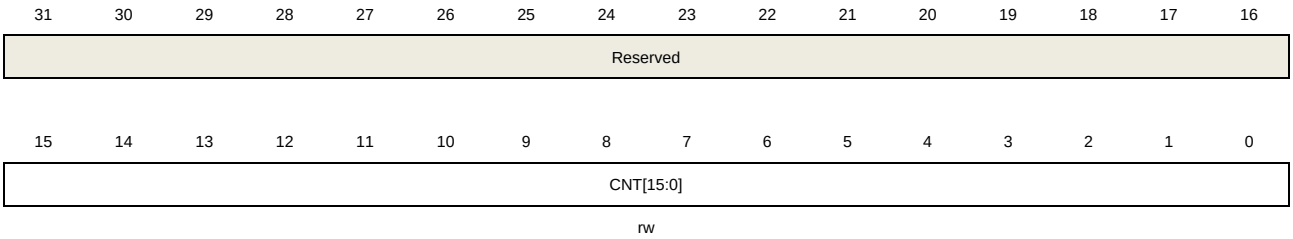
8.5.4. Channel x counter register (DMA_CHxCNT)

x = 0...6, where x is a channel number

Address offset: 0x0C + 0x14 × x

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CNT[15:0]	Transfer counter These bits can not be written when CHEN in the DMA_CHxCTL register is ‘1’. This register indicates how many transfers remain. Once the channel is enabled, it is read-only, and decreases after each DMA transfer. If the register is zero, no

transaction can be issued whether the channel is enabled or not. Once the transmission of the channel is complete, the register can be reloaded automatically by the previously programmed value if the channel is configured in circular mode.

8.5.5. Channel x peripheral base address register (DMA_CHxPADDR)

x = 0...6, where x is a channel number

Address offset: 0x10 + 0x14 × x

Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:0	PADDR[31:0]	Peripheral base address These bits can not be written when CHEN in the DMA_CHxCTL register is '1'. When PWIDTH is 01 (16-bit), the LSB of these bits is ignored. Access is automatically aligned to a half word address. When PWIDTH is 10 (32-bit), the two LSBs of these bits are ignored. Access is automatically aligned to a word address.

8.5.6. Channel x memory base address register (DMA_CHxMADDR)

x = 0...6, where x is a channel number

Address offset: 0x14 + 0x14 × x

Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:0	MADDR[31:0]	Memory base address These bits can not be written when CHEN in the DMA_CHxCTL register is '1'.

When MWIDTH in the DMA_CHxCTL register is 01 (16-bit), the LSB of these bits is ignored. Access is automatically aligned to a half word address.
When MWIDTH in the DMA_CHxCTL register is 10 (32-bit), the two LSBs of these bits are ignored. Access is automatically aligned to a word address.

9. DMA request multiplexer (DMAMUX)

9.1. Overview

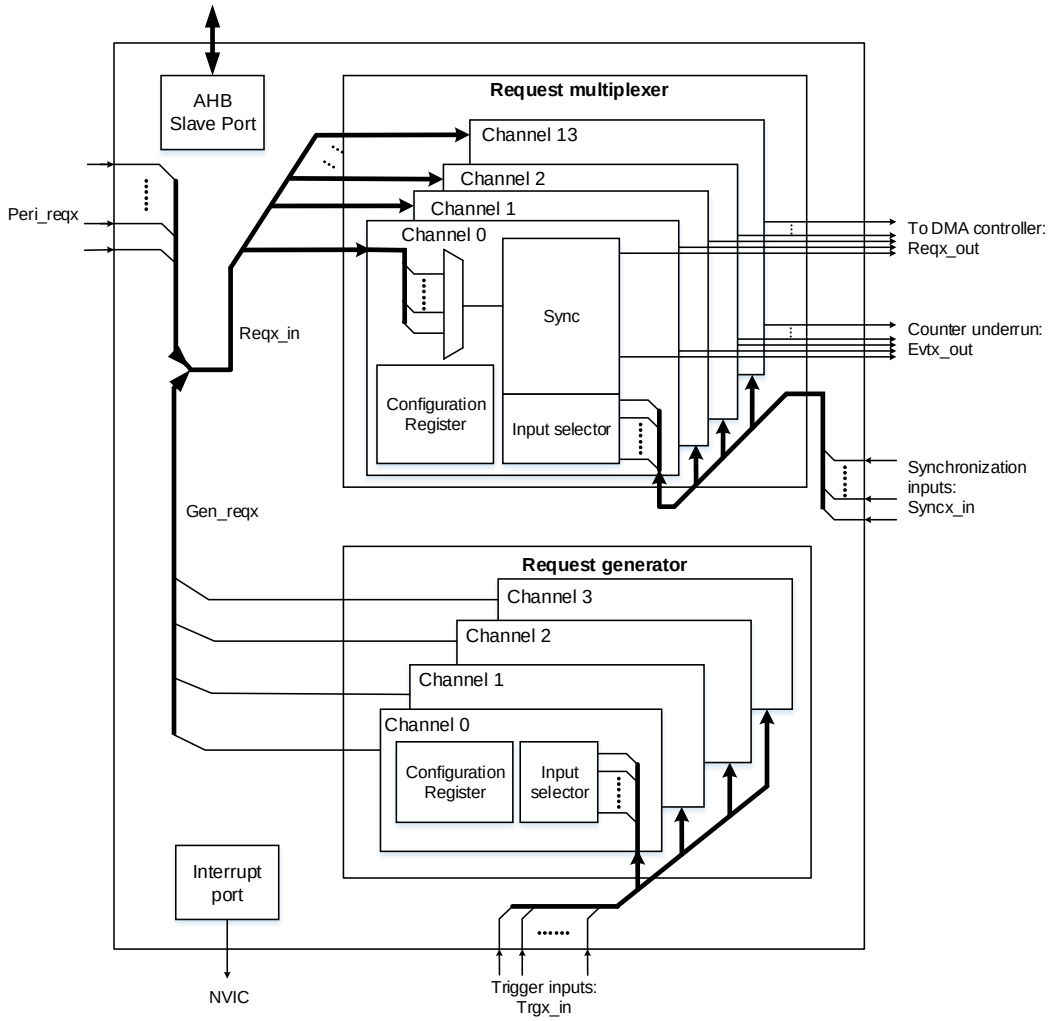
DMAMUX is a transmission scheduler for DMA requests. The DMAMUX request multiplexer is used for routing a DMA request line between the peripherals / generated DMA request (from the DMAMUX request generator) and the DMA controller. Each DMAMUX request multiplexer channel selects a unique DMA request line, 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).

9.2. Characteristics

- 14 channels for DMAMUX request multiplexer.
- 4 channels for DMAMUX request generator.
- Support 21 trigger inputs.
- Support 21 synchronization inputs.
- Each DMAMUX request generator channel:
  - DMA request trigger input selector
  - DMAMUX request generator counter
  - Trigger overrun flag
- Each DMAMUX request multiplexer channel:
  - 130 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

9.3. Block diagram

Figure 9-1. Block diagram of DMAMUX



9.4. Function overview

As shown in ***Figure 9-1. 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.

9.4.1. DMAMUX signals

Table 9-1. 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

9.4.2. DMAMUX request multiplexer

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

Request multiplexer channel

A DMA request input for the DMAMUX request multiplexer channel x is configured by the MUXID[6:0] bits in the DMAMUX_RM_CHxCFG register, sourced either from the peripherals or from the DMAMUX request generator, the sources can refer to ***Table 9-3. Request multiplexer input mapping***. A DMAMUX request multiplexer channel is connected and dedicated to one single channel of the DMA controller.

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

When synchronization mode is disabled

Each time the connected DMAMUX request is served by the DMA controller, the served DMA request is de-asserted, and the built-in DMAMUX request multiplexer counter is decremented. At the request multiplexer counter underrun, the built-in DMAMUX request multiplexer counter is automatically loaded with the value in NBR[4:0] bits of the DMAMUX_RM_CHxCFG register.

If the channel event generation is enabled by setting EVGEN bit, the number of DMA requests before an output event generation is NBR[4:0] + 1.

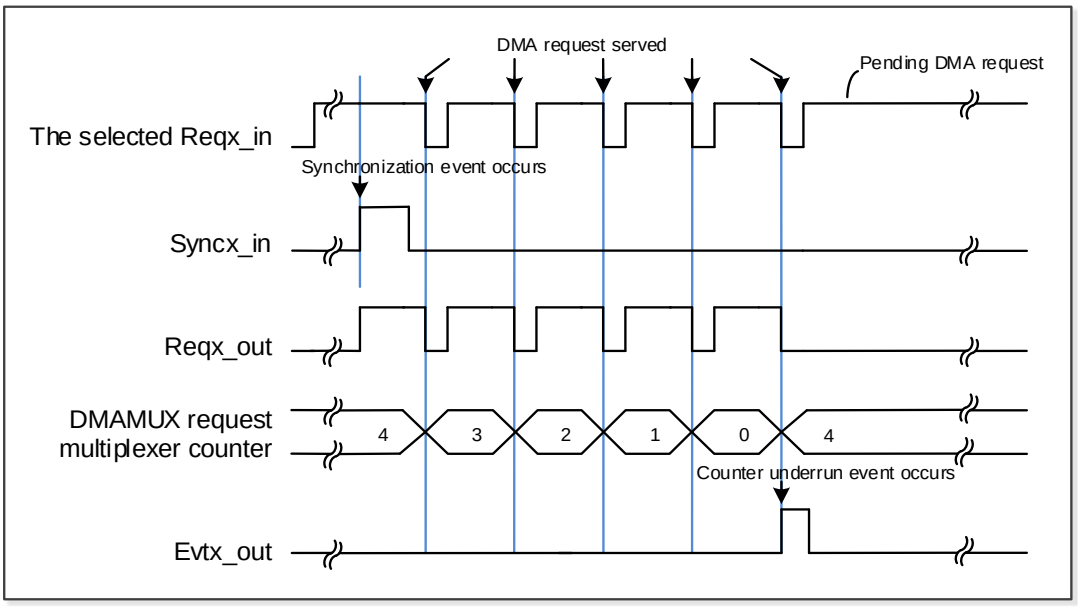
Note: The NBR[4:0] bits value shall only be written by software when both synchronization enable bit SYNCEN and event generation enable EVGEN bit of the corresponding request multiplexer channel x are disabled.

When synchronization mode is enabled

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

Figure 9-2. Synchronization mode shows an example when NBR[4:0]=4, SYNCEN=1, EVGEN=1, SYNCP[1:0]=1.

Figure 9-2. 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 9-5. Synchronization input mapping***. The synchronization input valid edge is configured by the SYNCP[1:0] bits of the DMAMUX_RM_CHxCFG register.

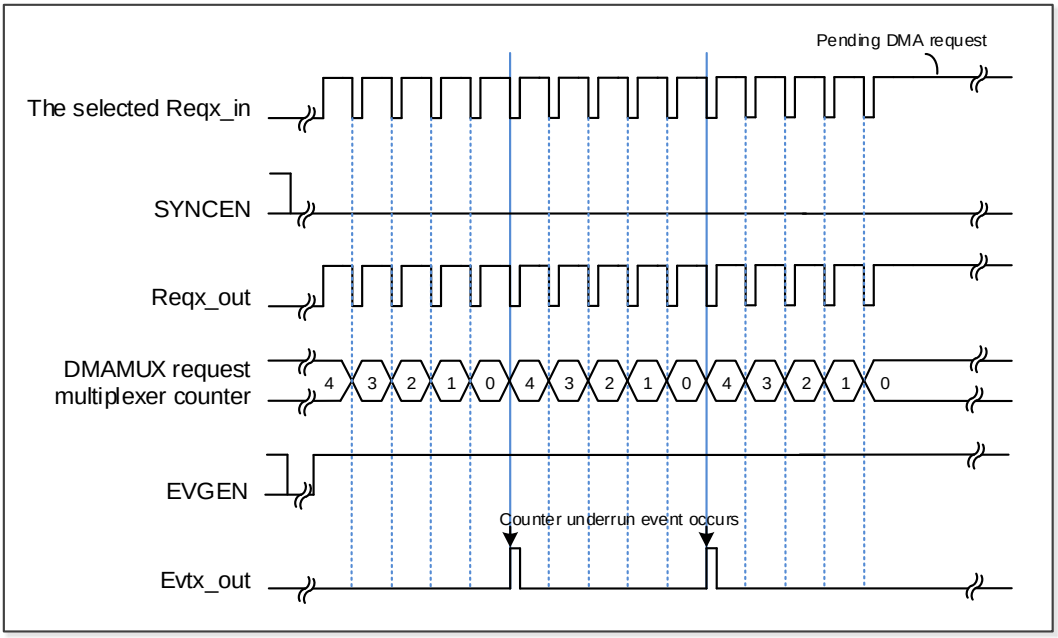
Note: If a synchronization input event occurs when there is no pending selected input DMA request line, the input event is discarded. The following asserted input request lines will not be routed to the DMAMUX multiplexer channel output until a synchronization input event occurs again.

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 9-3. Event generation shows an example when NBR[4:0]=4, SYNCEN=0, EVGEN=1.

Figure 9-3. 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.

9.4.3. 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 9-4. 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.

9.4.4. 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.

9.4.5. 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 9-2. Interrupt events***.

Table 9-2. 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.

9.4.6. DMAMUX mapping

Request multiplexer input mapping

A DMA request is sourced either from the peripherals or from the DMAMUX request generator, the sources can refer to ***Table 9-3. 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 9-3. 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
6	DAC0_CH0
7	DAC0_CH1
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	I2C2_RX
21	I2C2_TX
22	I2C3_RX
23	I2C3_TX
24	USART0_RX
25	USART0_TX
26	USART1_RX
27	USART1_TX
28	USART2_RX
29	USART2_TX

Request multiplexer channel input identification MUXID[7:0]	Source
30	UART3_RX
31	UART3_TX
32	UART4_RX
33	UART4_TX
34	ADC1
35	ADC2
36	ADC3
37	QSPI
38	DAC1_CH0
39	DAC1_CH1
40	TIMER0_CH0
41	TIMER0_CH1
42	TIMER0_CH2
43	TIMER0_CH3
44	TIMER0_CH0N
45	TIMER0_CH1N
46	TIMER0_CH2N
47	TIMER0_CH3N
48	TIMER0_ UP
49	TIMER0_TI
50	TIMER0_CO
51	TIMER7_CH0
52	TIMER7_CH1
53	TIMER7_CH2
54	TIMER7_CH3
55	TIMER7_CH0N
56	TIMER7_CH1N
57	TIMER7_CH2N
58	TIMER7_CH3N
59	TIMER7_ UP
60	TIMER7_TI
61	TIMER7_CO
62	TIMER1_CH0
63	TIMER1_CH1
64	TIMER1_CH2
65	TIMER1_CH3
66	TIMER1_ UP
67	TIMER1_TI
68	TIMER2_CH0

Request multiplexer channel input identification MUXID[7:0]	Source
69	TIMER2_CH1
70	TIMER2_CH2
71	TIMER2_CH3
72	TIMER2_ UP
73	TIMER2_TI
74	TIMER3_CH0
75	TIMER3_CH1
76	TIMER3_CH2
77	TIMER3_CH3
78	TIMER3_ UP
79	TIMER3_TI
80	TIMER4_CH0
81	TIMER4_CH1
82	TIMER4_CH2
83	TIMER4_CH3
84	TIMER4_ UP
85	TIMER4_TI
86	TIMER14_CH0
87	TIMER14_CH1
88	TIMER14_CH0N
89	TIMER14_ UP
90	TIMER14_ TI
91	TIMER14_ CO
92	TIMER15_CH0
93	TIMER15_CH0N
94	TIMER15_ UP
95	TIMER16_CH0
96	TIMER16_CH0N
97	TIMER16_ UP
98	TIMER19_CH0
99	TIMER19_CH1
100	TIMER19_CH2
101	TIMER19_CH3
102	TIMER19_CH0N
103	TIMER19_CH1N
104	TIMER19_CH2N
105	TIMER19_CH3N
106	TIMER19_ UP
107	TIMER19_ TI

Request multiplexer channel input identification MUXID[7:0]	Source
108	TIMER19_ CO
109	CAU_IN
110	CAU_OUT
111	HRTIMER_M
112	HRTIMER_0
113	HRTIMER_1
114	HRTIMER_2
115	HRTIMER_3
116	HRTIMER_4
117	HRTIMER_5
118	HRTIMER_6
119	HRTIMER_7
120	DAC2_CH0
121	DAC2_CH1
122	DAC3_CH0
123	DAC3_CH1
124	HPDF_FLT0
125	HPDF_FLT1
126	HPDF_FLT2
127	HPDF_FLT3
128	FAC_RD
129	FAC_WR
130	TMU_RD
131	TMU_WR
132	CAN0
133	CAN1
134	CAN2

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 9-4. Trigger input mapping***.

Table 9-4. Trigger input mapping

Trigger input identification TID[4:0]	Source
0	EXTI_0
1	EXTI_1
2	EXTI_2

Trigger input identification TID[4:0]	Source
3	EXTI_3
4	EXTI_4
5	EXTI_5
6	EXTI_6
7	EXTI_7
8	EXTI_8
9	EXTI_9
10	EXTI_10
11	EXTI_11
12	EXTI_12
13	EXTI_13
14	EXTI_14
15	EXTI_15
16	Evtx_out0
17	Evtx_out1
18	Evtx_out2
19	Evtx_out3
20	LPTIMER_OUT

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 9-5. Synchronization input mapping***.

Table 9-5. Synchronization input mapping

Synchronization input identification SYNCID[4:0]	Source
0	EXTI_0
1	EXTI_1
2	EXTI_2
3	EXTI_3
4	EXTI_4
5	EXTI_5
6	EXTI_6
7	EXTI_7
8	EXTI_8
9	EXTI_9
10	EXTI_10
11	EXTI_11
12	EXTI_12
13	EXTI_13

Synchronization input identification SYNCID[4:0]	Source
14	EXTI_14
15	EXTI_15
16	Evtx_out0
17	Evtx_out1
18	Evtx_out2
19	Evtx_out3
20	LPTIMER_OUT

9.5. Register definition

DMAMUX base address: 0x4002 0800

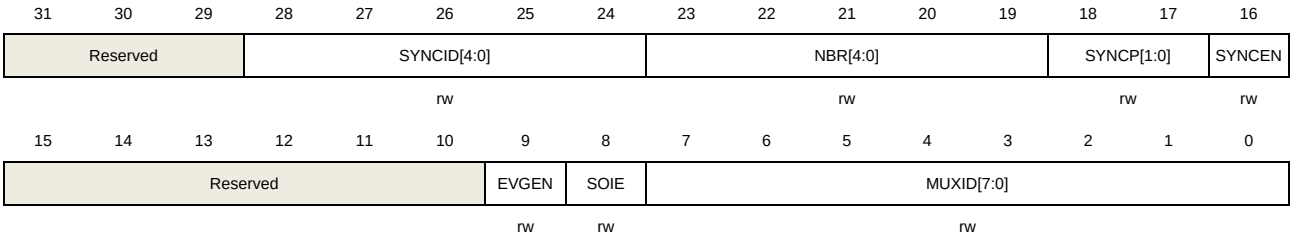
9.5.1. Request multiplexer channel x configuration register (DMAMUX_RM_CHxCFG)

x = 0...13, 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.

9.5.2. 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		SOIF13	SOIF12	SOIF11	SOIF10	SOIF9	SOIF8	SOIF7	SOIF6	SOIF5	SOIF4	SOIF3	SOIF2	SOIF1	SOIF0
		r	r	r	r	r	r	r	r	r	r	r	r	r	r

Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value.
13:0	SOIFx	Synchronization overrun event flag of request multiplexer channel x (x=0..13) 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.

9.5.3. 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		SOIFC13	SOIFC12	SOIFC11	SOIFC10	SOIFC9	SOIFC8	SOIFC7	SOIFC6	SOIFC5	SOIFC4	SOIFC3	SOIFC2	SOIFC1	SOIFC0
		w	w	w	w	w	w	w	w	w	w	w	w	w	w

Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value.
13:0	SOIFCx	Clear bit for synchronization overrun event flag of request multiplexer channel x (x=0..13) Writing 1 clears the corresponding overrun flag SOIFx in the DMAMUX_RM_INTF register.

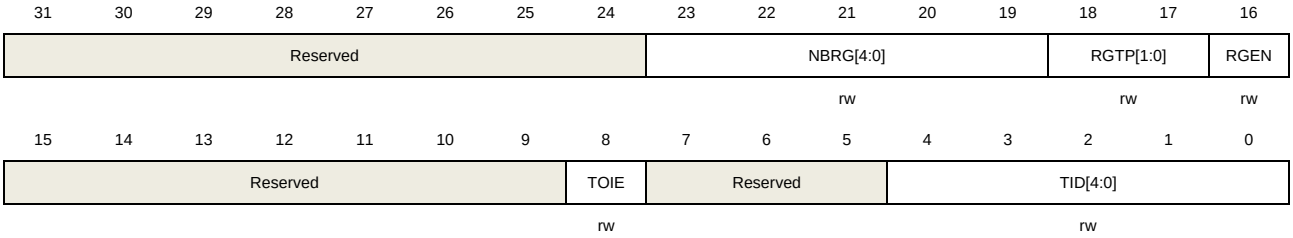
9.5.4. Request generator channel x configuration register (DMAMUX_RG_CHxCFG)

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

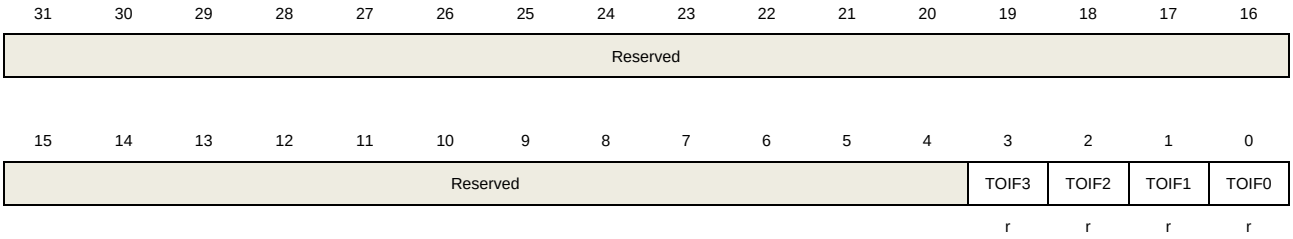
		1: Enable interrupt
7:5	Reserved	Must be kept at reset value.
4:0	TID[4:0]	Trigger input identification Selects the DMA request trigger input source.

9.5.5. 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).



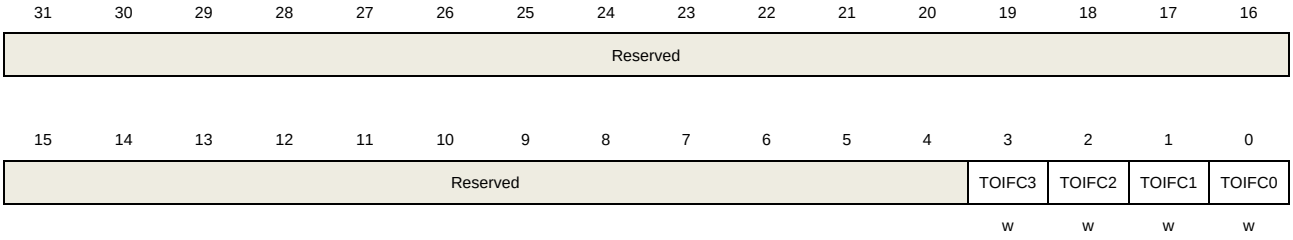
Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value.
3:0	TOIFx	Trigger overrun event flag of request generator channel x (x=0..3) The flag is set when a new trigger event occurs on DMA request generator channel x, before the request counter underrun (the internal request counter programmed 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.

9.5.6. 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).



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.

10. Cyclic redundancy checks management unit (CRC)

10.1. Overview

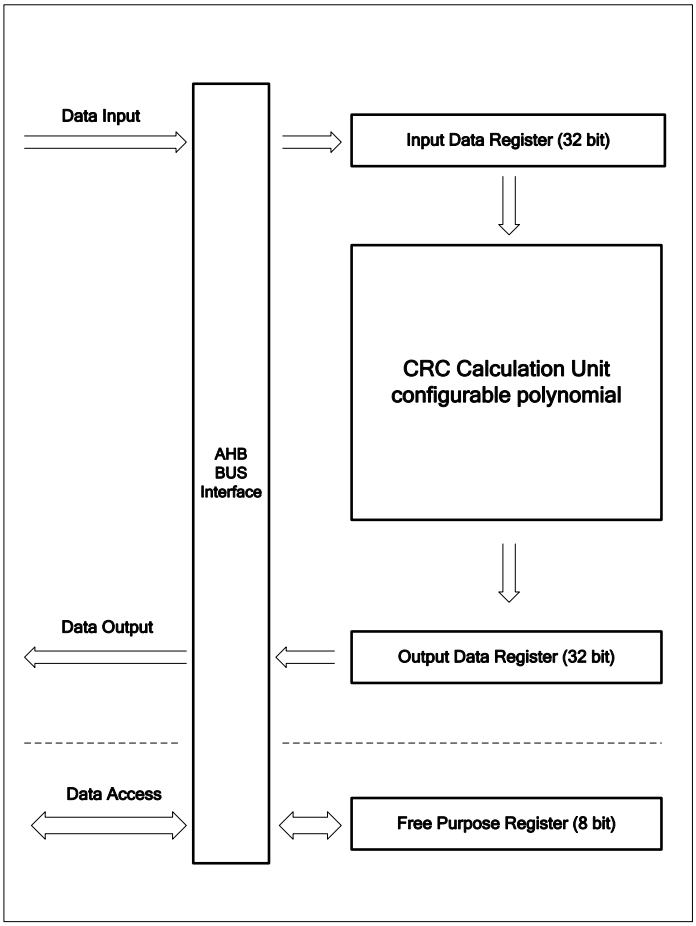
A cyclic redundancy check (CRC) is an error-detecting code commonly used in digital networks and storage devices to detect accidental changes to raw data.

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

10.2. Characteristics

- Supports 7 / 8 / 16 / 32 bit data input
- For 7(8) / 16 / 32 bit input data length, the calculation cycles are 1 / 2 / 4 AHB clock cycles
- User configurable polynomial value and size
- After CRC module·reset, user can configure initial value
- Free 8-bit register is unrelated to calculation and can be used for any other goals by any other peripheral devices

Figure 10-1. Block diagram of CRC calculation unit



10.3. Function overview

- CRC calculation unit is used to calculate the 32-bit raw data, and CRC_DATA register will receive the raw data and store the calculation result.

  If the CRC_DATA register has not been cleared by setting the CRC_CTL register, the new input raw data will be calculated based on the result of previous value of CRC_DATA.

  CRC calculation will spend 4 / 2 / 1 AHB clock cycles for 32 / 16 / 8(7) bit data size. During this period, AHB will not be hanged because of the existence of the 32bit input buffer.

- This module supplies an 8-bit free register CRC_FDATA.

  CRC_FDATA is unrelated to the CRC calculation. Independent read and write operations can be performed at any time.

- Reversible function can reverse the input data and output data.

  For input data, 3 reverse types can be selected.

Original data is 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. If the polynomial is less than 32 bit, the high bits of the input data and output data is unavailable. It is strongly recommend resetting the CRC calculation unit after change the PS[1:0] bits or polynomial.

10.4. Register definition

CRC base address: 0x4002 3000

10.4.1. Data register (CRC_DATA)

Address offset: 0x00
Reset value: 0xFFFF FFFF

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

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

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

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

10.4.2. Free data register (CRC_FDATA)

Address offset: 0x04
Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	FDATA[7:0]	Free data register bits Software writes and reads. These bits are unrelated with CRC calculation. This byte can be used for any goal

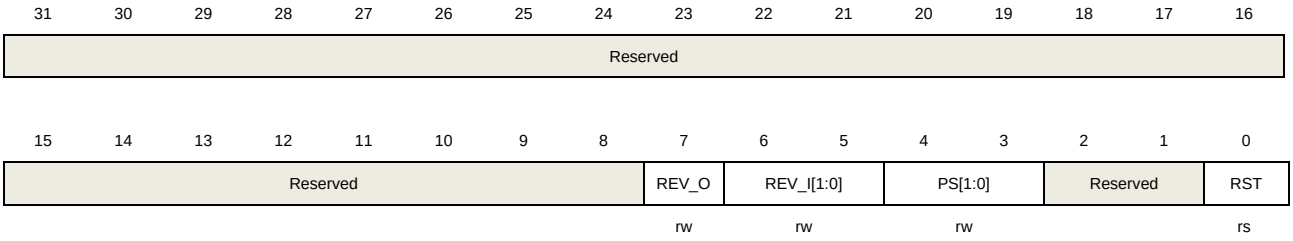
by any other peripheral. The CRC_CTL register will generate no effect to the byte.

10.4.3. Control register (CRC_CTL)

Address offset: 0x08

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7	REV_O	Reverse output data value in bit order 0:Not bit reversed for output data 1:Bit reversed for output data
6:5	REV_I[1:0]	Reverse type for input data 0: Dot not use reverse for input data 1: Reverse input data with every 8-bit length 2: Reverse input data with every 16-bit length 3: Reverse input data with whole 32-bit length
4:3	PS[1:0]	Size of polynomial 0: 32 bit 1: 16 bit (POLY [15:0] is used for calculation.) 2: 8 bit (POLY [7:0] is used for calculation.) 3: 7 bit (POLY [6:0] is used for calculation.)
2:1	Reserved	Must be kept at reset value.
0	RST	Software writes and reads. Set this bit can reset the CRC_DATA register. When set, the value of the CRC_DATA register is automatically initialized to the value in the CRC_IDATA register and then automatically cleared by hardware. This bit will take no effect to CRC_FDATA.

10.4.4. Initialization data register (CRC_IDATA)

Address offset: 0x10

Reset value: 0xFFFF FFFF

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

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.

10.4.5. Polynomial register (CRC_POLY)

Address offset: 0x14

Reset value: 0x04C1 1DB7

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

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

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

Bits	Fields	Descriptions
31:0	POLY[31:0]	User configurable polynomial value This value is used together with PS[1:0] bits.

11. Configurable Logic Array (CLA)

11.1. Overview

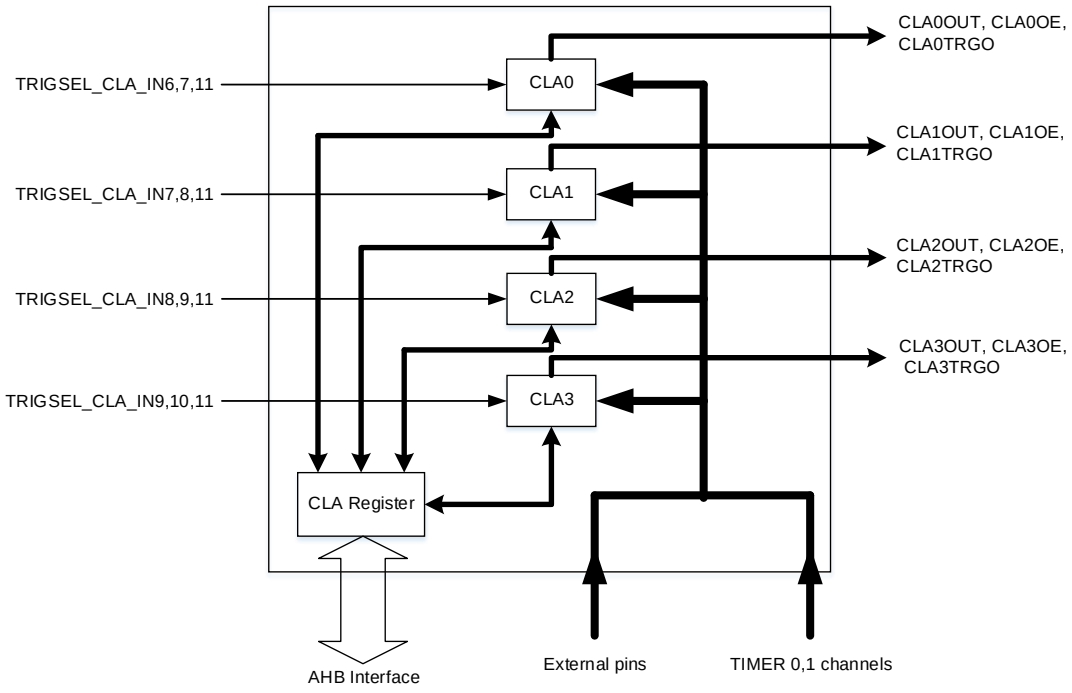
The configurable logic array provides 256 programmable digital logic operations for external pins, CMP, ADC and timers without intervention from the CPU. Four independent CLA units are implemented in this module. Each CLA unit supports configurable asynchronous and synchronous output for GPIO pins.

11.2. Characteristics

- Four independent CLA units, with two input Signal Selector(SIGS), supporting 16 input signals, including external pins, timer channels, CMP, ADC, and CLA asynchronous outputs.
- A Logic-Configure-Unit(LCU) providing 256 programmable digital logic functions is implemented in each CLA units.
- Programmable asynchronous and synchronous output
- CLA output can be configured to synchronize with external pins and timers.
- Four CLA units can be combined and support complicated logic operations.

11.3. Block diagram

Figure 11-1. Block diagram of CLA



11.4. Function overview

Four identical CLA units are inplemented in this module. Two SIGS are implemented in each CLA units. In addition, a LCU is included in each CLA units.

11.4.1. CLA input Signal Selector

Each CLA unit includes two Signal Selector: SIGS0 and SIGS1. The input of each SIGS include several external pins, timer channels(TIMERx_CHx), timer trigger output (TIMERx_TRGO), ADC conversion signal (ADCx_CONV) and CLA units asynchronuous output (CLAx_ASYNC_OUT). When another CLA units asynchoronous output is selected as SIGS input, a complicated combinatorial logic operation is realized. When the TIMERx_TRGO is selected as SIGS input, only the first HCLK cycle is valid, and the rest of the high TIMERx_TRGO is regarded as logic LOW.

Table 11-1. CLAxSIGS0 input selection and *Table 11-2. CLAxSIGS1 input selection* show the CLAxSIGS0 and CLAxSIGS1 input selection. TRIGSEL_CLA_IN0~TRIGSEL_CLA_IN11 signals come from the TRIGSEL module show in *Figure 11-1. Block diagram of CLA*.

Table 11-1. CLAxSIGS0 input selection

SIGS0[3:0]	CLA0SIGS0	CLA1SIGS0	CLA2SIGS0	CLA3SIGS0
0000	CLA0_ASYNC_OUT	CLA0_ASYNC_OUT	CLA0_ASYNC_OUT	CLA0_ASYNC_OUT
0001	CLA1_ASYNC_OUT	CLA1_ASYNC_OUT	CLA1_ASYNC_OUT	CLA1_ASYNC_OUT
0010	CLA2_ASYNC_OUT	CLA2_ASYNC_OUT	CLA2_ASYNC_OUT	CLA2_ASYNC_OUT
0011	CLA3_ASYNC_OUT	CLA3_ASYNC_OUT	CLA3_ASYNC_OUT	CLA3_ASYNC_OUT
0100	TRIGSEL_CLA_IN7	TRIGSEL_CLA_IN8	TRIGSEL_CLA_IN9	TRIGSEL_CLA_IN10
0101	TRIGSEL_CLA_IN0	TRIGSEL_CLA_IN0	TRIGSEL_CLA_IN1	TRIGSEL_CLA_IN2
0110	TRIGSEL_CLA_IN1	TRIGSEL_CLA_IN3	TRIGSEL_CLA_IN3	TRIGSEL_CLA_IN4
0111	TRIGSEL_CLA_IN2	TRIGSEL_CLA_IN4	TRIGSEL_CLA_IN5	TRIGSEL_CLA_IN5
1000	CLAIN0	CLAIN4	CLAIN0	CLAIN2
1001	CLAIN2	CLAIN5	CLAIN1	CLAIN3
1010	CLAIN4	CLAIN8	CLAIN8	CLAIN6
1011	CLAIN6	CLAIN10	CLAIN9	CLAIN7
1100	CLAIN8	CLAIN12	CLAIN14	CLAIN10
1101	CLAIN10	CLAIN13	CLAIN15	CLAIN11
1110	CLAIN12	CLAIN16	CLAIN16	CLAIN18
1111	CLAIN14	CLAIN18	CLAIN17	CLAIN19

Table 11-2. CLAxSIGS1 input selection

SIGS1[3:0]	CLA0SIGS1	CLA1SIGS1	CLA2SIGS1	CLA3SIGS1
0000	CLA0_ASYNC_OUT	CLA0_ASYNC_OUT	CLA0_ASYNC_OUT	CLA0_ASYNC_OUT
0001	CLA1_ASYNC_OUT	CLA1_ASYNC_OUT	CLA1_ASYNC_OUT	CLA1_ASYNC_OUT
0010	CLA2_ASYNC_OUT	CLA2_ASYNC_OUT	CLA2_ASYNC_OUT	CLA2_ASYNC_OUT

0011	CLA3_ASYNC_OUT	CLA3_ASYNC_OUT	CLA3_ASYNC_OUT	CLA3_ASYNC_OUT
0100	TRIGSEL_CLA_IN11	TRIGSEL_CLA_IN11	TRIGSEL_CLA_IN11	TRIGSEL_CLA_IN11
0101	TRIGSEL_CLA_IN3	TRIGSEL_CLA_IN1	TRIGSEL_CLA_IN0	TRIGSEL_CLA_IN0
0110	TRIGSEL_CLA_IN4	TRIGSEL_CLA_IN2	TRIGSEL_CLA_IN2	TRIGSEL_CLA_IN1
0111	TRIGSEL_CLA_IN5	TRIGSEL_CLA_IN5	TRIGSEL_CLA_IN4	TRIGSEL_CLA_IN3
1000	CLAIN1	CLAIN6	CLAIN2	CLAIN0
1001	CLAIN3	CLAIN7	CLAIN3	CLAIN1
1010	CLAIN5	CLAIN9	CLAIN10	CLAIN4
1011	CLAIN7	CLAIN11	CLAIN11	CLAIN5
1100	CLAIN9	CLAIN14	CLAIN12	CLAIN8
1101	CLAIN11	CLAIN15	CLAIN13	CLAIN9
1110	CLAIN13	CLAIN17	CLAIN18	CLAIN16
1111	CLAIN15	CLAIN19	CLAIN19	CLAIN17

11.4.2. LCU control

A LCU is implemented in each CLA unit, A LCU is implemented in each CLA unit which introduce 256 combinatorial logic functions controlled by LCU [7:0] bits in the CLAx_LCUCTL register. The LCU has three inputs, which have no effect on LCU output when the CLA unit is disabled.

- Input0: SIGS0 output
- Input1: SIGS1 output
- Input2: CLA result from CLA[x-1]

If the CLA unit is disabled, the LCU inputs are force to '0'.

The 8-bit LCU controls which logic function of the input 0(IN0), input 1(IN1) and input 2(IN2) can be effected on the output, as shown in ***Table 11-3. LCU control***.

Table 11-3. LCU control

LCU[7:0]	input 0	input 1	input 2
bit 0	0	0	0
bit 1	0	0	1
bit 2	0	1	0
bit 3	0	1	1
bit 4	1	0	0
bit 5	1	0	1
bit 6	1	1	0
bit 7	1	1	1

For example, if the logic function (IN0^IN1^IN2) is required, the LCU [7:0] in CLAx_LCUCTL shoud be configured like this: when {IN0, IN1 IN2} == 3'b000, the result of (IN0^IN1^IN2) is 1'b0, so bit 0 of LCU [7:0] = 0;
when {IN0, IN1 IN2} == 3'b001, the result of (IN0^IN1^IN2) is 1'b1, so bit 1 of LCU [7:0] = 1;

when {IN0, IN1 IN2} == 3'b010, the result of (IN0^IN1^IN2) is 1'b1, so bit 2 of LCU [7:0] = 1;
when {IN0, IN1 IN2} == 3'b011, the result of (IN0^IN1^IN2) is 1'b0, so bit 3 of LCU [7:0] = 0;
when {IN0, IN1 IN2} == 3'b100, the result of (IN0^IN1^IN2) is 1'b1, so bit 4 of LCU [7:0] = 1;
when {IN0, IN1 IN2} == 3'b101, the result of (IN0^IN1^IN2) is 1'b0, so bit 5 of LCU [7:0] = 0;
when {IN0, IN1 IN2} == 3'b110, the result of (IN0^IN1^IN2) is 1'b0, so bit 6 of LCU [7:0] = 0;
when {IN0, IN1 IN2} == 3'b111, the result of (IN0^IN1^IN2) is 1'b1, so bit 7 of LCU [7:0] = 1.
Thus, If you want to implement a logical function (IN0^IN1^IN2), LCU [7:0] = 8'b10010110.

11.4.3. CLA Output

Each CLA unit has an output to the selected GPIO pins, which can be the LCU result or the LCU result after a flip-flop, selected by the OSEL bit in the CLAx_CTL register.

The clock source and the clock polarity of the flip-flop can be selected by the CSEL[1:0] bits and CPOL bit of the CLAx_CTL register, respectively. There are four clocks which can be used as the clock source of flip-flop, they are:

- CLA[x-1] result: the LCU result of the previous CLA units (the CLA3 LCU result is sent to the CLA0).
- SIGS0 output: the output of SIGS0
- HCLK
- TIMER_TRGO: TRIGSEL_CLA_IN6 for CLA0, TRIGSEL_CLA_IN7 for CLA1, TRIGSEL_CLA_IN8 for CLA2, TRIGSEL_CLA_IN9 for CLA3.

Writing the FFRST bit in the CLAX_CTL register to "1" can reset the flip-flop output. If the flip-flop output is selected as the CLA output, it is recommended to set the FFRST bit before enable the CLA unit.

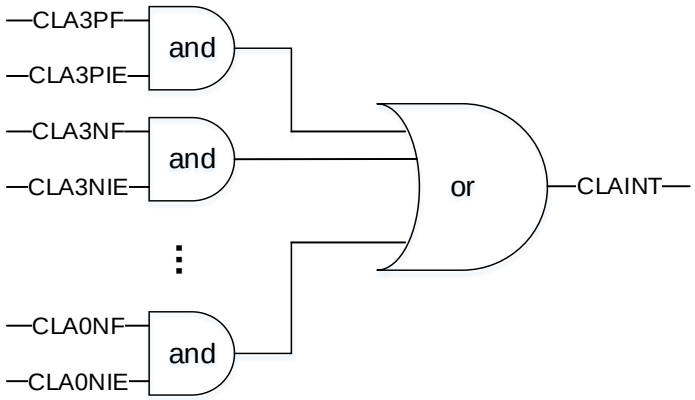
If the OEN bit is reset, the CLAxOE is reset, and the corresponding GPIO pins output function is disable.

Each CLA unit has an synchronizer, the HCLK is output to CLAx_TRGO after the synchronizer. which is sent to trigsel, and the trigsel allows software to select output to ADC,DAC and SYSCFG, as a trigger event.

11.4.4. Interrupt

Eight interrupt flags are implemented in the CLA_INTF register, including CLA3PF, CLA3NF, CLA2PF, CLA2NF, CLA1PF, CLA1NF, CLA0PF and CLA0NF. Each state flag has a dedicated enable bit in the CLA_GCTL register. The CLA interrupt logic is shown in ***Figure 11-2. CLA interrupt logic***, an interrupt can be produced when any flags is detected and the corresponding enable bit is enabled.

Figure 11-2. CLA interrupt logic



11.5. Register definition

CLA base address: 0x4003 8000

11.5.1. Global control register (CLA_GCTL)

Address offset: 0x00

Reset value: 0x0000 0000(must be power reset)

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												CLA3EN	CLA2EN	CLA1EN	CLA0EN
												rw	rw	rw	rw

Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value
3	CLA3EN	CLA3 unit enable Software set and cleared 0: CLA3 unit is disabled 1: CLA3 unit is enabled
2	CLA2EN	CLA2 unit enable Software set and cleared 0: CLA2 unit is disabled 1: CLA2 unit is enabled
1	CLA1EN	CLA1 unit enable Software set and cleared 0: CLA1 unit is disabled 1: CLA1 unit is enabled
0	CLA0EN	CLA0 unit enable Software set and cleared 0: CLA0 unit is disabled 1: CLA0 unit is enabled

11.5.2. Interrupt enable register (CLA_INTE)

Address offset: 0x04

Reset value: 0x0000 0000(must be power reset)

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								CLA3PIE	CLA3NIE	CLA2PIE	CLA2NIE	CLA1PIE	CLA1NIE	CLA0PIE	CLA0NIE
								rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value
7	CLA3PIE	CLA3 unit posedge interrupt enable Software set and cleared 0: CLA3 unit posedge interrupt is disabled 1: CLA3 unit posedge interrupt is enabled. An interrupt is generated when the CLA3PF bit is set.
6	CLA3NIE	CLA3 unit negedge interrupt enable Software set and cleared 0: CLA3 unit negedge interrupt is disabled 1: CLA3 unit negedge interrupt is enabled. An interrupt is generated when the CLA3NF bit is set.
5	CLA2PIE	CLA2 unit posedge interrupt enable Software set and cleared 0: CLA2 unit posedge interrupt is disabled 1: CLA2 unit posedge interrupt is enabled. An interrupt is generated when the CLA2PF bit is set.
4	CLA2NIE	CLA2 unit negedge interrupt enable Software set and cleared 0: CLA2 unit negedge interrupt is disabled 1: CLA2 unit negedge interrupt is enabled. An interrupt is generated when the CLA2NF bit is set.
3	CLA1PIE	CLA1 unit posedge interrupt enable Software set and cleared 0: CLA1 unit posedge interrupt is disabled 1: CLA1 unit posedge interrupt is enabled. An interrupt is generated when the CLA1PF bit is set.
2	CLA1NIE	CLA1 unit negedge interrupt enable Software set and cleared 0: CLA1 unit negedge interrupt is disabled 1: CLA1 unit negedge interrupt is enabled. An interrupt is generated when the CLA1NF bit is set.
1	CLA0PIE	CLA0 unit posedge interrupt enable

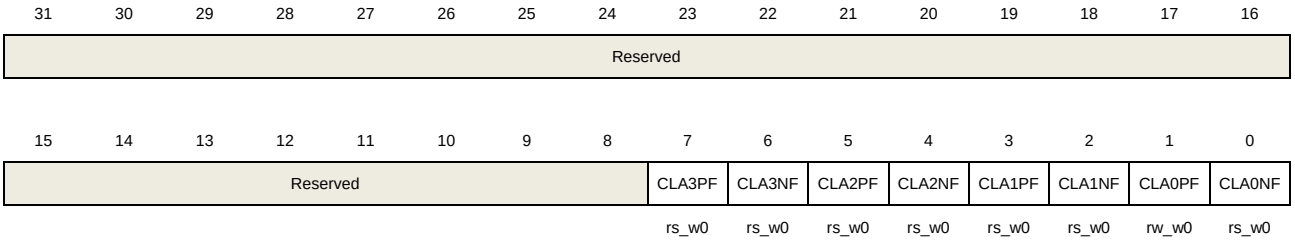
		Software set and cleared 0: CLA0 unit posedge interrupt is disabled 1: CLA0 unit posedge interrupt is enabled. An interrupt is generated when the CLA0PF bit is set.
0	CLA0NIE	CLA0 unit negedge interrupt enable Software set and cleared 0: CLA0 unit negedge interrupt is disabled 1: CLA0 unit negedge interrupt is enabled. An interrupt is generated when the CLA0NF bit is set.

11.5.3. Interrupt flag register (CLA_INTF)

Address offset: 0x08

Reset value: 0x0000 0000(must be power reset)

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



Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value
7	CLA3PF	CLA3 unit posedge flag Hareware set and software cleared 0: A CLA3 output posedge has not been detected 1: A CLA3 output posedge has been detected
6	CLA3NF	CLA3 unit negedge flag Hareware set and software cleared 0: A CLA3 output negedge has not been detected 1: A CLA3 output negedge has been detected
5	CLA2PF	CLA2 unit posedge flag Hareware set and software cleared 0: A CLA2 output posedge has not been detected 1: A CLA2 output posedge has been detected
4	CLA2NF	CLA2 unit negedge flag Hareware set and software cleared 0: A CLA2 output negedge has not been detected 1: A CLA2 output negedge has been detected

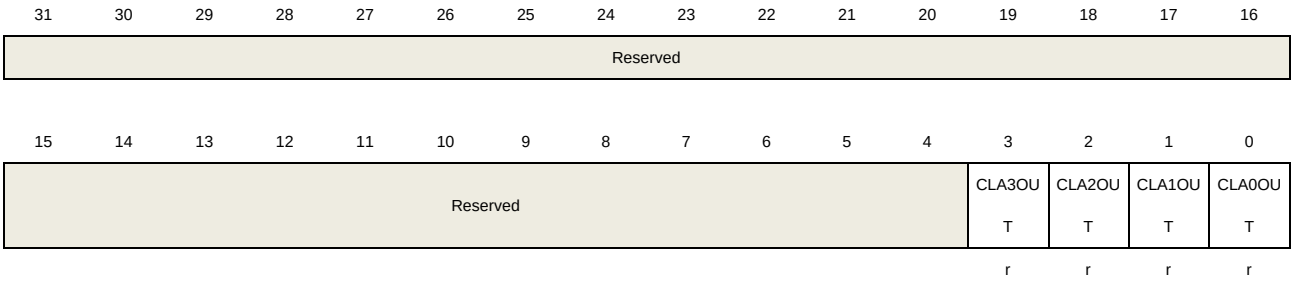
3	CLA1PF	CLA1 unit posedge flag Hareware set and software cleared 0: A CLA1 output posedge has not been detected 1: A CLA1 output posedge has been detected
2	CLA1NF	CLA1 unit negedge flag Hareware set and software cleared 0: A CLA1 output negedge has not been detected 1: A CLA1 output negedge has been detected
1	CLA0PF	CLA0 unit posedge flag Hareware set and software cleared 0: A CLA0 output posedge has not been detected 1: A CLA0 output posedge has been detected
0	CLA0NF	CLA0 unit negedge flag Hareware set and software cleared 0: A CLA0 output negedge has not been detected 1: A CLA0 output negedge has been detected

11.5.4. Status register (CLA_STAT)

Address offset: 0x0C

Reset value: 0x0000 0000(must be power reset)

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



Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value
3	CLA3OUT	CLA3 unit output state Hardware set and cleared 0: The current logic level of CLA3 unit is LOW 1: The current logic level of CLA3 unit is HIGH
2	CLA2OUT	CLA2 unit output state Hardware set and cleared 0: The current logic level of CLA2 unit is LOW 1: The current logic level of CLA2 unit is HIGH

1	CLA1OUT	CLA1 unit output state Hardware set and cleared 0: The current logic level of CLA1 unit is LOW 1: The current logic level of CLA1 unit is HIGH
0	CLA0OUT	CLA0 unit output state Hardware set and cleared 0: The current logic level of CLA0 unit is LOW 1: The current logic level of CLA0 unit is HIGH

11.5.5. Signal selection register (CLAx_SIGS) (x = 0...3)

Address offset: 0x10 + 0x0C * x

Reset value: 0x0000 0000(must be power reset)

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								SIGS0[3:0]				SIGS1[3:0]			
								rw				rw			

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value
7:4	SIGS0[3:0]	Signal Selector 0 input selection Software set and cleared These bits select the input for Signal Selector 0. Please reference ***Table 11-1. CLAxSIGS0 input selection***
3:0	SIGS1[3:0]	Signal Selector 1 input selection Software set and cleared These bits select the input for Signal Selector 1. Please reference ***Table 11-2. CLAxSIGS1 input selection***

11.5.6. LCU control register (CLAx_LCUCTL) (x = 0…3)

Address offset: 0x14 + 0x0C * x

Reset value: 0x0000 0000(must be power reset)

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

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

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

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value
7:0	LCU[7:0]	LCU control Software set and cleared These bits controll which logic function of the input0, input1 and input2 can be effected on the output Examples: IN1 | IN2: LCU = 8'b11101110 IN0 & (IN1 ^ IN2): LCU = 8'b01100000

11.5.7. Control register (CLAx_CTL) (x=0..3)

Address offset: 0x18 + 0x0C * x

Reset value: 0x0000 0000(must be power reset)

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								OSEL	OEN	Reserved		FFRST	CPOL	CSEL[1:0]	
								rw	rw			w	rw	rw	

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7	OSEL	Output selection Software set and cleared 0: Flip-flop output is selected as CLAx output 1: LCU result is selected as CLAx output
6	OEN	Output enable Software set and cleared 0: CLAx output is disabled 1: CLAx output is enabed
5:4	Reserved	Must be kept at reset value
3	FFRST	Flip-flop output reset 0: no effect

		1: Reset the flip-flop output asynchronously
2	CPOL	Clock polarity of Flip-flop Software set and cleared This bitfield must not be modified once the CLAx is enabled(CLAxEN = 1) 0: clock posedge is valid 1: clock negedge is valid
1:0	CSEL[1:0]	Flip-flop clock source selection Software set and cleared This bitfield must not be modified once the CLAx is enabled(CLAxEN = 1) 00: CLA[x-1] LCU result 01: SIGS0 output 10: HCLK 11: TIMER_TRGO

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 and it has been pre-certified NIST SP800-90B.

12.2. Characteristics

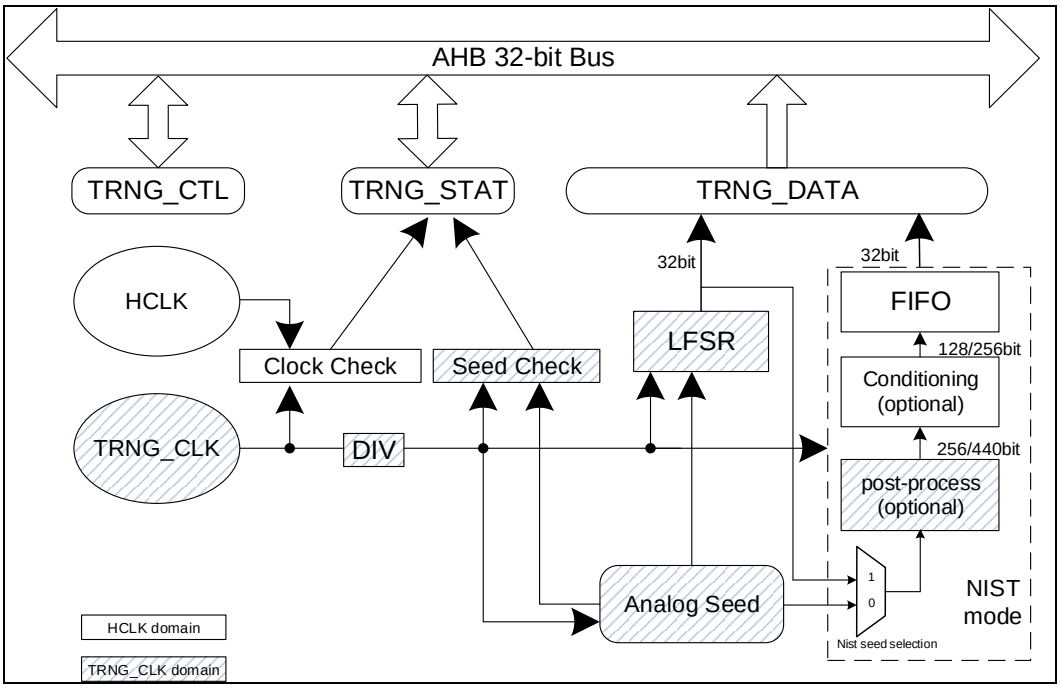
- LFSR mode and NIST mode to generate random number.
- About 40 periods of TRNG_CLK are needed between two consecutive random numbers in LFSR mode.
- 32bit random number is generated each time in LFSR mode.
- TRNG NIST mode follow the NIST SP800-90B.
- 32bit*4 or 32bit*8 random number is generated each time in NIST mode.
  - TRNG has the functions of startup and in-service self-check, associated with specific error flags.
- Disable TRNG module will significantly reduce the chip power consumption.
- 128-bit random value seed is generated from analog noise, so the random number is a true random number.

12.3. Limitation

The frequency of the HCLK should be more than twice as the frequency of diveded TRNG_CLK for internal clock synchron.

12.4. Function overview

Figure 12-1. TRNG block diagram



There are two modes in TRNG module, NIST mode and LFSR mode.

In the NIST mode, the analog seed of random number comes from 4 noise sources. The noise source signal is first operated by XOR and then digitized to obtain a 1-bit analog seed. This analog seed is then plugged into a conditioning logic(uint) to generate 128 or 256-bit data output. A HASH function compled with NIST.SP.800-90B is implemented in the conditioning stage to increase the entropy of random number. And the output data can be read out by reading the data register TRNG_DATA 4 times continuously after the data ready flag (DRDY) in TRNG_STAT register is set.

In the LFSR mode for the compatibility, the analog random number seed is plugged into a linear feedback shift register (LFSR), where a 32-bit width random number is generated. The 32-bit value of LFSR will transfer into TRNG_DATA register after a sufficient number of seeds have been sent to the LFSR.

The analog seed is generated by several ring oscillators. The seed generate model is driven by a configurable TRNG_CLK (refer to

chapter), so that the quality of the generated random number depends on TRNG_CLK exclusively, no matter what HCLK frequency was set or not.

In the NIST mode, the seed source plugged into the conditioning logic(unit) can be selected by setting the SEEDSEL bit in TRNG_CTL register. In addition to the analog seeds, the value of LFSR can also be used as seed source in the nist mode.

12.4.1. LFSR

A Linear Feedback Shift Register is a sequential shift register with combinational logic that causes it to pseudo-randomly cycle through a sequence of binary values. This operation will increase the entropy of the random number, and TRNG generates 32-bit data each time in this mode.

12.4.2. Post processing

When this function is enabled, half of the bits are taken from the sampled noise source, half of the bits are taken from inverted sampled noise source. And when the output data is ready, the post-process module will start a new random seed collection even if the random data haven't been read out. This operation will increase efficiency of TRNG module.

12.4.3. Conditioning

The conditioning component in the TRNG is a deterministic HASH function that increases the entropy rate of the resulting fixed-length bitstrings output. The NIST SP800-90B target is full entropy on the output.

The hash operation is controlled by ALGO[1:0] bits field and INIT bit in TRNG_CTL register. And the input random number bit width is controlled by register INMOD (256-bit or 440-bit), while the output bit width is controlled by OUTMOD (128-bit or 256-bit).

12.4.4. output FIFO

The width of data output FIFO is 4 or 8 (controlled by OUTMOD) words(32-bit) in NIST mode, and these data are stored in the output FIFO temporarily. When all words have been read from the output FIFO through the TRNG_DATA register, a new round of conditioning process is automatically started, then another 128-bit or 256-bit conditioning output data is pushed into the output FIFO and waiting for the next reading.

The DRDY bit in TRNG_STAT register will be set when a random number is available through the TRNG_DATA register. This bit remains set until output FIFO becomes empty after reading 4 or 8 times continuously from the TRNG_DATA register in NIST mode. This flag also remains setting until the one TRNG_DATA is read out when in LFSR mode.

In NIST mode, it will cost about input seed number(controlled by INMOD bit in the TRNG_CTL register) plus 10 TRNG_CLK periods and 70 HCLK periods. While in LFSR mode, generate a 32bit random number need about 40 TRNG_CLK periods.

The hash function with different ALGO setting in the TRNG_CTL register will produce results of different valid length. TRNG module will place the valid bits at high order of output FIFO, so the first read data is always valid data. And user should select the expect data and register setting according to the need. More details refer to ***Table 12-1. ALGO configurations***.

Table 12-1. ALGO configurations

ALGO	00	01	10	11
algorithm	SHA1	MD5	SHA224	SHA256
valid length	160	128	224	256

12.4.5. Health check

This component ensures the stable operation of TRNG and can quickly monitor the occurrence of errors.

The health tests features of TRNG module following NIST SP800-90B. For more details about thresholds, refer to TRNG_HTCFG register.

1) Start-up health tests: These tests performed after reset and before using TRNG to get random numbers for the first time.

   - Adaptive proportion test: The TRNG verifies that the first bit on the outputs of the noise source is not repeated more than a threshold value, this threshold value refer to APTTH bit fields in the TRNG_HTCFG register (default 691 times).
   - Repetition count test: If the noise source has provided more than a specified number of consecutive bits at a constant value(0 or 1), an error will occur and the relevant error flag will be set. This threshold value refer to RCTTH bit fields in TRNG_HTCFG register(default 40).
   - Replace test: The TRNG can replace the input of condition stage with a specific number, and compare the output to the golden answer according to different algorithm during this stage.

2) Continuous health tests: These tests are run indefinitely on the outputs of the noise source while the noise source is operating.

   - Adaptive proportion test: refer to the start-up health tests.
   - Repetition count test: refer to the start-up health tests.

3) GD-Defined continuous health tests: Additional health tests specified by vendor.

   - Transition count test: if the noise source provided more than 32 consecutive occurrence of two bits patterns(01 or 10), an error will occur and the relevant error flag will be set.
   - Clock detector: if the TRNG clock cycle before divided is smaller than AHB clock cycle divided by 16, an error will occur and the relevant error flag will be set.

4) On-demand test of the noise source output

   - The noise source output only support on-demand testing by restarting the entropy source and rerunning the startup tests.

Note:

- In NIST mode, ERRSTA bit in the TRNG_STAT register will be set when an error is

detected, and if the interrupt bit of TRNG is asserted, an interrupt is generated.

- When the replace test is enabled, the random numbers generated are only used to verify the functionality of the conditioning component. After the test is completed, the random numbers should be discarded and should not be used as true random numbers.

12.4.6. NIST mode FSM

The states of NIST mode are shown below:

1. The initial state of the TRNG is idle state.
2. It goes to warm-up state after enabling the TRNG by setting RNGEN bit in the TRNG_CTL register. This state is a period for analog initialition.
3. After counting 16 cycles of TRNG_CLK (before divider), the state goes to start-up state, and the start-up health tests are implemented, which will cost 1024 divided TRNG_CLK cycles.
4. Then the state changes to gen-samp (generate sample) state to generate random number sample. The TRNG module will generate a new set of random number after the output FIFO is empty.

If an error occurs when the TRNG is in start-up state or gen-samp state, the ERRSTA bit in the TRNG_STAT register will be set.

Note:

- Don’t change the CLKDIV[3:0] bits field in TRNG_CTL register when in operation.
- When the TRNG is in start-up state, a random number is also generated for the first sample generate time reduce consideration, even if this sample is suggested to discard according to the FIPS PUB140-2.
- If RTEN bit in the TRNG_CTL register is set, the first output number is a replace test result, and it should be discarded.

12.4.7. Operation flow

The following steps are recommended for using TRNG block:

1. Set CONDRST bit in TRNG_CTL register.
2. Write the required configuration in the TRNG_CTL register, such as module power consumption, clock frequency division factor, operating mode, input / output bit-width, algorithm, etc.
3. Enable the TRNGEN bit.
4. Clear the CONDRST bit for the written configuration to take effect.
5. Check the status register TRGN_STAT, if SEIF, CEIF, ERRSTA, SECS and CECS are all stay 0 and DRDY=1, then the random value in the data register could be read.

When the IE bit in the TRNG_CTL register is set, an interrupt is generated if:

- Successfully generated a random number, and the DRDY bit in the TRNG_STAT register

is set.

- A seed error occurs, and the SEIF and ERRSTA bits in the TRNG_STAT register are set.
- A clock error occurs, and the CEIF and ERRSTA bits in the TRNG_STAT register are set.

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.

12.4.8. Error flags

(1) 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.

(2) Seed error

When the analog seed is not changed or always changing exceed the threshold, the SECS and SEIF bit will be set. In this case, the random data in data register should not be used. The application needs a TRNG software reset by writing CONDRST bit at 1 and then at 0, clear the SEIF bit after reset TRNG, then wait for SECS bit in the TRNG_STAT register to be cleared by TRNG.

12.4.9. Low power usage

If customers concern about power consumption, besides configure the CLKDIV a large number, they can also disable the TRNG module after setting the DRDY bit in TRNG_STAT register. TRNG module will remain at the last state, and the random data still can be read through the register TRNG_DATA. If a new random number is required, enable TRNG to activate the random number generation again.

12.5. Register definition

TRNG base address: 0x4802 1800

12.5.1. Control register (TRNG_CTL)

Address offset: 0x00

Reset value: 0x0300 0410

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CTLLK	CONDRST	Reserved				NR[1:0]		Reserved				CLKDIV[3:0]			
rs	rw	rw										rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INMOD	OUTMOD	ALGO[1:0]		Reserved	CONDEN	PPEN	INIT	RTEN	SEEDSEL	CED	MODSEL	IE	TRNGEN	Reserved	
rw	rw	rw			rw	rw	rw	rw	rw	rw	rw	rw	rw		

Bits	Fields	Descriptions
31	CTLLK	TRNG_CTL register lock bit This bit can only be reset to 0 if TRNG is reset. 0: Write bits[29:4] are allowed 1: Lock bits[29:4] and write bits[29:4] are ignored
30	CONDRST	Reset conditioning logic Write 1 then write 0 to reset conditioning logic. It should be noted that TRNG_HTCFG register and bits[29:4] in the TRNG_CTL register can only be written when CONDRST is 1.
29:26	Reserved	Must be kept at reset value.
25:24	NR[1:0]	Analog trng power mode. Reset value:2'b 11 00: Ultra low; 01: Low 10: Medium 11: High
23:20	Reserved	Must be kept at reset value.
19:16	CLKDIV[3:0]	TRNG clock divider 0000: 2^0 TRNG clock cycle per internal TRNG clock 0001: 2^1 TRNG clock cycles per internal TRNG clock …… 1111: 2^{15} TRNG clock cycles per internal TRNG clock
15	INMOD	Select random seed number input to conditioning module

		0: 256 bits 1: 440 bits
14	OUTMOD	Select random data width output of conditioning module 0: 128-bit 1: 256-bit
13:12	ALGO[1:0]	conditioning module hash algorithm selection 00: SHA1 01: MD5 10: SHA224 11: SHA256
11	Reserved	Must be kept at reset value.
10	CONDEN	The enable bit of conditioning component 0: Disable conditioning component 1: Enable conditioning component
9	PPEN	The enable bit of post processing function 0: Disable pos processing function 1: Enable post processing function
8	INIT	Initialize hash algorithm when conditioning enabled. 0: Deinitialize hash algorithm 1: Initialize hash algo algorithm
7	RTEN	Replace test enable bit 0: Disable replace test 1: Enable replace test
6	SEEDSEL	NIST seed mode selection 0: NIST seed come from analog 1: NIST seed come from LFSR output **Note**: This bit must be configured in NIST mode.
5	CED	Clock error detection 0: Disable clock error detection 1: Enable clock error detection
4	MODSEL	LFSR or NIST mode selection 0: LFSR mode 1: NIST mode
3	IE	The enable bit of the TRNG interrupt. This bit controls the generation of an interrupt when the DRDY, SEIF, CEIF or ERRSTA bit was set. 0: Disable TRNG interrupt 1: Enable TRNG interrupt

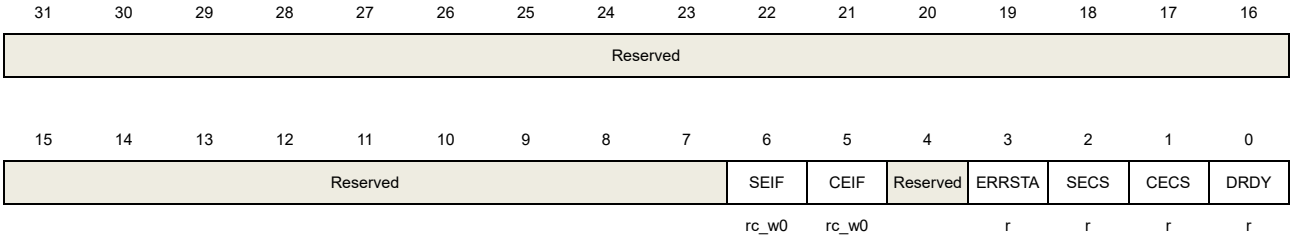
2	TRNGEN	The enabled bit of the TRNG. 0: Disable TRNG module (reduce power consuming) 1: Enable TRNG module
1:0	Reserved	Must be kept at reset value.

12.5.2. Status register (TRNG_STAT)

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	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	Reserved	Must be kept at reset value.
3	ERRSTA	NIST mode error flag, this bit could be reset by CONDRST 0: No error occurs in NIST mode 1: Error occurs in NIST mode
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

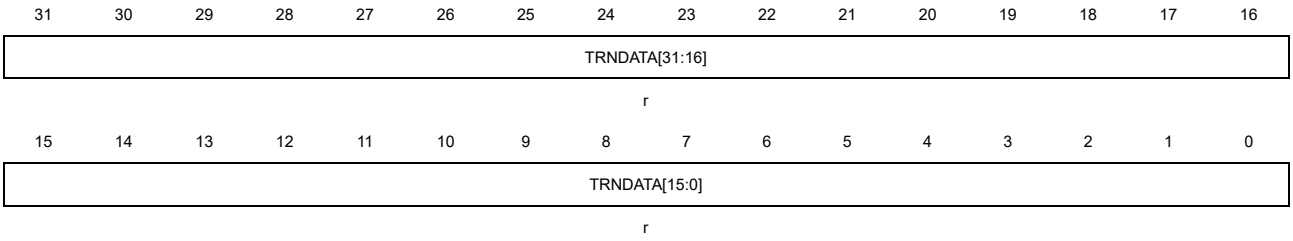
		means clock error has been detected before but now is recovered. 1: Clock error is detected at current time. TRNG_CLK frequency is lower than 1/16 HCLK frequency.
0	DRDY	Random data ready status bit. This bit is cleared by reading the TRNG_DATA register and set when a new random number is generated. 0: The content of TRNG data register is not available. 1: The content of TRNG data register is available.

12.5.3. Data register (TRNG_DATA)

Address offset: 0x08
Reset value: 0x0000 0000

Application must make sure DRDY bit in the TRNG_STAT register is set before reading this register.

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



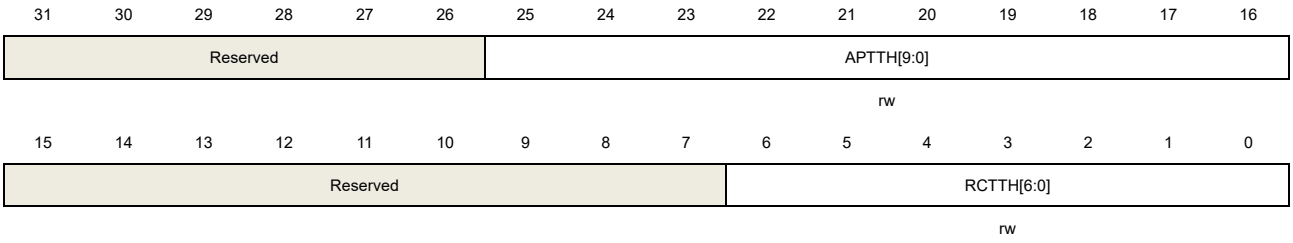
Bits	Fields	Descriptions
31:0	TRNDATA[31:0]	32-bit random data

12.5.4. Health tests configure register (TRNG_HTCFG)

Address offset: 0x10
Reset value: 0x02B3 0028

Set CONDRST bit and clear CTLLK bit in the TRNG_CTL before writing TRNG_HTCFG.

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



Bits	Fields	Descriptions
31:26	Reserved	Must be kept at reset value.

25:16	APTTH[9:0]	Adaptive proportion test threshold. Default 691.
15:7	Reserved	Must be kept at reset value.
6:0	RCTTH[6:0]	Repetition count test threshold. Default 40.

13. Cryptographic Acceleration Unit (CAU)

13.1. Overview

The cryptographic acceleration unit (CAU) is used to encipher and decipher data with DES, Triple-DES or AES (128, 192, or 256) algorithms. It is fully compliant implementation of the following standards:

- The Data Encryption Standard (DES) and the Triple Data Encryption Algorithm (TDEA) are announced by Federal Information Processing Standards Publication (FIPS) 46-3, October 25, 1999. It follows the American National Standards Institute (ANSI) X9.52 standard.
- The Advanced Encryption Standard (AES) is announced by Federal Information Processing Standards Publication 197, November 26, 2001.

DES/TDES/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

- DES, TDES and AES encryption/decryption algorithms are supported.
- Multiple modes are supported respectively in DES, TDES and AES, including Electronic codebook (ECB), Cipher block chaining (CBC), Counter mode (CTR), Galois/counter mode (GCM), Galois message authentication code mode (GMAC), Counter with CBC-MAC (CCM), Cipher Feedback mode (CFB) and Output Feedback mode(OFB).
- DMA transfer for incoming and outgoing data is supported.

DES/TDES

- Supports the ECB and CBC chaining algorithms.
- two 32-bit initialization vectors (IV) are used in CBC mode.
- 8*32-bit input and output FIFO.
- Multiple data types are supported, including No swapping, Half-word swapping Byte swapping and Bit swapping.
- Data are transferred by DMA, CPU during interrupts, or without both of them.

AES

- Supports the ECB, CBC, CTR, GCM, GMAC, CCM, CFB and OFB chaining algorithms.
- Supports 128-bit, 192-bit and 256-bit keys.
- four 32-bit initialization vectors (IV) are used in CBC, CTR, GCM, GMAC, CCM, CFB and OFB modes.

- 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 64/128 bits for DES/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. (For DES, the data block is two 32-bit words, please refer to the first two words data swapping in the figure).

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

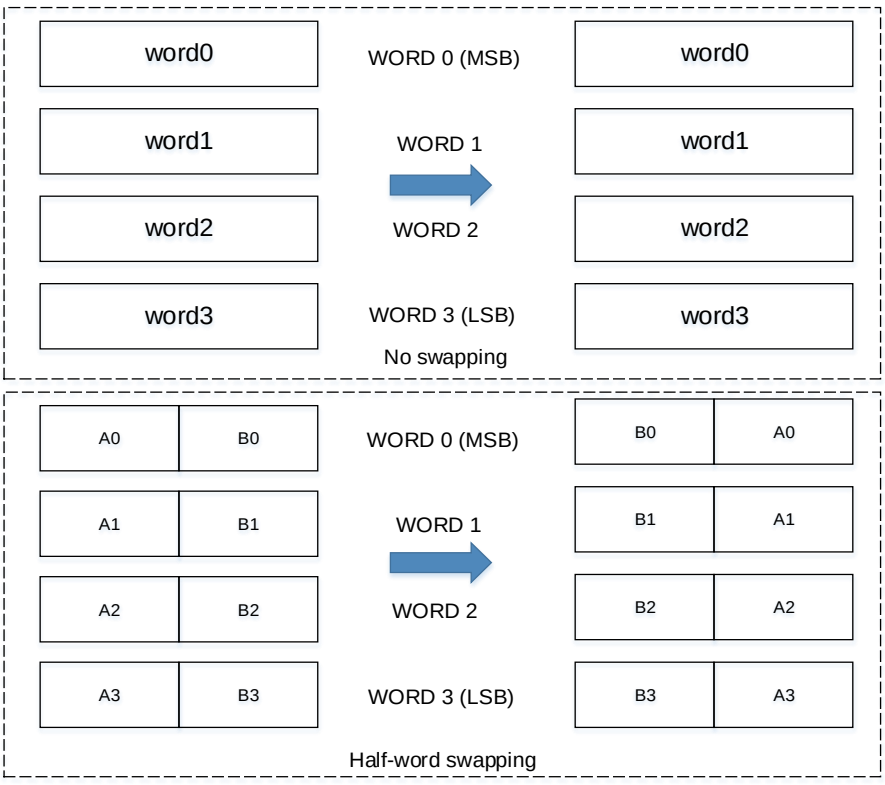
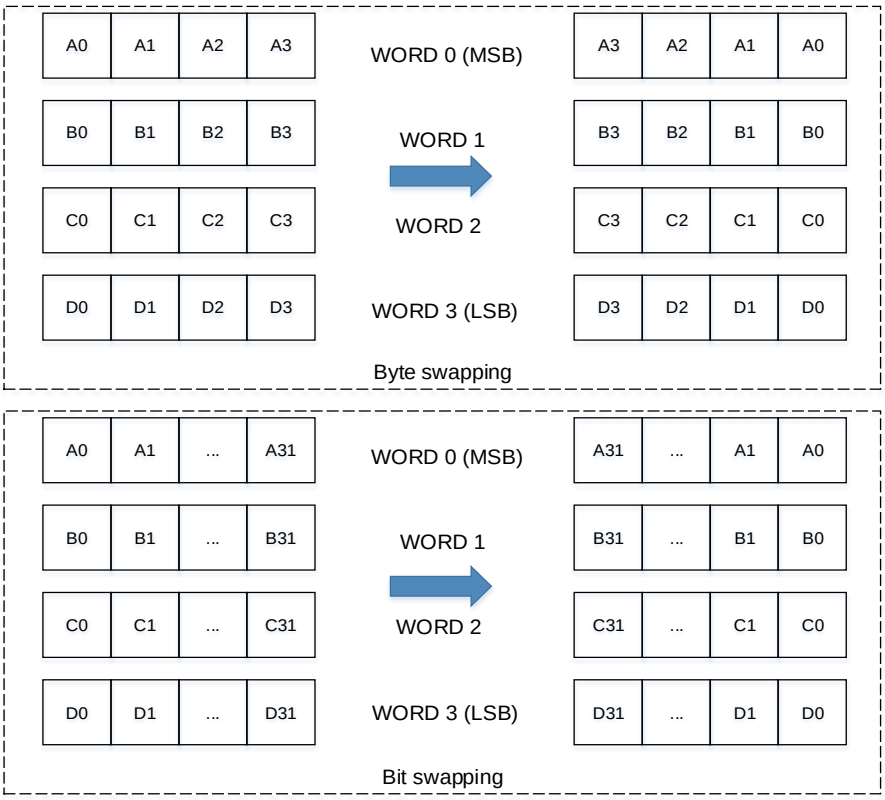


Figure 13-2. DATAM Byte swapping and Bit swapping



13.3.2. Initialization vectors

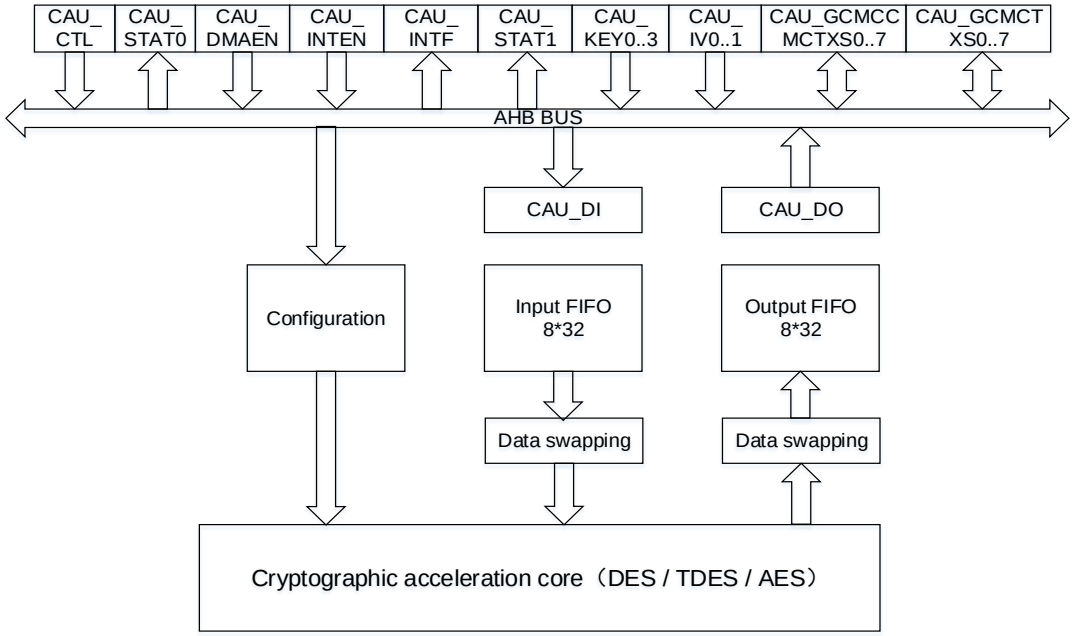
The initialization vectors are used in CBC, CTR, GCM, GMAC, CCM, CFB and OFB modes to XOR with data blocks. They are independent of plaintext and ciphertext, and the DATAM value will not affect them. Note the initialization vector registers CAU_IV0..1(H/L) can only be written when BUSY is 0, otherwise the write operations are invalid.

13.4. Cryptographic acceleration processor

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

The DES/TDES cryptographic acceleration processor contains the DES algorithm (DEA), cryptographic keys (1 for DES algorithm and 3 for TDES algorithm), and initialization vectors in CBC mode.

DES / TDES key

[KEY1] is used in DES and [KEY3 KEY2 KEY1] are used in TDES respectively.
When TDES algorithm is configured, three different keying options are allowed:

1. Three same keys

The three keys KEY3, KEY2 and KEY1 are completely equal, which means KEY3=KEY2=KEY1. FIPS PUB 46-3 – 1999 (and ANSI X9.52 -1998) refers to this option. It is easy to understand that this mode is equivalent to DES.

2. Two different keys

In this option, KEY2 is different from KEY1, and KEY3 is equal to KEY1, which means, KEY1 and KEY2 are independent while KEY3= KEY1. FIPS PUB 46-3 – 1999 (and ANSI X9.52 – 1998) refers to this option.

3. Three different keys

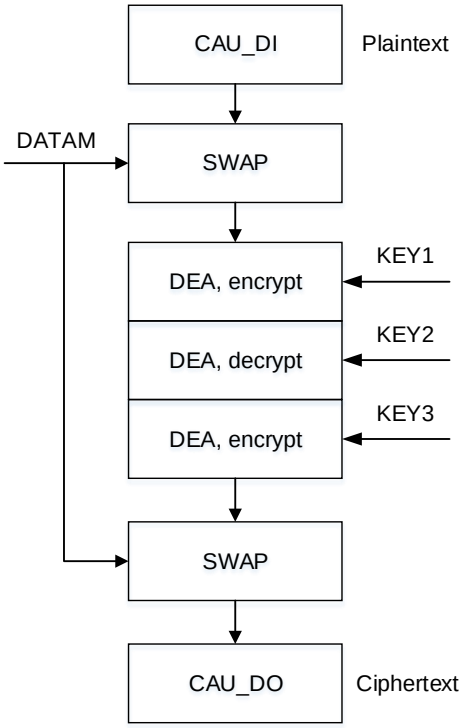
In this option, KEY1, KEY2 and KEY3 are completely independent. FIPS PUB 46-3 -1999 (and ANSI X9.52 – 1998) refers to this option.

More information of the thorough explanation of the key used in the DES/TDES please refer to FIPS PUB 46-3 (and ANSI X9.52 -1998), and the explanation process is omitted in this manual.

DES / TDES ECB encryption

The 64-bit input plaintext is first obtained after data swapping according to the data type. When the TDES algorithm is configured, the input data block is read in the DEA and encrypted using KEY1. The output is fed back directly to next DEA and then decrypted using KEY2. After that, the output is fed back directly to the last DEA and encrypted with KEY3. The output after above processes is then swapped back according to the data type again, and a 64-bit ciphertext is produced. When the DES algorithm is configured, the result of the first DEA encrypted using KEY1 is swapped directly according to the data type, and a 64-bit ciphertext is produced. The procedure of DES/TDES ECB mode encryption is illustrated in ***Figure 13-4. DES/TDES ECB encryption***.

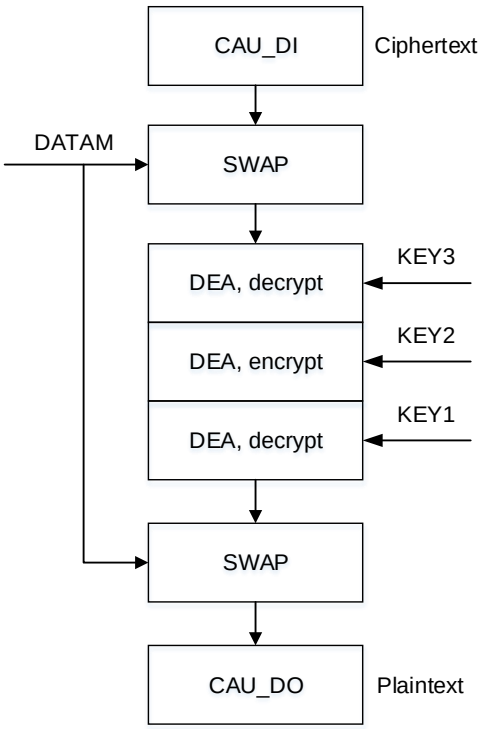
Figure 13-4. DES/TDES ECB encryption



DES / TDES ECB decryption

The 64-bit input ciphertext is first obtained after data swapping according to the data type. When the TDES algorithm is configured, the input data block is read in the DEA and decrypted using KEY3. The output is fed back directly to next DEA and then encrypted using KEY2. After that, the output is fed back directly to the last DEA and decrypted with KEY1. The output after above process is then swapped back according to the data type again, and a 64-bit plaintext is produced. When the DES algorithm is configured, the result of the first DEA decrypted using KEY1 is swapped directly according to the data type, and a 64-bit plaintext is produced. The procedure of DES/TDES ECB mode decryption is illustrated in ***Figure 13-5. DES/TDES ECB decryption***.

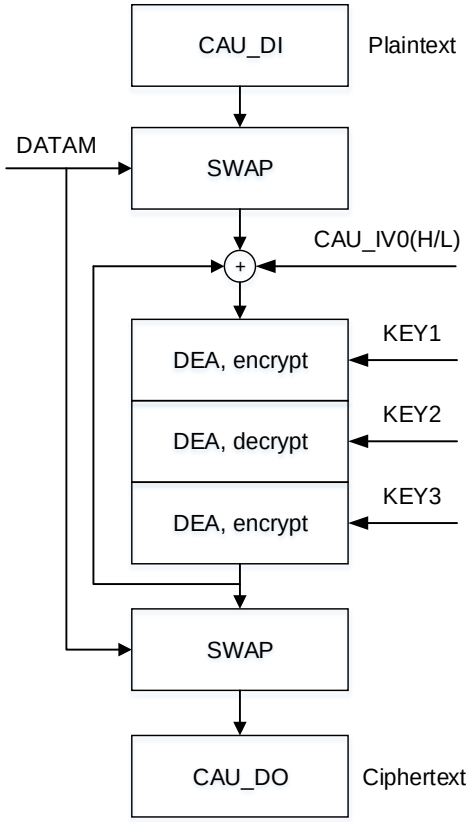
Figure 13-5. DES/TDES ECB decryption



DES/TDES CBC encryption

The input data of the DEA block in CBC mode consists of two aspects: the input plaintext after data swapping according to the data type, and the initialization vectors. When the TDES algorithm is configured, the XOR result of the swapped plaintext data block and the 64-bit initialization vector CAU_IV0..1 is read in the DEA and encrypted using KEY1. The output is fed back directly to next DEA and then decrypted using KEY2. After that, the output is fed back directly to the last DEA and encrypted with KEY3. The result is then used as the next initialization vector and exclusive-ORed with the next plaintext data block to process next encryption. The above operations are repeated until the last plaintext block is encrypted. Note if the plaintext message does not consist of an integral number of data blocks, the final partial data block should be encrypted in a specified manner. At last, the output ciphertext is also obtained after data swapping according to the data type. When the DES algorithm is configured, the state and process of the second and third block of DEA should be omitted. The procedure of DES/TDES CBC mode encryption is illustrated in ***Figure 13-6. DES/TDES CBC encryption***.

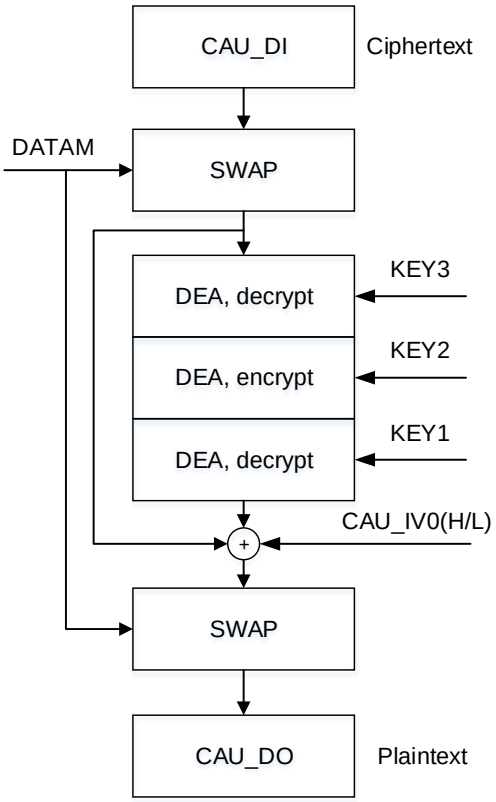
Figure 13-6. DES/TDES CBC encryption



DES / TDES CBC decryption

In DES/TDES CBC decryption, when the TDES algorithm is configured, the first ciphertext block is used directly after data swapping according to the data type, it is read in the DEA and decrypted using KEY3. The output is fed back directly to next DEA and then encrypted using KEY2. After that, the output is fed back directly to the last DEA and decrypted with KEY1. The first result of above process is then XORed with the initialization vector which is the same as that used during encryption. At the same time, the first ciphertext is then used as the next initialization vector and exclusive-ORed with the next result after DEA blocks. The above operations are repeated until the last ciphertext block is decrypted. Note if the ciphertext message does not consist of an integral number of data blocks, the final partial data block should be decrypted in a specified manner same to that in encryption. At last, the output plaintext is also obtained after data swapping according to the data type. When the DES algorithm is configured, the state and process of the second and third block of DEA should also be omitted. The procedure of DES / TDES CBC mode decryption is illustrated in ***Figure 13-7. DES/TDES CBC decryption***.

Figure 13-7. DES/TDES CBC decryption



13.4.2. 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.

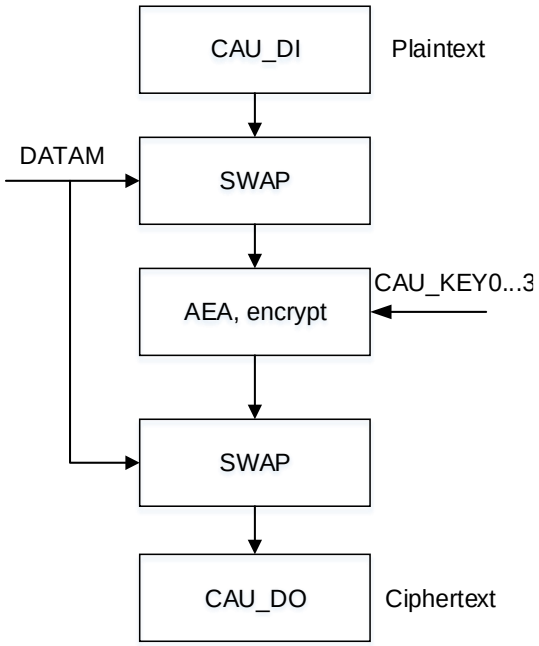
The AES key is used as [KEY3 KEY2] when the key size is configured as 128, [KEY3 KEY2 KEY1] when the key size is configured as 192 and [KEY3 KEY2 KEY1 KEY0] when the key size is configured as 256.

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 13-8. AES ECB encryption***.

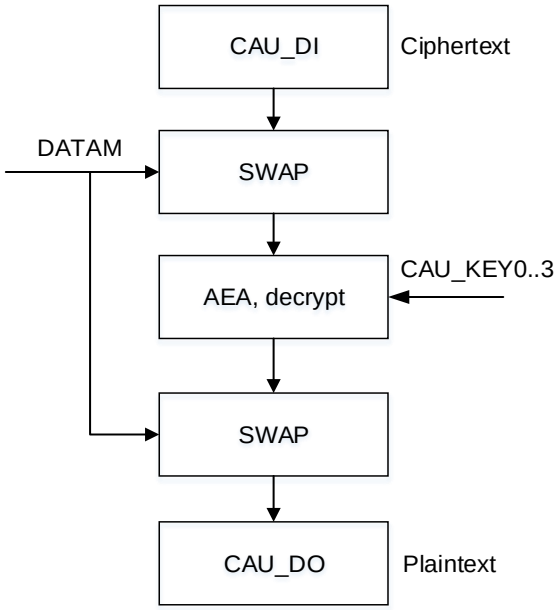
Figure 13-8. AES ECB encryption



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 13-9. AES ECB decryption***.

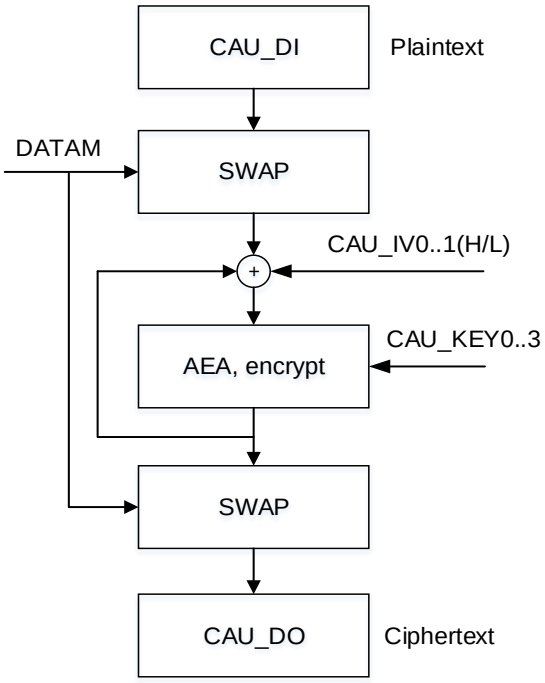
Figure 13-9. AES ECB decryption



AES-CBC mode encryption

The input data of the AEA block in CBC mode consists of two aspects: the input plaintext after data swapping according to the data type, and the initialization vectors. The XOR result of the swapped plaintext data block and the 128-bit initialization vector CAU_IV0..1 is read in the AEA and encrypted using the 128-, 192-, 256-bit key. The result is then used as the next initialization vector and exclusive-ORed with the next plaintext data block to process next encryption. The above operations are repeated until the last plaintext block is encrypted. Note if the plaintext message does not consist of an integral number of data blocks, the final partial data block should be encrypted in a specified manner. At last, the output ciphertext is also obtained after data swapping according to the data type. The procedure of AES CBC mode encryption is illustrated in ***Figure 13-10. AES CBC encryption***.

Figure 13-10. AES CBC encryption

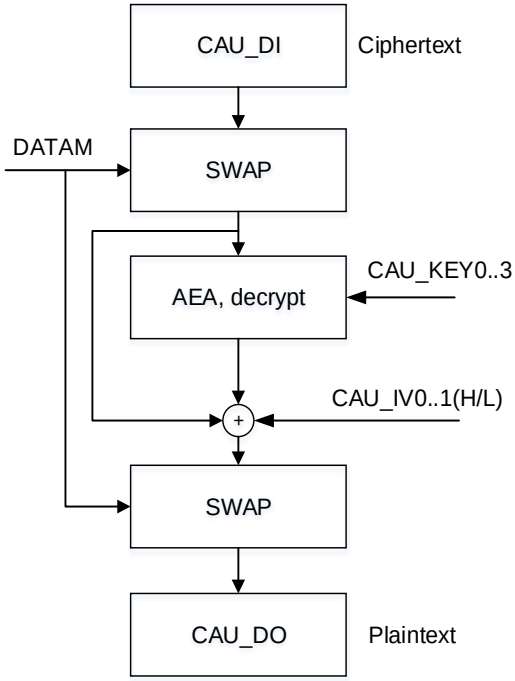


AES-CBC mode decryption

Similar to that in AES-ECB mode decryption, the key derivation also must be completed first to prepare the decryption keys, the input of the key schedule should be 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. At the same time, the first ciphertext is then used as the next initialization vector and exclusive-ORed with the next result after AEA blocks (The first initialization is obtained directly from the CAU_IV0..1 registers). The above operations are repeated until the last ciphertext block is decrypted. Note if the ciphertext message does not consist of an integral number of data blocks, the final partial data block should be decrypted in a specified manner same to that in encryption. At last, the output

plaintext is also obtained after data swapping according to the data type. The procedure of AES CBC mode decryption is illustrated in ***Figure 13-11. AES CBC decryption***.

Figure 13-11. AES CBC decryption



AES-CTR mode

In counter mode, a counter is used in addition with a nonce value to be encrypted and decrypted in AEA, and the result will be used for the XOR operation with the plaintext or the ciphertext. As the counter is incremented from the same initialized value for each block in encryption and decryption, the key schedule during the encryption and decryption are the same. Then decryption operation acts exactly in the same way as the encryption operation. Only the 32-bit LSB of the 128-bit initialization vector represents the counter, which means the other 96 bits are unchanged during the operation, and the initial value should be set to 1. Nonce is 32-bit single-use random value and should be updated to each communication block. And the 64-bit initialization vector is used to ensure that a given value is used only once for a given key. ***Figure 13-12. Counter block structure*** illustrates the counter block structure and ***Figure 13-13. AES CTR encryption/decryption*** shows the AES CTR encryption/decryption.

Figure 13-12. Counter block structure

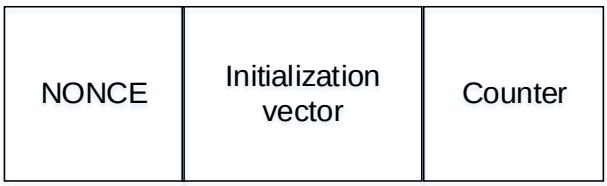
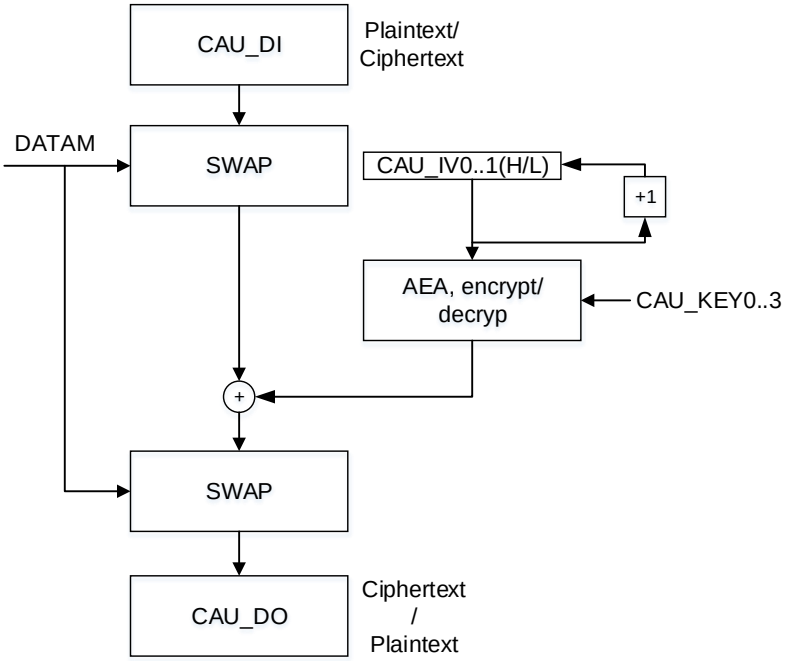


Figure 13-13. AES CTR encryption/decryption



AES-GCM mode

The AES Galois/counter mode (GCM) can be used to encrypt or authenticate message, and then ciphertext and tag can be obtained. This algorithm is based on AES CTR mode to ensure confidentiality. A multiplier over a fixed finite field is used to generate the tag.

In this mode, four steps are required to perform an encryption/decryption:

1. GCM prepare phase

   The hash key is calculated and saved internally to be used later.

   (a) Clear the CAUEN bit to make sure CAU is disabled.
   (b) Configure the ALGM[3:0] bits to '1000'.
   (c) Configure GCM_CCMPH[1:0] bits to '00'.
   (d) Configure key registers and initialization vectors.
   (e) Enable CAU by writing 1 to CAUEN bit.
   (f) Wait until CAUEN bit is cleared by hardware, and then enable CAU again for following phases.

2. GCM AAD (additional authenticated data) phase

   This phase must be performed after GCM prepare phase and also precede the encryption/decryption phase. In this phase, data is authenticated but not protected.

   (g) Configure GCM_CCMPH[1:0] bits to '01'.
   (h) Write data into CAU_DI register, INF and IEM flags can be used to determine if the input FIFO can receive data. The size of the AAD must be a multiple of 128bits. DMA can also be used.

(i) Repeat (h) until all AAD data are supplied, wait until BUSY bit is cleared.

3. GCM encryption/decryption phase

   This phase must be performed after GCM AAD phase. In this phase, the message is authenticated and encrypted/decrypted.

   (j) Configure GCM_CCMPH[1:0] bits to '10'.
   (k) Configure the computation direction in CAUDIR.
   (l) Write data into CAU_DI register, INF and IEM flags can be used to determine if the input FIFO can receive data. ONE and OFU flags can be used to check if the output FIFO is not empty. If so, read the CAU_DO register. DMA can also be used.
   (m) Repeat (l) step until all payload blocks are processed.

4. GCM tag phase

   In this phase, the final authentication tag is generated.

   (n) Configure GCM_CCMPH[1:0] bits to '11'.
   (o) Write the input into the CAU_DI register, 4 times write operation is needed. The input consists of the AAD data size (64bits) and the payload data size (64bits).
   (p) Wait until the ONE flag is set to 1, and then read CAU_DO 4 times. The output corresponds to the authentication tag.
   (q) Disable the CAU.

Note: The key should be prepared at the beginning when a decryption is performed.

AES-GMAC mode

The AES Galois message authentication code mode is also supported to authenticate the message. It is processing based on the AES-GCM mode, while the encryption/decryption phase is by-passed.

AES-CCM mode

The AES combined cipher machine mode, which is similar to AES-GCM mode, also allows encrypting and authenticating message. It is also based on AES-CTR mode to ensure confidentiality. In this mode, AES-CBC is used to generate a 128-bit tag.

The CCM standard (RFC 3610 Counter with CBC-MAC (CCM) dated September 2003) defines particular encoding rules for the first authentication block (B0 in the standard). In particular, the first block includes flags, a nonce and the payload length expressed in bytes. The CCM standard specifies another format, called A or counter, for encryption/decryption. The counter is incremented during the encryption/decryption phase and its 32 LSB bits are initialized to '1' during the tag generation (A0 packet in the CCM standard).

Note: The formatting operation of B0 packet should be handled by software.

In this mode, four steps are required to perform an encryption/decryption:

1. CCM prepare phase

   In this phase, B0 packet (the first packet) is programmed into the CAU_DI register. CAU_DO never contain data in this phase.

   (a) Clear the CAUEN bit to make sure CAU is disabled.
   (b) Configure the ALGM[3:0] bits to '1001'.
   (c) Configure GCM_CCMPH[1:0] bits to '00'.
   (d) Configure key registers and initialization vectors.
   (e) Enable CAU by writing 1 to CAUEN bit.
   (f) Program the B0 packet into the CAU_DI.
   (g) Wait until CAUEN is cleared by hardware, and then enable CAU again for following phases.

2. CCM AAD (additional authenticated data) phase

   This phase must be performed after CCM prepare phase and also precede the encryption/decryption phase. In this phase, CAU_DO never contain data in this phase.

   This phase can be by-passed if there is no additional authenticated data.

   (h) Configure GCM_CCMPH[1:0] bits to '01'.
   (i) Write data into CAU_DI register, INF and IEM flags can be used to determine if the input FIFO can receive data. The size of the AAD must be a multiple of 128 bits. DMA can also be used.
   (j) Repeat (i) until all header data are supplied, wait until BUSY bit is cleared.

3. CCM encryption / decryption phase

   This phase must be performed after CCM AAD phase. In this phase, the message is authenticated and encrypted/decrypted.

   Like GCM, the CCM chaining mode can be applied on a message composed only by plaintext authenticated data (that is, only AAD, no payload). Note that this way of using CCM is not called CMAC (it is not similar to GCM/GMAC).

   (k) Configure GCM_CCMPH[1:0] bits to '10'.
   (l) Configure the computation direction in CAUDIR.
   (m) Write data into CAU_DI register, INF and IEM flags can be used to determine if the input FIFO can receive data. ONE and OFU flags can be used to check if the output FIFO is not empty. If so, read the CAU_DO register. DMA can also be used.
   (n) Repeat (m) step until all payload blocks are processed.

4. CCM tag phase

   In this phase, the final authentication tag is generated.

   (o) Configure GCM_CCMPH[1:0] bits to '11'.
   (p) Write the 128 bit input into the CAU_DI register, 4 times of write operation to CAU_DI is needed. The input is the A0 value.

(q) Wait until the ONE flag is set to 1, and then read CAU_DO 4 times. The output corresponds to the authentication tag.

(r) Disable the CAU

AES-CFB mode

The Cipher Feedback (CFB) mode is a confidentiality mode that features the feedback of successive ciphertext segments into the input blocks of the forward cipher to generate output blocks that are exclusive-ORed with the plaintext to produce the ciphertext, and vice versa.

AES-OFB mode

The Output Feedback (OFB) mode is a confidentiality mode that features the iteration of the forward cipher on an IV to generate a sequence of output blocks that are exclusive-ORed with the plaintext to produce the ciphertext, and vice versa.

13.5. Operating modes

Encryption

1. Disable the CAU by resetting the CAUEN bit in the CAU_CTL register.
2. Enable CAU power domain by setting the CORE1WAKE bit in the PMU_CTL1 register, and then enable CAU clock.
3. Select and configure the key length with the KEYM bits in the CAU_CTL register if AES algorithm is chosen.
4. Configure the CAU_KEY0..3(H/L) registers according to the algorithm.
5. Configure the DATAM bit in the CAU_CTL register to select the data swapping type.
6. Configure the algorithm (DES / TDES / AES) and the chaining mode (ECB / CBC / CTR / GCM / GMAC / CCM / CFB / OFB) by writing the ALGM[3:0] bit in the CAU_CTL register.
7. Configure the encryption direction by writing 0 to the CAUDIR bit in the CAU_CTL register.
8. Configure the initialization vectors by writing the CAU_IV0..1 registers.
9. Flush the input FIFO and output FIFO by configure the FFLUSH bit in the CAU_CTL register when CAUEN is 0.
10. Enable the CAU by set the CAUEN bit as 1 in the CAU_CTL register.
11. 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.
12. 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.
13. Repeat steps 10, 11 until all data blocks has been encrypted.

Decryption

1. Disable the CAU by resetting the CAUEN bit in the CAU_CTL register.
2. Enable CAU power domain by setting the CORE1WAKE bit in the PMU_CTL1 register, and then enable CAU clock.
3. Select and configure the key length with the KEYM bits in the CAU_CTL register if AES algorithm is chosen.
4. Configure the CAU_KEY0..3(H/L) registers according to the algorithm.
5. Configure the DATAM bit in the CAU_CTL register to select the data swapping type.
6. Configure the ALGM[3:0] bits to “0111” in the CAU_CTL register to complete the key derivation.
7. Enable the CAU by set the CAUEN bit as 1.
8. Wait until the BUSY and CAUEN bit return to 0 to make sure that the decryption keys are prepared.
9. Configure the algorithm (DES / TDES / AES) and the chaining mode (ECB / CBC / CTR / GCM / GMAC / CCM / CFB / OFB) by writing the ALGM[3:0] bit in the CAU_CTL register.
10. Configure the decryption direction by writing 1 to the CAUDIR bit in the CAU_CTL register.
11. Configure the initialization vectors by writing the CAU_IV0..1 registers.
12. Flush the input FIFO and output FIFO by configure the FFLUSH bit in the CAU_CTL register when CAUEN is 0.
13. Enable the CAU by set the CAUEN bit as 1 in the CAU_CTL register.
14. 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.
15. 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.
16. Repeat steps 13, 14 until all data blocks has been decrypted.

Data append

For GCM payload encryption or CCM payload decryption, CAU supports non 128 bit integer multiple data block processing. When the last data block is less than 128bit, use ‘0’ to fill the remaining bits, and then configure the number of bytes to be filled in the NBPILB bitfield of the CAU_CTL register. AES will automatically remove the the number of filled pads and encrypt it. It should be noted that the NBPILB[3:0] bitfield should be configured after the encryption of the penultimate data block is completed.

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.

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.

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

The input FIFO interrupt is asserted when the number of words in the input FIFO is less than four words, then ISTA is asserted. And if the input FIFO interrupt is enabled by IINTEN with a 0 value, the IINTF is also asserted. Note if the CAUEN is low, then the ISTA and IINTF are also always low.

Output FIFO interrupt

The output FIFO interrupt is asserted when the number of words in the output FIFO is more than one words, then OSTA is asserted. And if the output FIFO interrupt is enabled by OINTEN with a 0 value, the OINTF is also asserted. Note Unlike that of Input FIFO interrupt, the value of CAUEN will never affect the situation of OSTA and OINTF.

13.8. CAU suspended mode

It is possible to suspend a data block if another new data block with a higher priority needs to be processed in CAU. The following steps can be performed to complete the encryption/decryption acceleration of the suspended data blocks.

When DMA transfer is used:

1. Stop the current input transfer. Clear the DMAIEN bit in the CAU_DMAEN register.
2. When it is DES or AES, wait until both the input and output FIFO are both empty if the input FIFO is not empty (IEM = 0), then write a word of data into CAU_DI register, do as so until the IEM is checked to be 1, then wait until the BUSY bit is cleared, so that the next data block will not be affected by the last one. Case of TDES is similar to that of AES except that it does not need to wait until the input FIFO is empty.
3. Stop the output transfer by clearing the DMAOEN bit in the CAU_DMAEN register. And disable the CAU by clearing the CAUEN bit in the CAU_CTL register.
4. Save the configuration, including the key size, data type, operation mode, direction, GCM CCM phase and the key values. When it is CBC, CTR, GCM, GMAC, CCM, CFB or OFB chaining mode, the initialization vectors should also be stored. When it is GCM, GMAC

or CCM mode, the context switch CAU_GCMCCMCTXSx (x = 0..7) and CAU_GCMCTXSx (x = 0..7) registers should also be stored.

5. Configure and process the new data block.
6. Restore the process before. Configure the CAU with the parameters stored before, and prepare the key and initialization vectors, and the context switch registers CAU_GCMCCMCTXSx (x = 0..7) and CAU_GCMCTXSx (x = 0..7) should also be restored. Then enable CAU by setting the CAUEN bit in the CAU_CTL register.

When data transfer is done by CPU access to CAU_DI and CAU_DO:

1. When the data transfer is done by CPU access, then wait for the fourth read of the CAU_DO register and before the next CAU_DI write access so that the message is suspended at the end of a block processing.
2. Disable the CAU by clearing the CAUEN bit in the CAU_CTL register.
3. Save the configuration, including the key size, data type, operation mode, direction, GCM CCM phase and the key values. When it is CBC, CTR, GCM, GMAC, CCM, CFB or OFB chaining mode, the initialization vectors should also be stored. When it is GCM, GMAC or CCM mode, the context switch CAU_GCMCCMCTXSx (x = 0..7) and CAU_GCMCTXSx (x = 0..7) registers should also be stored.
4. Configure and process the new data block.
5. Restore the process before. Configure the CAU with the parameters stored before, and prepare the key and initialization vectors, and the context switch registers CAU_GCMCCMCTXSx (x = 0..7) and CAU_GCMCTXSx (x = 0..7) should also be restored. Then enable CAU by setting the CAUEN bit in the CAU_CTL register.

13.9. Register definition

CAU base address: 0x4802 1000

13.9.1. Control register (CAU_CTL)

Address offset: 0x00

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								NBPILB[3:0]				ALGM[3]	Reserved	GCM_CCMPH[1:0]	
								rw				rw		rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAUEN	FFLUSH	Reserved				KEYM[1:0]		DATAM[1:0]		ALGM[2:0]			CAUDIR	Reserved	
rw	w					rw		rw		rw			rw		

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:20	NBPILB[3:0]	Number of bytes padding in last block of payload 0000: all bytes are valid (no padding) 0001: one padding byte of last block ... 1111: 15 padding bytes of last block
19	ALGM[3]	Encryption/decryption algorithm mode bit 3
18	Reserved	Must be kept at reset value.
17:16	GCM_CCMPH[1:0]	GCM CCM phase 00: prepare phase 01: AAD phase 10: encryption/decryption phase 11: tag phase
15	CAUEN	CAU Enable 0: CAU is disabled 1: CAU is enabled **Note:** the CAUEN can be cleared automatically when the key derivation (ALGM=0111b) is finished or the AES-GCM or AES-CCM initial phase 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 0000: TDES-ECB with CAU_KEY1, 2, 3. Initialization vectors (CAU_IV0..1) are not used 0001: TDES-CBC with CAU_KEY1, 2, 3. Initialization vectors (CAU_IV0) is used to XOR with data blocks 0010: DES-ECB with only CAU_KEY1 Initialization vectors (CAU_IV0..1) are not used 0011: DES-CBC with only CAU_KEY1 Initialization vectors (CAU_IV0) is used to XOR with data blocks 0100: AES-ECB with CAU_KEY0, 1, 2, 3. Initialization vectors (CAU_IV0..1) are not used 0101: AES-CBC with CAU_KEY0, 1, 2, 3. Initialization vectors (CAU_IV0..1) are used to XOR with data blocks 0110: AES_CTR with CAU_KEY0, 1, 2, 3. Initialization vectors (CAU_IV0..1) are used to XOR with data blocks In this mode, encryption and decryption are same, then the CAUDIR is disregarded. 0111: 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. 1000: Galois Counter Mode (GCM). This algorithm mode is also used for GMAC algorithm. 1001: Counter with CBC-MAC (CCM). 1010: Cipher Feedback (CFB) mode 1011: Output Feedback (OFB) mode
2	CAUDIR	CAU direction, must be configured when BUSY=0 0: encryption 1: decryption

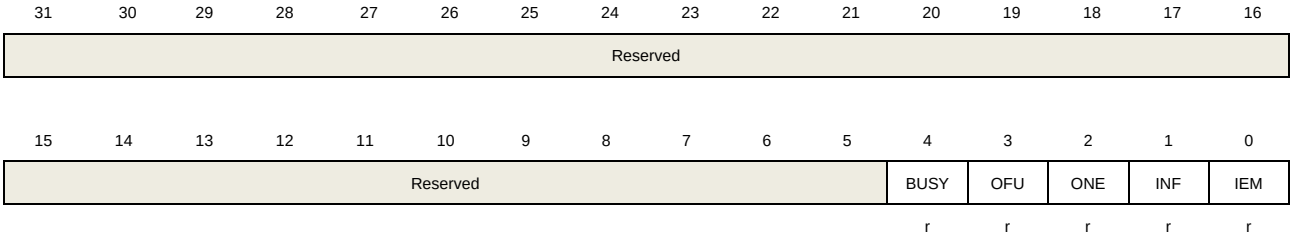
1:0	Reserved	Must be kept at reset value.

13.9.2. Status register 0 (CAU_STAT0)

Address offset: 0x04

Reset value: 0x0000 0003

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



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

13.9.3. Data input register (CAU_DI)

Address offset: 0x08

Reset value: 0x0000 0000

The data input register is used to transfer plaintext or ciphertext blocks into the input FIFO for processing. The MSB is firstly written into the FIFO and the LSB is the last one. If the CAUEN is 0 and the input FIFO is not empty, when it is read, then the first data in the FIFO is popped out and returned. If the CAUEN is 1, the returned value is undefined. Once it is read, then the FIFO must be flushed.

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

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

Bits	Fields	Descriptions
31:0	DI[31:0]	Data input Write these bits will write data to IN FIFO, read these bits will return IN FIFO value if CAUEN is 0, or it will return an undefined value

13.9.4. 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															

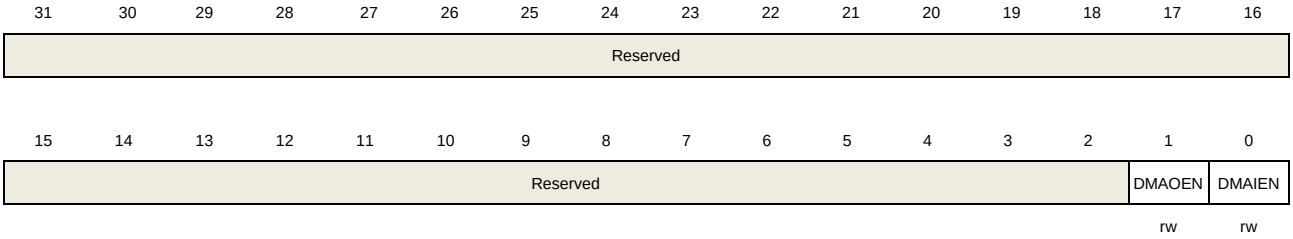
Bits	Fields	Descriptions
31:0	DO[31:0]	Data output These bits are read only, read these bits return OUT FIFO value.

13.9.5. DMA enable register (CAU_DMAEN)

Address offset: 0x10

Reset value: 0x0000 0000

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



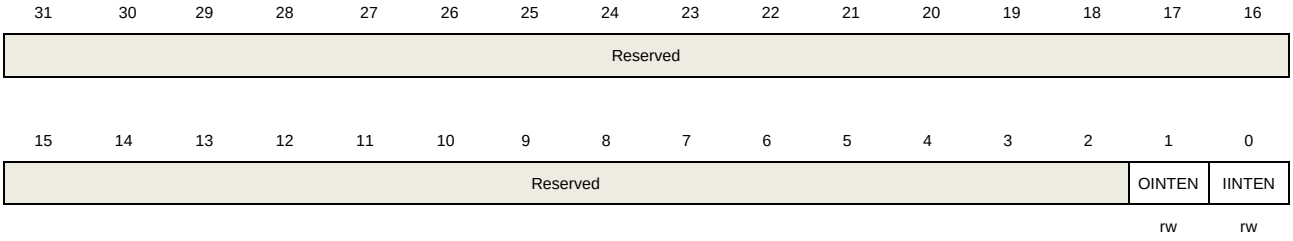
Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	DMAOEN	DMA output enable 0: DMA for OUT FIFO data is disabled 1: DMA for OUT FIFO data is enabled
0	DMAIEN	DMA input enable 0: DMA for IN FIFO data is disabled 1: DMA for IN FIFO data is enabled

13.9.6. Interrupt enable register (CAU_INTEN)

Address offset: 0x14

Reset value: 0x0000 0000

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



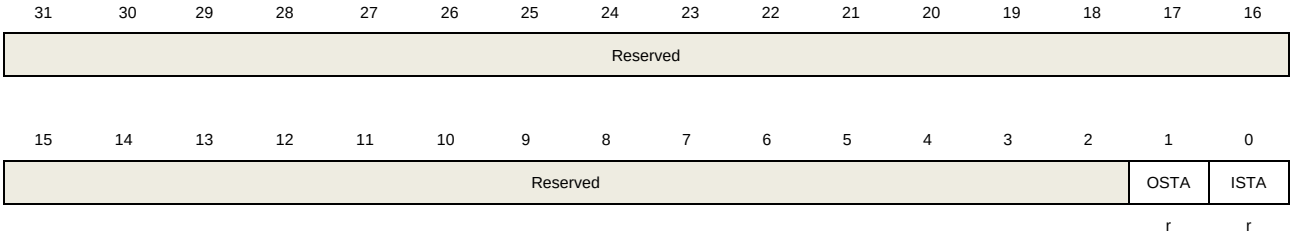
Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	OINTEN	OUT FIFO interrupt enable 0: OUT FIFO interrupt is disable 1: OUT FIFO interrupt is enable
0	IINTEN	IN FIFO interrupt enable 0: IN FIFO interrupt is disable 1: IN FIFO interrupt is enable

13.9.7. Status register 1 (CAU_STAT1)

Address offset: 0x18

Reset value: 0x0000 0001

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



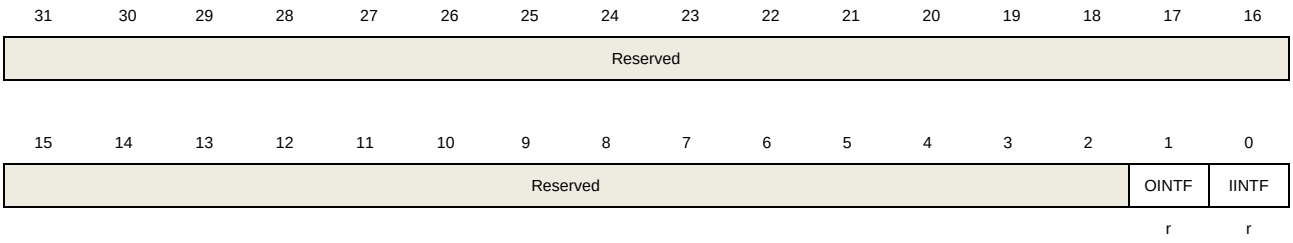
Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	OSTA	OUT FIFO interrupt status 0: OUT FIFO interrupt status not pending 1: OUT FIFO interrupt status pending
0	ISTA	IN FIFO interrupt status 0: IN FIFO interrupt not pending 1: IN FIFO interrupt flag pending

13.9.8. Interrupt flag register (CAU_INTF)

Address offset: 0x1C

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	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

13.9.9. Key registers (CAU_KEY0..3(H/L))

Address offset: 0x20 to 0x3C
Reset value: 0x0000 0000

This registers have to be accessed by word (32-bit), and all of them must be written when BUSY is 0.

In DES mode, only CAU_KEY1 is used.

In TDES mode, CAU_KEY1, CAU_KEY2 and CAU_KEY3 are used.

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

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

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

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

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

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

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

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

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 DES, TDES, AES

13.9.10. Initial vector registers (CAU_IV0..1(H/L))

Address offset: 0x40 to 0x4C

Reset value: 0x0000 0000

This registers have to be accessed by word (32-bit), and all of them must be written when BUSY is 0.

In DES/TDES mode, IV0H is the leftmost bits, and IV0L is the rightmost bits of the initialization vectors.

In AES mode, IV0H is the leftmost bits, and IV1L is the rightmost bits of the initialization vectors.

CAU_IV0H

Address offset: 0x40

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IV0H[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IV0H[15:0]															
rw															

CAU_IV0L

Address offset: 0x44

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IV0L[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IV0L[15:0]															
rw															

CAU_IV1H

Address offset: 0x48

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IV1H[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IV1H[15:0]															
rw															

CAU_IV1L

Address offset: 0x4C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IV1L[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IV1L[15:0]															
rw															

Bits	Fields	Descriptions
31:0	IV0...1(H/L)	The initialization vector for DES, TDES, AES

13.9.11. GCM or CCM mode context switch register x (CAU_GCMCCMCTXSx) (x=0..7)

Address offset: 0x50 to 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
CTXx[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTXx[15:0]															
rw															

Bits	Fields	Descriptions
31:0	CTXx[31:0]	The internal status of the CAU core. Read and save the register data when a high-priority task is coming to be processed, and restore the saved data back to the registers to resume the suspended processing. **Note:** These registers are used only when GCM, GMAC or CCM mode is selected.

13.9.12. GCM mode context switch register x (CAU_GCMCTXSx) (x=0..7)

Address offset: 0x70 to 0x8C
Reset value: 0x0000 0000

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

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

Bits	Fields	Descriptions
31:0	CTXx[31:0]	The internal status of the CAU core. Read and save the register data when a high-priority task is coming to be processed, and restore the saved data back to the registers to resume the suspended processing. **Note:** These registers are used only when GCM or GMAC mode is selected.

14. Trigonometric Math Unit (TMU)

14.1. Overview

The Trigonometric Math Unit (TMU) is a fully configurable block that execute common trigonometric and arithmetic operations. The TMU can reduce the burden of CPU, and it is usually used in motor control, signal processing and many other applications.

The TMU can be used to calculate total 10 kinds of functions. The input / output data support q1.31 or q1.15 fixed point format or IEEE754 32-bit single precision floating-point format.

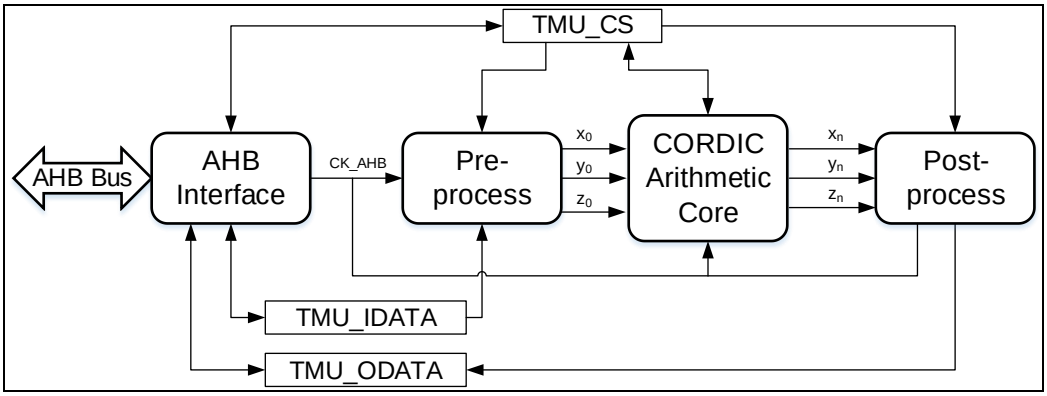
14.2. Characteristics

- 10 kinds of functions.
- Interrupt and DMA requests.
  - The fixed point q1.31 / q1.15 format or IEEE754 32-bit single precision floating-point format is configurable.
  - Programmable precision.
- CORDIC-algorithm core: circular system and hyperbolic system, rotation pattern and vectoring pattern.

14.3. Block diagram

Figure 14-1. TMU block diagram provides details of the internal configuration of the TMU.

Figure 14-1. TMU block diagram



The Pre-process module converts the contents in the TMU_IDATA register to obtain the initial data (x_0,y_0,z_0) required by the CORDIC-arithmetic core. The contents of the TMU_IDATA register are in the q1.31 / q1.15 format or IEEE754 32-bit single precision floating-point format.

After the initial data (x_0,y_0,z_0) is input to the CORDIC-algorithm core, it is iterated and calculated to obtain the (x_n,y_n,z_n). The CORDIC-algorithm core supports circular system and

hyperbolic system, and each system supports rotation pattern and vectoring pattern.

The Post-process module converts and scales the data (x_n,y_n,z_n) and writes the processed results into TMU_ODATA register. The contents of the TMU_ODATA register are in the fixed point q1.31 / q1.15 format or IEEE754 32-bit single precision floating-point format.

14.4. Function overview

14.4.1. Data format and configuration

When the IFLTEN / OFLTEN bit is reset, the input / output data of TMU module is in fixed point signed integer format (q1.31 and q1.15 format). When the IFLTEN / OFLTEN bit is set, the input / output data of TMU module is in IEEE754 32-bit single precision floating-point format.

In q1.31 format, the 31 bit is sign bit and 0~30 bits are fractional bits. The value range is [-1,1-2^{-31}], corresponding to [0x80000000,0x7FFFFFFF].

In q1.15 format, the 15 bit is sign bit and 0~14 bits are fractional bits. The value range is [-1,1-2^{-15}], corresponding to [0x8000,0x7FFF].

In IEEE754 32-bit single precision floating-point format, the value range is shown in ***Table 14-3 IEEE754 32-bit single precision floating-point format***.

When the input data is in fixed point format (IFLTEN=0), the IWIDTH bit in TMU_CS register is used to configure the fixed point format of the input data. When the input data is in floating-point format (IFLTEN=1), the configuration of the IWIDTH bit is invalid. Each mode requires a different number of input data (for example, mode 0 requires two inputs and mode 5 requires only one), and can be configured via the INUM bit of the TMU_CS register. Detailed configuration refer to ***Table 14-1. Input data configuration***.

Note: When the input data is configured in q1.15 format, the TMU_IDATA register only needs to be written once, the first input data in the low half word, the second input data in the high half word. If the mode only needs one input data, only the low half word is used, and the high half word is not used.

Table 14-1. Input data configuration

IWIDTH bit	INUM bit	IFLTEN bit	Data format	Write operation to TMU_IDATA
0	0	0	q1.31 fixed point	Only one write operation
0	1	0	q1.31 fixed point	Two successive write operation
1	0	0	q1.15 fixed point	Only one write operation
1	1	0	q1.15 fixed point	Not available
X	0	1	IEEE754 32-bit single precision floating-point	Only one write operation
X	1	1		Two successive write operation

When the output data is in fixed point format (OFLTEN=0), the OWIDTH bit in TMU_CS

register is used to configure the fixed point format of the output data. When the output data is in floating-point format (OFLTEN=1), the configuration of the OWIDTH bit is invalid. Each mode requires a different number of output data (for example, mode 0 has two output datas and mode 8 only has one), and can be configured via the ONUM bit of the TMU_CS register. Detailed configuration refer to ***Table 14-2. Output data configuration***.

Note: When the output data is configured in q1.15 format, the TMU_IDATA register only needs to be read once, the first output data in the low half word, the second output data in the high half word. If the mode only needs one output data, only the low half word is used, and the high half word is not used.

Table 14-2. Output data configuration

OWIDTH bit	ONUM bit	OFLTEN bit	Data format	Read operation to TMU_ODATA
0	0	0	q1.31 fixed point	Only one read operation
0	1	0	q1.31 fixed point	Two successive read operation
1	0	0	q1.15 fixed point	Only one read operation
1	1	0	q1.15 fixed point	Not available
X	0	1	IEEE754 32-bit single precision floating-point	Only one read operation
X	1	1		Two successive read operation

14.4.2. Floating-point data format

When the IFLTEN / OFLTEN bit is set, the input / output data of TMU module is IEEE754 32-bit single precision floating-point format, as shown in ***Table 14-3 IEEE754 32-bit single precision floating-point format***.

Table 14-3 IEEE754 32-bit single precision floating-point format

S [31]	E [30:23]	M [22:0]	Value (V)
0	0	0	Zero (V = 0)
1	0	0	Negative Zero (V = -0)
0 + ve 1 - ve	0	non zero	De-normalized (V=$(-1)^{s}$*$2^{(-126)}$* (0.M))
0 + ve 1 - ve	1 to 254	0 to 0x7FFFFF	Normal Range(V=$(-1)^{s}$*$2^{(E-127)}$* (1.M))
0	254	0x7FFFFF	Positive Max (V = +Max)
1	254	0x7FFFFF	Negative Max (V = -Max)
0	max=255	0	Positive Infinity (V = +Infinity)
1	max=255	0	Negative Infinity (V = -Infinity)
x	max=255	non zero	Not A Number (V = NaN)

The treatment of the various IEEE floating-point numerical formats for this TMU is given as below:

Negative Zero: All TMU operations generate a positive (S=0, E=0, M=0) zero, never a negative zero if the result of the operation is zero. All TMU operations treat negative zero operations as zero.

De-Normalized Numbers: A de-normalized operand (E=0, M!=0) input is treated as zero (E=0, M=0) by all TMU operations.

Not a Number (NaN): An NaN operand (E=max, M!=0) input is treated as Infinity (E=max, M=0) for all operations.

Overflow: When the IFLTEN and OFLTEN bits are both set, overflow occurs when the input data or an operation generates a value exceeds the domain of the corresponding mode. At the same time, the OVRF bit in the TMU_CS register is set to 1.

14.4.3. Mode configuration

The MODE[3:0] bit-field in TMU_CS register is used to configure the mode of the CORDIC-algorithm core. Different modes use different systems (circular or hyperbolic) and different patterns (rotation or vectoring). Detailed configuration refer to ***Table 14-4. TMU mode configuration***.

Table 14-4. TMU mode configuration

Mode	The first input data	The second input data	The first output data	The second output data	System and Pattern
Mode 0	θ	m	m*cos(θ)	m*sin(θ)	Circular, Rotation
Mode 1	θ	m	m*sin(θ)	m*cos(θ)	Circular, Rotation
Mode 2	x	y	atan2 (y,x)	$\sqrt{x^2+y^2}$	Circular, Vectoring
Mode 3	x	y	$\sqrt{x^2+y^2}$	atan2 (y,x)	Circular, Vectoring
Mode 4	x	None	$\tan^{-1}(x)$	None	Circular, Vectoring
Mode 5	x	None	cosh (x)	sinh (x)	Hyperbolic, Rotation
Mode 6	x	None	sinh (x)	cosh (x)	Hyperbolic, Rotation
Mode 7	x	None	$\tanh^{-1}(x)$	None	Hyperbolic, Vectoring
Mode 8	x	None	ln (x)	None	Hyperbolic, Vectoring
Mode 9	x	None	$\sqrt{x}$	None	Hyperbolic, Vectoring

Although TMU algorithm can only calculate a small number of functions directly, more functions can be obtained indirectly. For example, $e^x = \sinh(x) + \cosh(x)$.

Mode 0: m* cos (θ)

Mode 0 calculates the cosine of an angle. This mode takes two input datas and generates two ouput datas.

When IFLTEN = 1 and OFLTEN = 1, the detailed information refer to ***Table 14-5. Mode 0 description, when IFLTEN = 1 and OFLTEN = 1***.

Table 14-5. Mode 0 description, when IFLTEN = 1 and OFLTEN = 1

Parameter	Range	Description
First input data	$\frac{\theta}{\pi} \in$(-2^24,2^24)	The angle θ is in radians and overflow when exceed the range.
Second input data	m∈(-∞,+∞)	Overflow when m is too big for floating-point number
First output data	m* cos(θ) ∈(-∞,+∞)	--
Second output data	m* sin(θ) ∈(-∞,+∞)	--
FACTOR[2:0]	Not available	keep at reset value 3'b000.

When IFLTEN = 0 or OFLTEN = 0, the detailed information refer to ***Table 14-6. Mode 0 description, when IFLTEN = 0 or OFLTEN = 0***.

Table 14-6. Mode 0 description, when IFLTEN = 0 or OFLTEN = 0

Parameter	Range	Description
First input data	$\frac{\theta}{\pi} \in$[-1,1)	The angle θ in radians range from –π to π. The θ must be divide by π in software to convert it to the range [-1,1), and then it is written to TMU_IDATA register according to the format of q1.31, q1.15 or float.
Second input data	m∈[0,1)	If 0≤m<1, it is written to TMU_IDATA register according to the format of q1.31, q1.15 or float. if m≥1, a scaling must be applied in software to convert it to the range [-1,1), and then it is written to TMU_IDATA register according to the format of q1.31, q1.15 or float.
First output data	m* cos(θ) ∈[-1,1)	If the previous software has shrunk m, the output data needs to be scaled up to obtain the real result.
Second output data	m* sin(θ) ∈[-1,1)	
FACTOR[2:0]	Not available	keep at reset value 3'b000.

Note: When IFLTEN = 0 or OFLTEN = 0, if m>1, the software scaling value is optional.

For example, calculating $100^* \cos\left(\frac{\pi}{2}\right)$. When the IFLTEN and OFLTEN bits are reset, and IWIDTH and OWIDTH bits are set, the input and output datas are in q1.15 format. The steps to calculate the function are as follows:

1. Software processes the first input parameter $\frac{\pi}{2}$. Angle $\frac{\pi}{2}$ is divided by π: $\frac{\frac{\pi}{2}}{\pi}$=0.5, 0.5 is 0x4000 in q1.15 format. Software processes the first input parameter $\frac{\pi}{2}$. Angle $\frac{\pi}{2}$ is divided by π: $\frac{\frac{\pi}{2}}{\pi}$=0.5, 0.5 is 0x4000 in q1.15 format.
2. Software processes the second input parameter m. Modulus 100 is divided by 128: $\frac{100}{128}$=0.78125. 0.78125 is 0x6400 in q1.15 format. Software processes the second input

parameter m. Modulus 100 is divided by 128: $\frac{100}{128}$=0.78125. 0.78125 is 0x6400 in q1.15 format.

3. The first input data 0x4000 is written into TMU_IDATA.The first input data 0x4000 is written into TMU_IDATA.
4. The second input data 0x6400 is written into TMU_IDATA. Then the TMU calculation starts....The second input data 0x6400 is written into TMU_IDATA. Then the TMU calculation starts....
5. When the ENDF flag is set to 1, reading the TMU_ODATA register can get the first output data: $y_1=\frac{100}{128}*\cos\left(\frac{\pi}{2}\right)$, reading the TMU_ODATA register again can get the second output data: $y_2=\frac{100}{128}*\sin\left(\frac{\pi}{2}\right)$.
6. Software processes the result. Since the previous software has shrunk m by 128, the output data needs to be scaled up by 128 to obtain the real result: $100*\cos\left(\frac{\pi}{2}\right)=128*y_1$.

The scaling 128 is used for the input and output data in this example. Of course, other scaling value, such as 101, can also be used.

Mode 1: m*sin(θ)

Mode 1 calculates the sine of an angle. This mode take two input datas and generate two ouput datas.

When IFLTEN = 1 and OFLTEN = 1, the detailed information refer to ***Table 14-7. Mode 1 description, when IFLTEN = 1 and OFLTEN = 1***.

Table 14-7. Mode 1 description, when IFLTEN = 1 and OFLTEN = 1

Parameter	Range	Description
First input data	$\frac{\theta}{\pi}$∈(-2^24,2^24)	The angle θ is in radians and overflow when exceed the range.
Second input data	m∈(-∞,+∞)	Overflow when m is too big for floating-point number
First output data	m*sin(θ) ∈(-∞,+∞)	--
Second output data	m*cos(θ) ∈(-∞,+∞)	--
FACTOR[2:0]	Not available	keep at reset value 3'b000.

When IFLTEN = 0 or OFLTEN = 0, the detailed information refer to ***Table 14-8. Mode 1 description, when IFLTEN = 0 or OFLTEN = 0***.

Table 14-8. Mode 1 description, when IFLTEN = 0 or OFLTEN = 0

Parameter	Range	Description
First input data	$\frac{\theta}{\pi}$∈[-1,1)	The angle θ in radians range from –π to π. The θ must be divide by π in software to convert it to the range [-1,1), and then it is written to TMU_IDATA register according to the format of q1.31, q1.15 or float.

Parameter	Range	Description
Second input data	m∈[0,1)	If 0≤m<1, it is written to TMU_IDATA register according to the format of q1.31, q1.15 or float. if m≥1, a scaling must be applied in software to convert it to the range [-1,1), and then it is written to TMU_IDATA register according to the format of q1.31, q1.15 or float.
First output data	m* sin(θ) ∈[-1,1)	If the previous software has shrunk m, the output data needs to be scaled up to obtain the real result.
Second output data	m* cos(θ) ∈[-1,1)	
FACTOR[2:0]	Not available	keep at reset value 3'b000.

Note: When IFLTEN = 0 or OFLTEN = 0, if m>1, the software scaling value is optional.

For example, calculating $100*\sin\left(\frac{\pi}{2}\right)$. When the IFLTEN and OFLTEN bits are reset, and IWIDTH and OWIDTH bits are set, the input and output datas are in q1.15 format. The steps to calculate the function are as follows:

1. Software processes the first input parameter $\frac{\pi}{2}$. Angle $\frac{\pi}{2}$. is divided by π: $\frac{\frac{\pi}{2}}{\pi}$=0.5, 0.5 is 0x4000 in q1.15 format.....
2. Software processes the second input parameterm. Modulus 100 is divided by 128: $\frac{100}{128}$=0.78125. 0.78125 is 0x6400 in q1.15 format.....
3. The first input data 0x4000 is written into TMU_IDATA.The first input data 0x4000 is written into TMU_IDATA.
4. The second input data 0x6400 is written into TMU_IDATA. Then the TMU calculation starts....The second input data 0x6400 is written into TMU_IDATA. Then the TMU calculation starts....
5. When the ENDF flag is set to 1, reading the TMU_ODATA register can get the first output data: $y_1=\frac{100}{128}*\sin\left(\frac{\pi}{2}\right)$, and reading the TMU_ODATA register again can get the second output data:$y_2=\frac{100}{128}*\cos\left(\frac{\pi}{2}\right)$.
6. Software processes the result. Since the previous software has shrunk m by 128, the output data needs to be scaled up by 128 to obtain the real result: $100*\sin\left(\frac{\pi}{2}\right)=128*y_1$.

The scaling 128 is used for the input and output data in this example. Of course, other scaling value, such as 101, can also be used.

Mode 2: phase= atan2 (y,x)

Mode 2 calculates the atan2(y,x) of a vector (x,y). This mode take two input datas and generate two ouput datas.

When IFLTEN = 1 and OFLTEN = 1, the detailed information refer to ***Table 14-9. Mode 2 description, when IFLTEN = 1 and OFLTEN = 1***.

Table 14-9. Mode 2 description, when IFLTEN = 1 and OFLTEN = 1

Parameter	Range	Description
First input data	2^-24<|y/x|<2^24	Overflow when exceed the range.
Second input data		
First output data	θ∈[-1,1)	Angle, [-1,1) corresponding [-π, π). The output data is multiplied by π to get the real angle value.
Second output data	m∈[0,+∞)	Modulus, $m=\sqrt{x^2+y^2}$, overflow when m is too big for floating-point number
FACTOR[2:0]	Not available	keep at reset value 3'b000.

When IFLTEN = 0 or OFLTEN = 0, the detailed information refer to ***Table 14-10. Mode 2 description, when IFLTEN = 0 or OFLTEN = 0***.

Table 14-10. Mode 2 description, when IFLTEN = 0 or OFLTEN = 0

Parameter	Range	Description
First input data	x∈[-1,1)	The abscissa value in Cartesian coordinate system. If x≥1 or x<-1, software scaling is required.
Second input data	y∈[-1,1)	The ordinate value in Cartesian coordinate system. If y≥1 or y<-1, software scaling is required.
First output data	θ∈[-1,1)	Angle, [-1,1) corresponding [-π, π). The output data is multiplied by π to get the real angle value.
Second output data	m∈[0,1)	Modulus, $m=\sqrt{x^2+y^2}$. If x and y have been scaled before, the modulus needs to be scaled equally.
FACTOR[2:0]	Not available	keep at reset value 3'b000.

Note: When IFLTEN = 0 or OFLTEN = 0, as long as one of x and y is out of the range [-1,1), or the modulus $\sqrt{x^2+y^2}\geq 1$, x and y need to be scaled at the same scale at the same time to avoid going out of range of data. After scaling x and y in the same scale, the angle corresponding to the coordinates before and after scaling can be kept same.

For example, calculating θ=atan(5,80). When the IFLTEN and OFLTEN bits are reset, and IWIDTH and OWIDTH bits are set, the input and output are q1.15 format. The steps to calculate the function are as follows:

1. Software processes the input parameters (5,80). (5,80) is divided by 128. The result is (0.0390625,0.625). The q1.15 format is (0x0500,0x5000).
2. The first input data 0x0500 is written into TMU_IDATA.
3. The second input data 0x5000 is written into TMU_IDATA. Then the TMU calculation starts.
4. When the ENDF flag is set to 1, reading the TMU_ODATA register can get the first output data θ, and reading the TMU_ODATA register again can get the second output data modulus m.
5. Software processes the result. The first output data angle θ multiply π to get the real

radian. Since the previous software has shrunk the input datas by 128, the second output data m needs to be scaled up by 128 to obtain the real modulus.

The scaling 128 is used for the input datas and the modulus in this example. Of course, other scaling, such as 81, can also be used.

Mode 3: modulus=$\sqrt{x^2+y^2}$

Mode 3 calculates the modulus $\sqrt{x^2+y^2}$ of a vector (x,y). This mode takes two input datas and generates two ouput datas.

When IFLTEN = 1 and OFLTEN = 1, the detailed information refer to ***Table 14-11. Mode 3 description, when IFLTEN = 1 and OFLTEN = 1***.

Table 14-11. Mode 3 description, when IFLTEN = 1 and OFLTEN = 1

Parameter	Range	Description
First input data	2^-24<|y/x|<2^24	Overflow when exceed the range.
Second input data		
First output data	m∈[0,+∞)	Modulus, m=$\sqrt{x^2+y^2}$, overflow when m is too big for floating-point number.
Second output data	θ∈[-1,1)	Angle, [-1,1) corresponding [-π, π). The output data is multiplied by π to get the real angle value.
FACTOR[2:0]	Not available	keep at reset value 3'b000.

When IFLTEN = 0 or OFLTEN = 0, the detailed information refer to ***Table 14-12. Mode 3 description, when IFLTEN = 0 or OFLTEN = 0***.

Table 14-12. Mode 3 description, when IFLTEN = 0 or OFLTEN = 0

Parameter	Range	Description
First input data	x∈[-1,1)	The abscissa value in Cartesian coordinate system. If x≥1 or x<-1, software scaling is required.
Second input data	y∈[-1,1)	The ordinate value in Cartesian coordinate system. If y≥1 or y<-1, software scaling is required.
First output data	m∈[0,1)	Modulus, m=$\sqrt{x^2+y^2}$. If x and y have been scaled before, the modulus needs to be scaled equally.
Second output data	θ∈[-1,1)	Angle, [-1,1) corresponding [-π,π). The output data is multiplied by π to get the real angle value.
FACTOR[2:0]	Not available	Keep at reset value 3'b000.

Note: When IFLTEN = 0 or OFLTEN = 0, as long as one of x and y is out of the range [-1,1), or the modulus $\sqrt{x^2+y^2}$≥1, x and y need to be scaled at the same scale at the same time to avoid going out of range of data. After scaling x and y in the same scale, the angle corresponding to the coordinates before and after scaling can be kept same.

For example, calculating $\sqrt{5^2+80^2}$. When the IFLTEN and OFLTEN bits are reset, and

IWIDTH and OWIDTH bits are set, the input and output are q1.15 format. The steps to calculate the function are as follows:

1. Software processes the input parameters (5,80). (5,80) is divided by 128. The result is (0.0390625,0.625). The q1.15 format is (0x0500,0x5000).
2. The first input data 0x0500 is written into TMU_IDATA.
3. The second input data 0x5000 is written into TMU_IDATA. Then the TMU calculation starts.
4. When the ENDF flag is set to 1, reading the TMU_ODATA register can get the first output data modulus m, and reading the TMU_ODATA register again can get the second output data θ.
5. Software processes the result. Since the previous software has shrunk the input datas by 128, the first output data m needs to be scaled up by 128 to obtain the real modulus. The second output data angle θ multiply π to get the real radian.

The scaling 128 is used for the input datas and the modulus in this example. Of course, other scaling, such as 81, can also be used.

Mode 4: $\tan^{-1}(x)$

Mode 4 calculates the $\tan^{-1}(x)$. This mode takes one input data and generates one ouput data.

When IFLTEN = 1 and OFLTEN = 1, the detailed information refer to ***Table 14-13. Mode 4 description, when IFLTEN = 1 and OFLTEN = 1***.

Table 14-13. Mode 4 description, when IFLTEN = 1 and OFLTEN = 1

Parameter	Range	Description
Input data	x∈(-2^7,2^7)	Overflow when exceed the range.
Output data	$\theta\in(-\frac{1}{2},\frac{1}{2})$	Angle, [-1,1) corresponding [-π, π). The output data is multiplied by π and 2^f to get the real angle value θ.
FACTOR[2:0]	Not available	keep at reset value 3'b000

When IFLTEN = 0 or OFLTEN = 0, the detailed information refer to ***Table 14-14. Mode 4 description, when IFLTEN = 0 or OFLTEN = 0***.

Table 14-14. Mode 4 description, when IFLTEN = 0 or OFLTEN = 0

Parameter	Range	Description
Input data	$\frac{x}{2^f}\in[-1,1)$	If x∈[-1,1), the software does not need to process it, and the scaling factor is 2^0(FACTOR[2:0]=3'b000). If x is out of the range of [-1,1), it need to be scaled in software to ensure that $-1\le\frac{x}{2^f}<1$. Then write f to FACTOR[2:0] bit-field, and write the scaled data $\frac{x}{2^f}$ to TMU_IDATA in q1.31, q1.15 or float format.
Output data	$\frac{\theta}{2^f}\in(-\frac{1}{2},\frac{1}{2})$	Angle, [-1,1) corresponding [-π, π). The output data is multiplied by π and 2^f to get the real angle value θ.

Parameter	Range	Description
FACTOR[2:0]	f∈[0,7]	The bit-field FACTOR[2:0] is configured as f

Note: When IFLTEN = 0 or OFLTEN = 0, the scaling factor FACTOR[2:0] needs to be configured.

For example, calculating $\tan^{-1}(100)$. When the IFLTEN and OFLTEN bits are reset, and IWIDTH and OWIDTH bits are set, the input and output are q1.15 format. The steps to calculate the function are as follows:

1. Software processes the input parameter 100. 100 is divided by 128 (f=7=3'b111). The result is 0.78125 and the q1.15 format is 0x6400.
2. The scaling factor f=3'b111 is written into FACTOR[2:0] bit-field in TMU_CS register.
3. The input data 0x6400 is written into TMU_IDATA. Then the TMU calculation starts.
4. When the ENDF flag is set to 1, reading the TMU_ODATA register can get the output data $\frac{\theta}{2^7}$. The output data is q1.15 format.
5. Software processes the result. The output data $\frac{\theta}{2^7}$ needs to be multiplied by π and 2^7 to get the real radian.

Mode 5: cosh (x)

Mode 5 calculates calculates the hyperbolic cosine of a hyperbolic angle x. This mode takes one input data and generates two ouput datas.

When IFLTEN = 1 and OFLTEN = 1, the detailed information refer to ***Table 14-15. Mode 5 description, when IFLTEN = 1 and OFLTEN = 1***.

Table 14-15. Mode 5 description, when IFLTEN = 1 and OFLTEN = 1

Parameter	Range	Description
Input data	x∈[-1.118,1.118]	Overflow when exceed the range.
First output data	cosh (x) ∈[1,1.692]	--
Second output data	sinh (x) ∈[-1.366,1.366]	--
FACTOR[2:0]	Not available	keep at reset value 3'b000

When IFLTEN = 0 or OFLTEN = 0, the detailed information refer to ***Table 14-16. Mode 5 description, when IFLTEN = 0 or OFLTEN = 0.***

Table 14-16. Mode 5 description, when IFLTEN = 0 or OFLTEN = 0

Parameter	Range	Description
Input data	$\frac{x}{2}$∈[-0.559,0.559]	x∈[-1.118,1.118], a scaling factor $\frac{1}{2}$ is applied in softwate. Then write $\frac{x}{2}$ to TMU_IDATA in q1.31, q1.15 or float format.
First output data	$\frac{\cosh(x)}{2}$∈[0.5,0.846]	The output data is multiplied 2 to get the real hyperbolic cosine of a hyperbolic angle x.

Parameter	Range	Description
Second output data	$\frac{\sinh(x)}{2} \in [-0.683, 0.683]$	The output data is multiplied 2 to get the real hyperbolic sine of a hyperbolic angle x.
FACTOR[2:0]	1	The bit-field FACTOR[2:0] is configured as 3'b001

Note: When IFLTEN = 0 or OFLTEN = 0, the scaling factor FACTOR[2:0] must be 3'b001.

For example, calculating $\cosh(1.0)$. When the IFLTEN and OFLTEN bits are reset, and IWIDTH and OWIDTH bits are set, the input and output are q1.15 format. The steps to calculate the function are as follows:

1. Software processes the input parameter 1.0. 1.0 is divided by 2 (f=3'b001). The result is 0.5 and the q1.15 format is 0x4000.
2. The scaling factor (f=3'b001) is written into FACTOR[2:0] bit-field in TMU_CS register.
3. The input data 0x4000 is written into TMU_IDATA. Then the TMU calculation starts.
4. When the ENDF flag is set to 1, reading the TMU_ODATA register can get the first output data: $y_1 = \frac{\cosh(1.0)}{2}$, and reading the TMU_ODATA register again can get the second output data:$y_2 = \frac{\sinh(1.0)}{2}$.
5. Software processes the result. The two output datas are multiplied 2 to get the real hyperbolic cosine and sine of a hyperbolic angle 1.0.

Mode 6: sinh (x)

Mode 6 calculates calculates the hyperbolic sine of a hyperbolic angle x. This mode takes one input data and generates two ouput datas.

When IFLTEN = 1 and OFLTEN = 1, the detailed information refer to ***Table 14-17. Mode 6 description, when IFLTEN = 1 and OFLTEN = 1***.

Table 14-17. Mode 6 description, when IFLTEN = 1 and OFLTEN = 1

Parameter	Range	Description
Input data	$x \in [-1.118, 1.118]$	Overflow when exceed the range.
First output data	$\sinh(x) \in [-1.366, 1.366]$	--
Second output data	$\cosh(x) \in [1, 1.692]$	--
FACTOR[2:0]	Not available	keep at reset value 3'b000

When IFLTEN = 0 or OFLTEN = 0, the detailed information refer to ***Table 14-18. Mode 6 description, when IFLTEN = 0 or OFLTEN = 0.***

Table 14-18. Mode 6 description, when IFLTEN = 0 or OFLTEN = 0

Parameter	Range	Description
Input data	$\frac{x}{2} \in [-0.559, 0.559]$	$x \in [-1.118, 1.118]$, a scaling factor $\frac{1}{2}$ is applied in softwate. Then write $\frac{x}{2}$ to TMU_IDATA in q1.31,

Parameter	Range	Description
		q1.15 or float format.
First output data	$\frac{\sinh(x)}{2} \in [-0.683,0.683]$	The output data is multiplied 2 to get the real hyperbolic sine of a hyperbolic angle x.
Second output data	$\frac{\cosh(x)}{2} \in [0.5,0.846]$	The output data is multiplied 2 to get the real hyperbolic cosine of a hyperbolic angle x.
FACTOR[2:0]	1	The bit-field FACTOR[2:0] is configured as 3'b001

Note: When IFLTEN = 0 or OFLTEN = 0, the scaling factor FACTOR[2:0] must be 3'b001.

For example, calculating $\sinh(1.0)$. When the IFLTEN and OFLTEN bits are reset, and IWIDTH and OWIDTH bits are set, the input and output are q1.15 format. The steps to calculate the function are as follows:

1. Software processes the input parameter 1.0. 1.0 is divided by 2 (f=3'b001). The result is 0.5 and the q1.15 format is 0x4000.
2. The scaling factor f=3'b001 is written into FACTOR[2:0] bit-field in TMU_CS register.
3. The input data 0x4000 is written into TMU_IDATA. Then the TMU calculation starts.
4. When the ENDF flag is set to 1, reading the TMU_ODATA register can get the first output data: $y_1 = \frac{\sinh(1.0)}{2}$, and reading the TMU_ODATA register again can get the second output data:$y_2 = \frac{\cosh(1.0)}{2}$. The output data is q1.15 format.
5. Software processes the result. The two output datas are multiplied 2 to get the real hyperbolic cosine and sine of a hyperbolic angle 1.0.

Mode 7: $\tanh^{-1}(x)$

Mode 7 calculates calculates the hyperbolic arctangent of a hyperbolic angle x. This mode take one input data and generate one ouput data.

When IFLTEN = 1 and OFLTEN = 1, the detailed information refer to ***Table 14-19. Mode 7 description, when IFLTEN = 1 and OFLTEN = 1***.

Table 14-19. Mode 7 description, when IFLTEN = 1 and OFLTEN = 1

Parameter	Range	Description
Input data	$x \in [-0.806,0.806]$	Overflow when exceed the range.
Output data	$\tanh^{-1}(x) \in [-1.118,1.118]$	--
FACTOR[2:0]	Not available	keep at reset value 3'b000

When IFLTEN = 0 or OFLTEN = 0, the detailed information refer to ***Table 14-20. Mode 7 description, when IFLTEN = 0 or OFLTEN = 0***.

Table 14-20. Mode 7 description, when IFLTEN = 0 or OFLTEN = 0

Parameter	Range	Description
Input data	$\frac{x}{2} \in [-0.403,0.403]$	$x \in [-0.806,0.806]$, a scaling factor $\frac{1}{2}$ is applied in softwate. Then write $\frac{x}{2}$ to TMU_IDATA in q1.31,

Parameter	Range	Description
		q1.15 or float format.
Output data	$\frac{\tanh^{-1}(x)}{2} \in [-0.559,0.559]$	The output data is multiplied 2 to get the real hyperbolic arctangent of a hyperbolic angle x.
FACTOR[2:0]	1	The bit-field FACTOR[2:0] is configured as 3'b001

Note: When IFLTEN = 0 or OFLTEN = 0, the scaling factor FACTOR[2:0] must be 3'b001.

For example, calculating $\tanh^{-1}(0.5)$. When the IFLTEN and OFLTEN bits are reset, and IWIDTH and OWIDTH bits are set, the input and output are q1.15 format. The steps to calculate the function are as follows:

1. Software processes the input parameter 0.5. 0.5 is divided by 2 (f=3'b001). The result is 0.25 and the q1.15 format is 0x2000.
2. The scaling factor f=3'b001 is written into FACTOR[2:0] bit-field in TMU_CS register.
3. The input data 0x2000 is written into TMU_IDATA. Then the TMU calculation starts.
4. When the ENDF flag is set to 1, reading the TMU_ODATA register can get the output data: $y=\frac{\tanh^{-1}(0.5)}{2}$. The output data is q1.15 format.
5. Software processes the result. The output data is multiplied 2 to get the real hyperbolic arctangent of a hyperbolic angle 0.5.

Mode 8: ln (x)

Mode 8 calculates calculates the natural logarithm of the input parameter x. This mode takes one input data and generates one ouput data.

When IFLTEN = 1 and OFLTEN = 1, the detailed information refer to ***Table 14-21. Mode 8 description, when IFLTEN = 1 and OFLTEN = 1***.

Table 14-21. Mode 8 description, when IFLTEN = 1 and OFLTEN = 1

Parameter	Range	Description
Input data	$x \in [0.107,9.35]$	Overflow when exceed the range.
Output data	$\ln(x) \in (-2.235,2.235)$	--
FACTOR[2:0]	Not available	keep at reset value 3'b000

When IFLTEN = 0 or OFLTEN = 0, the detailed information refer to ***Table 14-22. Mode 8 description, when IFLTEN = 0 or OFLTEN = 0.***

Table 14-22. Mode 8 description, when IFLTEN = 0 or OFLTEN = 0

Parameter	Range	Description
Input data	$\frac{x}{2^f} \in [0.0535,0.875]$	$x \in [0.107,9.35]$, a scaling factor 2^{-f} is applied in softwate to ensure that $\frac{x}{2^f}<(1-\frac{1}{2^f})$. Then write $\frac{x}{2^f}$ to TMU_IDATA in q1.31, q1.15 or float format.

Parameter	Range	Description
Output data	$\frac{\ln(x)}{2^{(f+1)}} \in [-0.558, 0.137]$	The output data is multiplied $2^{(f+1)}$ to get the real $\ln(x)$.
FACTOR[2:0]	$f \in [1,4]$	The bit-field FACTOR[2:0] is configured as f

Note: When IFLTEN = 0 or OFLTEN = 0, the scaling factor FACTOR[2:0] needs to be configured.

For example, calculating $\ln(8)$. When the IFLTEN and OFLTEN bits are reset, and IWIDTH and OWIDTH bits are set, the input and output are q1.15 format. The steps to calculate the function are as follows:

1. Software processes the input parameter 8. 8 is divided by 16 (f=3'b100). The result is 0.5 and the q1.15 format is 0x4000.
2. The scaling factor f=3'b100 is written into FACTOR[2:0] bit-field in TMU_CS register.
3. The input data 0x4000 is written into TMU_IDATA. Then the TMU calculation starts.
4. When the ENDF flag is set to 1, reading the TMU_ODATA register can get the output data: $y=\frac{\ln(x)}{2^{(4+1)}}$. The output data is q1.15 format.
5. Software processes the result. The output data is multiplied $2^{(4+1)}$ to get the real the natural logarithm of 8.

In order to ensure the accuracy of calculation, it is recommended to use the scaling factors in ***Table 14-23. Recommended scaling factors in mode 8*** for different inputs.

Table 14-23. Recommended scaling factors in mode 8

Input Parameter x range	FACTOR[2:0]	Input data x range
0.107≤x<1	3'b001	[0.0535,0.5)
1≤x<3	3'b010	[0.25,0.75)
3≤x<7	3'b011	[0.375,0.875)
7≤x<9.35	3'b100	[0.4375,0.584)

Mode 9: $\sqrt{x}$

Mode 9 calculates the square root of the input parameter x. This mode takes one input data and generates one ouput data.

When IFLTEN = 1 and OFLTEN = 1, the detailed information refer to ***Table 14-24. Mode 9 description, when IFLTEN = 1 and OFLTEN = 1***.

Table 14-24. Mode 9 description, when IFLTEN = 1 and OFLTEN = 1

Parameter	Range	Description
Input data	$x \in [0,+\infty)$	Overflow when exceed the range.
Output data	$\sqrt{x} \in [0,+\infty)$	--
FACTOR[2:0]	Not available	keep at reset value 3'b000

When IFLTEN = 0 or OFLTEN = 0, the detailed information refer to ***Table 14-25. Mode 9***

description, when IFLTEN = 0 or OFLTEN = 0.

Table 14-25. Mode 9 description, when IFLTEN = 0 or OFLTEN = 0

Parameter	Range	Description
Input data	$\frac{x}{2^f} \in [0.027,0.875]$	x ∈ [0.027,2.34], a scaling factor 2^{-f} is applied in softwate to ensure that $\frac{x}{2^f} < (1-\frac{1}{2^{f+2}})$. Then write $\frac{x}{2^f}$ to TMU_IDATA in q1.31, q1.15 or float format.
Output data	$\frac{\sqrt{x}}{2^f} \in [0.164,0.866]$	The output data is multiplied 2^f to get the real $\sqrt{x}$.
FACTOR[2:0]	f∈[0,2]	The bit-field FACTOR[2:0] is configured as f

Note: When IFLTEN = 0 or OFLTEN = 0, the scaling factor FACTOR[2:0] needs to be configured.

For example, calculating $\sqrt{2}$. When the IFLTEN and OFLTEN bits are reset, and IWIDTH and OWIDTH bits are set, the input and output are q1.15 format. The steps to calculate the function are as follows:

1. Software processes the input parameter 2. 2 is divided by 4 (f=3'b010). The result is 0.5 and the q1.15 format is 0x4000.
2. The scaling factor f=3'b010 is written into FACTOR[2:0] bit-field in TMU_CS register.
3. The input data 0x4000 is written into TMU_IDATA. Then the TMU calculation starts.
4. When the ENDF flag is set to 1, reading the TMU_ODATA register can get the output data: $y=\frac{\sqrt{2}}{2^2}$. The output data is q1.15 format.
5. Software processes the result. The output data is multiplied 2^2 to get the real value of $\sqrt{2}$.

In order to ensure the accuracy of calculation, it is recommended to use the scaling factors in *Table 14-26. Recommended scaling factors in mode 9* for different inputs

Table 14-26. Recommended scaling factors in mode 9

Input Parameter x range	FACTOR[2:0]	Input data x range
0.027<x<0.75	3'b000	[0.027,0.75)
0.75≤x<1.75	3'b001	[0.375,0.875)
1.75≤x<2.341	3'b010	[0.4375,0.585)

14.4.4. TMU operation pending

If the TMU operation is ongoing, the further the contents written into TMU_IDATA register will be suspended. When the TMU operation completes (the result is read and the ENDF flag is cleared), if the number of suspended input data meets the configuration (defined in the suspended TMU_CS register), the TMU module will start a new TMU operation according to the pending configuration and data.

For example, if the configured TMU mode requires two 32-bit input data (IWIDTH = 0, INUM

= 1), the TMU operation is started as sonn as two input data are written into the TMU_IDATA. If the second input data does not change in the next TMU operation, the INUM bit can be set to 0 at this time. After the previous TMU operation is completed, only one input data is written into TMU_IDATA. The second input data still uses the previous value in the later TMU operation as long as the TMU mode does not change.

Note: After a reset, the second input data is +1 (0x7FFFFFFF).

While an TMU operation is pending, the further contents written into TMU_IDATA or TMU_CS register will cover the original contents. The new TMU contents will be suspended, and the suspension of the original contents will be invalid.

14.4.5. Zero-overhead mode

After a TMU operation starts, the output data register can be read directly. And the bus will automatically insert the waiting cycle before the result is returned. The following steps can be followed:

1. Configure TMU_CS register as needed.
2. Start the TMU operation by written the input data into TMU_IDATA.
3. Configure the next TMU mode as required and write the input data into TMU_IDATA.
4. Read the result from the TMU_ODATA register. The waiting cycles are automatically inserted into the bus. After reading TMU_ODATA operation is completed, the suspended TMU operation will start automatically.
5. Go to step3.

14.4.6. Interrupt and DMA requests

When ENDF flag is set to 1, if the RIE bit in TMU_CS register is set to 1, an interrupt request is generated. After the ENDF flag is cleared to 0, the interrupt request is also cleared.

When the OVRF flag is set to 1, if the OVRIE bit in TMU_CS register is set to 1, an interrupt request is generated. After the OVRF flag is cleared to 0, the interrupt request is also cleared.

If the WDEN bit in TMU_CS register is set to 1 and no TMU operation is pending, DMA request is generated. The number of DMA request depends on the INUM bit in TMU_CS register. If INUM is 0, only one DMA request is generated. If INUM is 1, two DMA requests are generated.

When ENDF flag is set to 1, if the RDEN bit in TMU_CS register is set to 1, DMA request is generated. The number of DMA request depends on the ONUM bit in TMU_CS register. If ONUM is 0, only one DMA request is generated. If ONUM is 1, two DMA requests are generated.

14.5. Registers definition

TMU base address: 0x4802 4400

14.5.1. Control and status register (TMU_CS)

Address offset: 0x00

Reset value: 0x0000 0050

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
ENDF	OVRF	OVRIE	Reserved						IWIDTH	OWIDTH	INUM	ONUM	WDEN	RDEN	RIE
r	rc_w1	rw							rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IFLTEN	OFLTEN	Reserved			FACTOR[2:0]			ITRTNUM[3:0]				MODE[3:0]			
rw	rw				rw			rw				rw			

Bits	Fields	Descriptions
31	ENDF	End of TMU operation flag 0: No TMU operation or TMU operation is ongoing 1: TMU operation ends and the output data has been written into TMU_ODATA register. This bit is set by hardware when TMU operation ends and the output data has been written into TMU_ODATA register. This bit is cleared by reading the TMU_ODATA register (ONUM + 1) times. **Note:** During this bit is set, no new TMU operation is started.
30	OVRF	The flag of overflow This bit is set by hardware, and cleared by software write 1. 0: No overflow 1: Overflow
29	OVRIE	Overflow interrupt enable 0: Overflow interrupt disable 1: Overflow interrupt enable
28:23	Reserved	Must be kept at reset value.
22	IWIDTH	Width of input data 0: 32-bit 1: 16-bit This bit decides the fixed point format of the input data. If 32-bit is configured, the input data in q1.31 format should be written into TMU_IDATA register. If 16-bit is configured, the input data in q1.15 format should be written into TMU_IDATA register. The first input data is written into the lower half-word , and the second input data is written into the upper half-word. If the TMU mode only needs one input data (INUM=0), the upper half-word of TMU_IDATA register is unused. **Note:** This bit-field takes effect if the input data is in the fixed point format (IFLTEN=0).

21	OWIDTH	Width of output data 0: 32-bit 1: 16-bit This bit decides the fixed point format of the output data. If 32-bit is configured, the TMU_ODATA contains the output data of TMU operation in q1.31 format. If 16-bit is configured, the TMU_ODATA contains the output data of TMU operation in q1.15 format. The the lower half-word of TMU_IDATA register contains the first output data, and the the upper half-word of TMU_IDATA register contains the second output data. **Note:** This bit-field takes effect if the output data is in the fixed point format (OFLTEN=0).
20	INUM	The number of times that the TMU_IDATA needs to be written 0: One 32-bit write operation. To start a new TMU operation, one 32-bit input data must be writen into TMU_IDATA register. 1: Two 32-bit write operation. To start a new TMU operation, two 32-bit input data must be written into TMU_IDATA register. **Note**: When the format of input data is q1.15 (IWIDTH=1, IFLTEN=0) and the TMU mode only needs one 32-bit write operation (INUM=0).
19	ONUM	The number of times that the TMU_ODATA needs to be read 0: One 32-bit read operation. When TMU operation completes, only one 32-bit result is transferred into TMU_ODATA register. Read the TMU_ODATA register once to clear ENDF flag. 1: Two 32-bit read operation. When TMU operation completes, two 32-bit results are transferred into TMU_ODATA register. Read the TMU_ODATA register twice to clear ENDF flag. **Note**: When the format of output data is q1.15 (OWIDTH=1, OFLTEN=0) and the TMU mode only needs one 32-bit read operation (ONUM=0).
18	WDEN	Enable DMA request to write TMU_IDATA 0: disabled. No DMA request is generated to write TMU_IDATA. 1: enabled. When no TMU operation is pending, the DMA request is generated.
17	RDEN	Enable DMA request to read TMU_ODATA 0: disabled. No DMA request is generated to read TMU_ODATA. 1: enabled. When ENDF is set, the DMA request is generated.
16	RIE	Enable interrupt request to read TMU_ODATA 0: disabled. No interrupt request is generated to read TMU_ODATA. 1: enabled. When ENDF is set, the interrupt request is generated.
15	IFLTEN	Input floating point format enable 0: Input data support fixed point signed integer format q1.15 or q1.31 1: Input data support 32-bit single precision floating point format

14	OFLTEN	Output floating point format enable 0: Output data support fixed point signed integer format q1.15 or q1.31 1: Output data support 32-bit single precision floating point format
13:11	Reserved	Must be kept at reset value.
10:8	FACTOR[2:0]	Scaling factor This bit-field defines the scaling factor: $2^{FACTOR[2:0]}$. 000: 2^0 001: 2^1 010: 2^2 … 110: 2^6 111: 2^7 When the actual input parameter exceeds the specified the input data range [-1,1), it is need to be divide by $2^{FACTOR[2:0]}$ and the output data is need to be multiplied by $2^{FACTOR[2:0]}$ to get the actual output result, details as follows: TMU_IDATA = the actual input parameter/$2^{FACTOR[2:0]}$ the actual output result=TMU_ODATA*$2^{FACTOR[2:0]}$. **Note:** For mode8 and mode9, this bit field recommends some configurations for different parameters. For mode0, mode1, mode2 and mode3, this bit field is recommended to be configured as 3'b000. For mode5, mode6, and mode7, this bit field is recommended to be configured as 3'b001.
7:4	ITRTNUM[3:0]	Number of iterations This bit-field defines the number of iterations: ITRTNUM[3:0]*4. 0000: Reserved 0001: 4 iteration steps 0010: 8 iteration steps … 0110: 24 iteration steps 0111~1111: Reserved **Note:** The higher the number of iterations, the higher the accuracy.
3:0	MODE[3:0]	Mode of TMU operation 0000: mode0,$m*\cos(\theta)$ 0001: mode1, $m*\sin(\theta)$ 0010: mode2, phase= $\text{atan2}(y,x)$ 0011: mode3, modulus=$\sqrt{x^2+y^2}$ 0100: mode4, $\tan^{-1}(x)$ 0101: mode5, $\cosh(x)$ 0110: mode6, $\sinh(x)$ 0111: mode7, $\tanh^{-1}(x)$ 1000: mode8, $\ln(x)$ 1001: mode9, $\sqrt{x}$

1010~1111: reserved

Note:

x,θ: the first input data

y,m: the second input data

14.5.2. Input data register (TMU_IDATA)

Address offset: 0x04

Reset value: 0xXXXX XXXX

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

Bits	Fields	Descriptions
31:0	IDATA[31:0]	The input data The input data is written into this register. For details, refer to ***Table 14-1. Input data configuration***. **Note:** 1. When no TMU operation is is ongoing and the required number of arguments has been written, a new operation will be started automatically. 2. When the TMU operation is ongoing, the written data is suspended until the end of the TMU operation and the output data is read. During this period, if new input data is written, the previous suspension will be cancelled, and the new data will overwrite the previous data and be suspended.

14.5.3. Output data register (TMU_ODATA)

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

Bits	Fields	Descriptions

31:0	ODATA[31:0]	The output data When TMU operation ends, the result is transferred into this register. For details, refer to ***Table 14-2. Output data configuration***. **Note:** 1. When the ENDF bit is 1, the result of TMU operation can be obtained by reading the register. 2. When the read operation meeting the configuration is completed, the ENDF bit is cleared.

15. Fast Fourier Transform (FFT)

15.1. Overview

The Fast Fourier Transform (FFT) is an efficient computation of the Discrete Fourier Transform (DFT). The module supports CPU to offload FFT operations. Compared to a software implementation, it can accelerate calculations and time critical tasks. The module supports 6 configruation FFT point number up to 1024, and input and output data should be IEEE-754 single precision float point complex number.

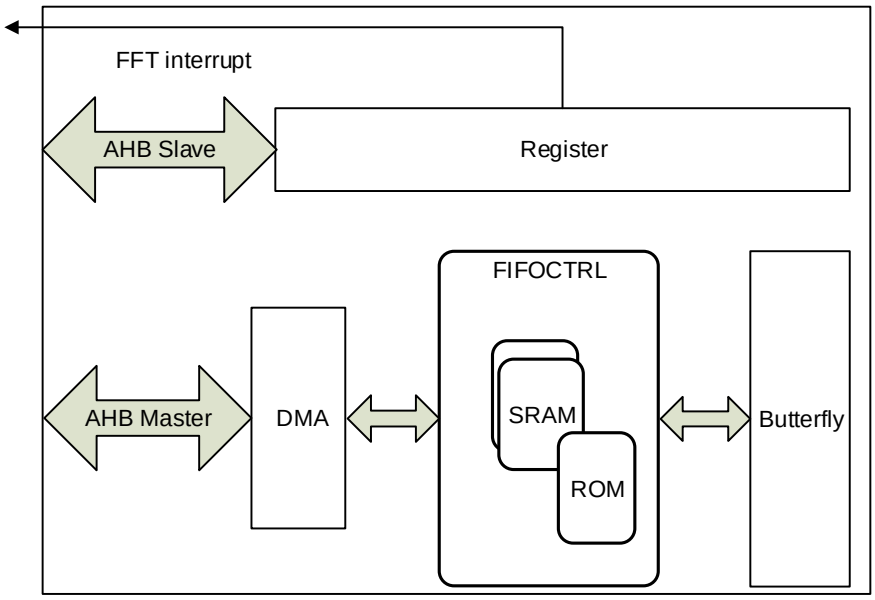
15.2. Characteristics

- Support 1024/512/256/128/64/32 points FFT
- Support IFFT mode
- IEEE-754 single precision float point complex number input and output data
- DMA master to load and store data
- Support window function configured in memory
- Support input down sample

15.3. Functional description

Figure 15-1. FFT module block diagram provides details of the internal configuration of the FFT module.

Figure 15-1. FFT module block diagram



The FFT module consists of DMA Master, FIFOCTRL submodule, Butterfly submodule and

register. The module is based on redix-2 FFT algorithm. The input data including real, image and window data in the format of IEEE754 single precision float point is loaded from memory, and after window function operation and input bits reverse operation they are written into internal SRAM. The rotation factor already be ready in ROM.

The butterfly is reused in FFT calculation, and calculate interatly (N / 2) * log_2N times. The calculation is same address calculation, that is, the output data of each calculation are write back to same address where the input data stored before.

The DMA master transfer the output data from internal SRAM to memory after all butterfly calculation interation finished.

15.4. Data format

The operation data and calculation result data format is given in ***Table 15-1. IEEE 32-Bit Single Precision Floating-Point Format***. They must meet IEEE 32-bit Single Precision Floating-Point.

Table 15-1. IEEE 32-Bit Single Precision Floating-Point Format

S [31]	E [30:23]	M [22:0]	Value (V)
0	0	0	Zero (V = 0)
1	0	0	Negative Zero (V = -0)
0 +ve 1 -ve	0	non zero	De-normalized (V=$(-1)^s$*$2^{(-126)}$* (0.M))
0 +ve 1 -ve	1 to 254	0 to 0x7FFFFF	Normal Range(V=$(-1)^s$*$2^{(E-127)}$* (1.M))
0	254	0x7FFFFF	Positive Max (V = +Max)
1	254	0x7FFFFF	Negative Max (V = -Max)
0	max=255	0	Positive Infinity (V = +Infinity)
1	max=255	0	Negative Infinity (V = -Infinity)
x	max=255	non zero	Not A Number (V = NaN)

The treatment of the various IEEE floating-point numerical formats for this module is given below:

De-Normalized Numbers: A de-normalized operand (E=0, M!=0) input is treated as zero (E=0, M=0).

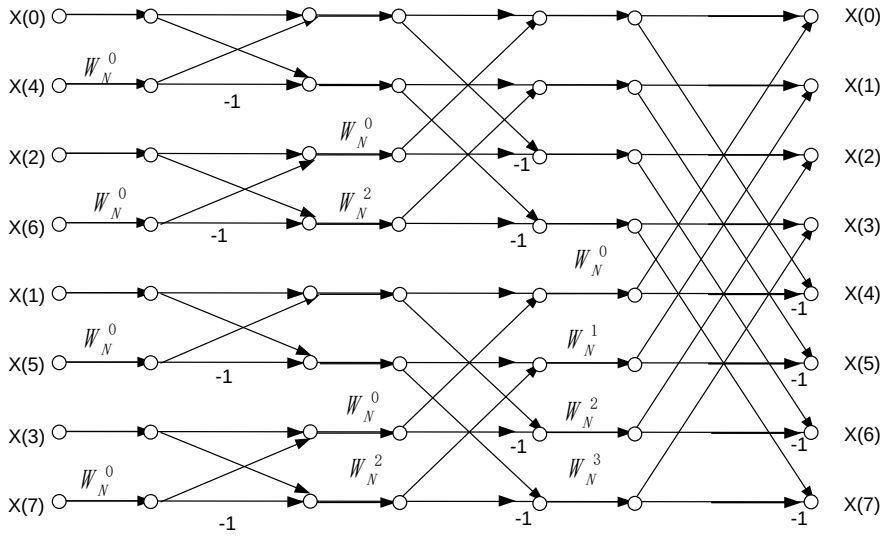
Overflow: Overflow occurs when an operation generates a value that is too large to represent in the given floating-point format.

Not a Number (NaN): An NaN operand (E=max, M!=0) input is treated as Infinity (E=max, M=0).

15.5. Radix-2 FFT

The FFT module uses the DIT(Decimation in Time) Radix-2 algorithms, which relies on the recursive decomposition of an N point transform into two (N/2) point transforms. For illustrative purposes, the 8 points (N=8) decimation in time algorithm is shown in the ***Figure 15-2 8 point DIT FFT flow digram***.

Figure 15-2 8 point DIT FFT flow digram



The 8 points input sequence occurs in bit-reversed order with respect to the output, as shown in ***Table 15-2. 8 points input sequence bits reverse operation***.

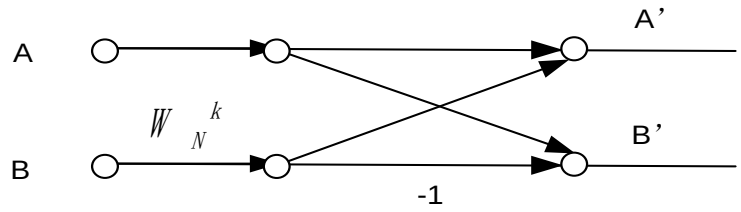
Table 15-2. 8 points input sequence bits reverse operation

index	Binary of index	Binary of bit-reversed index	bit-reversed index
0	000	000	0
1	001	100	4
2	010	010	2
3	011	110	6
4	100	001	1
5	101	101	5
6	110	011	3
7	111	111	7

15.6. Butterfly unit

The butterfly unit is the basic element of FFT calculation, and the progress are shown in the ***Figure 15-3. 8 points DIT FFT flow digram***.

Figure 15-3. 8 points DIT FFT flow digram



$$A' = A + BW = Re(A) + Re(BW) + j[Im(A) + Im(BW)]$$

$$B' = A - BW = Re(A) - Re(BW) + j[Im(A) - Im(BW)]$$

Where $W = W_N^k = e^{-j\frac{2\pi}{N}k}$

15.7. IFFT mode

IFFT converts frequency domain vector signal to time domain vector signal, and the main difference of realization between FFT and IFFT are:

- The twiddle factor conjugated.
- The result divide N, where N is num of FFT point.

15.8. FFT SRAM

The 8KB space of FFT SRAM(0x40025800~0x40027FFF) can be accessed by AHB when FFTEN in the FFT_CSR register is 0.

Note:

- When FFTEN in the FFT_CSR register is 1, a bus fault will occur if accessing FFT SRAM by AHB.
- If the FFT module has never been enabled, this FFT SRAM can be used for other purposes.

15.9. FFT loop mode

The FFT support loop mode. Each time FFTEN is set, one FFT operation is started, and the INDEX[15:0] is increased by 1 after FFT opration done.

The real part and image part start address increase and point to next input datas after every FFT opration done. That is, the INDEX[15:0] in FFT_LOOPLEN register increase 1 after every FFT opration done. The real part start address is FFT_RESADDR + {INDEX[15:0], 2'b00}, and the image part start address is FFT_IMSADDR + {INDEX[15:0], 2'b00}. When increasing to LENGTH[15:0], the INDEX[15:0] in FFT_LOOPLEN register clear to 0.

Note: The input data should be ready in corresponding address.

15.10. Operation guide

This section describes the advised operation guide for FFT.

1. Configure the FFT_IMSADDR to set the FFT image start address, if necessary, that is, the image data is not zero.
2. Configure the FFT_RESADDR to set the FFT real start address.
3. Configure the WINEN and FFT_WSADDR to set the FFT window start address, if necessary.
4. Configure the FFT_OSADDR to set the FFT output start address.
5. Configure the FFT_LOOPLEN, if necessary.
6. Configure the IFFTMODE bit, if necessary.
7. Configure the NUMPT[2:0] bits in the FFT_CSR register to set number of FFT points.
8. Configure FFTEN.
9. Wait FFTEN clear, or CCF set.

15.11. FFT interrupts

The interrupt can be produced on one of the flags:

- FFT calculation completion flag
- Transfer access error interrupt flag

These interrupts of FFT are mapped into the same interrupt vector IRQ96.

15.12. Register definition

FFT start address: 0x4002 5000

15.12.1. Control and status register (FFT_CSR)

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
DMABSY	CCF	CCIE	TAEIF	TAEIE	Reserved										
r	rc_w1	rw	rc_w1	rw											

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	IMSEL[1:0]		DOWNSAMP[3:0]				WINEN	Reserved			IFFTMODE	NUMPT[2:0]			FFTEN
	rw		rw				rw				rw	rw			rw

Bits	Fields	Descriptions
31	DMABSY	DMA Busy This bit reports that whether DMA is busy. This bit is read-only. 0: DMA is not busy 1: DMA is busy
30	CCF	FFT calculation completion flag The bit is cleared to 0 by writing 1. 0: FFT calculation is on going 1: FFT calculation completion
29	CCIE	FFT calculation completion interrupt enable 0: Disable calculation completed interrupt 1: Enable calculation completed interrupt
28	TAEIF	Transfer access error interrupt flag The bit is cleared to 0 by writing 1. 0: No transfer access error is detected. 1: A transfer access error is detected.
27	TAEIE	Transfer access error interrupt enable 0: Disable transfer access error interrupt 1: Enable transfer access error interrupt
26:15	Reserved	Must be kept at reset value
14:13	IMSEL[1:0]	Image input source select 00: Image input is from FFT_IMSADDR

		01: Image input is 0 10: Image input is the opposite sign value of image data from FFT_IMSADDR
12:9	DOWNSAMP[3:0]	Input data down sample selection 0000: Down sample 1 0001: Down sample 2 0010: Down sample 3 0011: Down sample 4 0100: Down sample 5 0101: Down sample 6 0110: Down sample 7 0111: Down sample 8 1000: Down sample 9 1001: Down sample 10 1010: Down sample 11 1011: Down sample 12 1100: Down sample 13 1101: Down sample 14 1110: Down sample 15 1111: Down sample 16 This bit can not be written when FFTEN in the FFT_CSR register is '1'.
8	WINEN	Window function enable 0: No window function 1: Window function from FFT_WSADDR This bit can not be written when FFTEN in the FFT_CSR register is '1'.
7:5	Reserved	Must be kept at reset value
4	IFFTMODE	IFFT mode enable 0: FFT mode 1: IFFT mode This bit can not be written when FFTEN in the FFT_CSR register is '1'.
3:1	NUMPT[2:0]	Number of FFT Point 000:32 001:64 010:128 011:256 100:512 101:1024 Others: Reserved This bit can not be written when FFTEN in the FFT_CSR register is '1'.
0	FFTEN	FFT enable When the FFT calculation end, this bit will be cleared automatically. Software shall

not clear this bit.

0: Disable FFT

1: Enable FFT

15.12.2. Real start address register(FFT_RESADDR)

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

Bits	Fields	Descriptions
31:0	RESADDR[31:0]	FFT real start address The address must be aligned to 32-bit. These bits can not be written when FFTEN in the FFT_CSR register is ‘1’.

15.12.3. Image start address register (FFT_IMSADDR)

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

Bits	Fields	Descriptions
31:0	IMSADDR[31:0]	FFT image start address The address must be aligned to 32-bit. These bits can not be written when FFTEN in the FFT_CSR register is ‘1’.

15.12.4. Window start address register (FFT_WSADDR)

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
FFT_WSADDR[31:16]															
rw															

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

Bits	Fields	Descriptions
31:0	FFT_WSADDR[31:0]	FFT window start address The address must be aligned to 32-bit. These bits can not be written when FFTEN in the FFT_CSR register is ‘1’.

15.12.5. Output start address register (FFT_OSADDR)

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
FFT_OSADDR[31:16]															
rw															

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

Bits	Fields	Descriptions
31:0	FFT_OSADDR[31:0]	FFT output start address The address must be aligned to 32-bit. These bits can not be written when FFTEN in the FFT_CSR register is ‘1’.

15.12.6. Loop length register (FFT_LOOPLEN)

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
INDEX[15:0]															
rw															

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

Bits	Fields	Descriptions
31:16	INDEX[15:0]	The index of DMA loop buffer The index can not more than LENGTH[15:0] and it ranges from 0 to LENGTH[15:0]. Add 1 at each end and clear 0 when increasing to LENGTH[15:0]. The DMA first address of actual real part is FFT_RESADDR + {INDEX[15:0], 2'b00}. The DMA first address of actual imaginary is FFT_IMSADDR + {INDEX[15:0], 2'b00}.
15:0	LENGTH[15:0]	The DMA loop buffer length of the FFT input data. These bits can not be written when FFTEN in the FFT_CSR register is '1'.

16. Debug (DBG)

16.1. Overview

The GD32G553 series provide a large variety of debug, trace and test features. They are implemented with a standard configuration of the Arm® CoreSight™ module together with a daisy chained standard TAP controller. Debug and trace functions are integrated into the ARM® Cortex®-M33. The debug system supports serial wire debug (SWD) and trace functions in addition to standard JTAG debug. The debug and trace functions refer to the following documents:

- Cortex®-M33 Technical Reference Manual
- ARM® Debug Interface v5 Architecture Specification

The DBG hold unit helps debugger to debug in power saving mode. When corresponding bit is set, debug module provide clock when in power saving mode or hold the state for TIMER, WWDGT, FWDGT, I2C, RTC, CAN, LPTIMER, HRTIMER.

16.2. JTAG / SW function overview

Debug capabilities can be accessed by a debug tool via serial wire (SW - Debug Port) or JTAG interface (JTAG - Debug Port).

16.2.1. 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.

16.2.2. Pin assignment

The JTAG interface provides 5-pin standard JTAG, known as JTAG clock (JTCK), JTAG mode selection (JTMS), JTAG data input (JTDI), JTAG data output (JTDO) and JTAG reset (NJTRST, active low). The serial wire debug (SWD) provide 2-pin SW interface, known as SW data input/output (SWDIO) and SW clock (SWCLK). The two SW pin are multiplexed with two of five JTAG pin, which is SWDIO multiplexed with JTMS, SWCLK multiplexed with JTCK. The JTDO is also used as trace async data output (TRACESWO) when async trace enabled.

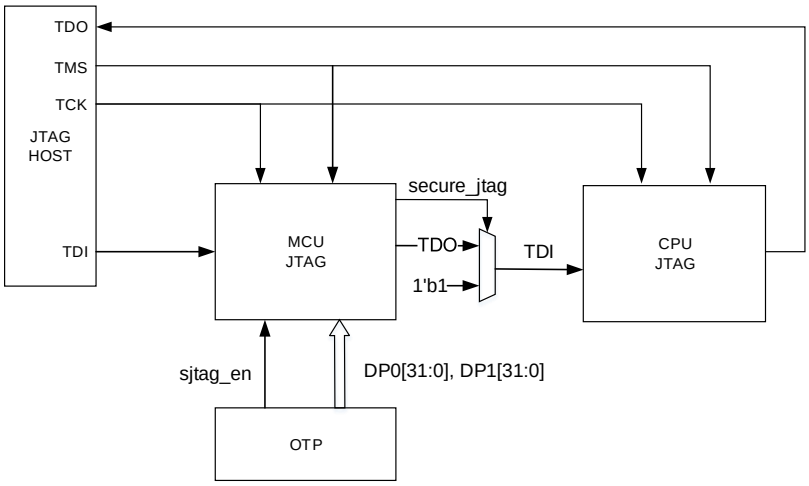
Table 16-1. Pin assignment

Pin	Debug interface
PA15	JTDI
PA14	JTCK / SWCLK
PA13	JTMS / SWDIO
PB3	JTDO
PB4	NJTRST

By default, 5-pin standard JTAG debug mode is chosen after reset. Users can also use JTAG function without NJTRST pin, then the PB4 can be used to other GPIO functions (NJTRST tied to 1 by hardware). If switch to SW debug mode, the PA15 / PB3 / PB4 are released to other GPIO functions. If JTAG and SW not used, all 5-pin can be released to other GPIO functions. Please refer to ***General-purpose and alternate-function I/Os (GPIO and AFIO)***.

16.2.3. JTAG

Figure 16-1. Block diagram of JTAG unit



JTAG daisy chained structure

The Cortex®-M33 JTAG TAP is connected to a boundary-scan (BSD) JTAG TAP. The BSD JTAG IR is 5-bit width, while the Cortec-M33 JTAG IR is 4-bit width. So when JTAG in IR shift step, it first shifts 5-bit BYPASS instruction (5'b 11111) for BSD JTAG, and then shift normal 4-bit instruction for Cortex®-M33 JTAG. Because of the data shift under BSD JTAG BYPASS mode, adding 1 extra bit to the data chain is needed.

The BSD JTAG IDCODE is 0x790007A3.

Secure JTAG

1. Secure JTAG only supports JTAG but not SW
2. OTP configuration:

OTP related bits: SWEN, NDBG[1:0], DPx[31:0](x=0,1)

Table 16-2. OTP JTAG bytes

Address	Name	abbreviations	Description
0x1fff f000	DP0[31:0]	w	secure JTAG password 0 Factor value: 0xFFFFFFFF
0x1fff f004	DP1[31:0]	w	secure JTAG password 1 Factor value: 0xFFFFFFFF
0x1fff f008	OTP_USER[31:0]	rw	[31:16]: LK Key value 0x3cc3 to lock OTP JTAG bytes, others values unlock Factor value: 0xFFFF [15:3]: Reserved [2:1]: NDBG 0x: No debug 10: Secure JTAG 11: No secure JTAG (factor value) [0]: SWEN 0: SW disable 1: SW enable (factor value)
0x1fff f00c	-	-	Reserved

Note: OTP JTAG bytes keep the factory value full FFFF. After LK[31:16] key value 0x3cc3 is set, the value of the OTP JTAG bytes cannot be modified. If the OTP JTAG bytes value has been modified, setting the LK[31:16] key value to 0x3cc3 is useless.

3. Using Secure JTAG

a) Configure OTP for secure JTAG mode: configure DPx[31:0] as the JTAG secure password, configure NDBG[1:0]= 2b'10' as secure JTAG.

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 following two passwords to the MCU JTAG to release the security mode. At this time, secure_jtag is 0, and the CPU can be operated through JTAG.

Write 5'b10101 to IR, Write DP0[31:0] to DR

Write 5'b10110 to IR, Write DP1[31:0] to DR

Note:

1 If the password is entered incorrectly, a power reset is required.

2 Any wrong input sequence occurrs, a power reset is required to re-decrypt.

d) Get written value and JTAG status:

Get the written value and JTAG status via JTAG:

Write 5'b11000 to IR, read value from DR: It can be judged whether the read IR value is the written 5'b11000

Write 5'b11001 to IR, read value from DR: It can be judged whether the read IR value is the written 5'b10110.

Write 5'b11010 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.

16.2.4. Debug reset

The JTAG-DP and SW-DP registers are in the power on reset domain. The system reset initializes the majority of the Cortex®-M33, excluding NVIC and debug logic, (FPB, DWT, and ITM). The NJTRST reset can reset JTAG TAP controller only. So, it can perform debug feature under system reset. Such as, halt-after-reset, which is the debugger sets halt under system reset, and the core halts immediately after the system reset is released.

16.2.5. JEDEC-106 ID code

The Cortex®-M33 integrates JEDEC-106 ID code, which is located in ROM table and mapped on the address of 0xE00FF000_0xE00FFFFF.

16.3. Debug hold function overview

16.3.1. Debug support for power saving mode

When STB_HOLD bit in DBG control register (DBG_CTL0) is set and entering the standby mode, the clock of AHB bus and system clock are provided by CK_IRC8M, and the debugger can debug in standby mode. When exit the standby mode, a system reset generated.

When DSLP_HOLD bit in DBG control register (DBG_CTL0) is set and entering the deep-

sleep mode, the clock of AHB bus and system clock are provided by CK_IRC8M, and the debugger can debug in deep-sleep mode.

When SLP_HOLD bit in DBG control register (DBG_CTL0) is set and entering the sleep mode, the clock of AHB bus for CPU is not closed, and the debugger can debug in sleep mode.

16.3.2. Debug support for TIMER, I2C, WWDGT, FWDGT, RTC, CAN, LPTIMER and HRTIMER

When the core halted and the corresponding bit in DBG control register x (DBG_CTLx, x=0, 1, 2) is set, the following behaved.

For TIMER/LPTIMER/HRTIMER, the timer counters stopped and hold for debug.

For I2C, SMBUS timeout hold for debug.

For WWDGT or FWDGT, the counter clock stopped for debug.

For RTC, the counter clock stopped for debug.

For CAN, the receive register stopped counting for debug.

16.4. Register definition

DBG base address: 0xE004 4000

16.4.1. ID code register (DBG_ID)

Address offset: 0x00
Reset value: 0xXXXX XXXX

This register attribute is read-only.

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

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

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

Bits	Fields	Descriptions
31:0	ID_CODE[31:0]	DBG ID code register These bits read by software. These bits are unchanged constant.

16.4.2. Control register0 (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															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										TRACE_IOEN	Reserved		STB_HOLD	DSLP_HOLD	SLP_HOLD
										rw			rw	rw	rw

Bits	Fields	Descriptions
31:6	Reserved	Must be kept at reset value.
5	TRACE_IOEN	Trace pin allocation enable This bit is set and reset by software. 0: Trace pin allocation disable 1: Trace pin allocation enable.
4:3	Reserved	Must be kept at reset value.

2	STB_HOLD	Standby mode hold bit This bit is set and reset by software. 0: No effect 1: In the standby mode, all active clocks continue to run, the debugger can debug in 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, all active clocks continue to run, the debugger can debug in deep-sleep mode.
0	SLP_HOLD	Sleep mode hold bit This bit is set and reset by software. 0: No effect 1: In the seep mode, all active clocks and oscillators continue to run, the debugger can debug in deep-sleep mode.

16.4.3. Control register1 (DBG_CTL1)

Address offset: 0x08
Reset value: 0x0000 0000, power reset only

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LPTIMER_HOLD	Reserved						I2C3_HOLD	I2C2_HOLD	I2C1_HOLD	I2C0_HOLD	Reserved				
rw							rw	rw	rw	rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			FWDGT_HOLD	WWDGT_HOLD	RTC_HOLD	Reserved				TIMER6_HOLD	TIMER5_HOLD	TIMER4_HOLD	TIMER3_HOLD	TIMER2_HOLD	TIMER1_HOLD
			rw	rw	rw					rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31	LPTIMER_HOLD	LPTIMER hold bit This bit is set and reset by software. 0: No effect 1: Hold the LPTIMER counter clock for debug when core is halted.
30:25	Reserved	Must be kept at reset value.
24	I2C3_HOLD	I2C3 hold bit This bit is set and reset by software. 0: No effect 1: Hold the I2C3 SMBUS timeout for debug when core is halted.
23	I2C2_HOLD	I2C2 hold bit

		This bit is set and reset by software. 0: No effect 1: Hold the I2C2 SMBUS timeout for debug when core is halted.
22	I2C1_HOLD	I2C1 hold bit This bit is set and reset by software. 0: No effect 1: Hold the I2C1 SMBUS timeout for debug when core is halted.
21	I2C0_HOLD	I2C0 hold bit This bit is set and reset by software. 0: No effect 1: Hold the I2C0 SMBUS timeout for debug when core is halted.
20:13	Reserved	Must be kept at reset value.
12	FWDGT_HOLD	FWDGT hold bit This bit is set and reset by software. 0: No effect 1: Hold the FWDGT counter clock for debug when core is halted.
11	WWDGT_HOLD	WWDGT hold bit This bit is set and reset by software. 0: No effect 1: Hold the WWDGT counter clock for debug when core is halted.
10	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 is halted.
9:6	Reserved	Must be kept at reset value.
5	TIMER6_HOLD	TIMER6 hold bit This bit is set and reset by software. 0: No effect 1: Hold the TIMER6 counter clock for debug when core is halted.
4	TIMER5_HOLD	TIMER5 hold bit This bit is set and reset by software. 0: No effect 1: Hold the TIMER5 counter clock for debug when core is halted.
3	TIMER4_HOLD	TIMER4 hold bit This bit is set and reset by software. 0: No effect 1: Hold the TIMER4 counter clock for debug when core is halted.
2	TIMER3_HOLD	TIMER3 hold bit

		This bit is set and reset by software. 0: No effect 1: Hold the TIMER3 counter clock for debug when core is halted.
1	TIMER2_HOLD	TIMER2 hold bit This bit is set and reset by software. 0: No effect 1: Hold the TIMER2 counter clock for debug when core is halted.
0	TIMER1_HOLD	TIMER1 hold bit This bit is set and reset by software. 0: No effect 1: Hold the TIMER1 counter clock for debug when core is halted.

16.4.4. Control register2 (DBG_CTL2)

Address offset: 0x0C
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					HRTIMER_HOLD	Reserved					TIMER19_HOLD	Reserved	TIMER16_HOLD	TIMER15_HOLD	TIMER14_HOLD
					rw						rw		rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		TIMER7_HOLD	Reserved	TIMER0_HOLD	Reserved								CAN2_HOLD	CAN1_HOLD	CAN0_HOLD
		rw		rw									rw	rw	rw

Bits	Fields	Descriptions
31:27	Reserved	Must be kept at reset value.
26	HRTIMER_HOLD	HRTIMER hold bit This bit is set and reset by software. 0: No effect 1: Hold the HRTIMER timeout for debug when core is halted.
25:21	Reserved	Must be kept at reset value.
20	TIMER19_HOLD	TIMER19 hold bit This bit is set and reset by software. 0: No effect 1: Hold the TIMER19 counter clock for debug when core is halted.
19	Reserved	Must be kept at reset value.
18	TIMER16_HOLD	TIMER16 hold bit This bit is set and reset by software.

		0: No effect 1: Hold the TIMER16 counter clock for debug when core is halted.
17	TIMER15_HOLD	TIMER15 hold bit This bit is set and reset by software. 0: No effect 1: Hold the TIMER15 counter clock for debug when core is halted.
16	TIMER14_HOLD	TIMER14 hold bit This bit is set and reset by software. 0: No effect 1: Hold the TIMER14 counter clock for debug when core is halted.
15:14	Reserved	Must be kept at reset value.
13	TIMER7_HOLD	TIMER7 hold bit This bit is set and reset by software. 0: No effect 1: Hold the TIMER7 counter clock for debug when core is halted.
12	Reserved	Must be kept at reset value.
11	TIMER0_HOLD	TIMER0 hold bit This bit is set and reset by software. 0: No effect 1: Hold the TIMER0 counter for debug when core is halted.
10:3	Reserved	Must be kept at reset value.
2	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 is halted.
1	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 is halted.
0	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 is halted.

17. Analog-to-digital converter (ADC)

17.1. Overview

A 12-bit successive approximation analog-to-digital converter module(ADC) is integrated on the MCU chip. ADC0 has 14 external channels, 5 internal channels (temperature sensor, the battery voltage, DAC0_OUT0, DAC0_OUT1, V_{REFINT} inputs channel), ADC1 has 16 external channels, 3 internal channels (DAC1_OUT0, DAC1_OUT1, V_{REFINT} inputs channel), ADC2 has 15 external channels, 5 internal channels (V_{REFINT} inputs channel, DAC2_OUT0, DAC2_OUT1, high-precision temperature sensor, the battery voltage), ADC3 has 18 external channels, 3 internal channels (DAC3_OUT0, DAC3_OUT1, V_{REFINT} inputs channel). 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(MSB) bit alignment. An on-chip hardware oversample scheme improves performances and reduces the computational burden of MCU.

17.2. Characteristics

- High performance:
  - ADC sampling resolution: 12-bit, 10-bit, 8-bit or 6-bit configurable resolution.
  - ADC sampling rate: 5.3 MSPs for 12-bit resolution, 6.15 MSPs for 10-bit resolution, 7.27 MSPs for 8-bit resolution, 8.89 MSPs for 6-bit resolution, faster sampling rate can be obtained by lowering the resolution.
  - Self-calibration time: 902 ADC clock periods.
  - Programmable sampling time.
  - Data storage mode: the most significant bit and the least significant bit
  - DMA support.
- Analog input channels:
  - 14 external analog inputs in ADC0, 16 external analog inputs in ADC1, 15 external analog inputs in ADC2, 18 external analog inputs in ADC3.
  - 1 channel for internal temperature sensor (V_{SENSE}).
  - 1 channel for internal reference voltage (V_{REFINT}).
  - 1 channel for external battery power supply pin (V_{BAT}).
  - 1 channel for internal high-precision temperature sensor (V_{SENSE2}).
  - Connection to DAC internal channels.
- 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 three ADCs).
- Conversion result threshold monitor function: analog watchdog.
- Interrupt generation at the end of routine sequence conversions, in case of analog watchdog event and overflow event.
- Oversampling:
  - 32-bit data register.
  - Oversampling ratio arbitrarily adjustable from 2x to 1024x.
  - Programmable data shift up to 11-bit.
- Channel input range: $V_{REFN} \leq V_{IN} \leq V_{REFP}$.
- Data can be routed to HPDF for post processing.

17.3. Pins and internal signals

Figure 17-1. ADC module block diagram shows the ADC block diagram. *Table 17-1. ADC internal input signals* and *Table 17-2. ADC input pins definition* give the ADC internal signals and pins description.

Table 17-1. ADC internal input signals

Internal signal name	Description
V_{SENSE}	Internal temperature sensor output voltage
V_{SENSE2}	Internal high-precision temperature sensor output voltage
V_{REFINT}	Internal voltage reference output voltage
V_{BAT}	External battery voltage

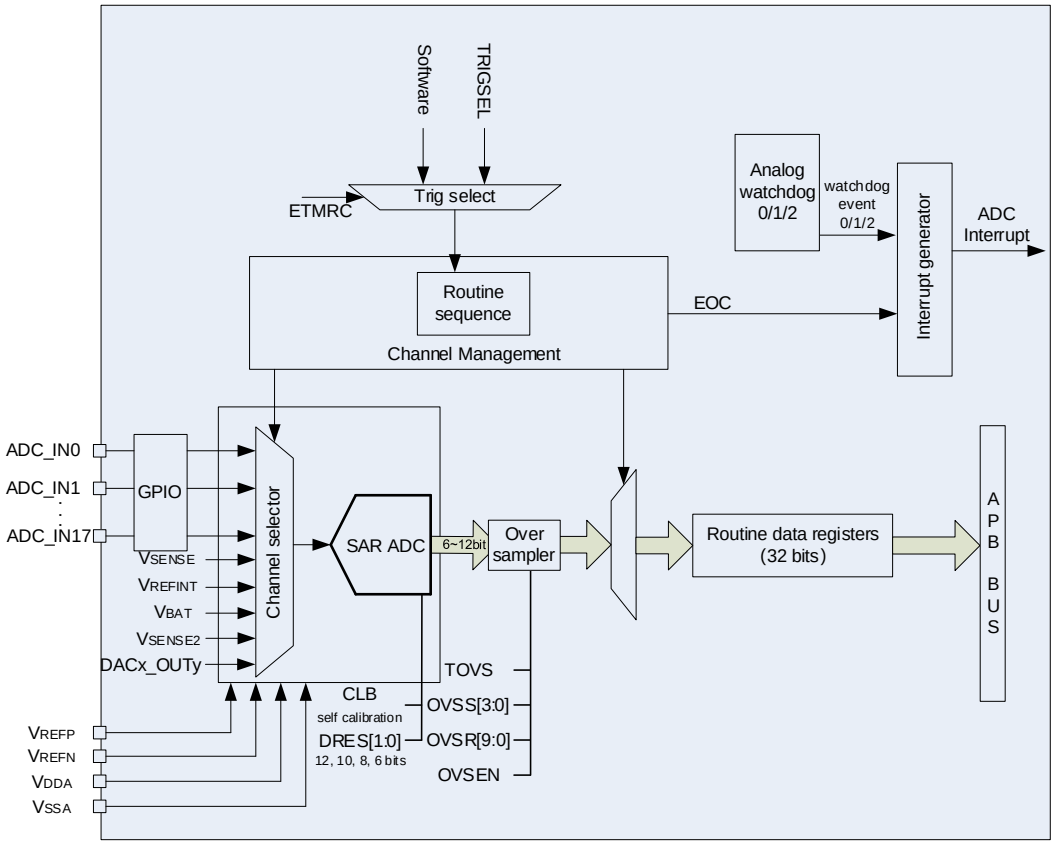
Table 17-2. ADC input pins definition

Name	Description
V_{DDA}	Analog power supply equal to V_{DD}
V_{SSA}	Ground for analog power supply equal to V_{SS}
V_{REFP}	The positive reference voltage for the ADC
V_{REFN}	The negative reference voltage for the ADC
ADCx_IN[17:0]	Up to 18 external channels

Note: V_{DDA} and V_{SSA} have to be connected to V_{DD} and V_{SS}, respectively.

17.4. Function overview

Figure 17-1. ADC module block diagram



17.4.1. Foreground calibration function

During the foreground calibration procedure, the ADC calculates a calibration factor which is internally applied to the ADC until the next ADC power-off. The application must not use the ADC during calibration and must wait until it is completed. Calibration should be performed before starting A/D conversion. The calibration is initiated by software by setting bit CLB=1. CLB bit stays at 1 during all the calibration sequence. It is then cleared by hardware as soon as the calibration is completed.

When the ADC operating conditions change (such as supply power voltage V_{DDA}, positive reference voltage V_{REFP}, temperature and so on), it is recommended to re-run a calibration cycle.

The internal analog calibration can be reset by setting the RSTCLB bit in ADC_CTL1 register.

Calibration software procedure:

1. Ensure that ADCON=1.
2. Delay 14 CK_ADC to wait for ADC stability.
3. Set RSTCLB (optional).

4. Set CLB=1.
5. Wait until CLB=0.

17.4.2. Dual clock domain architecture

The CK_ADC clock provided by the clock controller is synchronous with the AHB clock. In this mode, ADCSCK[3:0] in ADC_SYNCCTL should be set different from 0000.The divide factor can be 2, 4, 6, 8, 10, 12, 14,16. Thus the maximum frequency is 80 MHz.

The CK_ADC can also be feed by CK_PLLR or CK_SYS, which can be asynchronous and independent from the AHB clock. In this mode, ADCSCK[3:0] in ADC_SYNCCTL should be set to 0000. The divide factor can be configured through to ADCCK[3:0] of ADC_SYNCCTL.

The RCU controller has a dedicated programmable prescaler for the ADC clock.

Note: The ADC1 and ADC2 clock shares the ADC0 clock. When using the ADC1 and ADC2, ADC0 clock needs to be opened, and the clock frequency division can only be configured through ADC0.

17.4.3. ADC enable

The ADCON bit on the ADC_CTL1 register is the enable switch of the ADC module. The ADC module will keep in reset state if this bit is 0. For power saving, when this bit is reset, the analog sub-module will be put into power off mode. After ADC is enabled, you need delay t_{SU} time for sampling, the value of t_{SU} please refer to the device datasheet.

17.4.4. Single-enden and differential input channels

By writing to bits DIFCTL[21:0] in the ADC_DIFCTL register, the user can configure channels as differential input or single-ended input, and the ADC must be disabled (ADCON = 0) when the user configures these bits.

The channel n voltage is the difference between positive input and negative input. The positive input is external voltage V_{INn}, and there is a difference of the negative input between single-ended mode and differential input mode. In single-ended input mode, the negative input is V_{REFN}, in differential input mode, the negative input is V_{INm}. Differential channel pin shown in ***Table 17-3. ADC differential channel pin matching***.

Table 17-3. ADC differential channel pin matching

Differential channel n number	ADC0		ADC1		ADC2		ADC3	
	V_{INn}	V_{INm}	V_{INn}	V_{INm}	V_{INn}	V_{INm}	V_{INn}	V_{INm}
0	PA0	PA1	PA0	PA1	PB1	PE9	PE14	PE15
1	PA1	PA2	PA1	PA6	PE9	PE13	PE15	PB12
2	PA2	PA3	PA6	PA7	PE13	PE7	PB12	PB14
3	PA3	PB14	PA7	PC4	PE7	PB13	PB14	PB15

4	PB14	PC0	PC4	PC0	PB13	PE8	PB15	PE8
5	PC0	PC1	PC0	PC1	PE8	PD10	PE8	PD10
6	PC1	PC2	PC1	PC2	PD10	PD11	PD10	PD11
7	PC2	PC3	PC2	PC3	PD11	PD12	PD11	PD12
8	PC3	PF0	PC3	PF1	PD12	PD13	PD12	PD13
9	PF0	PB12	PF1	PC5	PD13	PD14	PD13	PD14
10	PB12	PB1	PC5	PB2	PD14	PB0	PD14	PD8
11	PB1	PB0	PB2	PA5	PE11	PE10	PD8	PD9
12	PB0	PB1	PA5	PB11	PE10	PE11	PD9	PE10
13	PB11	PB0	PB11	PB15	PE12	PE11	PE10	PE11
14	PB12	PB11	PA5	PA4	PB1	PB0	PE11	PE12
15	PB11	PB12	PA4	PA5	PB0	PB1	PE12	PE14
16	PC1	PC0	PC1	PC0	PE10	PE9	PA8	PA9
17	PC2	PC1	PC2	PC1	PE9	PE10	PA9	PA8
18	PC3	PC2	PC3	PC2	PD14	PD13	PB15	PD8
19	PA1	PA0	PA1	PA0	PD13	PD12	PD8	PB15
20	PA2	PA1	PC4	PA7	PD12	PD11	PD9	PD8
21	PA3	PA2	PA7	PA6	PD11	PD10	PE15	PE14

When the channel is used in differential input mode, the input voltages should be differential signals (common mode voltage is V_{REFP}/ 2), and the input ranges are still (V_{REFN} ~ V_{REFP}).

Taking the right-aligned, 12-bit resolution as an example:

1) When V_{INn} is V_{REFP} and V_{INm} is V_{REFN}, the conversion result of channel n is 0x0FFF;
2) When V_{INn} is V_{REFN} and V_{INm} is V_{REFP}, The conversion result of channel n is 0x0000;
3) When V_{INn} is V_{REFP} / 2 and V_{INm} is V_{REFP} / 2, the conversion result of channel n is 0x07FF.

D_{out} is the conversion result of channel n, then the differential voltage is:

$$V_{INn}-V_{INm} = V_{REFP}*(2*D_{out}/4095-1) \quad (17\text{-}1)$$

17.4.5. Routine sequence

The channel management circuit can organize the sampling conversion channels into a sequence: routine sequence. The routine sequence supports up to 22 channels, and each channel is called routine channel.

The ADC_RSQ0~ADC_RSQ8 registers specify the selected channels of the routine sequence. The RL[3:0] bits in the ADC_RSQ0 register specify the total conversion sequence length.

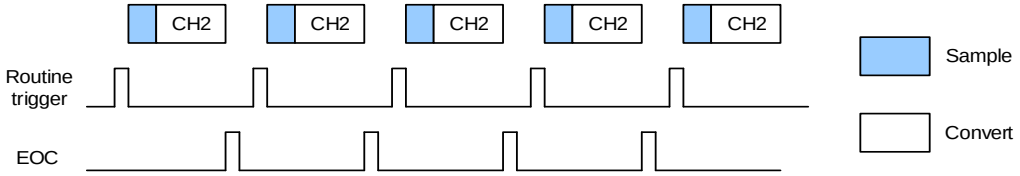
Note: Although the ADC supports 22 multiplexed channels, the maximum length of the sequence is only 16.

17.4.6. Operation modes

Single operation mode

In the single operation mode, the ADC performs conversion on the channel specified in the RSQ0[4:0] bits of ADC_RSQ8 register at a routine trigger. 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 17-2. Single operation mode



After conversion of a single routine channel, the conversion data will be stored in the ADC_RDATA register, the EOC will be set. An interrupt will be generated if the EOCIE bit is set.

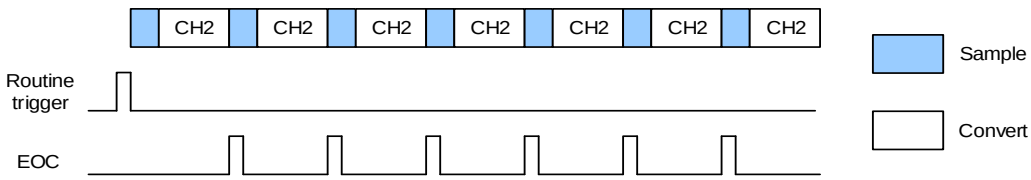
Software procedure for a single operation mode of a routine channel:

1. Make sure the DISRC, SM in the ADC_CTL0 register and CTN bit in the ADC_CTL1 register are reset.
2. Configure RSQ0 with the analog channel number.
3. Configure ADC_RSQx 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 EOC flag to be set.
7. Read the converted in the ADC_RDATA register.
8. Clear the EOC flag by writing 0 to it.

Continuous operation mode

This mode can be run on the 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. 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 17-3. Continuous operation mode



Software procedure for continuous operation mode on a routine channel:

1. Set the CTN bit in the ADC_CTL1 register.
2. Configure RSQ0 with the analog channel number.
3. Configure ADC_RSQx 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 EOC flag to be set.
7. Read the converted in the ADC_RDATA register.
8. Clear the EOC flag by writing 0 to it.
9. Repeat steps 6~8 as soon as the conversion is in need.

To get rid of checking, DMA can be used to transfer the converted data:

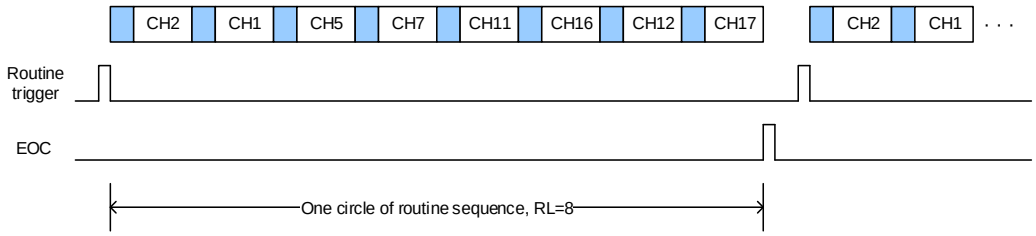
1. Set the CTN and DMA bit in the ADC_CTL1 register.
2. Configure RSQ0 with the analog channel number.
3. Configure ADC_RSQx register.
4. Configure ETMRC[1:0] bits in the ADC_CTL1 register if in need.
5. Prepare the DMA module to transfer data from the ADC_RDATA (refer to the spec of the DMA module).
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 all channels with a specific routine sequence specified in the ADC_RSQ0~ADC_RSQ8 registers. When the ADCON has been set high, the ADC samples and converts specified channels one by one in the routine sequence till the end of the sequence, once the corresponding software trigger or TRIGSEL trigger is active. The conversion data will be stored in the ADC_RDATA register. After conversion of the routine sequence, the EOC will be set. An interrupt will be generated if the EOCIE bit is set. The DMA bit in ADC_CTL1 register must be set when the routine sequence works in scan mode.

After conversion of a routine sequence, the conversion can be restarted automatically if the CTN bit in the ADC_CTL1 register is set.

Figure 17-4. Scan operation mode, continuous disable

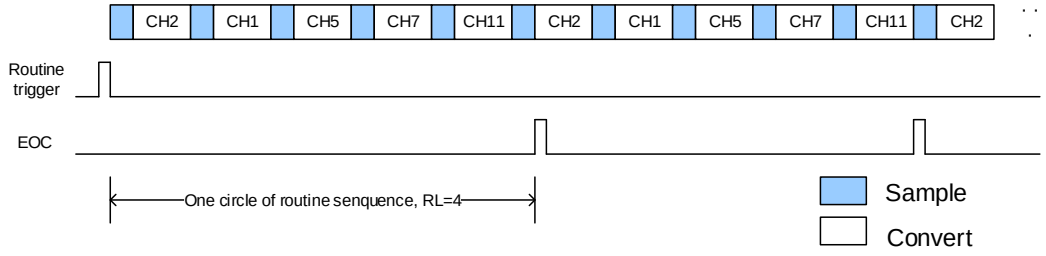


Software procedure for scan operation mode on a routine sequence:

1. Set the SM bit in the ADC_CTL0 register and the DMA bit in the ADC_CTL1 register
2. Configure ADC_RSQx 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 (refer to the spec of the DMA module).
5. Set the SWRCST bit, or generate an TRIGSEL trigger for the routine sequence.
6. Wait the EOC flag to be set.
7. Clear the EOC flag by writing 0 to it.

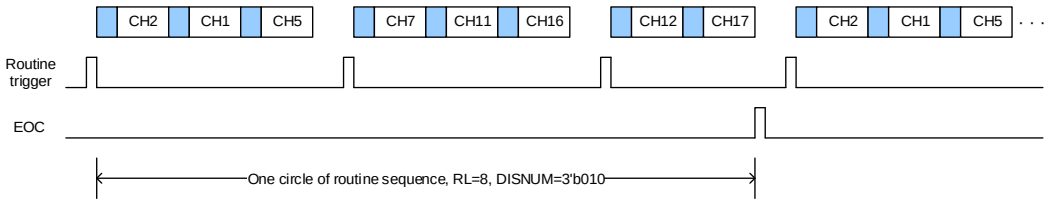
Figure 17-5. Scan operation mode, continuous enable



Discontinuous operation mode

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 sequence selected in the ADC_RSQ0~ADC_RSQ8 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_RSQ8 registers until all the channels of routine sequence are done. The EOC will be set after every circle of the routine sequence. An interrupt will be generated if the EOCIE bit is set.

Figure 17-6. Discontinuous operation mode



Software procedure for discontinuous operation mode on a routine sequence:

1. Set the DISRC bit in the ADC_CTL0 register and the DMA bit in the ADC_CTL1 register.
2. Configure DISNUM[2:0] bits in the ADC_CTL0 register.
3. Configure ADC_RSQx 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.
8. Wait the EOC flag to be set.

Clear the EOC flag by writing 0 to it.

17.4.7. Conversion result threshold monitor function

Analog watchdog 0

The analog watchdog 0 is enabled when the RWD0EN bit in the ADC_CTL0 register is set for routine sequence. This function is used to monitor whether the conversion result exceeds the set thresholds, and when the analog voltage converted by the ADC is below a low threshold or above a high threshold, the WDE0 bit in ADC_STAT register will be set. An interrupt will be generated if the WDE0IE bit is set. The ADC_WDHT0 and ADC_WDLT0 registers are used to specify the high and low threshold. The comparison is done before the alignment, so the threshold values are independent of the alignment, which is specified by the DAL bit in the ADC_CTL1 register. One or more channels, which are select by the RWD0EN, WD0SC and WD0CHSEL[4:0] bits in ADC_CTL0 register, can be monitored by the analog watchdog 0.

Analog watchdog 1/2

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_WDLT1, ADC_WDHT1, ADC_WDLT2 and ADC_WDHT2 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.

17.4.8. Data storage mode

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

Figure 17-7. 12-bit Data storage mode

Routine channel data

0	0	0	0	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

DAL=0

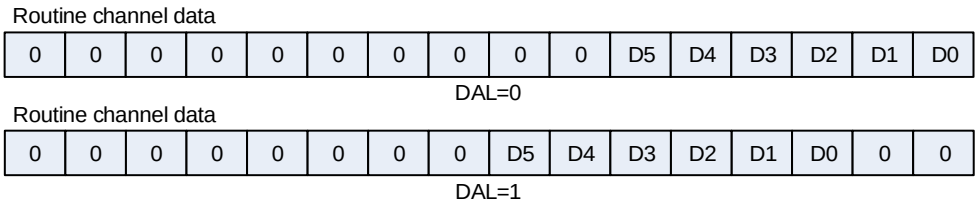
Routine channel data

D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0

DAL=1

6-bit resolution data storage mode is different from 12-bit/10-bit/8-bit resolution data storage mode, shown as ***Figure 17-8. 6-bit data storage mode***.

Figure 17-8. 6-bit data storage mode



NOTE: When OVSEN bit in the ADC_OVSAMPCTL register is set, the DAL bit value in the ADC_CTL1 register is ignored and the ADC only support LSB storage mode.

17.4.9. Sample time configuration

The number of CK_ADC cycles which is used to sample the input voltage can be specified by the RSMPn[9:0] bits in the ADC_RSQ0~ ADC_RSQ8 registers. A different sample time can be specified for each sequence. For 12-bits resolution, the total sampling and conversion time is “sampling time + 12.5” CK_ADC cycles.

Example:

CK_ADC = 40MHz and sample time is 2.5 cycles, the total conversion time is “2.5+12.5” CK_ADC cycles, that means 0.375us.

17.4.10. External trigger configuration

The conversion of routine sequence can be triggered by rising edge of TRIGSEL or software. The trigger source of routine sequence is controlled by the ETMRC[1:0] bits in the ADC_CTL1 register.

Table 17-4. Trigger source for routine channels

ETMRC[1:0]	Trigger Source	Trigger Type
01, 10, 11	TRIGSEL	Signal from TRIGSEL
00	SWRCST	Software trigger

17.4.11. DMA request

The DMA request, which is enabled by the DMA bit of ADC_CTL1 register, is used to transfer data of routine sequence for conversion of more than one channel. The ADC generates a DMA request at the end of conversion of a routine channel. When this request is received, the DMA will transfer the converted data from the ADC_RDATA register to the destination location which is specified by the user.

17.4.12. Overflow detection

Overflow detection is enabled when DMA is enabled or EOCM bit in ADC_CTL1 is set. An overflow event occurs when a routine conversion is done before the prior routine data has

been read out. The ROVF bit of the ADC_STAT is set. Overflow interrupt is generated if the ROVFIE bit in the ADC_CTL0 is set.

It is recommended to reinitialize the DMA module to recover the ADC from ROVF state. To ensure the routine converted data are transferred correctly, the internal state machine is reset. The ADC conversion will be stalled until the ROVF bit is cleared.

Software procedure for recovering the ADC from ROVF state:

1. Clear DMA 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 bit of ADC_STAT to 0.
5. Set CHEN bit of DMA_CHxCTL to 1.
6. Set DMA 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.

17.4.13. ADC internal channels

When the TSVEN1 bit of ADC_CTL1 register is set, the temperature sensor channel (ADC0_IN14) is enabled. When the TSVEN2 bit of ADC_CTL1 register is set, the high-precision temperature sensor channel (ADC2_IN18) is enabled when the INREFEN bit of ADC_CTL1 register is set, the VREFINT channel (ADC0_IN18 / ADC1_IN18 / ADC2_IN15 / ADC3_IN20) 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 or the high-precision 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 TSVEN1 or TSVEN2 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 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.

The internal voltage reference (V_{REFINT}) provides a stable (bandgap) voltage output for the ADC and Comparators. V_{REFINT} is internally connected to the ADC0_IN18, ADC1_IN18, ADC2_IN15, ADC3_IN20 input channel.

To use the normal temperature sensor:

1. Configure the conversion sequence (ADC0_IN14) and the sampling time(t_{s_temp} μs) for the channel.
2. Enable the temperature sensor by setting the TSVEN1 bit in the ADC control register 1 (ADC_CTL1).
3. Start the ADC conversion by setting the ADCON bit or by the triggers.

4. Read the temperature data($V_{temperature}$) in the ADC data register, and get the temperature with the following equation.

   Temperature (°C) = {(V_{25} – $V_{temperature}$ (digit)) / Avg_Slope} + 25.

   V_{25}: $V_{temperature}$ value at 25°C, the typical value please refer to the datasheet.

   Avg_Slope: Average Slope for curve between Temperature vs. $V_{temperature}$, the typical value please refer to the datasheet.

To use the high precision temperature sensor:

1. Configure the ADC clock(not greater than 5MHz).
2. Configure the conversion sequence (ADC2_IN18) and the sampling time(t_{s_temp} μs) for the channel.
3. Enable the temperature sensor by setting the TSVEN2 bit in the ADC control register 1 (ADC_CTL1).
4. Start the ADC conversion by setting the ADCON bit or by the triggers.
5. Read the temperature data($V_{temperature}$) in the ADC data register, and get the temperature with the following equation.

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

   V_{25}: $V_{temperature}$ value at 25°C, the typical value please refer to the datasheet.

   Avg_Slope: Average Slope for curve between Temperature vs. $V_{temperature}$, the typical value please refer to the datasheet.

Note:

1) After the high precision temperature sensor is enabled, it is necessary to wait for at least 3 ADC sampling cycles before the ADC conversion code value is considered valid, and the first 3 conversion data should be discarded;
2) The sampling accuracy of high precision temperature sensor can be improved by means of hardware on chip over sampling or software averaging.

17.4.14. Battery voltage monitoring

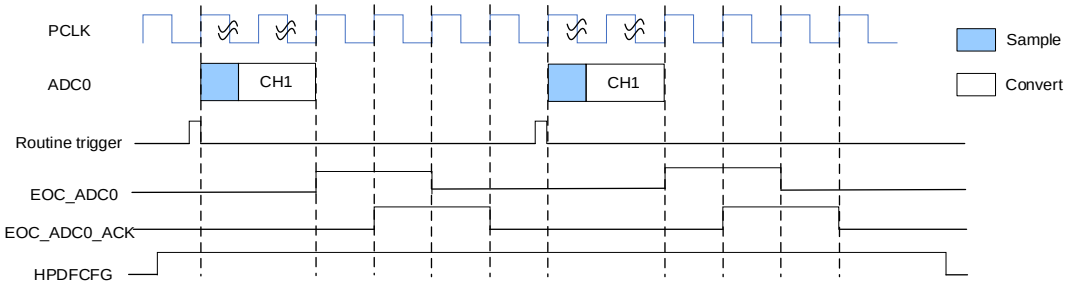
The V_{BAT} channel can be used to measure the backup battery voltage on the V_{BAT} pin. When the VBATEN bit in ADC_CTL1 register is set, V_{BAT} channel (ADC0_IN15 / ADC2_IN19) is enabled and a bridge divider by 3 integrated on the V_{BAT} pin is also enabled automatically with it. As V_{BAT} may be higher than V_{DDA}, this bridge is used to ensure the ADC correct operation. And it connects V_{BAT} / 3 to the ADC0_IN15 / ADC2_IN19 input channel. So, the converted digital value is V_{BAT} / 3. In order to prevent unnecessary battery energy consumption, it is recommended that the bridge will be enabled only when it is required.

17.4.15. Using HPDF to managing the conversion results

High-Performance Digital Filter (HPDF)can be used to manage the ADC conversion results.

In this situation, The HPDFCFG bit must be set to 1 and DMA bit, DDM bit, SYNCDMA[1:0] bits and SYNCDDM bit must be cleared to 0. If DMA and HPDF work in parallel,only DMA takes effect. The ADC transfers 16 least significant bits of the routine data register data to the HPDF, which in turns will reset the EOC flag once the transfer is complete. As shown in ***Figure 17-9. Schematic diagram of handshake signal between HFDF and ADC module***.

Figure 17-9. Schematic diagram of handshake signal between HFDF and ADC module



17.4.16. Programmable resolution (DRES)

The resolution can be configured to be either 12, 10, 8, or 6 bits by programming the DRES[1:0] bits in the ADC_CTL0 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. Lower resolution reduces the conversion time needed for the successive approximation steps as shown in ***Table 17-5. tCONV timings depending on resolution for ADC***.

Table 17-5. t_{CONV} timings depending on resolution for ADC

DRES[1:0] bits	t_{CONV} (ADC clock cycles)	t_{CONV}(ns) at f_{ADC}=40MHz	t_{SMPL}(min) (ADC clock cycles)	t_{ADC} (ADC clock cycles)	t_{ADC}(ns) at f_{ADC}=40MHz
12	12.5	312.5ns	2.5	15	375 ns
10	10.5	262.5 ns	2.5	13	325 ns
8	8.5	212.5ns	2.5	11	275 ns
6	6.5	162.5 ns	2.5	9	225 ns

17.4.17. 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 32-bit in all ADC. 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 Dout(n) is the n-th output digital signal of the ADC:

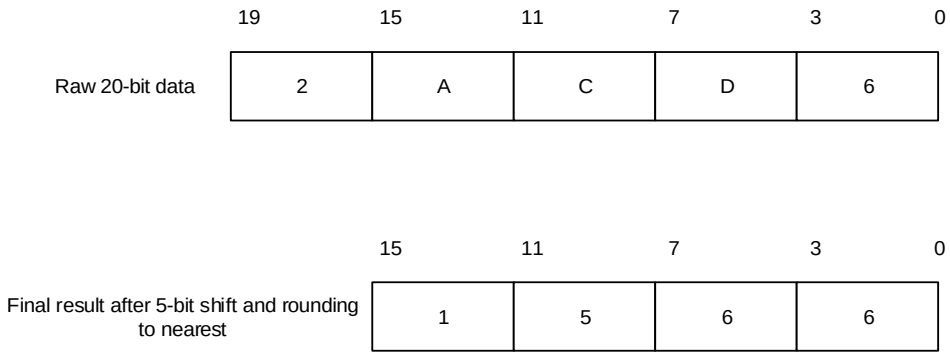
$$Result = \frac{1}{M} * \sum_{n=0}^{N-1} D_{out}(n) \quad (17\text{-}2)$$

The on-chip hardware oversampling circuit performs the following functions: summing and bit

right shifting. The oversampling ratio N is defined by the OVSR[9:0] bits in the ADC_OVSAMPCTL register. It can range from 2x to1024x. The division coefficient M means bit right shifting up to 11-bit. It is configured through the OVSS[3:0] bits in the ADC_OVSAMPCTL register.

The summation unit can produce up to 22 bits (1024 x 12bit), which is first shifted right. Then store the data into register.

Figure 17-10. Numerical example with 5-bits shift and rounding



When compared to standard conversion mode, the conversion timings of oversampling mode do not change, and the sampling time is maintained the same as that of standard conversion mode during the whole oversampling sequence. New data is supplied every N conversions, and the equivalent delay is equal to:

$$N \times t_{ADC} = N \times (t_{SMPL} + t_{CONV}) \quad (17\text{-}3)$$

17.4.18. Gain mode

When the GAINEN bit is set in the ADC_CTL1 register, the gain calibration is activated on all ADC converted data. After each conversion, calculate the data using the following formula:

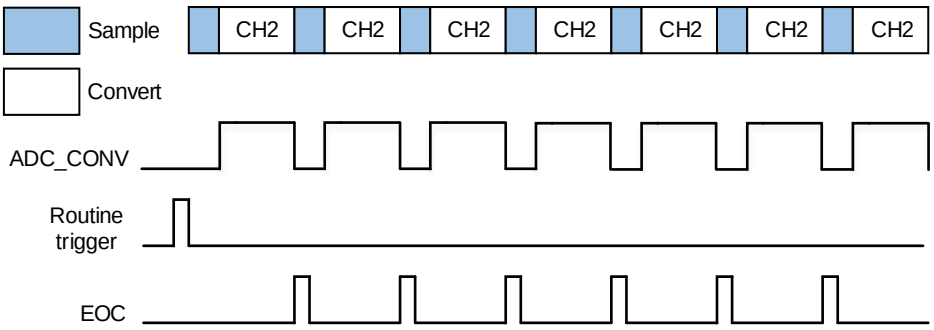
$$DATA(\text{new ADC result}) = DATA(\text{original ADC result}) \times (GAIN/4096) \quad (17\text{-}4)$$

The programmable range of GAIN is 0 to 16383, and the actual gain factor (GAIN / 4096) is 0 to 3.999756.

17.4.19. ADC convertsion signal

The ADC conversion signal ADC_CONV is kept high during ADC channel conversion and low at other times. The ADC_CONV signal can be internally connected to the CLAx module via TRIGSEL as an input to the CLAx multiplexer.

Figure 17-11. ADC convertsion signal in continuous operation mode



17.5. 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/ADC1/ADC2, according to the sync mode configurated by the SYNCM[4:0] bits in ADC_SYNCCTL 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 and ADC2. The converted data of routine channel is stored in the ADC sync routine data register (ADC_SYNCDATA).

The following modes can be configured in ***Table 17-6. ADC sync mode table***.

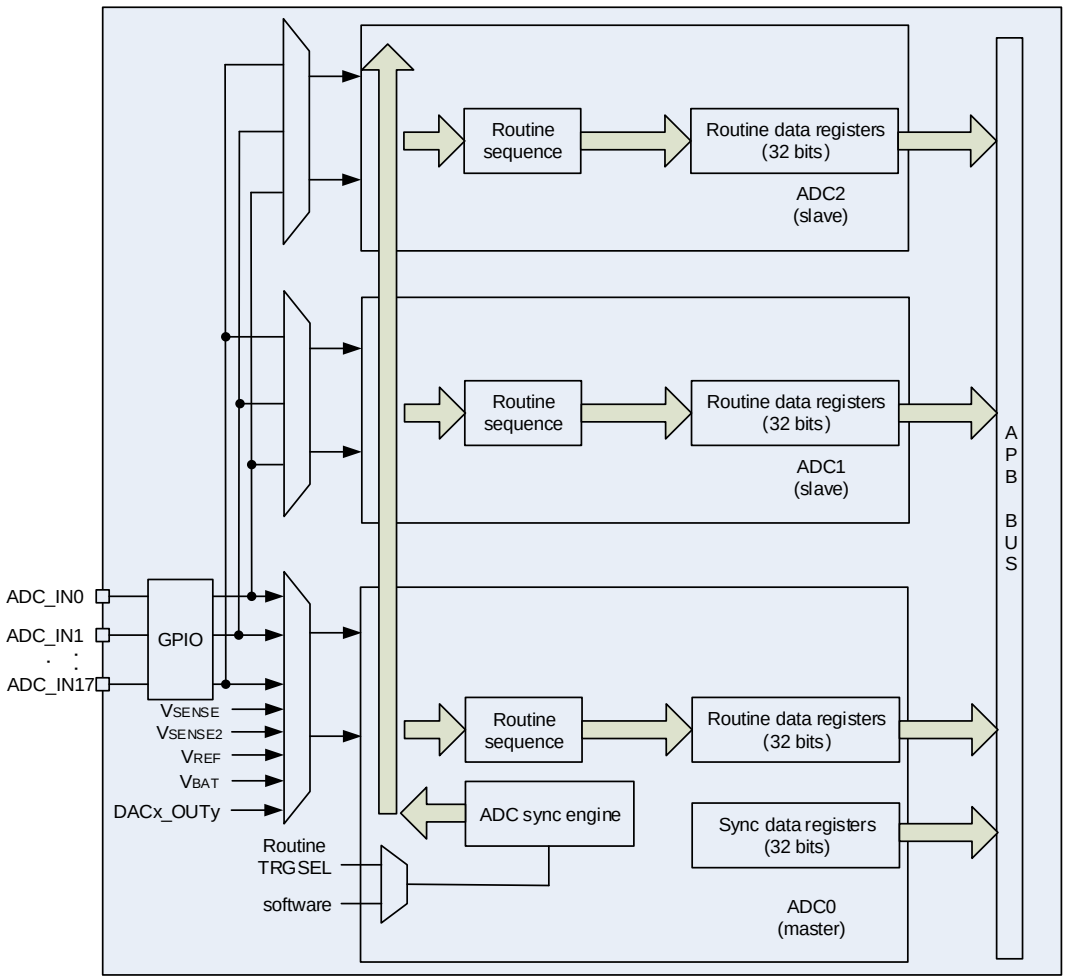
Table 17-6. ADC sync mode table

SYNCM[4: 0]	mode
00000	Free mode
00110	ADC0 and ADC1 work in routine parallel mode
00111	ADC0 and ADC1 work in routine follow-up mode
10110	ADC0, ADC1 and ADC2 work in routine parallel mode
10111	ADC0, ADC1 and ADC2 work in follow-up mode

When the ADCs are in a sync mode other than free mode, they should be configured to free mode before being configured to another sync mode.

The ADC sync scheme is shown in ***Figure 17-12. ADC sync block diagram*** .

Figure 17-12. ADC sync block diagram



17.5.1. Free mode

In this mode, the ADC synchronization is bypassed, and each ADC works freely.

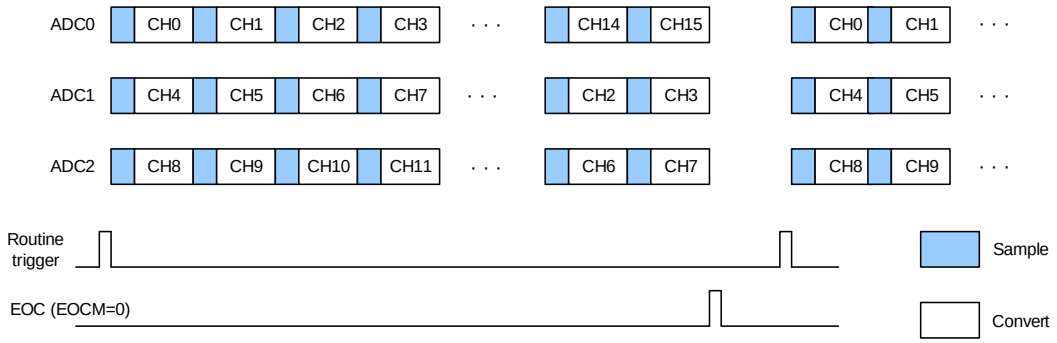
17.5.2. Routine parallel mode

The routine parallel mode is enabled by setting the SYNCM[4:0] bits in the ADC_SYNCCTL register to 00110 or 10110. In the routine parallel mode, all of the ADCs convert the routine sequence parallelly at the selected external trigger of ADC0. The triggers is selected by configuring the ETMRC[1:0] bits in the ADC_CTL1 register of ADC0.

EOC interrupts (if enabled on the ADC interfaces) are generated at the end of conversion events according to the EOCM bit in the ADC_CTL1 register. The behavior of routine parallel

mode is shown in the ***Figure 17-13. Routine parallel mode on 16 channels*** .

Figure 17-13. Routine parallel mode on 16 channels



Note:

1. Do not convert the same channel on two ADCs at a given time (no overlapping sampling times for the ADCs when converting the same channel).
2. Make sure to trigger the ADCs when none of them is converting (do not trigger ADC0 when some of the conversions are not finished).
3. ADC2 works freely if SYNCM=00110.

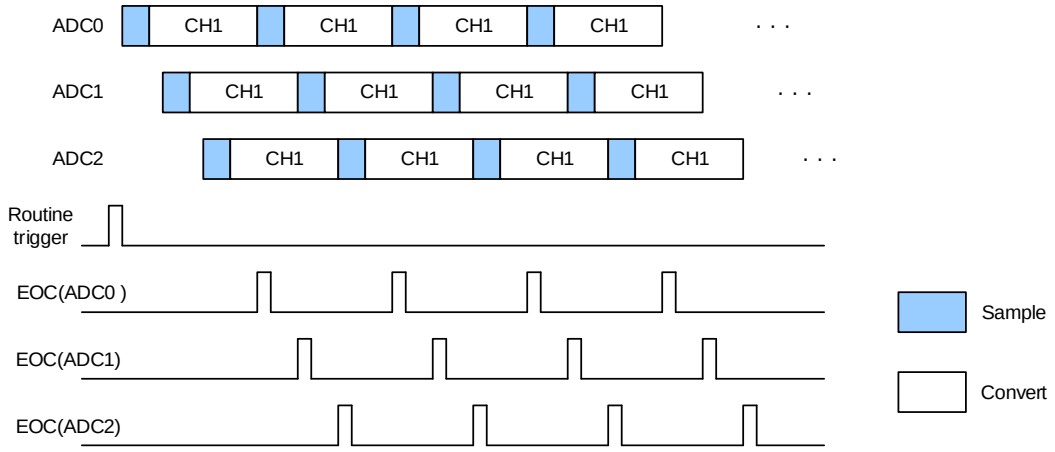
17.5.3. Follow-up mode

The follow-up mode is enabled by setting the SYNCM[4:0] bits in the ADC_SYNCCTL register to 00111 or 10111. In the follow-up mode, ADC0 converts the routine sequence at the selected external trigger. The triggers are selected by configuring the ETMRC[1:0] bits in the ADC_CTL1 register of ADC0. After a delay time, ADC1 converts the routine sequence. After another delay time, ADC2 converts the routine sequence. The routine sequence in above descriptions only includes one routine channel.

The delay time between two consecutive sample phase is configured by the SYNCDLY[3:0] bits in the ADC_SYNCCTL register. To prevent more than one ADCs from sampling the same channel at a given time, if the delay time configured by the SYNCDLY bits is shorter than the sample time, the delay time of (sample time + 2) CK_ADC cycles will be used.

If the CNT bit in ADC_CTL1 register is set, the selected routine channels are continuously converted. EOC interrupts (if enabled on the ADC interfaces) are generated at the end of conversion events according to the EOCM bit in the ADC_CTL1 register. The behavior of follow-up mode is shown in the ***Figure 17-14. Follow-up mode on 1 channel in continuous operation mode*** .

Figure 17-14. Follow-up mode on 1 channel in continuous operation mode



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. Make sure to trigger the ADCs when none of them is converting (do not trigger ADC0 when some of the conversions are not finished).
3. ADC2 works freely if SYNCM=00111.

17.5.4. Use DMA in ADC sync mode

In ADC sync mode, the converted data of routine sequence are stored in the ADC sync routine data register (ADC_SYNCDATA). DMA can be used to transfer data from ADC_SYNCDATA register. There are two DMA work modes, which can work well with the various ADC sync modes.

ADC sync DMA mode 0

In ADC sync DMA mode 0, the bit width of DMA transfer is 32. One DMA request transfers one data, which is selected from the routine data of the ADCs in turn. For every request, the source address of the DMA channel should be fixed to the ADC_SYNCDATA register, while the content of the ADC_SYNCDATA changes to the data that is to be transferred. When ADC0 and ADC2 work in SYNC mode, the transfer sequence is ADC0_RDATA[31:0] -> ADC1_RDATA[31:0] -> ADC0_RDATA[31:0] -> ADC1_RDATA[31:0]. When ADC0, ADC1 and ADC2 work in SYNC mode, the transfer sequence is ADC0_RDATA[31:0] -> ADC1_RDATA[31:0] -> ADC2_RDATA[31:0] ->ADC0_RDATA[31:0] -> ADC1_RDATA[31:0] -> ADC2_RDATA[31:0].

The ADC Sync DMA mode 0 is properly for:

- ADC0 and ADC1 work in routine parallel mode (SYNCM=00110).
- ADC0, ADC1 and ADC2 work in routine parallel mode (SYNCM=10110)

ADC sync DMA mode 1

In ADC sync DMA mode 1, the bit width of DMA transfer is 32. One DMA request transfers two data, which are selected from the routine data of the ADCs in turn. For every request, the source address of the DMA channel should be fixed to the ADC_SYNCDATA register, while the content of the ADC_SYNCDATA changes to the data that is to be transferred. When ADC0 and ADC1 works in SYNC mode, the transfer data are always {ADC1_RDATA[15:0], ADC0_RDATA[15:0]}. When ADC0, ADC1 and ADC2 works in SYNC mode, the transfer data are {ADC1_RDATA[15:0],ADC0_RDATA[15:0]} -> {ADC0_RDATA[15:0],ADC2_RDATA[15:0]} -> {ADC2_RDATA[15:0],ADC1_RDATA[15:0]} -> {ADC1_RDATA[15:0],ADC0_RDATA[15:0]}.

The ADC Sync DMA mode 1 is properly for:

- ADC0 and ADC1 work in routine parallel mode (SYNCM=00110).
- ADC0 and ADC1 work in follow-up mode (SYNCM=00111).
- ADC0, ADC1 and ADC2 work in follow-up mode (SYNCM=10111).

17.6. ADC interrupts

The interrupt can be produced on one of the events:

- End of conversion for routine sequence.
- The analog watchdog event.
- Overflow event.

The interrupts of ADC0 and ADC1 are mapped into the same interrupt vector IRQ18. The ADC2 is mapped into interrupt vector IRQ47. The ADC3 is mapped into interrupt vector IRQ61.

17.7. Register definition

ADC0 base address: 0x5000 0000

ADC1 base address: 0x5000 0400

ADC2 base address: 0x5000 0800

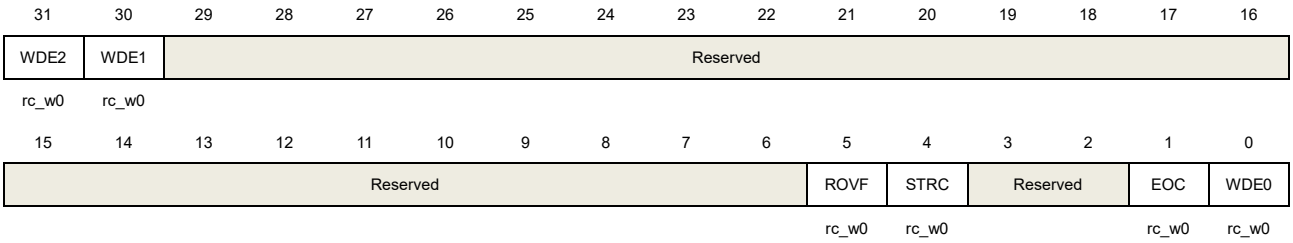
ADC3 base address: 0x5000 0C00

17.7.1. Status register (ADC_STAT)

Address offset: 0x00

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31	WDE2	Analog watchdog 2 event flag 0: No analog watchdog 2 event 1: Analog watchdog 2 event Set by hardware when the converted voltage crosses the values programmed in the ADC_WDLT2 and ADC_WDLT2 register. Cleared by software writing 0 to it.
30	WDE1	Analog watchdog 1 event flag 0: No analog watchdog 1 event 1: Analog watchdog 1 event Set by hardware when the converted voltage crosses the values programmed in the ADC_WDLT1 and ADC_WDLT1 register. Cleared by software writing 0 to it.
29:6	Reserved	Must be kept at reset value.
5	ROVF	Routine data register overflow 0: Routine data register not overflow 1: Routine data register overflow This bit is set by hardware when the routine data registers are overflow, in single mode or multi mode. This flag is only set when DMA is enabled or end of conversion mode is set to 1(EOCM=1). The recent routine data is lost when this bit is set. Cleared by software writing 0 to it.

4	STRC	Start flag of routine sequence 0: Conversion is not started 1: Conversion is started Set by hardware when routine sequence conversion starts. Cleared by software writing 0 to it.
3:2	Reserved	Must be kept at reset value.
1	EOC	End of sequence conversion flag 0: No end of sequence conversion 1: End of sequence conversion Set by hardware at the end of a routine sequence channel conversion. Cleared by software writing 0 to it or by reading the ADC_RDATA register.
0	WDE0	Analog watchdog 0 event flag 0: No analog watchdog 0 event 1: Analog watchdog 0 event Set by hardware when the converted voltage crosses the values programmed in the ADC_WDLT and ADC_WDHT registers. Cleared by software writing 0 to it.

17.7.2. 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
WDE2IE	WDE1IE	Reserved			ROVFIE	DRES[1:0]		RWD0EN	Reserved						
rw	rw				rw	rw		rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DISNUM[2:0]			Reserved	DISRC	Reserved	WD0SC	SM	Reserved	WDE0IE	EOCIE	WD0CHSEL[4:0]				
rw				rw		rw	rw		rw	rw	rw				

Bits	Fields	Descriptions
31	WDE2IE	Interrupt enable for WDE2 0: WDE2 interrupt disable 1: WDE2 interrupt enable
30	WDE1IE	Interrupt enable for WDE1 0: WDE1 interrupt disable 1: WDE1 interrupt enable
29:27	Reserved	Must be kept at reset value.
26	ROVFIE	Interrupt enable for ROVF 0: ROVF interrupt disable

		1: ROVF interrupt enable
25:24	DRES[1:0]	ADC data resolution 00: 12bit 01: 10bit 10: 8bit 11: 6bit
23	RWD0EN	Routine channel analog watchdog 0 enable 0: Routine channel analog watchdog 0 disable 1: Routine channel analog watchdog 0 enable
22:16	Reserved	Must be kept at reset value.
15:13	DISNUM[2:0]	Number of conversions in discontinuous mode The number of channels to be converted after a trigger will be DISNUM+1
12	Reserved	Must be kept at reset value.
11	DISRC	Discontinuous mode on routine channels 0: Discontinuous operation mode disable 1: Discontinuous operation mode enable
10	Reserved	Must be kept at reset value.
9	WD0SC	When in scan mode, analog watchdog 0 is effective on a single channel 0: Analog watchdog 0 is effective on all channels 1: Analog watchdog 0 is effective on a single channel
8	SM	Scan mode 0: Scan operation mode disable 1: Scan operation mode enable
7	Reserved	Must be kept at reset value.
6	WDE0IE	Interrupt enable for WDE0 0: Interrupt disable 1: Interrupt enable
5	EOCIE	Interrupt enable for EOC 00: Interrupt disable 1: Interrupt enable
4:0	WD0CHSEL[4:0]	Analog watchdog 0 channel select 00000: ADC channel0 00001: ADC channel1 00010: ADC channel2 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
10101: ADC channel 21
Other values are reserved.

Note: ADC0 analog inputs channel14, channel15, channel16, channel17, channel18 are internally connected to tempeture sensor, the battery, DAC0_OUT0, DAC0_OUT1, V_{REFINT} inputs.

ADC1 analog inputs channel16, channel17, channel18 are internally connected to DAC1_OUT0, DAC1_OUT1, V_{REFINT} inputs.

ADC2 analog inputs channel15, channel16, channel17, channel18, channel19 are internally connected to V_{REFINT} inputs, DAC2_OUT0, DAC2_OUT1, high-precision tempeture sensor, the battery.

ADC3 analog inputs channel18, channel19, channel20 are internally connected to DAC3_OUT0, DAC3_OUT1, V_{REFINT} inputs.

17.7.3. 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
TSVEN2	SWRCST	ETMRC[1:0]		CALMOD	Reserved	VBATEN	INREFEN	TSVEN1	Reserved						GAINEN
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			HPDFCFG	DAL	EOCM	DDM	DMA	Reserved	CALNUM[2:0]			RSTCLB	CLB	CTN	ADCON
			rw	rw	rw	rw	rw		rw			rw	rw	rw	rw

Bits	Fields	Descriptions
31	TSVEN2	This bit can be set or cleared by software in ADC2. Channel 18(high-precision temperature sensor) enable of ADC2.

		0: high-precision temperature sensor Channel disable 1: high-precision temperature sensor Channel enable
30	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.
29:28	ETMRC[1:0]	External trigger mode for routine channel 00: External trigger for routine channel disable 01: Rising edge of external trigger for routine channel enable 01: Falling edge of external trigger for routine channel enable 11: Rising and falling edge of external trigger for routine channel enable
27	CALMOD	ADC calibration mode 0：offset, mismatch mode 1：offset mode
26	Reserved	Must be kept at reset value.
25	VBATEN	This bit can be set or cleared by software in ADC0 and ADC2. Channel 15 (1/3 voltage of external battery) enable of ADC0 Channel 19 (1/3 voltage of external battery) enable of ADC2 0: 1/3 voltage of external battery Channel disable 1: 1/3 voltage of external battery Channel enable
24	INREFEN	This bit can be set or cleared by software in ADC0, ADC1, ADC2 and ADC3. Channel 18 (internal reference voltage) enable of ADC0. Channel 18 (internal reference voltage) enable of ADC1. Channel 15 (internal reference voltage) enable of ADC2. Channel 20 (internal reference voltage) enable of ADC3. 0: internal reference voltage Channel disable 1: internal reference voltage Channel enable
23	TSVEN1	This bit can be set or cleared by software in ADC0. Channel 14(temperature sensor) enable of ADC0. 0: temperature sensor Channel disable 1: temperature sensor Channel enable
22:17	Reserved	Must be kept at reset value.
16	GAINEN	Gain mode enable This bit can be set or cleared by software to the gain mode. 0: Gain mode disabled 1: Gain mode enabled and effect on all channels **Note**: Software is allowed to write this bit must ensure that no conversion is ongoing.
15:13	Reserved	Must be kept at reset value.

12	HPDFCFG	HPDF mode configuration To enable the HPDF mode, this bit is set and cleared by software. It is only valid when DMA=0. 0: HPDF mode disabled 1: HPDF mode enabled
11	DAL	Data alignment 0: LSB alignment 1: MSB alignment
10	EOCM	End of conversion mode 0: Only at the end of a sequence of routine conversions, the EOC bit is set. Overflow detection is disabled unless DMA=1. 1: At the end of each routine conversion, the EOC bit is set. Overflow is detected automatically
9	DDM	DMA disable mode This bit configures the DMA disable mode for single ADC mode 0: The DMA engine is disabled after the end of transfer signal from DMA controller is detected. 1: When DMA=1, the DMA engine issues a request at end of each routine conversion.
8	DMA	DMA request enable. 0: DMA request disable 1: DMA request enable
7	Reserved	Must be kept at reset value.
6:4	CALNUM[2:0]	Calibration Times These bits define the calibration times for ADC. 000:1 time 001:2 times 010:4times 011:8times 100:16times Others: reserved.
3	RSTCLB	Reset calibration This bit is set by software and cleared by hardware after the calibration registers are initialized. 0: Calibration register initialize done. 1: Initialize calibration register start
2	CLB	ADC calibration 0: Calibration done 1: Calibration start

1	CTN	Continuous mode 0: Continuous operation mode disable 1: Continuous operation mode enable
0	ADCON	ADC ON. The ADC will be wake up when this bit is changed from low to high and take a stabilization time. For power saving, when this bit is reset, the analog submodule will be put into power down mode. 0: ADC disable and power down 1: ADC enable

17.7.4. Watchdog high threshold register0 (ADC_WDHT0)

Address offset: 0x1C

Reset value: 0x003F 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										WDHT0[21:16]					
										rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDHT0[15:0]															
rw															

Bits	Fields	Descriptions
31:22	Reserved	Must be kept at reset value.
21:0	WDHT0[21:0]	High threshold for analog watchdog 0 These bits define the high threshold for the analog watchdog 0.

17.7.5. Watchdog low threshold register0 (ADC_WDLT0)

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										WDLT0[21:16]					
										rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDLT0[15:0]															
rw															

Bits	Fields	Descriptions
31:22	Reserved	Must be kept at reset value.

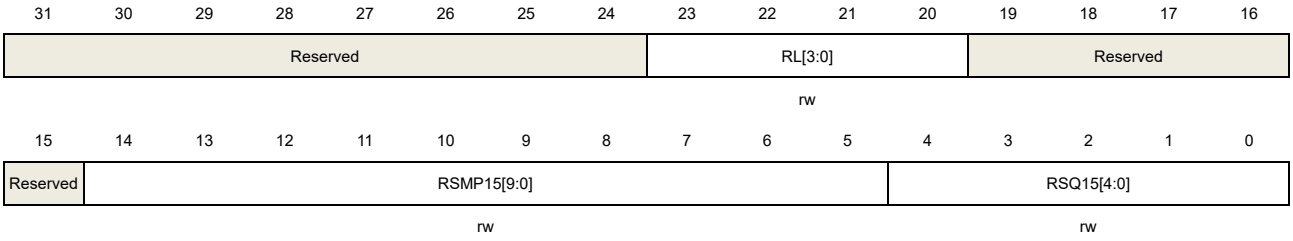
21:0	WDLT0[21:0]	Low threshold for analog watchdog 0 These bits define the low threshold for the analog watchdog.

17.7.6. Routine sequence register 0 (ADC_RSQ0)

Address offset: 0x24

Reset value: 0x0000 0000

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



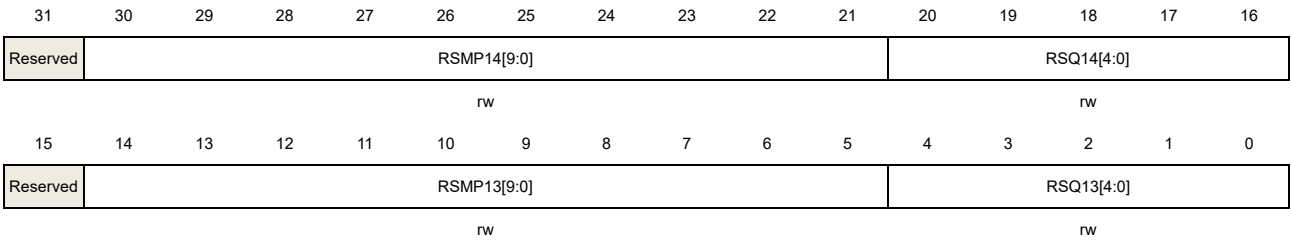
Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:20	RL[3:0]	Routine sequence length. The total number of conversion in routine sequence equals to RL[3:0]+1.
19:15	Reserved	Must be kept at reset value.
14:5	RSMP15[9:0]	Routine channel sample time Refer to RSMP0[9:0] description.
4:0	RSQ15[4:0]	refer to RSQ0[4:0] description

17.7.7. Routine sequence register 1 (ADC_RSQ1)

Address offset: 0x28

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.
30:21	RSMP14[9:0]	Routine channel sample time

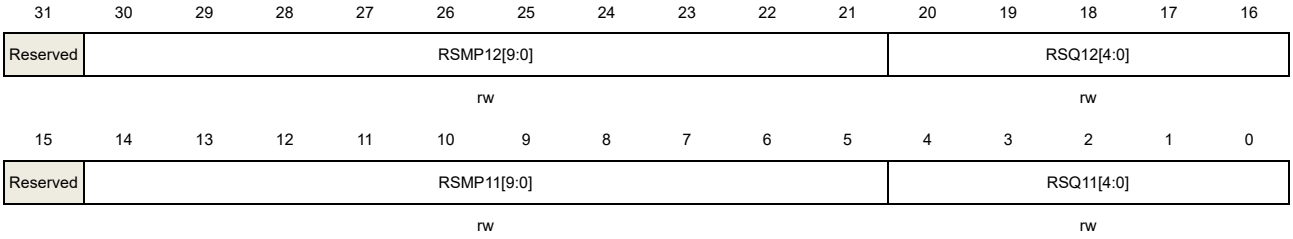
		Refer to RSMP0[9:0] description.
20:16	RSQ14[4:0]	refer to RSQ0[4:0] description
15	Reserved	Must be kept at reset value.
14:5	RSMP13[9:0]	Routine channel sample time Refer to RSMP0[9:0] description.
4:0	RSQ13[4:0]	refer to RSQ0[4:0] description

17.7.8. Routine sequence register 2 (ADC_RSQ2)

Address offset: 0x2C

Reset value: 0x0000 0000

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



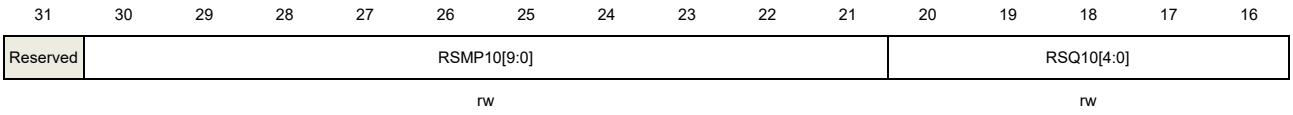
Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30:21	RSMP12[9:0]	Routine channel sample time Refer to RSMP0[9:0] description.
20:16	RSQ12[4:0]	refer to RSQ0[4:0] description
15	Reserved	Must be kept at reset value.
14:5	RSMP11[9:0]	Routine channel sample time Refer to RSMP0[9:0] description.
4:0	RSQ11[4:0]	refer to RSQ0[4:0] description

17.7.9. Routine sequence register 3 (ADC_RSQ3)

Address offset: 0x30

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	RSMP9[9:0]										RSQ9[4:0]				
	rw										rw				

Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30:21	RSMP10[9:0]	Routine channel sample time Refer to RSMP0[9:0] description.
20:16	RSQ10[4:0]	refer to RSQ0[4:0] description
15	Reserved	Must be kept at reset value.
14:5	RSMP9[9:0]	Routine channel sample time Refer to RSMP0[9:0] description.
4:0	RSQ9[4:0]	refer to RSQ0[4:0] description

17.7.10. Routine sequence register 4 (ADC_RSQ4)

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	RSMP8[9:0]										RSQ8[4:0]				
	rw										rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	RSMP7[9:0]										RSQ7[4:0]				
	rw										rw				

Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30:21	RSMP8[9:0]	Routine channel sample time Refer to RSMP0[9:0] description.
20:16	RSQ8[4:0]	refer to RSQ0[4:0] description
15	Reserved	Must be kept at reset value.
14:5	RSMP7[9:0]	Routine channel sample time Refer to RSMP0[9:0] description.
4:0	RSQ7[4:0]	refer to RSQ0[4:0] description

17.7.11. Routine sequence register 5 (ADC_RSQ5)

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	RSMP6[9:0]										RSQ6[4:0]				
	rw										rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	RSMP5[9:0]										RSQ5[4:0]				
	rw										rw				

Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30:21	RSMP6[9:0]	Routine channel sample time Refer to RSMP0[9:0] description.
20:16	RSQ6[4:0]	refer to RSQ0[4:0] description
15	Reserved	Must be kept at reset value.
14:5	RSMP5[9:0]	Routine channel sample time Refer to RSMP0[9:0] description.
4:0	RSQ5[4:0]	refer to RSQ0[4:0] description

17.7.12. Routine sequence register 6 (ADC_RSQ6)

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	RSMP4[9:0]										RSQ4[4:0]				
	rw										rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	RSMP3[9:0]										RSQ3[4:0]				
	rw										rw				

Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30:21	RSMP4[9:0]	Routine channel sample time Refer to RSMP0[9:0] description.

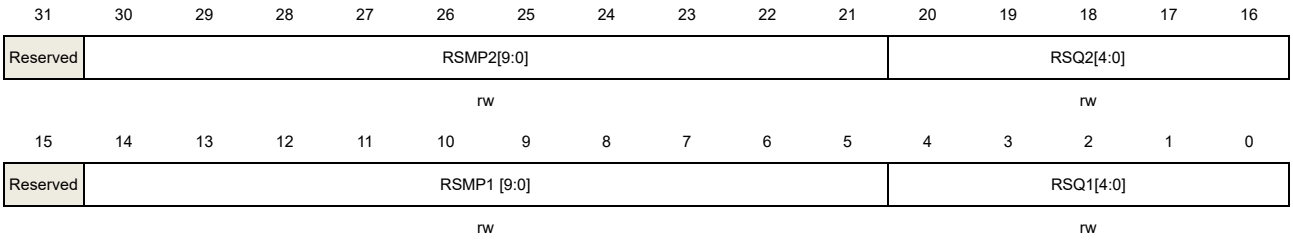
20:16	RSQ4[4:0]	refer to RSQ0[4:0] description
15	Reserved	Must be kept at reset value.
14:5	RSMP3[9:0]	Routine channel sample time Refer to RSMP0[9:0] description.
4:0	RSQ3[4:0]	refer to RSQ0[4:0] description

17.7.13. Routine sequence register 7 (ADC_RSQ7)

Address offset: 0x40

Reset value: 0x0000 0000

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



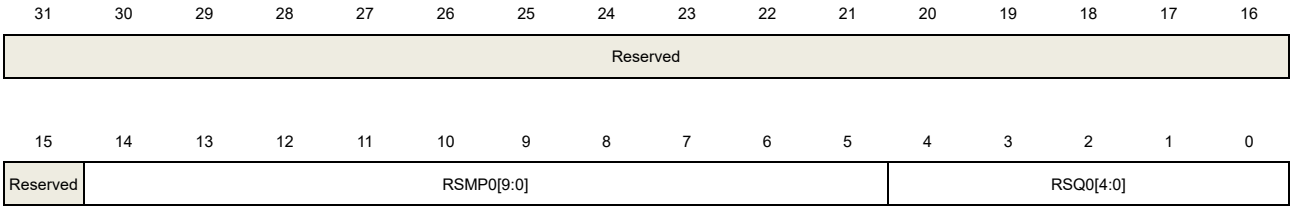
Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30:21	RSMP2[9:0]	Routine channel sample time Refer to RSMP0[9:0] description.
20:16	RSQ2[4:0]	refer to RSQ0[4:0] description
15	Reserved	Must be kept at reset value.
14:5	RSMP1[9:0]	Routine channel sample time Refer to RSMP0[9:0] description.
4:0	RSQ1[4:0]	refer to RSQ0[4:0] description

17.7.14. Routine sequence register 8 (ADC_RSQ8)

Address offset: 0x44

Reset value: 0x0000 0000

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



rw rw

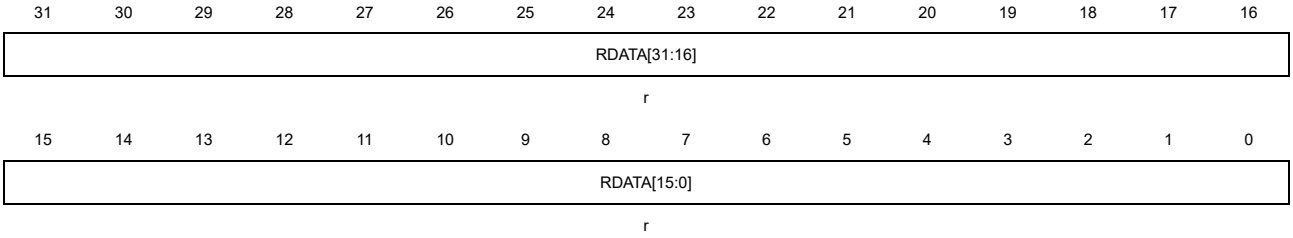
Bits	Fields	Descriptions
31:15	Reserved	Must be kept at reset value.
14:5	RSMP0[9:0]	Routine channel sample time ADC sample time is RSMP0[9:0] + 2.5 cycles. For example, RSMP0[9:0] = 100, the sample time is 102.5 cycles. RSMP0[9:0] max is 638, all other values are reserved. **Note**: if RSMPx[9:0]>638,the work behavior in adc will not be guaranteed.
4:0	RSQ0[4:0]	The channel number (0..21) is written to these bits to select a channel as the nth conversion in the routine sequence.

17.7.15. Routine data register (ADC_RDATA)

Address offset: 0x64

Reset value: 0x0000 0000

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



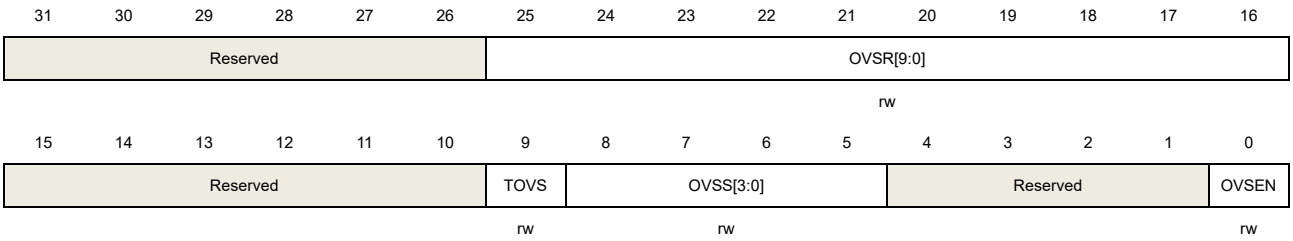
Bits	Fields	Descriptions
31:0	RDATA[31:0]	Routine channel data. These bits contain the conversion result from routine channel, which is read only.

17.7.16. 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

31:26	Reserved	Must be kept at reset value.
25:16	OVSR[9:0]	Oversampling ratio This bit filed defines the number of oversampling ratio. 10'd0: 1x (no oversampling) 10'd1: 2x 10'd2: 3x …… 10'd1023:1024x **Note:** The software allows this bit to be written only when ADCON = 0 (this ensures that no conversion is in progress).
15:10	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[9:0]). **Note:** The software allows this bit to be written only when ADCON = 0 (this ensures that no conversion is in progress).
8:5	OVSS[3:0]	Oversampling shift This bit is set and cleared by software. 0000: No shift 0001: Shift 1-bit 0010: Shift 2-bits 0011: Shift 3-bits 0100: Shift 4-bits 0101: Shift 5-bits 0110: Shift 6-bits 0111: Shift 7-bits 1000: Shift 8-bits 1001: Shift 9-bits 1010: Shift 10-bits 1011: Shift 11-bits Other codes reserved **Note:** The software allows this bit to be written only when ADCON = 0 (this ensures that no conversion is in progress).
4:1	Reserved	Must be kept at reset value.
0	OVSEN	Oversampling Enable This bit is set and cleared by software. 0: Oversampling disabled 1: Oversampling enabled

Note: The software allows this bit to be written only when ADCON = 0 (this ensures that no conversion is in progress).

17.7.17. Watchdog 1 Channel Selection Register (ADC_WD1SR)

Address offset: 0xA0

Reset value: 0x00000000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										AWD1CS[21:16]					
										rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AWD1CS[15:0]															
rw															

Bits	Fields	Descriptions
31:22	Reserved	Must be kept at reset value.
21:0	AWD1CS[21:0]	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[21:0] = 000..0, the analog Watchdog 1 is disabled **Note:** 1) The channels selected by AWD1CS must be also selected into the ADC_RSQn registers. 2) Software is allowed to write these bits only when the ADC is disabled (ADCON =0).

17.7.18. Watchdog 2 Channel Selection Register (ADC_WD2SR)

Address offset: 0xA4

Reset value: 0x00000000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										AWD2CS[21:16]					
										rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AWD2CS[15:0]															
rw															

Bits	Fields	Descriptions
31:22	Reserved	Must be kept at reset value.
21:0	AWD2CS[21:0]	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[21:0] = 000..0, the analog Watchdog 2 is disabled **Note:** 1) The channels selected by AWD2CS must be also selected into the ADC_RSQn registers. 2) Software is allowed to write these bits only when the ADC is disabled (ADCON =0).

17.7.19. Watchdog high threshold register1 (ADC_WDHT1)

Address offset: 0xA8
Reset value: 0x003F 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										WDHT1[21:16]					
										rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDHT1[15:0]															
rw															

Bits	Fields	Descriptions
31:22	Reserved	Must be kept at reset value.
21:0	WDHT1[21:0]	High threshold for analog watchdog 1 These bits define the high threshold for the analog watchdog 1. **Note:** Software is allowed to write these bits only when the ADC is disabled (ADCON =0).

17.7.20. Watchdog low threshold register1 (ADC_WDLT1)

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	WDLT1[21:16]
	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDLT1[15:0]															
rw															

Bits	Fields	Descriptions
31:22	Reserved	Must be kept at reset value.
21:0	WDLT1[23:0]	Low threshold for analog watchdog 1 These bits define the high threshold for the analog watchdog 1. **Note:** Software is allowed to write these bits only when the ADC is disabled (ADCON =0).

17.7.21. Watchdog high threshold register2 (ADC_WDHT2)

Address offset: 0xB0
Reset value: 0x003F 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										WDHT2[21:16]					
										rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDHT2[15:0]															
rw															

Bits	Fields	Descriptions
31:22	Reserved	Must be kept at reset value.
21:0	WDHT2[21:0]	High threshold for analog watchdog 2 These bits define the high threshold for the analog watchdog 2. **Note:** Software is allowed to write these bits only when the ADC is disabled (ADCON =0).

17.7.22. Watchdog low threshold register2 (ADC_WDLT2)

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										WDLT2[21:16]					
										rw					

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

WDLT2[15:0]
rw

Bits	Fields	Descriptions
31:22	Reserved	Must be kept at reset value.
21:0	WDLT2[21:0]	Low threshold for analog watchdog 2 These bits define the high threshold for the analog watchdog 2. **Note:** Software is allowed to write these bits only when the ADC is disabled (ADCON =0).

17.7.23. Differential mode control register (ADC_DIFCTL)

Address offset: 0XB8
Reset value: 0x0000 0000

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

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

Bits	Fields	Descriptions
31:22	Reserved	Must be kept at reset value.
21:0	DIFCTL[21:0]	Differential mode for channel 21..0. These bits are configured to select whether a channel is in single-ended or differential mode. DIFCTL[i] = 0: ADC analog input channel-i is configured in single-ended mode DIFCTL[i] = 1: ADC analog input channel-i is configured in differential mode **Note:** Software is allowed to write these bits only when the ADC is disabled (ADCON =0).

17.7.24. Gain configure register (ADC_GAINCFG)

Address offset: 0xC0
Reset value: 0x0000 0000
This register has to be accessed by word (32-bit).

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

Reserved	GAIN[13:0]
	rw

Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value
13:0	GAIN[13:0]	Gain factor These bits can be set or cleared by software to program the gain factor. The programmable range of GAIN is 0 to 16383, and the actual gain calculation factor (GAIN / 4096) is 0 to 3.999756. 00 0000 0000 0000: The gain is 0, the value of (GAIN /4096) is 0. ... 01 0000 0000 0000: The gain is 4096, the value of (GAIN / 4096) is 1. ... 10 0000 0000 0000: The gain is 8192, the value of (GAIN / 4096) is 2. ... 11 0000 0000 0000: The gain is 12288, the value of (GAIN / 4096) is 3. ... 11 1111 1111 1111: gain factor of 3.999756 **Note**: The GAINEN bit of ADC_CTL1 register must set 1 and then the gain factor can be effected.

17.7.25. Summary status register (ADC_SSTAT)

Address offset: 0x300 (for ADC0 base address)
Reset value: 0x0000 0000

This register is read only and provides a summary of the three ADCs. This register is not available in ADC1, ADC2 and ADC3.

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADC3_ROVF	ADC3_STRC	Reserved		ADC3_EOC	ADC3_WDE2	ADC3_WDE1	ADC3_WDE0	ADC2_ROVF	ADC2_STRC	Reserved		ADC2_EOC	ADC2_WDE2	ADC2_WDE1	ADC2_WDE0
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
ADC1_ROVF	ADC1_STRC	Reserved		ADC1_EOC	ADC1_WDE2	ADC1_WDE1	ADC1_WDE0	ADC0_ROVF	ADC0_STRC	Reserved		ADC0_EOC	ADC0_WDE2	ADC0_WDE1	ADC0_WDE0
r	r			r	r	r	r	r	r			r	r	r	r

Bits	Fields	Descriptions
31	ADC3_ROVF	This bit is the mirror image of the ROVF bit of ADC3
30	ADC3_STRC	This bit is the mirror image of the STRC bit of ADC3

29:28	Reserved	Must be kept at reset value
27	ADC3_EOC	This bit is the mirror image of the EOC bit of ADC3
26	ADC3_WDE2	This bit is the mirror image of the WDE2 bit of ADC3
25	ADC3_WDE1	This bit is the mirror image of the WDE1 bit of ADC3
24	ADC3_WDE0	This bit is the mirror image of the WDE0 bit of ADC3
23	ADC2_ROVF	This bit is the mirror image of the ROVF bit of ADC2
22	ADC2_STRC	This bit is the mirror image of the STRC bit of ADC2
21:20	Reserved	Must be kept at reset value
19	ADC2_EOC	This bit is the mirror image of the EOC bit of ADC2
18	ADC2_WDE2	This bit is the mirror image of the WDE2 bit of ADC2
17	ADC2_WDE1	This bit is the mirror image of the WDE1 bit of ADC2
16	ADC2_WDE0	This bit is the mirror image of the WDE0 bit of ADC2
15	ADC1_ROVF	This bit is the mirror image of the ROVF bit of ADC1
14	ADC1_STRC	This bit is the mirror image of the STRC bit of ADC1
13:12	Reserved	Must be kept at reset value
11	ADC1_EOC	This bit is the mirror image of the EOC bit of ADC1
10	ADC1_WDE2	This bit is the mirror image of the WDE2 bit of ADC1
9	ADC1_WDE1	This bit is the mirror image of the WDE1 bit of ADC1
8	ADC1_WDE0	This bit is the mirror image of the WDE0 bit of ADC1
7	ADC0_ROVF	This bit is the mirror image of the ROVF bit of ADC0
6	ADC0_STRC	This bit is the mirror image of the STRC bit of ADC0
5:4	Reserved	Must be kept at reset value
3	ADC0_EOC	This bit is the mirror image of the EOC bit of ADC0
2	ADC0_WDE2	This bit is the mirror image of the WDE2 bit of ADC0
1	ADC0_WDE1	This bit is the mirror image of the WDE1 bit of ADC0
0	ADC0_WDE0	This bit is the mirror image of the WDE0 bit of ADC0

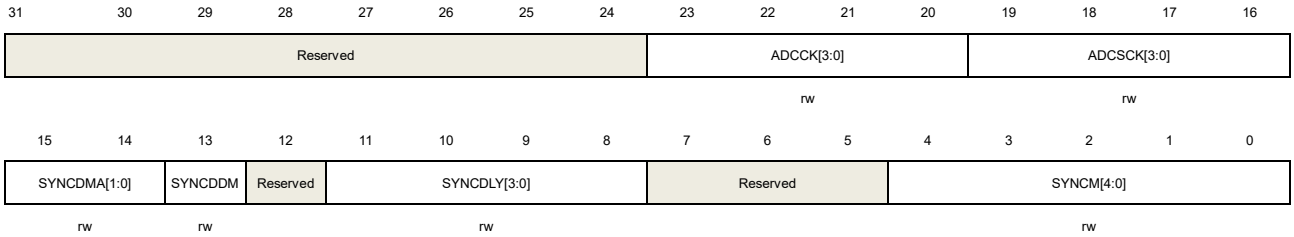
17.7.26. Sync control register (ADC_SYNCCTL)

Address offset: 0x304

Reset value: 0x0000 0000

This register is not available in ADC1, ADC2.

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



Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:20	ADCCK[3:0]	ADC clock prescaler. ADC Prescaler These bits are set and cleared by software to select the frequency of the ADC clock. All ADCs are common. 4'b0000: ADC clock div1 4'b0001: ADC clock div2 4'b0010: ADC clock div4 4'b0011: ADC clock div6 4'b0100: ADC clock div8 4'b0101: ADC clock div10 4'b0110: ADC clock div12 4'b0111: ADC clock div16 4'b1000: ADC clock div32 4'b1001: ADC clock div64 4'b1010: ADC clock div128 4'b1011: ADC clock div256 All other values are reserved.
19:16	ADCSCK[3:0]	ADC sync clock mode These bits are set and cleared by by software to define the ADC sync clock mode. All ADCs are common. 4'b0000: CLK_ADC (async clock mode) 4'b1000: HCLK div2 (sync clock mode) 4'b1001: HCLK div4 (sync clock mode) 4'b1010: HCLK div6 (sync clock mode) 4'b1011: HCLK div8 (sync clock mode) 4'b1100: HCLK div10 (sync clock mode) 4'b1101: HCLK div12 (sync clock mode) 4'b1110: HCLK div14 (sync clock mode) 4'b1111: HCLK div16 (sync clock mode) All other values are reserved.

15:14	SYNCDMA[1:0]	ADC sync DMA mode selection 00: ADC sync DMA disabled 01: ADC sync DMA mode 0 10: ADC sync DMA mode 1 11: reserved
13	SYNCDDM	ADC sync DMA disable mode This bit configures the DMA disable mode for ADC sync mode 0: The DMA engine is disabled after the end of transfer signal from DMA controller is detected. 1: When SYNCDMA is not equal to 2'b00, the DMA engine issues requests according to the SYNCDMA bits.
12	Reserved	Must be kept at reset value.
11:8	SYNCDLY[3:0]	ADC sync delay These bits are used to configure the delay between 2 sampling phases in ADC sync modes to (5+SYNCDLY) ADC clock cycles.
7:5	Reserved	Must be kept at reset value.
4:0	SYNCM[4:0]	ADC sync mode When ADC sync mode is enabled these bits should be set to 00000 firstly before change to another value. When ADC sync mode is enabled these bits should be set to 00000 firstly before change to another value. 00000: ADC sync mode disabled. All the ADCs work independently. 00110: ADC0 and ADC1 work in routine parallel mode. 00111: ADC0 and ADC1 work in follow-up mode. 10110: ADC0, ADC1 and ADC2 work in routine parallel mode. 10111: ADC0, ADC1 and ADC2 work in follow-up mode. All other values are reserved

17.7.27. Sync routine data register (ADC_SYNCDATA)

Address offset: 0x308 (for ADC0 base address)
Reset value: 0x0000 0000

This register is not available in ADC1, ADC2 and ADC3.

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SYNCDATA1[31:16]															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYNCDATA0[15:0]															
r															

Bits	Fields	Descriptions
31:16	SYNCDATA1[31:16]	Routine data 1 in ADC sync mode
15:0	SYNCDATA0[15:0]	Routine data 0 in ADC sync mode

18. Digital-to-analog converter (DAC)

18.1. 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, and DAC output buffer can be calibrated to improve output accuracy.

The sample and keep mode can reduce the power consumption of DAC.

The two DAC units can work independently or concurrently.

18.2. Characteristics

DAC main features are as follows:

- 8-bit or 12-bit resolution.
- Left or right data alignment.
- DMA capability for each unit and underrun function.
- Conversion update synchronously.
- Conversion trigged by external triggers.
- Configurable internal buffer.
- External voltage reference, V_{REFP}.
- Output buffer calibration.
- Using sample and keep mode to reduce the power consumption.
- Noise wave generation (LFSR noise mode and Triangle noise mode).
- Sawtooth wave generation.
- Two DAC units in concurrent mode.
- Output can be configured to persist when system reset.

Figure 18-1. DAC block diagram and *Table 18-1. DAC I/O description* show the block diagram of DAC and the pin description of DAC, respectively.

Figure 18-1. DAC block diagram

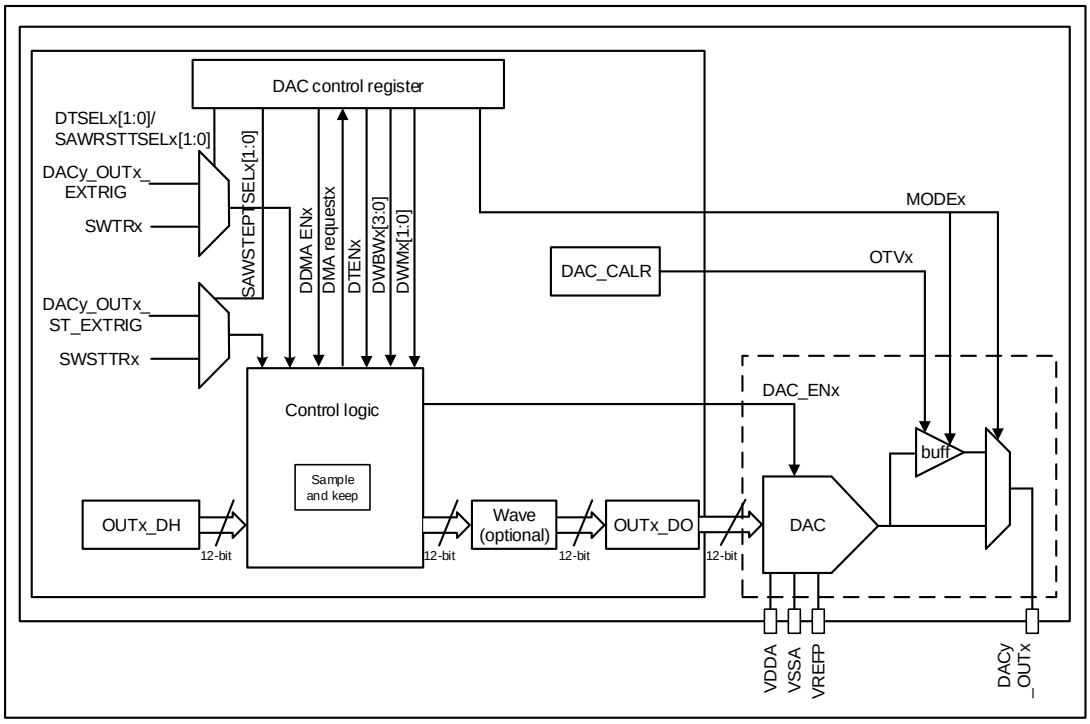


Table 18-1. DAC I/O description

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 18-2. DAC triggers and outputs summary

	DAC0		DAC1		DAC2		DAC3	
Unit	**Unit0**	**Unit1**	**Unit0**	**Unit1**	**Unit0**	**Unit1**	**Unit0**	**Unit1**
DAC outputs connected to I / Os	PA4	PA5	PA6	PA7	/	/	/	/
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.

18.3. Function overview

18.3.1. 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.

18.3.2. 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 MODEx bits in the DAC_MDCR register.

Note: DAC2 and DAC3 don't have output buffer.

18.3.3. 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 fored to 4'b0000.

The data written to DAC_OUTx_R12DH, DAC_OUTx_L12DH or DAC_OUTx_R8DH register are handled as unsigned format by default and can also be signed format (2's complement, Q1.15, Q1.11 or Q1.7 format) by setting DHFMTx bit in the DAC_MDCR register to '1'. No matter the signed/unsigned data format, the DAC output voltage range is 0 to VREFP. And the data transferred to DAC_OUTx_DO is converted to unsigned format. Which is shown in the below table.

Table 18-3 DAC data format (12-bit data)

DHFMTx bit	DATA written to DAC_OUTx_DH register	Decimal value	DATA transfered to DAC_OUTx_DO register	Output voltage
0	0x000	0	0x000	0
0	0xFFF	4095	0xFFF	V_{REFP}
1	0x7FF	2047	0xFFF	V_{REFP}
1	0x000	0	0x800	$\frac{V_{REFP}}{2}$
1	0xFFF	-1	0x7FF	$\frac{V_{REFP}}{2} - \frac{V_{REFP}}{4096}$

DHFMTx bit	DATA written to DAC_OUTx_DH register	Decimal value	DATA transfered to DAC_OUTx_DO register	Output voltage
1	0x800	-2048	0x000	0

18.3.4. 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 18-4. Triggers of DAC***.

Table 18-4. Triggers of DAC

DTSELx[1:0]	Trigger Source	Trigger Type
2b'00	External trigger from DAC_OUTx_EXTRIG in TRIGSEL	Hardware trigger
2b'01	SWTRx in DAC_SWT	Software trigger
2b'10	Reserved	Reserved
2b'11		

The external trigger is generated from the TRIGSEL, while the software trigger can be generated by setting the SWTRx bits in the DAC_SWT register.

18.3.5. 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.

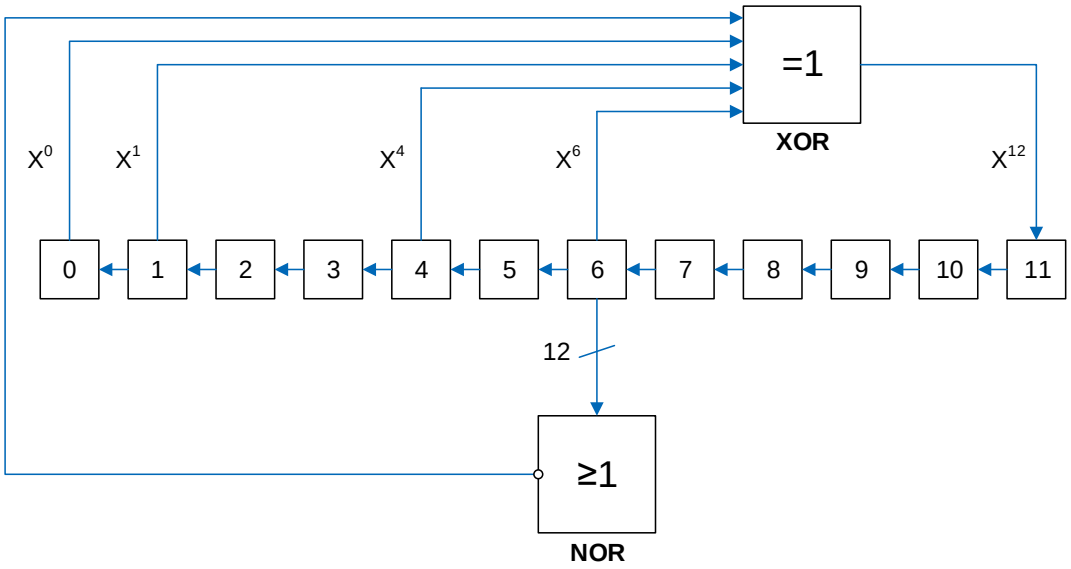
18.3.6. DAC noise wave

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

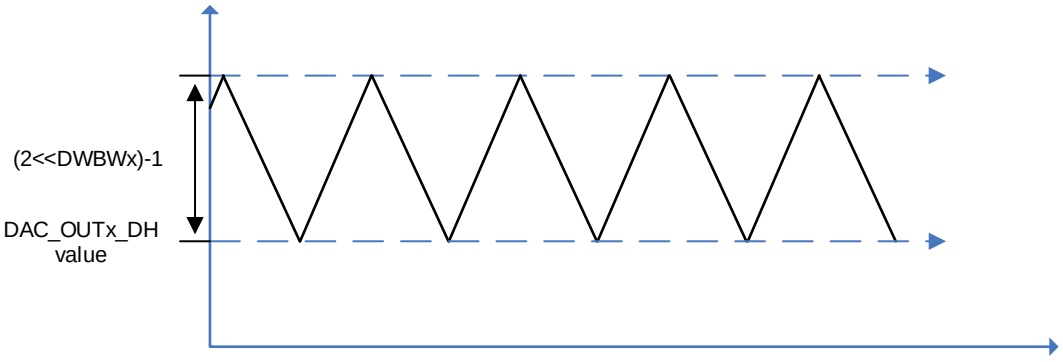
width is less than 12, the noise signal equals to the LSB DWBWx bits of the LFSR register, while the MSB bits are masked.

Figure 18-2. DAC LFSR algorithm



Triangle noise mode: in this 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 18-3. DAC triangle noise wave



18.3.7. DAC sawtooth wave

To generate a sawtooth waveform, a 16-bit sawtooth counter is implemented with reset trigger and step-up/down trigger.

The initial value is specified by SAWINITx[11:0] bits in the DAC_OUTx_SAW register, the bits 12 to 15 are set to 4b'0000.

The step direction is defined by SAWDIRx bit in the DAC_OUTx_SAW register, which decides the counter up or counter down.

The step value is defined by SAWSTEPx[15:0] bits in the DAC_OUTx_SAW register, the 12

MSB of sawtooth counter value will be transferred to DAC_OUTx_DO register to output voltage.

The sawtooth counter steps up/down(resets) at each rising edge of step(reset) trigger signal, which are selected by SAWRSTTSELx and SAWSTEPTSELx bits in DAC_SAWMDR register, respectively.

Table 18-5. Reset triggers of DAC sawtooth wave generation

SAWRSTTSELx[1:0]	Trigger Source	Trigger Type
2b'00	External trigger from DAC_OUTx_EXTRIG in TRIGSEL	Hardware trigger
2b'01	SWTRx in DAC_SWT	Software trigger
2b'10	Reserved	Reserved
2b'11		

Table 18-6. Step triggers of DAC sawtooth wave generation

SAWSTEPTSELx[1:0]	Trigger Source	Trigger Type
2b'00	External trigger from DAC_OUTx_ST_EXTRIG in TRIGSEL	Hardware trigger
2b'01	SWSTTRx in DAC_SWT	Software trigger
2b'10	Reserved	Reserved
2b'11		

Figure 18-4. DAC sawtooth wave (SAWDIRx = 1)

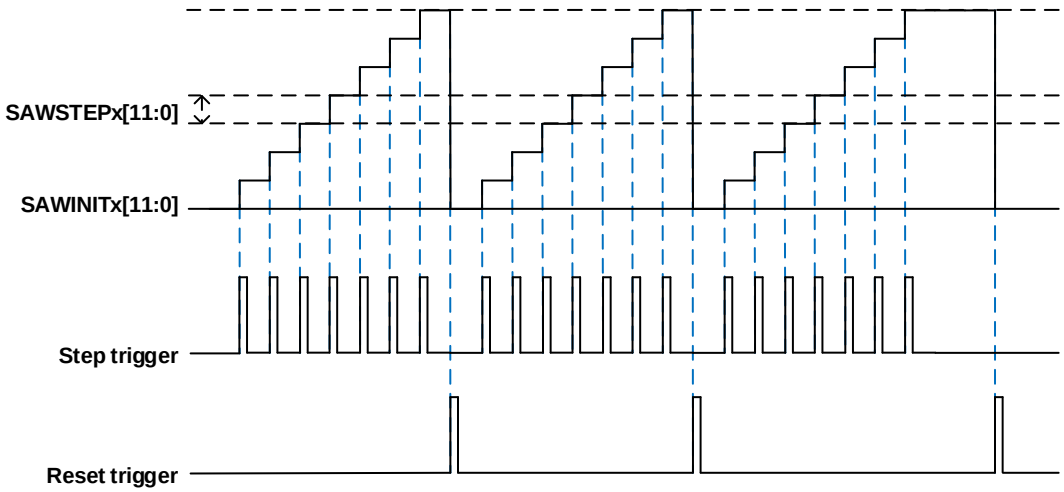
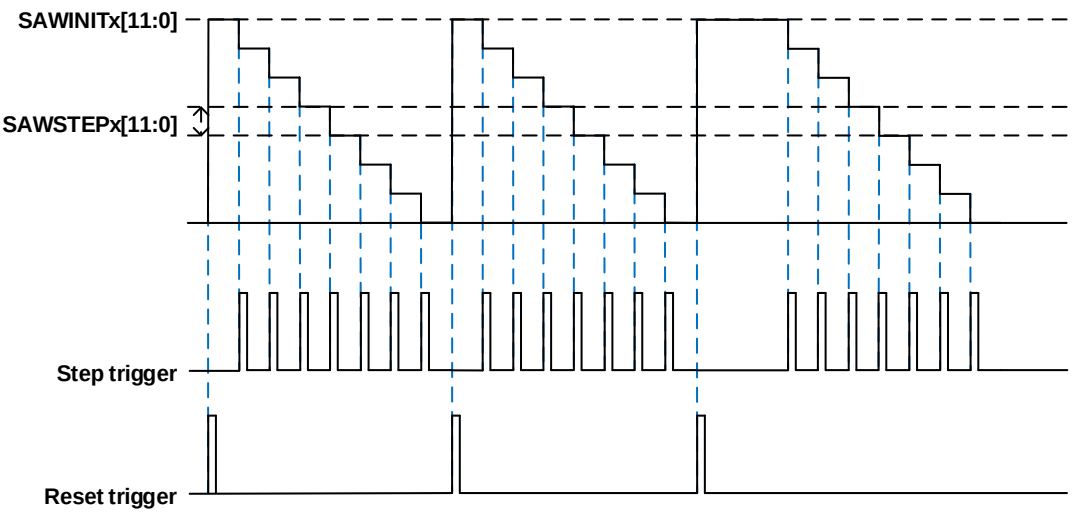


Figure 18-5. DAC sawtooth wave (SAWDIRx = 0)



Note:

The interval between two triggers (step and reset) of sawtooth wave mode output cannot be less than 1 hclk cycle or 1 pclk cycle.

18.3.8. DAC output voltage

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

$$V_{DAC_OUT}=V_{REFP}*OUTx_DO/4096 \quad (18\text{-}1)$$

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

18.3.9. DMA request

When the external trigger is enabled, the DMA request is enabled by setting the DDMAENx bits of the DAC_CTL0 register. A DMA request will be generated when an external hardware trigger (not a software trigger) occurs.

If the second external trigger arrives before confirming the previous request, the new request will not be serviced, and an underrun error event occurs. The DDUDRx bit in the DAC_STAT0 register is set, an interrupt will be generated if the DDUDRIEx bit in the DAC_CTL0 register is set. The DMA request will be stalled until the DDUDRx bit is cleared.

18.3.10. DAC concurrent conversion

When the two output units work at the same time, for maximum bus bandwidth utilization in specific applications, two output units 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 units 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.

18.3.11. DAC reset persist mode

The reset persist mode can be enabled by setting the DRSTMDx bit in the DAC_CTL0 register. Output units will persist through all resets except for power-on-reset.

When DRSTMDx is set to 1, the register DAC_CTL0(bit DRSTMDx, DTENx, DBOFFx and DENx), DAC_OUTx_DH, DAC_OUTx_DO can only be reset by power on reset.

Note: When DACx use internal reference voltage (VREF_EN = 1), this mode can be realized by enabling the reset persist function of VREF.

18.3.12. 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:

$$V_{out}=(D/2^{N-1})*G*V_{REFP}+V_{of} \quad (18\text{-}2)$$

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 enable. During the calibration:

- Buffer disconnect from I/O pin and other peripherals, and enter tri-stated.
- The buffer will be used as a comparator to detect the intermediate code value 0x800,and compare it with $V_{REFP}/2$ through the internal bridge. The CALFx bit in DAC_STAT0 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 OTV0[4:0] bits in DAC_CALR register is loaded when DAC reset.
- User calibration
  - When the operating conditions are different from the nominal factory calibration conditions, especially when the VDDA, V_{REFP} and temperature change, the user can calibrate at any point in the application process through the software.

The steps for the user to adjust the calibration are as follows:

- Writing 0 to DENx bit in DAC_CTL0 register to disable DAC output.
- Writing 1 to CALENx bit in DAC_CTL0 register to enable DAC calibration.
- Process of calibration algorithm
  - Writing a code (staring by 0x00000b) into OTVx[4:0].
  - Waiting for T_{cal} delay.
  - Checking the flag CALFx in DAC_STAT0 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 is found.

The contents of OTVx[4:0] bits can be calculated in a faster way by using successive approximation or dichotomy techniques.

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.

18.3.13. DAC modes

DAC two units can be set to normal mode or sample and keep mode. The DAC out can be connected to I/O pin or other peripherals.

Normal mode

When the MODEx[2] bit in the DAC_MDCR register is 0, DAC is in normal mode.

Sample and keep mode

When the MODEx[2] bit in the DAC_MDCR register is 1, DAC is in sample and keep mode, the DAC core converts the data after triggering the conversion, and then keeps the converted voltage on the capacitor (internal or external) in sample and keep mode. The DAC core is closed between the two samples. Without converting, the DAC output is tri-stated, so the overall power consumption can be reduced. In this mode, IRC32K clock drives all the corresponding logic and the DAC core and registers, so that DAC can be used in Deep-sleep mode.

The operation of sample and keep mode can be divided into three stages:

Sample stage

The sample and keep element is charged to the required voltage. The charging time is determined by the capacitance value. The sampling time is configured by the TSAMPx[9:0] bits in DAC_SKSTRx register. The BWTx bit in DAC_SATA0 is set to 1 when writing the TSAMPx[9:0] bits which indicates the synchronization between AHB clock and IRC32K. The BWTx is cleared by hardware when writing is completed. Then user can write TSAMPx[9:0] bits again. The TSAMPx[9:0] bits can be changed by software during DAC output operation.

Keep stage

At this stage, the DAC core is closed due to reduced power consumption. The keep time is configured by the TKEEPx[9:0] bits in DAC_SKKTR register. DAC out is tri-stated.

Refresh stage

In this stage, DAC core is turn on again to charge the declined voltage to target value. The refresh time is determined by the TREFx[7: 0] bits in DAC_SKRTR register.

When a new OUTx_DH is updated (a trigger when DTENx=1 or update when DTENx = 0), the operation stage will go to sample stage and the DAC core converts the new data to the required voltage. In sample and keep mode, it should take more than 3 cycles of IRC32K between two continuous data update operation for synchronous.

Note:

If select the sawtooth wave generation (DWMx=2’b11) during the sample and keep mode,

SAWRSTTSELx and DTSELx (x= 0,1) should keep same.

Time calculation

The calculation of the time for the three stages above are based on IRC32K clock periods. To configure enough sample and refresh time, refer to the following formula:

Table 18-7. Formula of sample and refresh time

Buffer State	t_{sample} (1) (2)	$t_{refresh}$ (1) (2) (3)	t_{keep} (3)
ON	$t_{wakeup} + R_{BON}*C_{SK}*ln(2^{N+1})$	$t_{wakeup} + R_{BON}*C_{SK}*ln(2*N_{LSB})$	$(V_{REFP}/2^{N})*N_{LSB}*C_{SK}/I_{leak}$
OFF	$t_{wakeup} + R_{BOFF}*C_{SK}*ln(2^{N+1})$	$t_{wakeup} + R_{BOFF}*C_{SK}*ln(2*N_{LSB})$	$(V_{REFP}/2^{N})*N_{LSB}*C_{SK}/I_{leak}$

Note:

(1) In the above formula, the t_{wakeup} is wakeup time from off state to the DAC output reaches final value, the charge time is calculated with 1/2 LSB error accuracy to desired output voltage. N is the DAC resolution 12 bits or 8 bits.

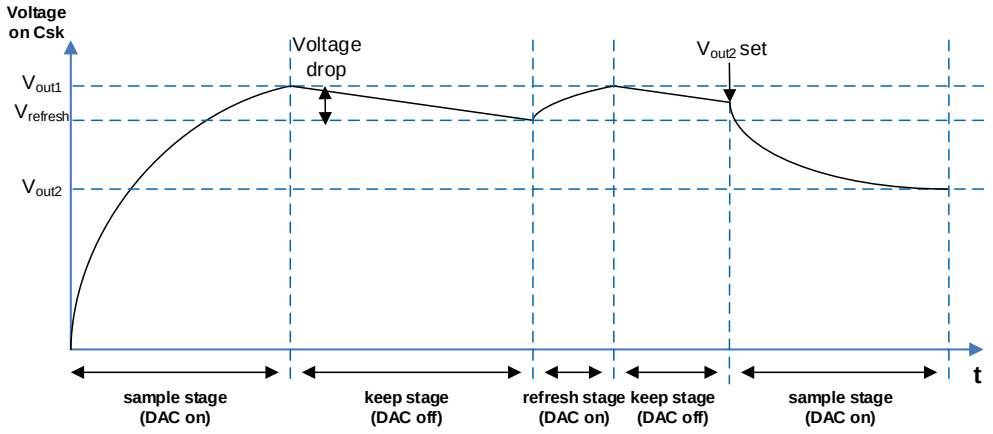
(2) R_{BON}/R_{BOFF} is the output impedence when buffer on or off, C_{SK} is the sample and keep capacitor (internal or external). When MODEx[2:0] bits in DAC_MDCR register are 3'b111, the internal capacitor is used to keep the DAC output voltage for on-chip peripherals.

(3) The keep time depends on the tolerance voltage drop during keep stage due to the capacitor discharging with the output leakage current. The number of LSBs N_{LSB} represents the voltage drop, and I_{leak} is the leakage current.

(4) The value of R_{BON}, R_{BOFF}, C_{SK} and t_{wakeup} please refer to device datasheet.

The sample and keep mode stage diagram is shown as below.

Figure 18-6. DAC sample and keep mode stage diagram



18.3.14. DAC low-power modes

Sleep mode

In Sleep mode, DAC can work normally, and can be used with DMA.

Deep-sleep mode

In Deep-sleep mode, if sample and keep mode is enabled before entering Deep-sleep mode, DAC can still hold the static output, otherwise DAC stops working.

Standby mode

In Standby mode, DAC stops working. When exting from the standby mode, the DAC need to be reinitialized to work again.

18.4. Register definition

DAC0 base address: 0x5000 1000
DAC1 base address: 0x5000 1400
DAC2 base address: 0x5000 1800
DAC3 base address: 0x5000 1C00

18.4.1. 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
DRSTMD1	CALEN1	DDUDRIE1	DDMAEN1	DWBW1[3:0]				DWM1[1:0]		Reserved		DTSEL1[1:0]		DTEN1	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
DRSTMD0	CALEN0	DDUDRIE0	DDMAEN0	DWBW0[3:0]				DWM0[1:0]		Reserved		DTSEL0[1:0]		DTEN0	DEN0
rw	rw	rw	rw	rw				rw				rw		rw	rw

Bits	Fields	Descriptions
31	DRSTMD1	DACx_OUT1 reset mode. This bit can only be reset by power on reset. 0: Normal mode. All resets will reset DACx_OUT1 and its associated registers. 1: Reset persist mode. DACx_OUT1 output will persist through all resets except for power-on-resets.
30	CALEN1	DACx_OUT1 calibration enable. 0: DACx_OUT1 calibration mode disabled. 1: DACx_OUT1 calibration mode enabled. CALEN1 can be written to 1 only when DEN1 is 0.
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
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 10: Triangle noise mode 11: Sawtooth noise mode
21:20	Reserved	Must be kept at reset value.
19:18	DTSEL1[1:0]	DACx_OUT1 trigger selection These bits are only used if bit DTEN = 1 and select the external event used to trigger DAC. 00: External trigger from DACx_OUT1_EXTRIG in TRIGSEL 01: Software trigger All other values: reserved.
17	DTEN1	DACx_OUT1 trigger enable 0: DACx_OUT1 trigger disabled 1: DACx_OUT1 trigger enabled
16	DEN1	DACx_OUT1 enable 0: DACx_OUT1 disabled 1: DACx_OUT1 enabled
15	DRSTMD0	DACx_OUT0 reset mode This bit can only be reset by power on reset 0: Normal mode. All resets will reset DACx_OUT0 and its associated registers 1: Reset persist mode. DACx_OUT0 output will persist through all resets except for power-on-resets
14	CALEN0	DACx_OUT0 calibration enable 0: DACx_OUT0 calibration mode disabled

		1: DACx_OUT0 calibration mode enabled CALEN0 can be written to 1 only when DEN0 is 0.
13	DDUDRIE0	DACx_OUT0 DMA underrun interrupt enable 0: DACx_OUT0 DMA underrun interrupt disabled 1: DACx_OUT0 DMA underrun interrupt enabled
12	DDMAEN0	DACx_OUT0 DMA enable 0: DACx_OUT0 DMA mode disabled 1: DACx_OUT0 DMA mode enabled
11:8	DWBW0[3:0]	DACx_OUT0 noise wave bit width These bits specify bit width of the noise wave signal of DACx_OUT0. These bits indicate that unmask LFSR bit [n-1, 0] in LFSR noise mode or the amplitude of the triangle is ((2<<(n-1))-1) in triangle noise mode, where n is the bit width of wave. 0000: The bit width of the wave signal is 1 0001: The bit width of the wave signal is 2 0010: The bit width of the wave signal is 3 0011: The bit width of the wave signal is 4 0100: The bit width of the wave signal is 5 0101: The bit width of the wave signal is 6 0110: The bit width of the wave signal is 7 0111: The bit width of the wave signal is 8 1000: The bit width of the wave signal is 9 1001: The bit width of the wave signal is 10 1010: The bit width of the wave signal is 11 ≥1011: The bit width of the wave signal is 12
7:6	DWM0[1:0]	DACx_OUT0 noise wave mode These bits specify the mode selection of the noise wave signal of DACx_OUT0 when external trigger of DACx_OUT0 is enabled (DTEN0=1). 00: wave disabled 01: LFSR noise mode 10: Triangle noise mode 11: Sawtooth noise mode
5:4	Reserved	Must be kept at reset value.
3:2	DTSEL0[1:0]	DACx_OUT0 trigger selection These bits are only used if bit DTEN = 1 and select the external event used to trigger DAC. 00: External trigger from DACx_OUT0_EXTRIG in TRIGSEL. 01: Software trigger. All other values: reserved.
1	DTEN0	DACx_OUT0 trigger enable 0: DACx_OUT0 trigger disabled

		1: DACx_OUT0 trigger enabled
0	DEN0	DACx_OUT0 enable 0: DACx_OUT0 disabled 1: DACx_OUT0 enabled

18.4.2. DACx software trigger register (DAC_SWT)

Address offset: 0x04

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved														SWSTTR1	SWSTTR0
														w	w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														SWTR1	SWTR0
														w	w

Bits	Fields	Descriptions
31:18	Reserved	Must be kept at reset value.
17	SWSTTR1	DACx_OUT1 sawtooth step-up/down software trigger. This bit is set by software to trigger the DAC in software trigger mode (sawtooth generation). It is cleared by hardware. 0: Software trigger for sawtooth step-up/down disabled 1: Software trigger for sawtooth step-up/down enabled
16	SWSTTR0	DACx_OUT0 sawtooth step-up/down software trigger. This bit is set by software to trigger the DAC in software trigger mode (sawtooth generation). It is cleared by hardware. 0: Software trigger for sawtooth step-up/down disabled 1: Software trigger for sawtooth step-up/down enabled
15:2	Reserved	Must be kept at reset value.
1	SWTR1	DACx_OUT1 conversion or sawtooth reset software trigger, cleared by hardware. 0: Software trigger disabled 1: Software trigger enabled
0	SWTR0	DACx_OUT0 conversion or sawtooth reset software trigger, cleared by hardware. 0: Software trigger disabled 1: Software trigger enabled

18.4.3. DACx_OUT0 12-bit right-aligned data holding register (DAC_OUT0_R12DH)

Address offset: 0x08

Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				OUT0_DH[11:0]											
				rw											

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

18.4.4. 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).

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OUT0_DH[11:0]												Reserved			
rw															

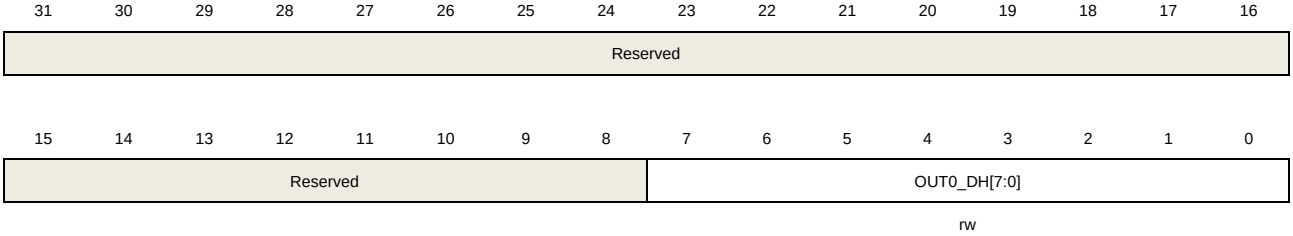
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:4	OUT0_DH[11:0]	DACx_OUT0 12-bit left-aligned data. These bits specify the data that is to be converted by DACx_OUT0.
3:0	Reserved	Must be kept at reset value.

18.4.5. 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).



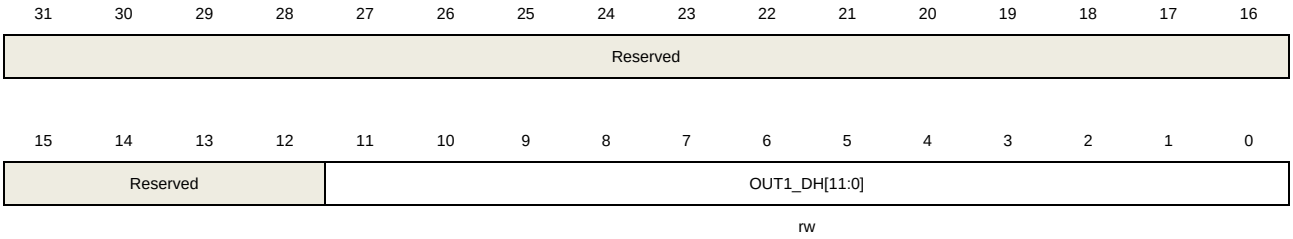
Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	OUT0_DH[7:0]	DACx_OUT0 8-bit right-aligned data. These bits specify the MSB 8 bits of the data that is to be converted by DACx_OUT0.

18.4.6. 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).



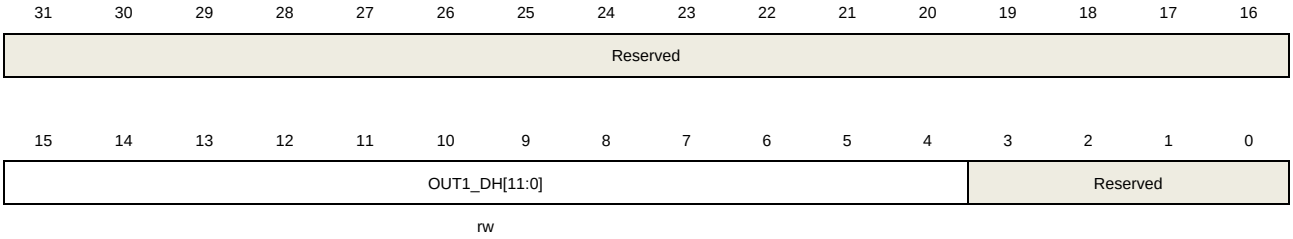
Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11:0	OUT1_DH[11:0]	DACx_OUT1 12-bit right-aligned data. These bits specify the data that is to be converted by DACx_OUT1.

18.4.7. 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).



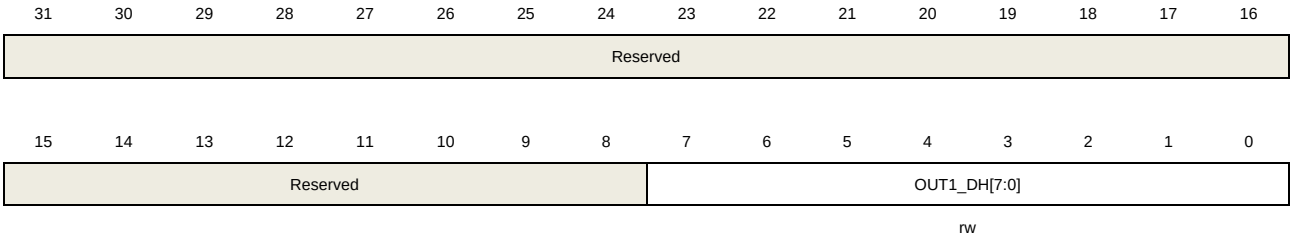
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:4	OUT1_DH[11:0]	DACx_OUT1 12-bit left-aligned data. These bits specify the data that is to be converted by DACx_OUT1.
3:0	Reserved	Must be kept at reset value.

18.4.8. 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 bits of the data that is to be converted by DACx_OUT1.

18.4.9. 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).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				OUT1_DH[11:0]											
				rw											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				OUT0_DH[11:0]											
				rw											

Bits	Fields	Descriptions
31:28	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.
15:12	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.

18.4.10. 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)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OUT1_DH[11:0]												Reserved			
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OUT0_DH[11:0]												Reserved			
rw															

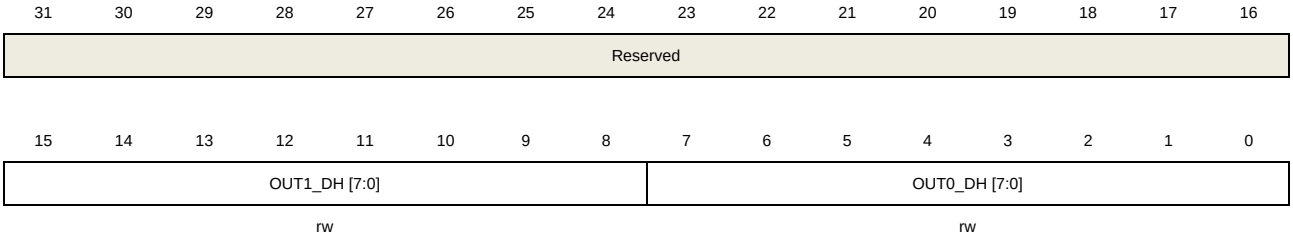
Bits	Fields	Descriptions
31:20	OUT1_DH[11:0]	DACx_OUT1 12-bit left-aligned data These bits specify the data that is to be converted by DACx_OUT1.
19:16	Reserved	Must be kept at reset value.
15:4	OUT0_DH[11:0]	DACx_OUT0 12-bit left-aligned data These bits specify the data that is to be converted by DACx_OUT0.
3:0	Reserved	Must be kept at reset value.

18.4.11. 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).



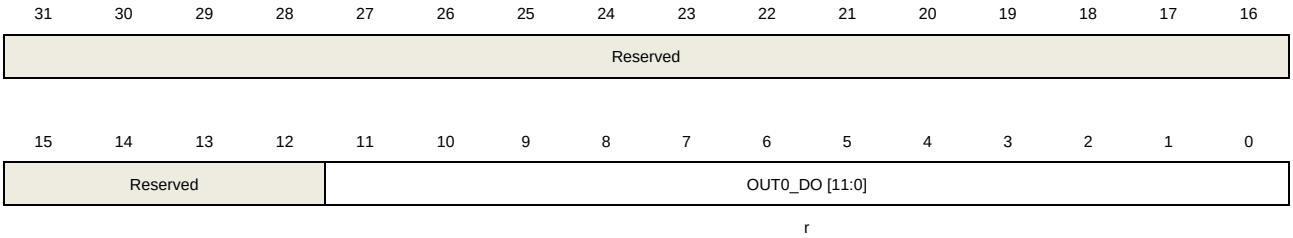
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:8	OUT1_DH[7:0]	DACx_OUT1 8-bit right-aligned data These bits specify the MSB 8-bit of the data that is to be converted by DACx_OUT1.
7:0	OUT0_DH[7:0]	DACx_OUT0 8-bit right-aligned data These bits specify the MSB 8-bit of the data that is to be converted by DACx_OUT0.

18.4.12. 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 data output These bits, which are read only, storage the data that is being converted by DACx_OUT0.

18.4.13. 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).

31	30	29	28	27	26	25	24	23	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				OUT1_DO [11:0]											
				r											

Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11:0	OUT1_DO [11:0]	DACx_OUT1 data output These bits, which are read only, storage the data that is being converted by DACx_OUT1.

18.4.14. DACx status register 0 (DAC_STAT0)

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
BWT1	CALF1	DDUDR1	Reserved												
r	r	rc_w1													

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BWT0	CALF0	DDUDR0	Reserved												
r	r	rc_w1													

Bits	Fields	Descriptions
31	BWT1	DAC_OUT1 TSAMP1[9:0] writing busy flag. This bit is set by the system when the sample and keep mode is enabled. When the TSAMP1[9:0] is writing, this bit is set. This bit is cleared by hardware when the write operation is complete. 0: There is no writing operation of TSAMP1[9:0]. 1: There is a writing operation of TSAMP1[9:0].
30	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.

29	DDUDR1	DACx_OUT1 DMA underrun flag, set by hardware, cleared by software write 1. 0: No underrun occurred. 1: Underrun occurred (Speed of DAC trigger is high than the DMA transfer).
28:16	Reserved	Must be kept at reset value.
15	BWT0	DAC_OUT0 TSAMP0[9:0] writing busy flag. This bit is set by the system when the sample and keep mode is enabled. When the TSAMP0[9:0] is writing, this bit is set. This bit is cleared by hardware when the write operation is complete. 0: There is no writing operation of TSAMP0[9:0]. 1: There is a writing operation of TSAMP0[9:0].
14	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.
13	DDUDR0	DACx_OUT0 DMA underrun flag, set by hardware, cleared by software write 1. 0: No underrun occurred. 1: Underrun occurred (Speed of DAC trigger is high than the DMA transfer).
12:0	Reserved	Must be kept at reset value.

18.4.15. DACx calibration register (DAC_CALR)

This register is only available on DACs with output buffer.

Address offset: 0x38

Reset value: 0x00XX 00XX

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											OTV1				
											rw				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											OTV0				
											rw				

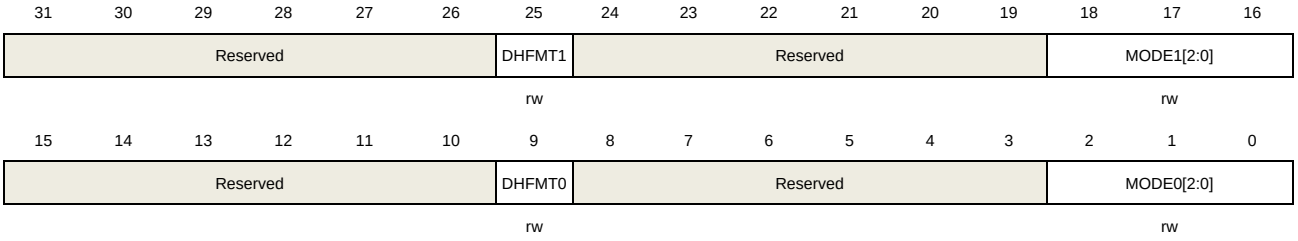
Bits	Fields	Descriptions
31:21	Reserved	Must be kept at reset value.
20:16	OTV1[4:0]	DACx_OUT1 offset calibration value.
15:5	Reserved	Must be kept at reset value.
4:0	OTV0[4:0]	DACx_OUT0 offset calibration value.

18.4.16. DACx mode control register (DAC_MDCR)

Address offset: 0x3C

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	DHFMT1	DACx_OUT1 data format written in data holding register. 0: Written data is handled as unsigned format. 1: Written data is handled as signed format (2's complement).
24:19	Reserved	Must be kept at reset value.
18:16	MODE1[2:0]	DACx_OUT1 mode. These bits can be written when bit DEN1=0 and bit CALEN1=0 in the DAC_CTL0 register, the write operation is invalid when DEN1=1 or CALEN1=1. DACx_OUT1 in normal mode. 000: Buffer is enabled and DACx_OUT1 is connected to I/O pin. 001: Buffer is enabled and DACx_OUT1 is connected to other peripherals and to I/O pin. 010: Buffer is disabled and DACx_OUT1 is connected to I/O pin. 011: Buffer is disabled and DACx_OUT1 is connected to other peripherals. DACx_OUT1 in sample and keep mode. 100: Buffer is enabled and DACx_OUT1 is connected to I/O pin. 101: Buffer is enabled and DACx_OUT1 is connected to other peripherals and to I/O pin. 110: Buffer is disabled and DACx_OUT1 is connected to other peripherals and to I/O pin. 111: Buffer is disabled and DACx_OUT1 is connected to other peripherals. Note: Only MODE1[2] bit is available for DACx(x=2,3) without output buffer to select normal mode or sample and keep mode.
15:10	Reserved	Must be kept at reset value.
9	DHFMT0	DACx_OUT0 data format written in data holding register. 0: Written data is handled as unsigned format 1: Written data is handled as signed format (2's complement).

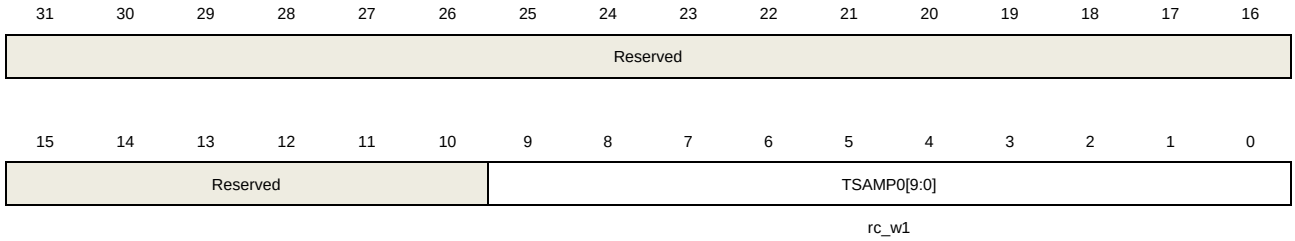
8:3	Reserved	Must be kept at reset value.
2:0	MODE0[2:0]	DACx_OUT0 mode. These bits can be written when bit DEN0=0 and bit CALEN0=0 in the DAC_CTL0 register, the write operation is invalid when DEN0=1 or CALEN0=1. DACx_OUT0 in normal mode. 000: Buffer is enabled and DACx_OUT0 is connected to I/O pin. 001: Buffer is enabled and DACx_OUT0 is connected to other peripherals and to I/O pin. 010: Buffer is disabled and DACx_OUT0 is connected to I/O pin. 011: Buffer is disabled and DACx_OUT0 is connected to other peripherals. DACx_OUT0 in sample and keep mode. 100: Buffer is enabled and DACx_OUT0 is connected to I/O pin. 101: Buffer is enabled and DACx_OUT0 is connected to other peripherals and to I/O pin. 110: Buffer is disabled and DACx_OUT0 is connected to other peripherals and to I/O pin. 111: Buffer is disabled and DACx_OUT0 is connected to other peripherals. Note: Only MODE[2] bit is available for DACx(x=2,3) without output buffer to select normal mode or sample and keep mode.

18.4.17. DACx sample and keep sample time register 0 (DAC_SKSTR0)

Address offset: 0x40

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.
9:0	TSAMP0[9:0]	DACx_OUT0 sample time.

18.4.18. DACx sample and keep sample time register 1 (DAC_SKSTR1)

Address offset: 0x44

Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						TSAMP1[9:0]									
						rc_w1									

Bits	Fields	Descriptions
31:10	Reserved	Must be kept at reset value.
9:0	TSAMP1[9:0]	DACx_OUT1 sample time.

18.4.19. DACx sample and keep keep time register (DACx_SKKTR)

Address offset: 0x48

Reset value: 0x0001 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						TKEEP1[9: 0]									
						rc_w1									

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						TKEEP0[9: 0]									
						rc_w1									

Bits	Fields	Descriptions
31:26	Reserved	Must be kept at reset value.
25:16	TKEEP1[9: 0]	DACx_OUT1 keep time (only available in sample and keep mode).
15:10	Reserved	Must be kept at reset value.
9:0	TKEEP0[9: 0]	DACx_OUT0 keep time (only available in sample and keep mode).

18.4.20. DACx sample and keep refresh time register (DACx_SKRTR)

Address offset: 0x4C

Reset value: 0x0001 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								TREF1[7:0]							
								rc_w1							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								TREF0[7:0]							
								rc_w1							

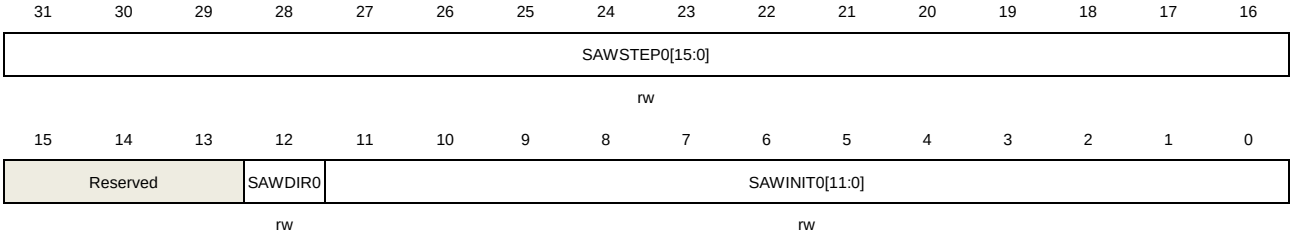
Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:16	TREF1[7: 0]	DACx_OUT1 refresh time (only available in sample and keep mode).
15:8	Reserved	Must be kept at reset value.
7:0	TREF0[7: 0]	DACx_OUT refresh time (only available in sample and keep mode).

18.4.21. DACx_OUT0 sawtooth register (DAC_OUT0_SAW)

Address offset: 0x58

Reset value: 0x0000 0000

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



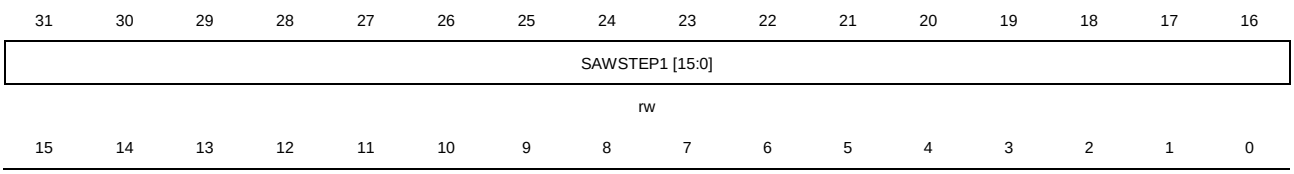
Bits	Fields	Descriptions
31:16	SAWSTEP0[15:0]	DACx_OUT0 sawtooth step value.
15:13	Reserved	Must be kept at reset value.
12	SAWDIR0	DACx_OUT0 sawtooth step direction. This bit is written by software to select the direction of sawtooth step. 0: Step-down 1: Step-up
11:0	SAWINIT0[11:0]	DACx_OUT0 sawtooth initial value

18.4.22. DACx_OUT1 sawtooth register (DAC_OUT1_SAW)

Address offset: 0x5C

Reset value: 0x0000 0000

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



Reserved	SAWDIR1	SAWINIT1 [11:0]
	rw	rw

Bits	Fields	Descriptions
31:16	SAWSTEP1[15:0]	DACx_OUT1 sawtooth step value.
15:13	Reserved	Must be kept at reset value.
12	SAWDIR1	DACx_OUT1 sawtooth step direction. This bit is written by software to select the direction of sawtooth step. 0: Step-down 1: Step-up
11:0	SAWINIT1[11:0]	DACx_OUT1 sawtooth initial value.

18.4.23. DACx sawtooth mode register (DACx_SAWMDR)

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
Reserved						SAWSTEPTSEL1[1:0]		Reserved						SAWRSTTSEL1[1:0]	
						rw								rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						SAWSTEPTSEL0[1:0]		Reserved						SAWRSTTSEL0[1:0]	
						rw								rw	

Bits	Fields	Descriptions
31:26	Reserved	Must be kept at reset value.
25:24	SAWSTEPTSEL1[1:0]	DACx_OUT1 sawtooth step-up/down trigger selection. 00: External trigger from DACx_OUT1_ST_EXTRIG in TRIGSEL 01: Software trigger All other values: reserved.
23:18	Reserved	Must be kept at reset value.
17:16	SAWRSTTSEL1[1:0]	DACx_OUT1 sawtooth reset to initial value trigger selection. 00: External trigger from DACx_OUT1_EXTRIG in TRIGSEL 01: Software trigger All other values: reserved.
15:10	Reserved	Must be kept at reset value.
9:8	SAWSTEPTSEL0[1:0]	DACx_OUT0 sawtooth step-up/down trigger selection 00: External trigger from DACx_OUT0_ST_EXTRIG in TRIGSEL

		01: Software trigger All other values: reserved.
7:2	Reserved	Must be kept at reset value.
1:0	SAWRSTTSEL0[1:0]	DACx_OUT0 sawtooth reset to initial value trigger selection 00: External trigger from DACx_OUT0_EXTRIG in TRIGSEL 01: Software trigger All other values: reserved.

19. Comparator (CMP)

19.1. 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.

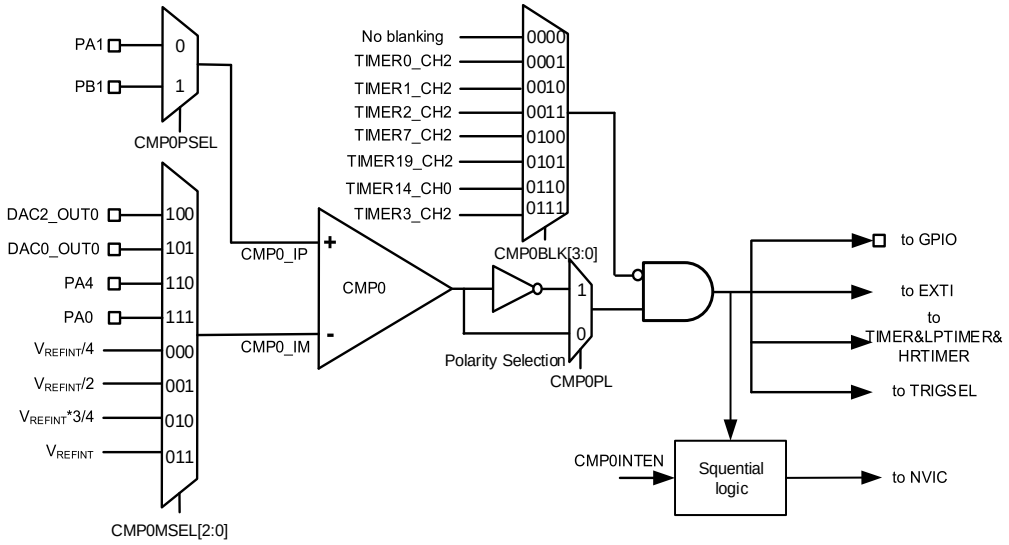
19.2. Characteristic

- Rail-to-rail comparators.
- Configurable hysteresis.
- 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.
- Outputs to EXTI.
- Outputs to NVIC.
- Outputs to TRIGSEL.
- Outputs can be configured to persist when system reset.

19.3. Function overview

The block diagram of CMP is shown below:

Figure 19-1. CMP block diagram



Note:

1) V_{REFINT} is 1.2V.

2) CMPx(x=1,2..7) as the same CMP0, CMPxPSEL, CMPxBLK, CMPxMSEL etc. refer to table ***Table 19-1. CMP inputs and outputs summary*** and register.

19.3.1. CMP clock

The clock of the CMP which is connected to APB bus.

19.3.2. 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, the polarity selection logic and the output redirection to the port work independently from PCLK.

Table 19-1. CMP inputs and outputs summary details the inputs and outputs of the CMP.

Table 19-1. CMP inputs and outputs summary

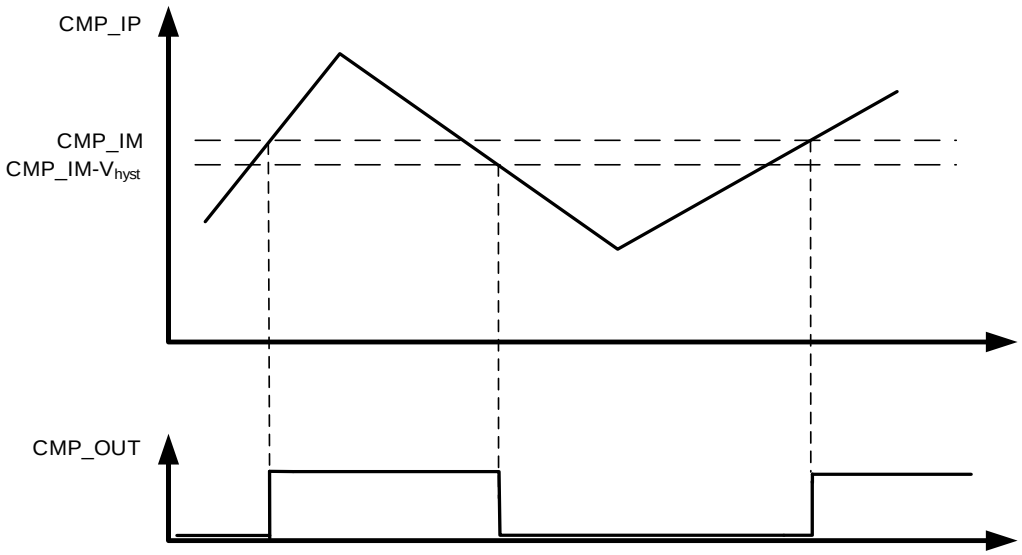
	CMP0	CMP1	CMP2	CMP3	CMP4	CMP5	CMP6	CMP7
CMP non inverting	PA1 PB1	PA7 PA3	PA0 PC1	PB0 PE7	PB13 PD12	PB11 PD11	PB14 PD14	PC2 PE9

	CMP0	CMP1	CMP2	CMP3	CMP4	CMP5	CMP6	CMP7
inputs connected to I/Os								
CMP inverting inputs connected to I/Os	PA4 PA0	PA5 PA2	PF1 PC0	PE8 PB2	PB10 PD13	PD10 PB15	PD15 PB12	PD8 PD9
CMP inverting inputs connected to internal signals	V_{REFINT}/4, V_{REFINT}/2, V_{REFINT}*3/4, V_{REFINT}, DAC2_OUT0 DAC0_OUT0	V_{REFINT}/4, V_{REFINT}/2, V_{REFINT}*3/4, V_{REFINT}, DAC2_OUT1 DAC0_OUT1	V_{REFINT}/4, V_{REFINT}/2, V_{REFINT}*3/4, V_{REFINT}, DAC2_OUT0 DAC0_OUT0	V_{REFINT}/4, V_{REFINT}/2, V_{REFINT}*3/4, V_{REFINT}, DAC2_OUT1 DAC0_OUT0	V_{REFINT}/4, V_{REFINT}/2, V_{REFINT}*3/4, V_{REFINT}, DAC3_OUT0 DAC0_OUT1	V_{REFINT}/4, V_{REFINT}/2, V_{REFINT}*3/4, V_{REFINT}, DAC3_OUT1 DAC1_OUT0	V_{REFINT}/4, V_{REFINT}/2, V_{REFINT}*3/4, V_{REFINT}, DAC3_OUT0 DAC1_OUT0	V_{REFINT}/4, V_{REFINT}/2, V_{REFINT}*3/4, V_{REFINT}, DAC3_OUT1 DAC1_OUT1
CMP outputs connected to I/Os	PA0 PA6 PA11 PB8 PF4	PA2 PA7 PA12 PB9	PB7 PB15 PC2	PB1 PB6 PB14	PA9 PC7	PC6 PA10	PC8 PA8	PA13 PA14
CMP outputs connected to EXTI	•							
CMP outputs connected to TRIGSEL	•							
CMP outputs connected to NVIC	•							
CMP outputs connected to internal signals	TIMER0, TIMER1, TIMER2, TIMER3, TIMER4, TIMER7, TIMER19, LPTIMER, HRTIMER							
CMP outputs connected to internal signals	BREAK0(TIMER0, TIMER7, TIMER14, TIMER15, TIMER16, TIMER19)							
	BREAK1(TIMER0, TIMER7, TIMER19)							

19.3.3. 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 19-2. CMP hysteresis



19.3.4. CMP register write protection

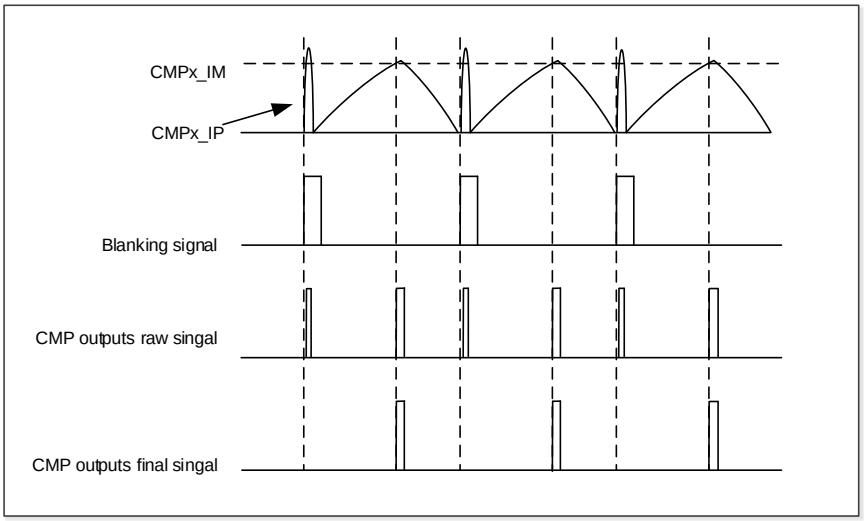
The CMP control and status register (CMPx_CS) can be protected from writing by setting CMPxLK bit to 1. The CMPx_CS register, including the CMPxLK bit will be read-only, and can only be reset by the MCU reset.

19.3.5. 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[2:0] bits in the CMPx_CS register are setting to an available value, the CMP output final signal is obtained by ANDing the complementary signal of the selected blanking signal with the raw output of the comparator. The blanking function can be used for false overcurrent detection in motor control applications.

Figure 19-3. The CMP outputs signal blanking shows the comparator output blank function.

Figure 19-3. The CMP outputs signal blanking



19.3.6. CMP voltage scaler function

The voltage scaler function can provide selectable 1 / 4, 1 / 2, 3 / 4 reference voltage for CMP input. It is controlled by CMPxSEN and CMPxBEN bits in CMPx control / status register. The CMPxSEN and CMPxBEN bits are used to enable the V_{REFINT} voltage output and the divider circuit, respectively, to generate the selected voltage.

19.3.7. CMP interrupt

The CMP output is connected to the EXTI and the EXTI line is exclusive to CMP. With this function, CMP can generate either interrupt or event which could be used to exit from low-power mode.

The CMP also can generate an interrupt to NVIC. It is a sequential logic signal, so the PCLK is needed.

19.3.8. CMP reset persist mode

The reset persist mode can be enabled by setting the CMPxRSTMD bit in the CMPx_CS register. Output channels will persist through all resets except for power-on-reset.

19.4. Register definition

CMP base address: 0x4001 7C00

19.4.1. CMP status register (CMP_STAT)

Address offset: 0x00

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								CMP7IF	CMP6IF	CMP5IF	CMP4IF	CMP3IF	CMP2IF	CMP1IF	CMP0IF
								r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								CMP7O	CMP6O	CMP5O	CMP4O	CMP3O	CMP2O	CMP1O	CMP0O
								r	r	r	r	r	r	r	r

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23	CMP7IF	CMP7 interrupt flag 0: No CMP7 output interrupt 1: CMP7 output interrupt Set by hardware when the CMP7 output is set. Cleared by software writing 1 to CMP7IC bit in the CMP_IFC register.
22	CMP6IF	CMP6 interrupt flag 0: No CMP6 output interrupt 1: CMP6 output interrupt Set by hardware when the CMP6 output is set. Cleared by software writing 1 to CMP6IC bit in the CMP_IFC register.
21	CMP5IF	CMP5 interrupt flag 0: No CMP5 output interrupt 1: CMP5 output interrupt Set by hardware when the CMP5output is set. Cleared by software writing 1 to CMP5IC bit in the CMP_IFC register.
20	CMP4IF	CMP4 interrupt flag 0: No CMP4 output interrupt 1: CMP4 output interrupt Set by hardware when the CMP4 output is set. Cleared by software writing 1 to CMP4IC bit in the CMP_IFC register.
19	CMP3IF	CMP3 interrupt flag 0: No CMP3 output interrupt

		1: CMP3 output interrupt Set by hardware when the CMP3 output is set. Cleared by software writing 1 to CMP3IC bit in the CMP_IFC register.
18	CMP2IF	CMP2 interrupt flag 0: No CMP2 output interrupt 1: CMP2 output interrupt Set by hardware when the CMP2 output is set. Cleared by software writing 1 to CMP2IC bit in the CMP_IFC register.
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.
15:8	Reserved	Must be kept at reset value.
7	CMP7O	CMP7 output state This bit is a copy of CMP7 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
6	CMP6O	CMP6 output state This bit is a copy of CMP6 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
5	CMP5O	CMP5 output state This bit is a copy of CMP5 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
4	CMP4O	CMP4 output state This bit is a copy of CMP4 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
3	CMP3O	CMP3 output state This bit is a copy of CMP3 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

2	CMP2O	CMP2 output state This bit is a copy of CMP2 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
1	CMP1O	CMP1 output state This bit is a copy of CMP1 output state, which is read only. 0: Non-inverting input below inverting input and the output is low 1: Non-inverting input above inverting input and the output is high
0	CMP0O	CMP0 output state This bit is a copy of CMP0 output state, which is read only. 0: Non-inverting input below inverting input and the output is low 1: Non-inverting input above inverting input and the output is high

19.4.2. CMP interrupt flag clear register (CMP_IFC)

Address offset: 0x04

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								CMP7IC	CMP6IC	CMP5IC	CMP4IC	CMP3IC	CMP2IC	CMP1IC	CMP0IC
								w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23	CMP7IC	CMP7 interrupt flag clear 0: Not clear CMP7 interrupt flag 1: Clear CMP7 interrupt flag
22	CMP6IC	CMP6 interrupt flag clear 0: Not clear CMP6 interrupt flag 1: Clear CMP6 interrupt flag
21	CMP5IC	CMP5 interrupt flag clear 0: Not clear CMP5 interrupt flag 1: Clear CMP5 interrupt flag
20	CMP4IC	CMP4 interrupt flag clear 0: Not clear CMP4 interrupt flag 1: Clear CMP4 interrupt flag

19	CMP3IC	CMP3 interrupt flag clear 0: Not clear CMP3 interrupt flag 1: Clear CMP3 interrupt flag
18	CMP2IC	CMP2 interrupt flag clear 0: Not clear CMP2 interrupt flag 1: Clear CMP2 interrupt flag
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.

19.4.3. CMP0 control/status register (CMP0_CS)

Address offset: 0x08

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CMP0LK	Reserved			CMP0BLK[3:0]				Reserved			CMP0PSEL	Reserved	CMP0MSEL[2:0]		
rw				rw							rw		rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					CMP0HST[2:0]			CMP0RSTMD	CMP0INTEN	Reserved		CMP0PL	CMP0SEN	CMP0BEN	CMP0EN
					rw			rw	rw			rw	rw	rw	rw

Bits	Fields	Descriptions
31	CMP0LK	CMP0 lock This bit allows to have all control bits of CMP0 as read-only. It can only be set once by software and cleared by a system reset. 0: CMP0_CS bits are read-write 1: CMP0_CS bits are read-only
30:28	Reserved	Must be kept at reset value.
27:24	CMP0BLK[3:0]	CMP0 output blanking source This bit is used to select which timer output controls the comparator output blanking. 0000: No blanking 0001: Select TIMER0_CH2 output compare signal as blanking source 0010: Select TIMER1_CH2 output compare signal as blanking source

		0011: Select TIMER2_CH2 output compare signal as blanking source 0100: Select TIMER7_CH2 output compare signal as blanking source 0101: Select TIMER19_CH2 output compare signal as blanking source 0110: Select TIMER14_CH0 output compare signal as blanking source 0111: Select TIMER3_CH2 output compare signal as blanking source All other values: reserved.
23:21	Reserved	Must be kept at reset value.
20	CMP0PSEL	CMP0_IP input selection This bit is used to select the source connected to the CMP0_IP input of the CMP0. 0: PA1 1: PB1
19	Reserved	Must be kept at reset value.
18:16	CMP0MSEL[2:0]	CMP0_IM internal input selection These bits are used to select the internal source connected to the CMP0_IM input of the CMP0. 000: V_{REFINT} / 4 001: V_{REFINT} / 2 010: V_{REFINT}*3 / 4 011: V_{REFINT} 100: DAC2_OUT0 101: DAC0_OUT0 110: PA4 111: PA0
15:11	Reserved	Must be kept at reset value.
10:8	CMP0HST[2:0]	CMP0 hysteresis These bits are used to control the hysteresis level. 000: No hysteresis 001: 10 mv 010: 20 mv 011: 30 mv 100: 40 mv 101: 50 mv 110: 60 mv 111: 70 mv
7	CMP0RSTMD	CMP0 reset mode. This bit can only be reset by power on reset 0: Normal mode. All resets will reset CMP0 and its associated registers 1: Reset persist mode. CMP0 output will persist through all resets except for power-on-resets
6	CMP0INTEN	CMP0 interrupt enable

		0: Disabled 1: Enabled
5:4	Reserved	Must be kept at reset value.
3	CMP0PL	Polarity of CMP0 output This bit is used to select the polarity of CMP0 output. 0: Output is not inverted 1: Output is inverted
2	CMP0SEN	Voltage scaler enable bit This bit is set and cleared by software. This bit enables the outputs of the VREFINT divider, which is treated as the minus input of the comparator. 0: Disable bandgap scaler in case that CMP1SEN bit of CMP1_CS is also reset 1: Enable bandgap scaler enable
1	CMP0BEN	Scaler bridge enable bit 0: Disable scaler resistor bridge in case that CMP1BEN bit of CMP01_CS is also reset 1: Enable scaler resistor bridge
0	CMP0EN	CMP0 enable 0: CMP0 disabled 1: CMP0 enabled

19.4.4. CMP1 control/status register (CMP1_CS)

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
CMP1LK	Reserved			CMP1BLK[3:0]				Reserved			CMP1PSEL	Reserved	CMP1MSEL[2:0]		
rw				rw							rw		rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					CMP1HST[2:0]			CMP1RSTMD	CMP1INTEN	Reserved		CMP1PL	CMP1SEN	CMP1BEN	CMP1EN
					rw			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. It can only be set once by software and cleared by a system reset. 0: CMP1_CS bits are read-write 1: CMP1_CS bits are read-only

30:28	Reserved	Must be kept at reset value.
27:24	CMP1BLK[3:0]	CMP1 output blanking source This bit is used to select which timer output controls the comparator output blanking. 0000: No blanking 0001: Select TIMER0_CH2 output compare signal as blanking source 0010: Select TIMER1_CH2 output compare signal as blanking source 0011: Select TIMER2_CH2 output compare signal as blanking source 0100: Select TIMER7_CH2 output compare signal as blanking source 0101: Select TIMER19_CH2 output compare signal as blanking source 0110: Select TIMER14_CH0 output compare signal as blanking source 0111: Select TIMER3_CH2 output compare signal as blanking source All other values: reserved.
23:21	Reserved	Must be kept at reset value.
20	CMP1PSEL	CMP1_IP input selection This bit is used to select the source connected to the CMP1_IP input of the CMP1. 0: PA7 1: PA3
19	Reserved	Must be kept at reset value.
18:16	CMP1MSEL[2:0]	CMP1_IM internal input selection These bits are used to select the internal source connected to the CMP1_IM input of the CMP1. 000: V_{REFINT} / 4 001: V_{REFINT} / 2 010: V_{REFINT}*3 / 4 011: V_{REFINT} 100: DAC2_OUT1 101: DAC0_OUT1 110: PA5 111: PA2
15:11	Reserved	Must be kept at reset value.
10:8	CMP1HST[2:0]	CMP1 hysteresis These bits are used to control the hysteresis level. 000: No hysteresis 001: 10 mv 010: 20 mv 011: 30 mv 100: 40 mv 101: 50 mv 110: 60 mv

		111: 70 mv
7	CMP1RSTMD	CMP1 reset mode. This bit can only be reset by power on reset 0: Normal mode. All resets will reset CMP1 and its associated registers 1: Reset persist mode. CMP1 output will persist through all resets except for power-on-resets
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	CMP1SEN	Voltage scaler enable bit This bit is set and cleared by software. This bit enables the outputs of the VREFINT divider, which is treated as the minus input of the comparator. 0: Disable bandgap scaler disable in case that CMP0SEN bit of CMP0_CS is also reset 1: Enable bandgap scaler enable
1	CMP1BEN	Scaler bridge enable bit 0: Disable scaler resistor bridge in case that CMP0BEN bit of CMP0_CS is also reset 1: Enable scaler resistor bridge
0	CMP1EN	CMP1 enable 0: CMP1 disabled 1: CMP1 enabled

19.4.5. CMP2 control/status register (CMP2_CS)

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
CMP2LK	Reserved			CMP2BLK[3:0]				Reserved			CMP2PSEL	Reserved	CMP2MSEL[2:0]		
rw				rw							rw		rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					CMP2HST[2:0]			CMP2RSTMD	CMP2INTEN	Reserved		CMP2PL	CMP2SEN	CMP2BEN	CMP2EN

rw rw rw rw rw rw rw

Bits	Fields	Descriptions
31	CMP2LK	CMP2 lock This bit allows to have all control bits of CMP2 as read-only. It can only be set once by software and cleared by a system reset. 0: CMP2_CS bits are read-write 1: CMP2_CS bits are read-only
30:28	Reserved	Must be kept at reset value.
27:24	CMP2BLK[3:0]	CMP2 output blanking source This bit is used to select which timer output controls the comparator output blanking. 0000: No blanking 0001: Select TIMER0_CH2 output compare signal as blanking source 0010: Select TIMER2_CH2 output compare signal as blanking source 0011: Select TIMER1_CH3 output compare signal as blanking source 0100: Select TIMER7_CH2 output compare signal as blanking source 0101: Select TIMER19_CH2 output compare signal as blanking source 0110: Select TIMER14_CH0 output compare signal as blanking source 0111: Select TIMER3_CH2 output compare signal as blanking source All other values: reserved.
23:21	Reserved	Must be kept at reset value.
20	CMP2PSEL	CMP2_IP input selection This bit is used to select the source connected to the CMP2_IP input of the CMP2. 0: PA0 1: PC1
19	Reserved	Must be kept at reset value.
18:16	CMP2MSEL[2:0]	CMP2_IM internal input selection These bits are used to select the internal source connected to the CMP2_IM input of the CMP2. 000: V_{REFINT} / 4 001: V_{REFINT} / 2 010: V_{REFINT}*3 / 4 011: V_{REFINT} 100: DAC2_OUT0 101: DAC0_OUT0 110: PF1 111: PC0
15:11	Reserved	Must be kept at reset value.
10:8	CMP2HST[2:0]	CMP2 hysteresis

		These bits are used to control the hysteresis level. 000: No hysteresis 001: 10 mv 010: 20 mv 011: 30 mv 100: 40 mv 101: 50 mv 110: 60 mv 111: 70 mv
7	CMP2RSTMD	CMP2 reset mode. This bit can only be reset by power on reset 0: Normal mode. All resets will reset CMP2 and its associated registers 1: Reset persist mode. CMP2 output will persist through all resets except for power-on-resets
6	CMP2INTEN	CMP2 interrupt enable 0: Disabled 1: Enabled
5:4	Reserved	Must be kept at reset value.
3	CMP2PL	Polarity of CMP2 output This bit is used to select the polarity of CMP2 output. 0: Output is not inverted 1: Output is inverted
2	CMP2SEN	Voltage scaler enable bit This bit is set and cleared by software. This bit enables the outputs of the VREFINT divider, which is treated as the minus input of the comparator. 0: Disable bandgap scaler in case that CMP3SEN bit of CMP3_CS is also reset 1: Enable bandgap scaler enable
1	CMP2BEN	Scaler bridge enable bit 0: Disable scaler resistor bridge in case that CMP3BEN bit of CMP3_CS is also reset 1: Enable scaler resistor bridge
0	CMP2EN	CMP2 enable 0: CMP2 disabled 1: CMP2 enabled

19.4.6. CMP3 control/status register (CMP3_CS)

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
CMP3LK	Reserved			CMP3BLK[3:0]				Reserved			CMP3PSEL	Reserved	CMP3MSEL[2:0]		
rw				rw							rw		rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					CMP3HST[2:0]			CMP3RSTMD	CMP3INTEN	Reserved		CMP3PL	CMP3SEN	CMP3BEN	CMP3EN
					rw			rw	rw			rw	rw	rw	rw

Bits	Fields	Descriptions
31	CMP3LK	CMP3 lock This bit allows to have all control bits of CMP3 as read-only. It can only be set once by software and cleared by a system reset. 0: CMP3_CS bits are read-write 1: CMP3_CS bits are read-only
30:28	Reserved	Must be kept at reset value.
27:24	CMP3BLK[3:0]	CMP3 output blanking source This bit is used to select which timer output controls the comparator output blanking. 0000: No blanking 0001: Select TIMER2_CH3 output compare signal as blanking source 0010: Select TIMER7_CH2 output compare signal as blanking source 0011: Select TIMER14_CH1 output compare signal as blanking source 0100: Select TIMER0_CH2 output compare signal as blanking source 0101: Select TIMER19_CH2 output compare signal as blanking source 0110: Select TIMER14_CH0 output compare signal as blanking source 0111: Select TIMER3_CH2 output compare signal as blanking source All other values: reserved.
23:21	Reserved	Must be kept at reset value.
20	CMP3PSEL	CMP3_IP input selection This bit is used to select the source connected to the CMP3_IP input of the CMP3. 0: PB0 1: PE7
19	Reserved	Must be kept at reset value.
18:16	CMP3MSEL[2:0]	CMP3_IM internal input selection These bits are used to select the internal source connected to the CMP3_IM input of the CMP3. 000: V_{REFINT} / 4 001: V_{REFINT} / 2 010: V_{REFINT}*3 / 4 011: V_{REFINT}

		100: DAC2_OUT1 101: DAC0_OUT0 110: PE8 111: PB2
15:11	Reserved	Must be kept at reset value.
10:8	CMP3HST[2:0]	CMP3 hysteresis These bits are used to control the hysteresis level. 000: No hysteresis 001: 10 mv 010: 20 mv 011: 30 mv 100: 40 mv 101: 50 mv 110: 60 mv 111: 70 mv
7	CMP3RSTMD	CMP3 reset mode. This bit can only be reset by power on reset 0: Normal mode. All resets will reset CMP3 and its associated registers 1: Reset persist mode. CMP3 output will persist through all resets except for power-on-resets
6	CMP3INTEN	CMP3 interrupt enable 0: Disabled 1: Enabled
5:4	Reserved	Must be kept at reset value.
3	CMP3PL	Polarity of CMP3 output This bit is used to select the polarity of CMP3 output. 0: Output is not inverted 1: Output is inverted
2	CMP3SEN	Voltage scaler enable bit This bit is set and cleared by software. This bit enables the outputs of the VREFINT divider, which is treated as the minus input of the comparator. 0: Disable bandgap scaler disable in case that CMP2SEN bit of CMP2_CS is also reset 1: Enable bandgap scaler enable
1	CMP3BEN	Scaler bridge enable bit 0: Disable scaler resistor bridge in case that CMP2BEN bit of CMP2_CS is also reset 1: Enable scaler resistor bridge
0	CMP3EN	CMP3 enable 0: CMP3 disabled

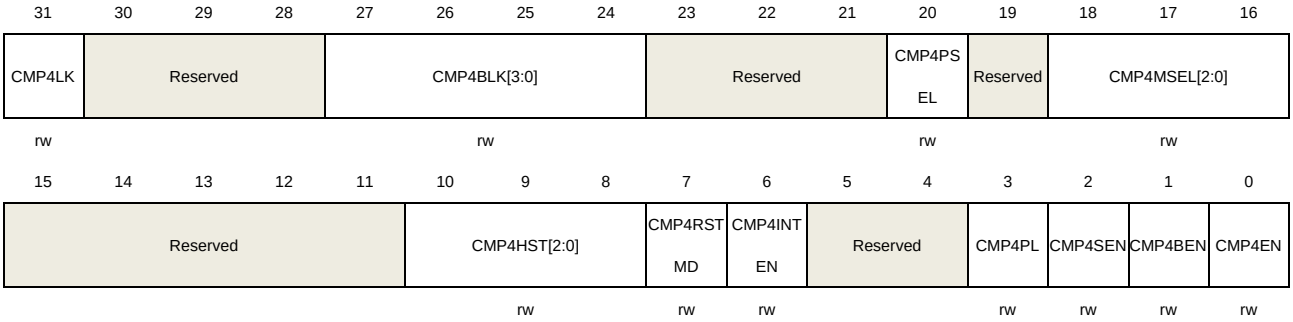
1: CMP3 enabled

19.4.7. CMP4 control/status register (CMP4_CS)

Address offset: 0x18

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31	CMP4LK	CMP4 lock This bit allows to have all control bits of CMP4 as read-only. It can only be set once by software and cleared by a system reset. 0: CMP4_CS bits are read-write 1: CMP4_CS bits are read-only
30:28	Reserved	Must be kept at reset value.
27:24	CMP4BLK[3:0]	CMP4 output blanking source This bit is used to select which timer output controls the comparator output blanking. 0000: No blanking 0001: Select TIMER1_CH2 output compare signal as blanking source 0010: Select TIMER7_CH2 output compare signal as blanking source 0011: Select TIMER2_CH2 output compare signal as blanking source 0100: Select TIMER0_CH2 output compare signal as blanking source 0101: Select TIMER19_CH2 output compare signal as blanking source 0110: Select TIMER14_CH0 output compare signal as blanking source 0111: Select TIMER3_CH2 output compare signal as blanking source All other values: reserved.
23:21	Reserved	Must be kept at reset value.
20	CMP4PSEL	CMP4_IP input selection This bit is used to select the source connected to the CMP4_IP input of the CMP4. 0: PB13 1: PD12
19	Reserved	Must be kept at reset value.

18:16	CMP4MSEL[2:0]	CMP4_IM internal input selection These bits are used to select the internal source connected to the CMP4_IM input of the CMP4. 000: V_{REFINT} / 4 001: V_{REFINT} / 2 010: V_{REFINT}*3 / 4 011: V_{REFINT} 100: DAC3_OUT0 101: DAC0_OUT1 110: PB10 111: PD13
15:11	Reserved	Must be kept at reset value.
10:8	CMP4HST[2:0]	CMP4 hysteresis These bits are used to control the hysteresis level. 000: No hysteresis 001: 10 mv 010: 20 mv 011: 30 mv 100: 40 mv 101: 50 mv 110: 60 mv 111: 70 mv
7	CMP4RSTMD	CMP4 reset mode. This bit can only be reset by power on reset 0: Normal mode. All resets will reset CMP4 and its associated registers 1: Reset persist mode. CMP4 output will persist through all resets except for power-on-resets
6	CMP4INTEN	CMP4 interrupt enable 0: Disabled 1: Enabled
5:4	Reserved	Must be kept at reset value.
3	CMP4PL	Polarity of CMP4 output This bit is used to select the polarity of CMP4 output. 0: Output is not inverted 1: Output is inverted
2	CMP3SEN	Voltage scaler enable bit This bit is set and cleared by software. This bit enables the outputs of the VREFINT divider, which is treated as the minus input of the comparator. 0: Disable bandgap scaler disable in case that CMP5SEN bit of CMP5_CS is also reset

		1: Enable bandgap scaler enable
1	CMP3BEN	Scaler bridge enable bit 0: Disable scaler resistor bridge in case that CMP5BEN bit of CMP5_CS is also reset 1: Enable scaler resistor bridge
0	CMP4EN	CMP4 enable 0: CMP4 disabled 1: CMP4 enabled

19.4.8. CMP5 control/status register (CMP5_CS)

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
CMP5LK	Reserved			CMP5BLK[3:0]				Reserved			CMP5PSEL	Reserved	CMP5MSEL[2:0]		
rw				rw							rw		rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					CMP5HST[2:0]			CMP5RSTMD	CMP5INTEN	Reserved		CMP5PL	CMP5SEN	CMP5BEN	CMP5EN
					rw			rw	rw			rw	rw	rw	rw

Bits	Fields	Descriptions
31	CMP5LK	CMP5 lock This bit allows to have all control bits of CMP5 as read-only. It can only be set once by software and cleared by a system reset. 0: CMP5_CS bits are read-write 1: CMP5_CS bits are read-only
30:28	Reserved	Must be kept at reset value.
27:24	CMP5BLK[3:0]	CMP5 output blanking source This bit is used to select which timer output controls the comparator output blanking. 0000: No blanking 0001: Select TIMER7_CH2 output compare signal as blanking source 0010: Select TIMER1_CH3 output compare signal as blanking source 0011: Select TIMER14_CH1 output compare signal as blanking source 0100: Select TIMER0_CH2 output compare signal as blanking source 0101: Select TIMER19_CH2 output compare signal as blanking source 0110: Select TIMER14_CH0 output compare signal as blanking source 0111: Select TIMER3_CH2 output compare signal as blanking source

		All other values: reserved.
23:21	Reserved	Must be kept at reset value.
20	CMP5PSEL	CMP5_IP input selection This bit is used to select the source connected to the CMP5_IP input of the CMP5. 0: PB11 1: PD11
19	Reserved	Must be kept at reset value.
18:16	CMP5MSEL[2:0]	CMP5_IM internal input selection These bits are used to select the internal source connected to the CMP5_IM input of the CMP5. 000: V_{REFINT} / 4 001: V_{REFINT} / 2 010: V_{REFINT}*3 / 4 011: V_{REFINT} 100: DAC3_OUT1 101: DAC1_OUT0 110: PD10 111: PB15
15:11	Reserved	Must be kept at reset value.
10:8	CMP5HST[2:0]	CMP5 hysteresis These bits are used to control the hysteresis level. 000: No hysteresis 001: 10 mv 010: 20 mv 011: 30 mv 100: 40 mv 101: 50 mv 110: 60 mv 111: 70 mv
7	CMP5RSTMD	CMP5 reset mode. This bit can only be reset by power on reset 0: Normal mode. All resets will reset CMP5 and its associated registers 1: Reset persist mode. CMP5 output will persist through all resets except for power-on-resets
6	CMP5INTEN	CMP5 interrupt enable 0: Disabled 1: Enabled
5:4	Reserved	Must be kept at reset value.
3	CMP5PL	Polarity of CMP5 output

		This bit is used to select the polarity of CMP5 output. 0: Output is not inverted 1: Output is inverted
2	CMP5SEN	Voltage scaler enable bit This bit is set and cleared by software. This bit enables the outputs of the VREFINT divider, which is treated as the minus input of the comparator. 0: Disable bandgap scaler disable in case that CMP4SEN bit of CMP4_CS is also reset 1: Enable bandgap scaler enable
1	CMP5BEN	Scaler bridge enable bit 0: Disable scaler resistor bridge in case that CMP4BEN bit of CMP4_CS is also reset 1: Enable scaler resistor bridge
0	CMP5EN	CMP5 enable 0: CMP5 disabled 1: CMP5 enabled

19.4.9. CMP6 control/status register (CMP6_CS)

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
CMP6LK	Reserved			CMP6BLK[3:0]				Reserved			CMP6PSEL	Reserved	CMP6MSEL[2:0]		
rw				rw							rw		rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					CMP6HST[2:0]			CMP6RSTMD	CMP6INTEN	Reserved		CMP6PL	CMP6SEN	CMP6BEN	CMP6EN
					rw			rw	rw			rw	rw	rw	rw

Bits	Fields	Descriptions
31	CMP6LK	CMP6 lock This bit allows to have all control bits of CMP6 as read-only. It can only be set once by software and cleared by a system reset. 0: CMP6_CS bits are read-write 1: CMP6_CS bits are read-only
30:28	Reserved	Must be kept at reset value.
27:24	CMP6BLK[3:0]	CMP6 output blanking source This bit is used to select which timer output controls the comparator output blanking.

		0000: No blanking 0001: Select TIMER0_CH2output compare signal as blanking source 0010: Select TIMER7_CH2 output compare signal as blanking source 0011: Select TIMER2_CH2 output compare signal as blanking source 0100: Select TIMER14_CH1 output compare signal as blanking source 0101: Select TIMER19_CH2 output compare signal as blanking source 0110: Select TIMER14_CH0 output compare signal as blanking source 0111: Select TIMER3_CH2 output compare signal as blanking source All other values: reserved.
23:21	Reserved	Must be kept at reset value.
20	CMP6PSEL	CMP6_IP input selection This bit is used to select the source connected to the CMP6_IP input of the CMP6. 0: PB14 1: PD14
19	Reserved	Must be kept at reset value.
18:16	CMP6MSEL[2:0]	CMP6_IM internal input selection These bits are used to select the internal source connected to the CMP6_IM input of the CMP6. 000: V_{REFINT} / 4 001: V_{REFINT} / 2 010: V_{REFINT}*3 / 4 011: V_{REFINT} 100: DAC3_OUT0 101: DAC1_OUT0 110: PD15 111: PB12
15:11	Reserved	Must be kept at reset value.
10:8	CMP6HST[2:0]	CMP6 hysteresis These bits are used to control the hysteresis level. 000: No hysteresis 001: 10 mv 010: 20 mv 011: 30 mv 100: 40 mv 101: 50 mv 110: 60 mv 111: 70 mv
7	CMP6RSTMD	CMP6 reset mode. This bit can only be reset by power on reset 0: Normal mode. All resets will reset CMP6 and its associated registers 1: Reset persist mode. CMP6 output will persist through all resets except for power-

		on-resets
6	CMP6INTEN	CMP6 interrupt enable 0: Disabled 1: Enabled
5:4	Reserved	Must be kept at reset value.
3	CMP6PL	Polarity of CMP6 output This bit is used to select the polarity of CMP6 output. 0: Output is not inverted 1: Output is inverted
2	CMP6SEN	Voltage scaler enable bit This bit is set and cleared by software. This bit enables the outputs of the VREFINT divider, which is treated as the minus input of the comparator. 0: Disable bandgap scaler disable in case that CMP7SEN bit of CMP7_CS is also reset 1: Enable bandgap scaler enable
1	CMP6BEN	Scaler bridge enable bit 0: Disable scaler resistor bridge in case that CMP7BEN bit of CMP7_CS is also reset 1: Enable scaler resistor bridge
0	CMP6EN	CMP6 enable 0: CMP6 disabled 1: CMP6 enabled

19.4.10. CMP7 control/status register (CMP7_CS)

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
CMP7LK	Reserved			CMP7BLK[3:0]				Reserved		CMP7PSEL[1:0]		Reserved	CMP7MSEL[2:0]		
rw				rw						rw			rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					CMP7HST[2:0]			CMP7RSTMD	CMP7INTEN	Reserved		CMP7PL	CMP7SEN	CMP7BEN	CMP7EN
					rw			rw	rw			rw	rw	rw	rw

Bits	Fields	Descriptions
31	CMP7LK	CMP7 lock This bit allows to have all control bits of CMP7 as read-only. It can only be set once by software and cleared by a system reset.

		0: CMP7_CS bits are read-write 1: CMP7_CS bits are read-only
30:28	Reserved	Must be kept at reset value.
27:24	CMP7BLK[3:0]	CMP7 output blanking source This bit is used to select which timer output controls the comparator output blanking. 0000: No blanking 0001: Select TIMER1_CH2output compare signal as blanking source 0010: Select TIMER7_CH2 output compare signal as blanking source 0011: Select TIMER2_CH2 output compare signal as blanking source 0100: Select TIMER0_CH2 output compare signal as blanking source 0101: Select TIMER19_CH2 output compare signal as blanking source 0110: Select TIMER14_CH0 output compare signal as blanking source 0111: Select TIMER3_CH2 output compare signal as blanking source All other values: reserved.
23:21	Reserved	Must be kept at reset value.
20	CMP7PSEL	CMP7_IP input selection This bit is used to select the source connected to the CMP7_IP input of the CMP7. 0: PC2 1: PE9
19	Reserved	Must be kept at reset value.
18:16	CMP7MSEL[2:0]	CMP7_IM internal input selection These bits are used to select the internal source connected to the CMP7_IM input of the CMP7. 000: V_{REFINT} / 4 001: V_{REFINT} / 2 010: V_{REFINT}*3 / 4 011: V_{REFINT} 100: DAC3_OUT1 101: DAC1_OUT1 110: PD8 111: PD9
15:11	Reserved	Must be kept at reset value.
10:8	CMP7HST[2:0]	CMP7 hysteresis These bits are used to control the hysteresis level. 000: No hysteresis 001: 10 mv 010: 20 mv 011: 30 mv 100: 40 mv

		101: 50 mv 110: 60 mv 111: 70 mv
7	CMP7RSTMD	CMP7 reset mode. This bit can only be reset by power on reset 0: Normal mode. All resets will reset CMP7 and its associated registers 1: Reset persist mode. CMP7 output will persist through all resets except for power-on-resets
6	CMP7INTEN	CMP7 interrupt enable 0: Disabled 1: Enabled
5:4	Reserved	Must be kept at reset value.
3	CMP7PL	Polarity of CMP7 output This bit is used to select the polarity of CMP7 output. 0: Output is not inverted 1: Output is inverted
2	CMP7SEN	Voltage scaler enable bit This bit is set and cleared by software. This bit enables the outputs of the VREFINT divider, which is treated as the minus input of the comparator. 0: Disable bandgap scaler disable in case that CMP6SEN bit of CMP6_CS is also reset 1: Enable bandgap scaler enable
1	CMP7BEN	Scaler bridge enable bit 0: Disable scaler resistor bridge in case that CMP6BEN bit of CMP6_CS is also reset 1: Enable scaler resistor bridge
0	CMP7EN	CMP7 enable 0: CMP7 disabled 1: CMP7 enabled

20. VREF

20.1. Overview

A precision internal reference circuit is inside. The internal voltage reference unit is used to provide voltage reference for ADC / DAC, or used by off-chip circuit connecting to VREFP pin.

20.2. Characteristics

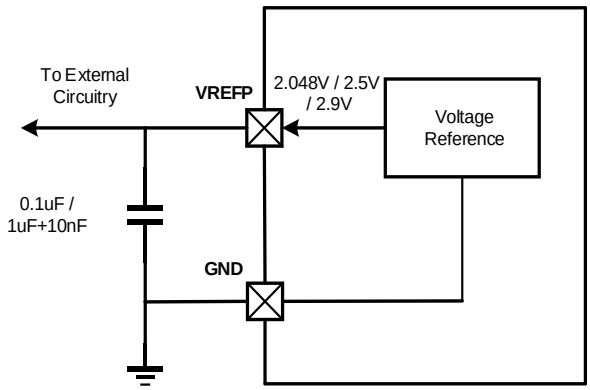
- Stable voltage, and product calibrated.
- Connects to VREFP pin to source off-chip circuits.
- 2.048V, 2.5V or 2.9V configurable reference voltage output.

20.3. Function overview

The VREF is enabled by set the VREFEN bit in VREF_CS register, and the VREF output can be configured to be either 2.048V, 2.5V, or 2.9V by programming the VREFS[1:0] bits. When VREF is enabled and the HIPM bit is reset, the internal voltage reference can be connected to VREFP pin. When VREF is disabled and the HIPM bit is set, off-chip voltage reference can be injected to VREFP pin to source ADC / DAC. If there is no VREFP pin (refer to datasheet), the VREFP is connected to VDDA and the VREFEN bit must keep 0.

When using precision internal voltage reference, and a bypass capacitor about 0.1uF (or 0.1uF and 10nF connected in parallel) which is recommended to ground is required.

Figure 20-1.Precision reference connection



As shown in ***Table 20-1. VREF modes***, the precision internal reference voltage unit can work in four kinds of mode by programming the VREFEN and HIPM bits in the VREF_CS register.

Table 20-1. VREF modes

VREFEN	HIPM	Mode
0	0	VREF disabled, VREFP pin pulled-down to V_{SSA}.
0	1	External voltage reference mode (default): VREF disabled. off-chip reference voltage injected from VREFP pin.
1	0	Internal voltage reference mode: VREF enabled. VREFP pin inside connected to VREF output.
1	1	Hold mode: VREF disabled. VREFP pin floating. The voltage is maintained by the external capacitor. VREFRDY detection disabled and VREFRDY bit keeps last state.

When VREF is configured in internal voltage reference mode by setting VREFEN bit and reset HIPM bit in the VREF_CS register, the user must wait before VREFRDY bit is set, indicating that the VREF output has attained the set value.

20.3.1. VREF calibration

During the course of production test, the internal voltage reference has been calibrated, and each voltage(2.048V, 2.5V, and 2.9V) has its corresponding calibration value saved in the Flash memory. On reset, or when the selected voltage reference changes (controlled by the VREFS[1:0] bits), the VREFCAL[5:0] bits in the VREF_CALIB register will be automatically initialized to the corresponding calibration value.

Writing to VREFCAL[5:0] bits can adjusts the output of VREF. After writing to these bits, the calibration value will not be automatically initialized when the selected voltage reference changes until the MCU is reset.

20.4. Register definition

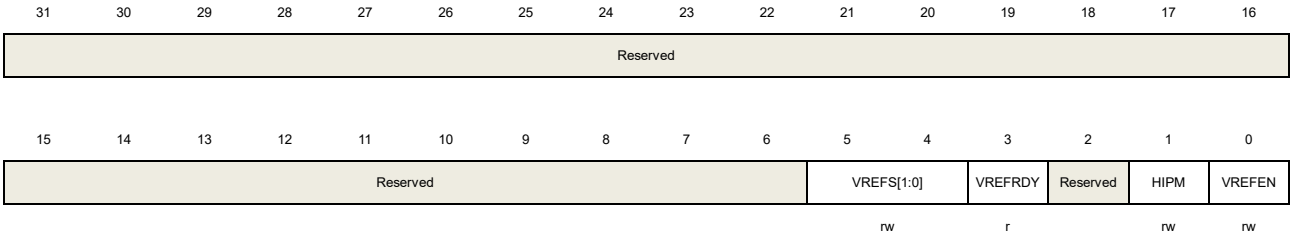
VREF base address: 0x4001 7800

When the DRSTMDy(y=0,1) bit of any DACx(x=0,1,2,3) is set, the VREF registers bit will be retained across all reset events except POR.

20.4.1. Control and status register (VREF_CS)

Address offset: 0x00
Reset value: 0x0000 0002

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



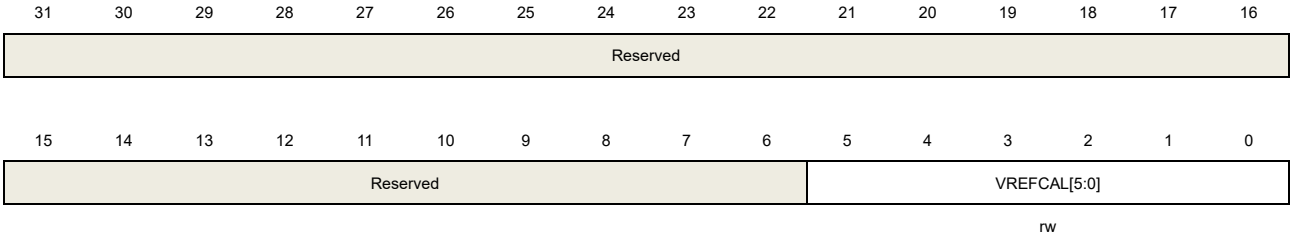
Bits	Fields	Descriptions
31:6	Reserved	Must be kept at reset value.
5:4	VREFS[1:0]	Voltage reference select These bits set the value of voltage reference output by the VREF. 00: Voltage reference set to 2.048 V 01: Voltage reference set to 2.5 V 10: Voltage reference set to 2.90 V 11: Reserved **Note:**This bit can be modified only when the VREF is disabled (VREFEN = 0).
3	VREFRDY	VREF ready 0: The output of the VREF does not attain the set value 1: The output of the VREF attains the set value
2	Reserved	Must be kept at reset value.
1	HIPM	High impedance state 0: The VREFP pin is inside connected to the VREF output 1: The VREFP pin is set to high impedance state
0	VREFEN	VREF enable 0: VREF is disabled 1: VREF is enabled

20.4.2. Calibration register (VREF_CALIB)

Address offset: 0x04

Reset value: 0x0000 00XX

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



Bits	Fields	Descriptions
31:6	Reserved	Must be kept at reset value.
5:0	VREFCAL[5:0]	VREF calibration On reset, or when the selected voltage reference changes (controlled by the VREFS[1:0] bits), these bits will be automatically initialized to the corresponding calibration value saved in the Flash memory during the course of production test. Writing to these bits also can adjust the output of the VREF. After writing to these bits, the calibration value will not be automatically initialized when the selected voltage reference changes until the MCU is reset. **Note:** If the user performs the calibration procedure, the VREF calibration must increase progressively from 0x00 to 0x3F.

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 (or an interrupt in window watchdog timer) 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 (IRC32K). Thereupon the FWDGT can operate even if the main clock fails. It's suitable for the situation that requires an independent environment and lower timing accuracy.

The Free watchdog timer generate a reset when the internal down counter reaches 0 or the counter is refreshed when the value of the counter is greater than the window register value. The register write protection function in free watchdog can be enabled to prevent it from changing the configuration unexpectedly.

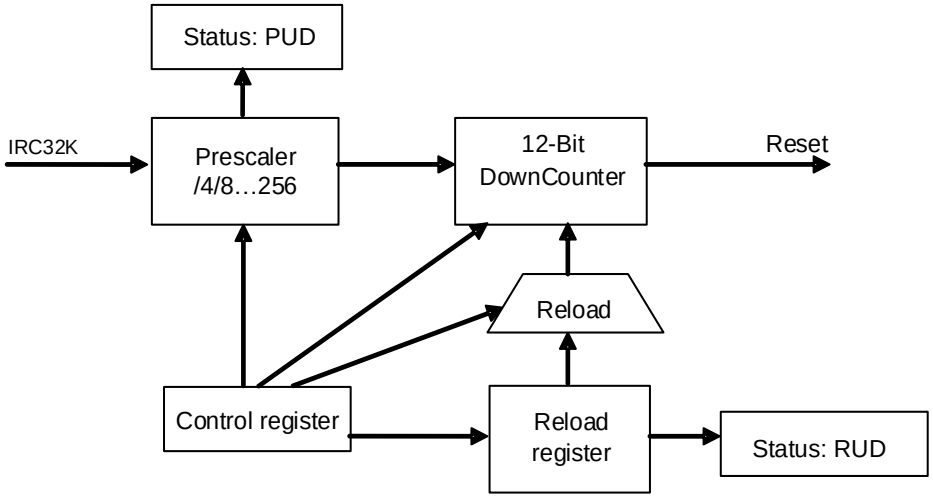
21.1.2. Characteristics

- 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 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.
- Configure FWDGSPD_STDBY or FWDGSPD_DPSLP, FWDGT would stop working, or else wake up the device and continue to work, in standby or deep sleep 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 bit in the device option bits is set. To avoid reset, the software should reload the counter before the counter reaches 0x000.

The FWDGT_PSC register, the FWDGT_RLD register and the FWDGT_WND register are write protected. Before writing these registers, the software should write the value (0x5555) to the FWDGT_CTL register. These registers will be protected again by writing any other value to the FWDGT_CTL register. When an update operation of the prescaler register (FWDGT_PSC), window register (FWDGT_WND) or the reload value register (FWDGT_RLD) is ongoing, the status bits in the FWDGT_STAT register are set.

If the FWDGT_HOLD bit in DBG module is cleared, the FWDGT continues to work even the Cortex®-M33 core halted (Debug mode). The FWDGT stops in Debug mode if the FWDGT_HOLD bit is set.

Table 21-1. Min/max FWDGT timeout period at 32KHz (IRC32K)

Prescaler divider	PSC[2:0] bits	Min timeout (ms) RLD[11:0]= 0x000	Max timeout (ms) RLD[11:0]= 0xFFF
1/4	000	0.125	512
1/8	001	0.25	1024
1/16	010	0.5	2048
1/32	011	1.0	4096
1/64	100	2.0	8192
1/128	101	4.0	16384
1/256	110 or 111	8.0	32768

The FWDGT timeout can be more accurately by calibrating the IRC32K.

Note: When after the execution of watchdog reload operation, if the MCU needs enter the deepsleep/standby mode immediately, more than 3 IRC32K clock intervals must be inserted in the middle of reload and deepsleep/standby mode commands by software setting.

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 half-word(16-bit) or word(32-bit).

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

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

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CMD[15:0]	Write only. Several different functions are realized by writing these bits with different values. 0x5555: Disable the FWDGT_PSC, FWDGT_RLD and FWDGT_WND write protection. 0xCCCC: Start the free watchdog timer counter. When the counter reduces to 0, the free watchdog generates a reset 0xAAAA: Reload the counter

Prescaler register (FWDGT_PSC)

Address offset: 0x04
Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value.
2:0	PSC[2:0]	Free watchdog timer prescaler selection. Write 0x5555 in the FWDGT_CTL register before writing these bits. During a write operation to this register, the PUD bit in the

FWDGT_STAT register is set and the value read from this register is invalid.

000: 1/4

001: 1/8

010: 1/16

011: 1/32

100: 1/64

101: 1/128

110: 1/256

111: 1/256

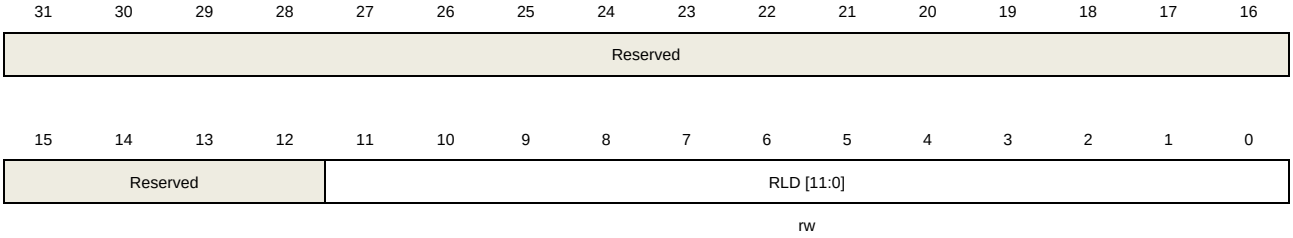
If several prescaler values are used by the application, it is mandatory to wait until PUD bit has been reset before changing the prescaler value. If the prescaler value has been updated, it is not necessary to wait until PUD has been reset before continuing code execution (Before entering low-power mode, it is necessary to wait until PUD is reset).

Reload register (FWDGT_RLD)

Address offset: 0x08

Reset value: 0x0000 0FFF

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



Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11:0	RLD[11:0]	Free watchdog timer counter reload value. Write 0xAAAA in the FWDGT_CTL register will reload the FWDGT conter with the RLD value. These bits are write-protected. Write 0X5555 to the FWDGT_CTL register before writing these bits. During a write operation to this register, the RUD bit in the FWDGT_STAT register is set and the value read from this register is invalid. If several reload values are used by the application, it is mandatory to wait until RUD bit has been reset before changing the reload value. If the reload value has been updated, it is not necessary to wait until RUD has been reset before continuing code execution (Before entering low-power mode, it is necessary to wait until RUD is reset).

Status register (FWDGT_STAT)

Address offset: 0x0C

Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value.
2	WUD	Watchdog counter window value update When a write operation to FWDGT_WND register ongoing, this bit is set and the value read from FWDGT_WND register is invalid.
1	RUD	Free watchdog timer counter reload value update During a write operation to FWDGT_RLD register, this bit is set and the value read from FWDGT_RLD register is invalid.
0	PUD	Free watchdog timer prescaler value update During a write operation to FWDGT_PSC register, this bit is set and the value read from FWDGT_PSC register is invalid.

Window register (FWDGT_WND)

Address offset: 0x10

Reset value: 0x0000 0FFF

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

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

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

Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11:0	WND[11:0]	Watchdog counter window value. These bits are used to contain the high limit of the window value to be compared to the downcounter. A reset will occur if the reload operation is performed while the counter is greater than the value stored in this register. The WUD bit in the FWDGT_STAT register must be reset in order to be able to change the reload value.

These bits are write protected. Write 0x5555 in the FWDGT_CTL register before writing these bits.

If several window values are used by the application, it is mandatory to wait until WUD bit has been reset before changing the window value. However, after updating the window value it is not necessary to wait until WUD is reset before continuing code execution except in case of low-power mode entry(Before entering low-power mode, it is necessary to wait until WUD is reset).

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, interrupt occurs if it is enabled.

The window watchdog timer clock is prescaled from the APB1 clock. The window watchdog timer is suitable for the situation that requires an accurate timing.

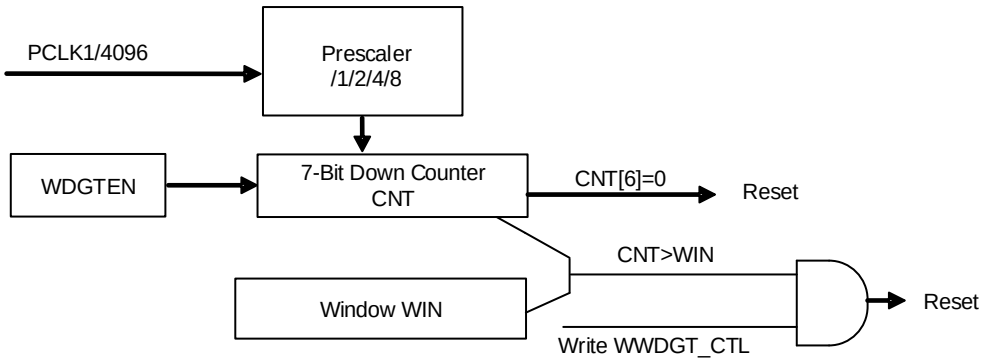
21.2.2. Characteristics

- 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): if the watchdog is started and the interrupt is enabled, the interrupt occurs when the counter reaches 0x40.
- WWDGT debug mode, the WWDGT can stop or continue to work in debug mode.

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



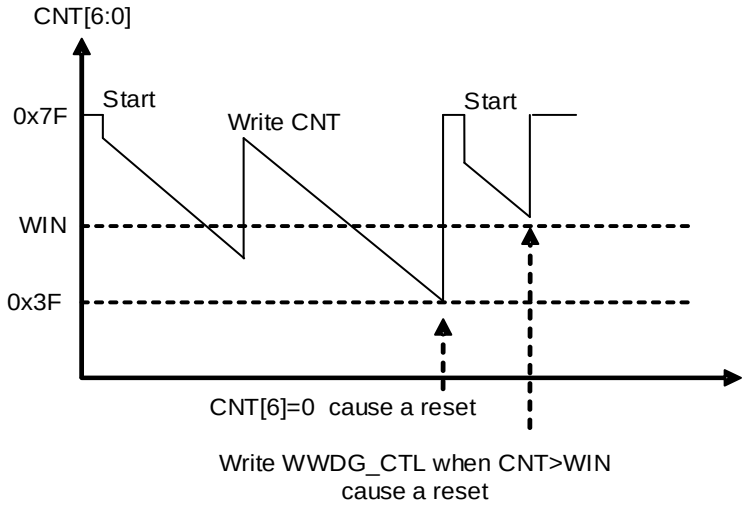
The watchdog is always disabled after power on reset. The software starts the watchdog by setting the WDGTEN bit in the WWDGT_CTL register. When window watchdog timer is enabled, the counter counts down all the time, the configured value of the counter should be greater than 0x3F(it implies that the CNT[6] bit should be set). The CNT[5:0] determine the maximum time interval between two reloading. The count down speed depends on the APB1 clock and the prescaler (PSC[1:0] bits in the WWDGT_CFG register).

The WIN[6:0] bits in the configuration register (WWDGT_CFG) specifies the window value. The software can prevent the reset event by reloading the down counter. The counter value is less than the window value and greater than 0x3F, otherwise the watchdog causes a reset.

The early wakeup interrupt (EWI) is enabled by setting the EWIE bit in the WWDGT_CFG register, and the interrupt will be generated when the counter reaches 0x40. The software can do something such as communication or data logging in the interrupt service routine (ISR) in order to analyse the reason of software malfunctions or save the important data before resetting the device. Moreover the software can reload the counter in ISR to manage a software system check and so on. In this case, the WWDGT will never generate a WWDGT reset but can be used for other things.

The EWI interrupt is cleared by writing '0' to the EWIF bit in the WWDGT_STAT register.

Figure 21-3. Window watchdog timing diagram



Calculate the WWDGT timeout by using the formula below.

$$t_{WWDGT}=t_{PCLK1} \times 4096 \times 2^{PSC} \times (CNT[6:0]+1)\ (ms) \qquad (21\text{-}1)$$

where:

t_{WWDGT}: WWDGT timeout

t_{PCLK1}: APB1 clock period measured in ms

The ***Table 21-2. Min-max timeout value at 216 MHz (fPCLK1)*** shows the minimum and maximum values of the t_{WWDGT}.

Table 21-2. Min-max timeout value at 216 MHz (f_{PCLK1})

Prescaler divider	PSC[1:0]	Min timeout value CNT[6:0] =0x40	Max timeout value CNT[6:0]=0x7F
1/1	00	18.96 µs	1.21 ms
1/2	01	37.93 µs	2.43 ms
1/4	10	75.85 µs	4.85 ms
1/8	11	151.70µs	9.71 ms

If the WWDGT_HOLD bit in DBG module is cleared, the WWDGT continues to work even the Cortex®-M33 core halted (Debug mode). While the WWDGT_HOLD bit is set, the WWDGT stops in Debug mode.

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 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								WDGTEN	CNT[6:0]						
								rs	rw						

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

Configuration register (WWDGT_CFG)

Address offset: 0x04

Reset value: 0x0000 007F

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

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
Reserved						EWIE	PSC[1:0]		WIN[6:0]						
						rs	rw		rw						

Bits	Fields	Descriptions
31:10	Reserved	Must be kept at reset value.
9	EWIE	Early wakeup interrupt enable. If the bit is set, an interrupt occurs when the counter reaches 0x40. It can be cleared by a hardware reset or software clock reset. A write

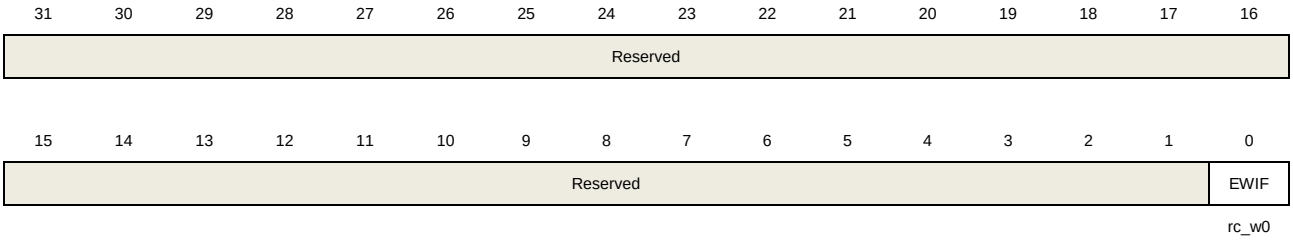
		operation of 0 has no effect.
8:7	PSC[1:0]	Prescaler. The time base of the watchdog counter 00: (PCLK1 / 4096) / 1 01: (PCLK1 / 4096) / 2 10: (PCLK1 / 4096) / 4 11: (PCLK1 / 4096) / 8
6:0	WIN[6:0]	The Window value. A reset occur if the watchdog counter (CNT bits in WWDGT_CTL) is written when the value of the watchdog counter is greater than the Window value.

Status register (WWDGT_STAT)

Address offset: 0x08

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:1	Reserved	Must be kept at reset value.
0	EWIF	Early wakeup interrupt flag. When the counter reaches 0x40 or refreshes before it reaches the window value, 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 a time which includes hour/minute/second/sub-second and a calendar includes year/month/day/week day. The time and calendar are expressed in BCD code except sub-second. Sub-second is expressed in binary code. Hour adjust for daylight saving time. Working in power saving mode and smart wakeup is software configurable. Support improving the calendar accuracy using extern accurate low frequency clock.

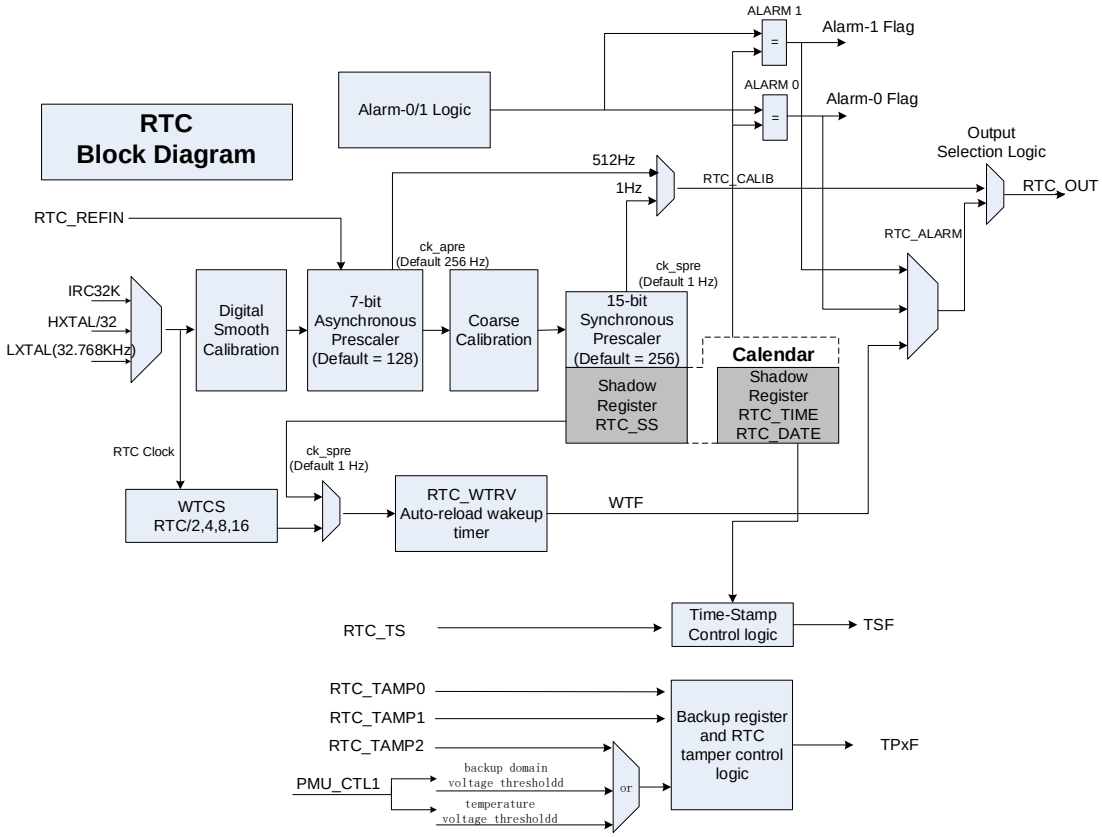
22.2. Characteristics

- Support calendar function, which can support year, month, date, day, hours, minutes, seconds and subseconds (date is the day of week and day is the day of month).
- Daylight saving compensation supported, which is realized through software
- External high-accurate low frequency(50Hz or 60Hz) clock used to achieve higher calendar accuracy performed by reference clock detection option function.
- Atomic clock adjust (max adjust accuracy is 0.95PPM) for calendar calibration performed by digital calibration function.
- Sub-second adjustment by shift function
- Time-stamp function for saving event time
- Three tamper sources can be chosen and tamper type is configurable (RTC_TAMP0, RTC_TAMP1 and RTC_TAMP2).
- Programmable calendar and two field maskable alarms.
- Maskable interrupt source:
  - Alarm 0 and Alarm 1
  - Time-stamp detection
  - Tamper detection
  - Auto wakeup event
- Thirty-two 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 timestamp event.
- When tamper detection event happen, the backup registers will be erased
- When power switch is switched to the VBAT, there will generate a timestamp event.
- 32-bit backup registers, which number increased to 32
- Optional RTC output function:
  - 512Hz (default prescale) :RTC_OUT(PC13 or PB2)
  - 1Hz(default prescale): 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 event detection: RTC_TS(PC13)
  - tamper 0 event detection: RTC_TAMP0(PC13) (works under VDD and VBAT supply)
  - tamper 1 event detection: RTC_TAMP1(PA0) (only works under VDD supply)
  - tamper 2 event detection: RTC_TAMP2(PE6) (only works under VDD supply)

- reference clock input: RTC_REFIN(PB15 or PA1)
- tamper 2 event detection not only can be generated by I/O, when VBTMEN bit in PMU_CTL1 is set, the backup domain voltage thresholds or temperature voltage thresholds also can generate tamper 2 event detection.

22.3.2. Clock source and prescalers

RTC unit has three independent clock sources: LXTAL, IRC32K and HXTAL with divided by 32(configured in RCU_CFG register).

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 value. The ck_spre clock is used to driven the calendar registers. Each clock will make second plus one.

22.3.3. Shadow registers introduction

BPSHAD control bit decides the location when APB bus accesses the RTC calendar register RTC_DATE, RTC_TIME and RTC_SS. By default, the BPSHAD is cleared, and APB bus accesses the shadow calendar registers. Shadow calendar registers is updated 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 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. 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.

22.3.6. RTC initialization and configuration

RTC register write protection

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

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

Following below steps will unlock the write protection:

1. Write ‘0xCA’ into the RTC_WPK register
2. Write ‘0x53’ into the RTC_WPK register

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

are not:

RTC_TIME, RTC_DATE, RTC_CTL, RTC_STAT, RTC_PSC, RTC_WUT, RTC_ALRM0TD, RTC_ALRM1TD, RTC_SHIFTCTL, RTC_HRFC, RTC_ALRM0SS, RTC_ALRM1SS, RTC_CFG

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.

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.

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)

22.3.7. 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_SSTS/RTC_TTS/RTC_DTS)
- RTC Tamper register (RTC_TAMP)
- RTC Backup registers (RTC_BKPx, 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 value in RTC_PSC. The higher FACTOR_S, the higher adjust precision.

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

Note: Before using shift function, the software must check the MSB of SSC in RTC_SS

(SSC[15]) and confirm it is 0.

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

Shift operation only works correctly when REFEN=0.

Software must not write to RTC_SHIFTCTL if REFEN=1.

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=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 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 211/210/29(32/16/8 seconds) RTC clock insert one cycle.

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

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

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

$$f_{cal}=f_{rtcclk} \times (1+\frac{FREQI\times 512-CMSK}{2^N+CMSK-FREQI\times 512}) \quad (22\text{-}3)$$

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

Calibration when FACTOR_A < 3

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

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

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

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

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

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

$$f_{cal}=f_{rtcclk} \times (1+\frac{256-CMSK}{2^N+CMSK-256}) \quad (22\text{-}4)$$

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

Re-calibration on-the-fly

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

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

22.3.12. Time-stamp function

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

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

When an internal time-stamp event detected (ITSEN = 1), the calendar value will be saved in time-stamp registers (RTC_DTS/RTC_TTS/RTC_SSTS), the time-stamp flag (TSF) 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 TSF=1.

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

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

22.3.13. 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
- It can wakeup from DeepSleep and Standby modes, and generate an interrupt.

RTC backup registers (RTC_BKPx)

The RTC backup registers are located in the VDD 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 is different from 0x0 (Level detection mode with configurable filtering), there are three ck_apre cycles.
- When TPTS is set (Timestamp on tamper event), there are three ck_apre cycles.
- When FLT 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.

Timestamp on tamper event

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

Edge detection mode on tamper input detection

When FLT bit is set to 0x0, the tamper detection is set to edge detection mode and TPxEG bit determines the rising edge or falling edge is the detecting edge. When tamper detection is

under edge detection mode, the internal pull-up resistors on the tamper detection input pin are deactivated.

Because of detecting the tamper event will reset the backup registers (RTC_BKPx), writing to the backup register should ensure that the tamper event reset and the writing operation will not occur at the same time, a recommend way to avoid this situation is disable the tamper detection before writing to the backup register and re-enable tamper detection after finish writing.

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

Level detection mode with configurable filtering on tamper input detection

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

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

22.3.14. Calibration clock output

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

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

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

$$f_{rtc_calib} = \frac{f_{rtcclk}}{(FACTOR_A+1)\times(FACTOR_S+1)} \quad (22\text{-}5)$$

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

22.3.15. Alarm output

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

22.3.16. RTC pin configuration

RTC_OUT, RTC_TS 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 (tamper enabled)	TSEN (time stamp enabled)	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	-	-	-
TAMP0 input floating	00	0	1	0	-
TIMESTAMP and TAMP0 input floating	00	0	1	1	Don't care
TIMESTAMP input floating	00	0	0	1	Don't care
Standard GPIO	00	0	0	0	Don't care

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[23] in the RTC_CTL register.
- RTC_TAMP0: tamper event detection
- RTC_TS: time stamp event detection

ALRMOUTTYPE in RTC_CFG is used to select whether the RTC_ALARM is output in push-pull or open-drain mode.

It is possible to output RTC_OUT on PB2 or PC13 pin thanks to OUT2EN bit in RTC_CFG[31]. This output is not available in VBAT / Standby / Shutdown mode.

22.3.17. RTC power saving mode management

Table 22-2 RTC power saving mode management

Mode	Active in Mode	Exit Mode
Sleep	Yes	RTC Interrupts
Deep-sleep	Yes: if clock source is LXTAL or IRC32K	RTC Alarm / Tamper Event / Timestamp Event / Wake up
Standby	Yes: if clock source is LXTAL or IRC32K	RTC Alarm / Tamper Event / Timestamp Event / Wake up

22.3.18. RTC interrupts

All RTC interrupts are connected to the EXTI controller.

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

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

Table 22-3 RTC interrupts control

Interrupt	Event flag	Control Bit	Exit Sleep	Exit Deep-sleep	Exit Standby
Alarm 0	ALRM0F	ALRM0IE	Y	Y(1)	Y(1)
Alarm 1	ALRM1F	ALRM1IE	Y	Y(1)	Y(1)
Wakeup	WTF	WTIE	Y	Y(1)	Y(1)
Timestamp	TSF	TSIE	Y	Y(1)	Y(1)
Tamper 0	TP0F	TPIE	Y	Y(1)	Y(1)
Tamper 1	TP1F	TPIE	Y	Y(1)	N(1)
Tamper 2	TP2F	TPIE	Y	Y(1)	N(1)

(1) Only active when RTC clock source is LXTAL or IRC32K.

22.4. Register definition

RTC base address: 0x4000 2800

22.4.1. Time register (RTC_TIME)

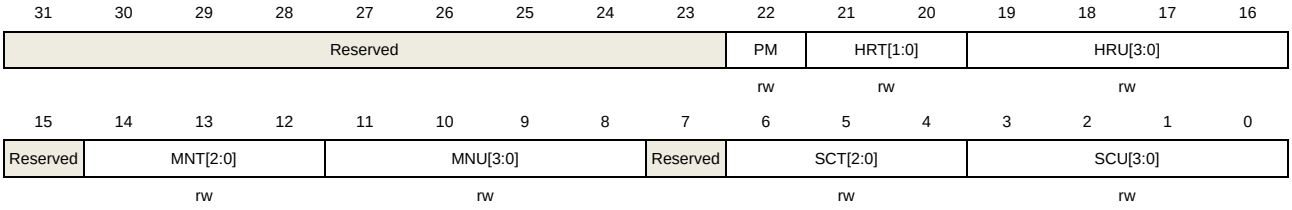
Address offset: 0x00

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

This register has 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	Year tens in BCD code
19:16	YRU[3:0]	Year units in BCD code
15:13	DOW[2:0]	Days of the week 0x0: Reserved 0x1: Monday … 0x7: Sunday
12	MONT	Month tens in BCD code
11:8	MONU[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: 0x0000 0000
This register is writing protected

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						TAMPCLK	ITSEN	COEN	OS[1:0]		OPOL	COS	DSM	S1H	A1H
						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	TSEN	WTEN	ALRM1EN	ALRM0EN	Reserved	CS	BPSHAD	REFEN	TSEG	WTCS[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw		

Bits	Fields	Descriptions
31:26	Reserved	Must be kept at reset value.
25	TAMPCLK	edge tamper needs RTC Clok 0: need RTC Clok

		1: not need RTC Clok
24	ITSEN	Internal timestamp event enable 0: Disable Internal timestamp event 1: Enable Internal timestamp 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 0x0: Disable output RTC_ALARM 0x1: Enable alarm0 flag output 0x2: Enable alarm1 flag output 0x3: 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	TSEN	Time-stamp function enable 0: Disable time-stamp function 1: Enable time-stamp function
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 must be 0x00FF
3	TSEG	Valid event edge of time-stamp 0: rising edge is valid event edge for time-stamp event 1: falling edge is valid event edge for time-stamp event
2:0	WTCS[2:0]	Auto-wakeup timer clock selection 0x0:RTC Clock divided by 16 0x1:RTC Clock divided by 8

0x2:RTC Clock divided by 4

0x3:RTC Clock divided by 2

0x4:0x5: ck_spre (default 1Hz) clock

0x6:0x7: 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														ITSF	SCPF
														rc_w0	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TP2F	TP1F	TP0F	TSOVRF	TSF	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	rc_w0	rw	r	rc_w0	r	r	r	r	r

Bits	Fields	Descriptions
31:18	Reserved	Must be kept at reset value.
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 TSF 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	TP2F	RTC_TAMP2 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	TP1F	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.
13	TP0F	RTC_TAMP0 detected flag Set to 1 by hardware when tamper detection is found on tamper0 input pin. Software can clear this bit by writing 0 into this bit.
12	TSOVRF	Time-stamp overflow flag This bit is set by hardware when a time-stamp event is detected if TSF bit is set before.

		Cleared by software writing 0.
11	TSF	Time-stamp flag Set by hardware when time-stamp event is detected. Cleared by software writing 0.
10	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)

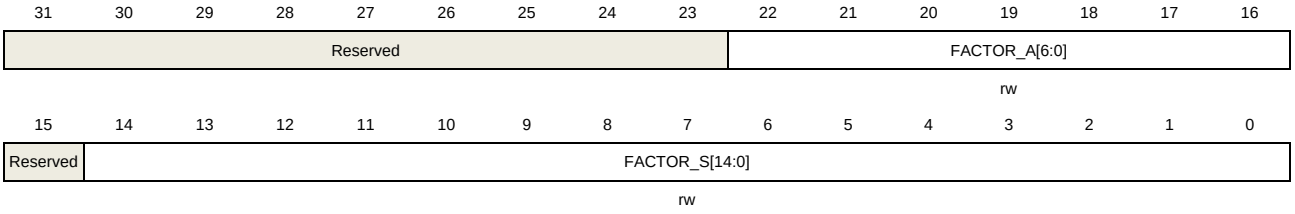
Address offset: 0x10

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)

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

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

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.

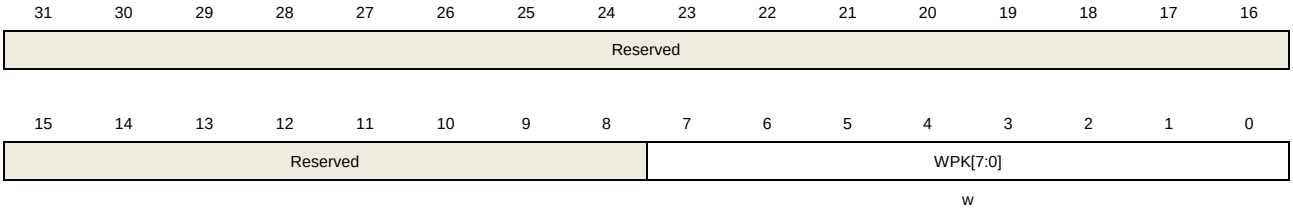
29:28	DAYT[1:0]	Day tens in BCD code
27:24	DAYU[3:0]	Day units or week day in BCD code
23	MSKH	Alarm hour mask bit 0: Not mask hour field 1: Mask hour field
22	PM	AM/PM flag 0: AM or 24-hour format 1: PM
21:20	HRT[1:0]	Hour tens in BCD code
19:16	HRU[3:0]	Hour units in BCD code
15	MSKM	Alarm minutes 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)



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

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
A1S	Reserved														
w															

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

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

down counter:

Delay (seconds) = SFS / (FACTOR_S + 1)

When jointly using A1S and SFS, the clock will advance:

Advance (seconds) = (1 - (SFS / (FACTOR_S + 1)))

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

22.4.12. Time of time stamp register (RTC_TTS)

Address offset: 0x30

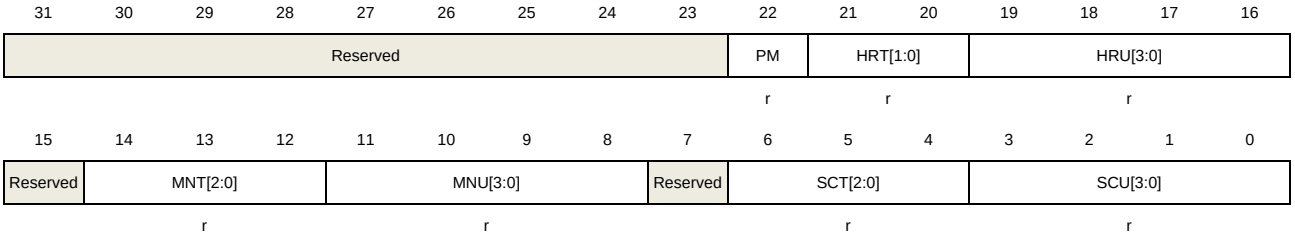
Backup domain reset value: 0x0000 0000

System reset: no effect

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

Reset TSF bit will also clear this register.

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



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 register (RTC_DTS)

Address offset: 0x34

Backup domain reset value: 0x0000 0000

System reset: no effect

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

Reset TSF bit will also clear this register.

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

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

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

Address offset: 0x38

Backup domain reset: 0x0000 0000

System reset: no effect

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

Reset TSF bit will also clear this register.

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

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

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

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

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	Reserved				CMSK[8:0]								
rw	rw	rw					rw								

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	FREQI	Increase RTC frequency by 488.5PPM 0: No effect 1: One RTCCLK pulse is inserted every 2^{11} pulses. This bit should be used in conjunction with CMSK bit. If the input clock frequency is 32.768KHz, the number of RTCCLK pulses added during 32s calibration window is (512 * FREQI) - CMSK
14	CWND8	Frequency compensation window 8 second selected 0: No effect 1: Calibration window is 8 second Note: When CWND8=1, CMSK[1:0] are stuck at “00”.
13	CWND16	Frequency compensation window 16 second selected 0: No effect 1: Calibration window is 16 second Note: When CWND16=1, CMSK[0] are stuck at “0”.
12:9	Reserved	Must be kept at reset value.
8:0	CMSK[8:0]	Calibration mask number The number of mask pulse out of 2^{20} RTCCLK pulse. This feature will decrease the frequency of calendar with a resolution of 0.9537 PPM.

22.4.16. Tamper register (RTC_TAMP)

Address offset: 0x40

Backup domain reset: 0x0000 0000

System reset: no effect

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		TP2IE	TP1IE	TP0IE	Reserved	TP2MASK	TP1MASK	TP0MASK	Reserved	TP2 NOERASE	TP1 NOERASE	TP0 NOERASE	Reserved		TP2_DISPIN
		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
DISPU	PRCH[1:0]		FLT[1:0]		FREQ[2:0]			TPTS	TP2EG	TP2EN	TP1EG	TP1EN	TPIE	TP0EG	TP0EN
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	TP2IE	Tamper 2 interrupt enable 0: Disable tamper 2 interrupt 1: Enable tamper 2 interrupt
28	TP1IE	Tamper 1 interrupt enable 0: Disable tamper 1 interrupt 1: Enable tamper 1 interrupt
27	TP0IE	Tamper 0 interrupt enable 0: Disable tamper 0 interrupt 1: Enable tamper 0 interrupt
26	Reserved	Must be kept at reset value.
25	TP2MASK	Tamper 2 mask flag 0: Tamper 2 event generates a trigger event and TP2F must be cleared by software to allow next tamper event detection. 1: Tamper 2 event generates a trigger event. TP2F is masked and internally cleared by hardware. The backup registers are not erased. **Note:** The Tamper 2 interrupt must not be enabled when TP2MASK is set.
24	TP1MASK	Tamper 1 mask flag 0: Tamper 1 event generates a trigger event and TP1F must be cleared by software to allow next tamper event detection. 1: Tamper 1 event generates a trigger event. TP1F is masked and internally cleared by hardware. The backup registers are not erased. **Note:** The Tamper 1 interrupt must not be enabled when TP1MASK is set.
23	TP0MASK	Tamper 0 mask flag 0: Tamper 0 event generates a trigger event and TP0F must be cleared by software to allow next tamper event detection. 1: Tamper 0 event generates a trigger event. TP0F is masked and internally cleared by hardware. The backup registers are not erased. **Note:** The Tamper 0 interrupt must not be enabled when TP0MASK is set.
22	Reserved	Must be kept at reset value.
21	TP2 NOERASE	Tamper 2 no erase

		0: Tamper 2 event erases the backup registers 1: Tamper 2 event does not erase the backup registers
20	TP1 NOERASE	Tamper 1 no erase 0:Tamper 1 event erases the backup registers 1:Tamper 1 event does not erase the backup registers
19	TP0 NOERASE	Tamper 0 no erase 0:Tamper 0 event erases the backup registers 1:Tamper 0 event does not erase the backup registers
18:17	Reserved	Must be kept at reset value.
16	TP2_DISPIN	Tamper 2 selection 0: pin trigger tamper 2 1: pin don not trigger tamper 2
15	DISPU	RTC_TAMPx pull up disable bit 0: Enable inner pull-up before sampling for pre-charge RTC_TAMPx pin. 1: Disable pre-charge duration
14:13	PRCH[1:0]	Pre-charge duration time of RTC_TAMPx This setting determines the pre-charge time before each sampling. 0x0: 1 RTC clock 0x1: 2 RTC clock 0x2: 4 RTC clock 0x3: 8 RTC clock
12:11	FLT[1:0]	RTC_TAMPx filter count setting This bit determines the tamper sampling type and the number of consecutive sample. 0x0: Detecting tamper event using edge mode. Pre-charge duration is disabled automatically. 0x1: Detecting tamper event using level mode.2 consecutive valid level samples will make an effective tamper event. 0x2: Detecting tamper event using level mode.4 consecutive valid level samples will make an effective tamper event. 0x3: Detecting tamper event using level mode.8 consecutive valid level samples will make an effective tamper event.
10:8	FREQ[2:0]	Sampling frequency of tamper event detection 0x0: Sample once every 32768 RTCCLK(1Hz if RTCCLK=32.768KHz) 0x1: Sample once every 16384 RTCCLK(2Hz if RTCCLK=32.768KHz) 0x2: Sample once every 8192 RTCCLK(4Hz if RTCCLK=32.768KHz) 0x3: Sample once every 4096 RTCCLK(8Hz if RTCCLK=32.768KHz) 0x4: Sample once every 2048 RTCCLK(16Hz if RTCCLK=32.768KHz) 0x5: Sample once every 1024 RTCCLK(32Hz if RTCCLK=32.768KHz)

		0x6: Sample once every 512 RTCCLK(64Hz if RTCCLK=32.768KHz) 0x7: Sample once every 256 RTCCLK(128Hz if RTCCLK=32.768KHz)
7	TPTS	Make tamper function used for timestamp function 0:No effect 1:TSF is set when tamper event detected even TSEN=0
6	TP2EG	Tamper 2 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	TP2EN	Tamper 2 detection enable 0:Disable tamper 2 detection function 1:Enable tamper 2 detection function
4	TP1EG	Tamper 1 event trigger edge If tamper detection is in edge mode(FLT =0): 0: Rising edge triggers a tamper detection event 1: Falling edge triggers a tamper detection event If tamper detection is in level mode(FLT !=0): 0: Low level triggers a tamper detection event 1: High level triggers a tamper detection event
3	TP1EN	Tamper 1 detection enable 0:Disable tamper 1 detection function 1:Enable tamper 1 detection function
2	TPIE	Tamper detection interrupt enable 0: Disable tamper interrupt 1: Enable tamper interrupt
1	TP0EG	Tamper 0 event trigger edge If tamper detection is in edge mode(FLT =0): 0: Rising edge triggers a tamper detection event 1: Falling edge triggers a tamper detection event If tamper detection is in level mode(FLT !=0): 0: Low level triggers a tamper detection event 1: High level triggers a tamper detection event
0	TP0EN	Tamper 0 detection enable 0:Disable tamper 0 detection function 1:Enable tamper 0 detection function

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

22.4.17. Alarm 0 sub second register (RTC_ALRM0SS)

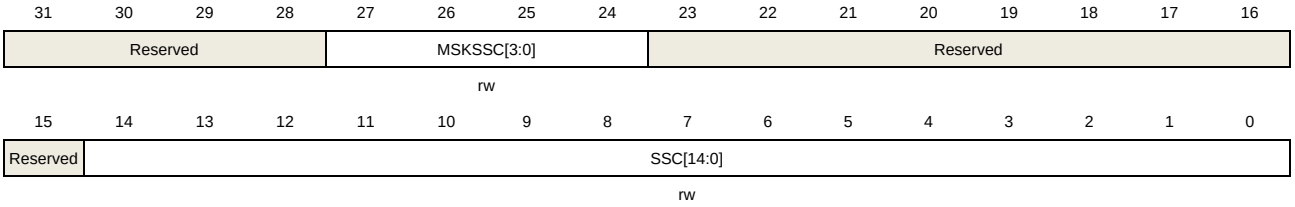
Address offset: 0x44

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

22.4.18. Alarm 1 sub second register (RTC_ALRM1SS)

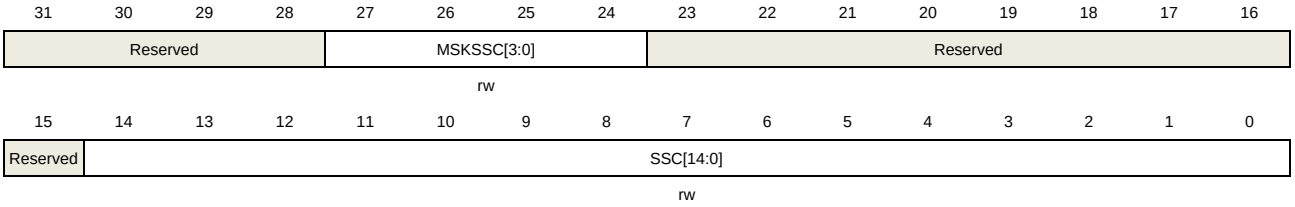
Address offset: 0x48

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

22.4.19. Configuration register (RTC_CFG)

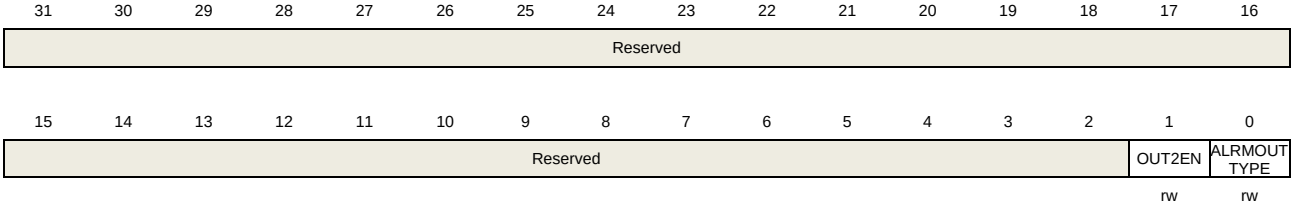
Address offset: 0x4C

Backup domain reset: 0x0000 0000

System reset: no effect

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
31:2	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

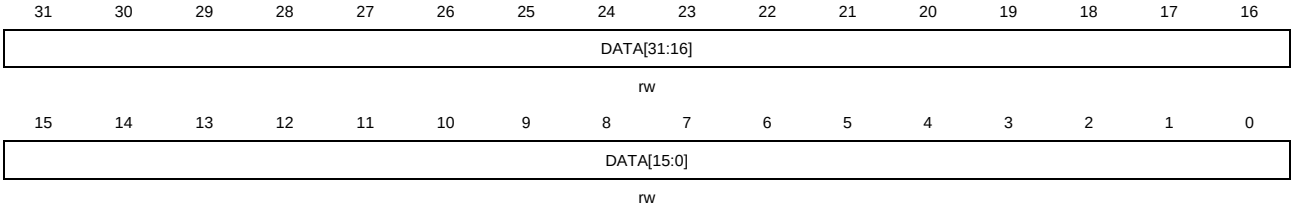
22.4.20. Backup registers (RTC_BKPx) (x=0..31)

Address offset: 0x50~0xCC

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.

23. TIMER (TIMER)

Table 23-1. Timers (TIMERx) are divided into five sorts

TIMER		TIMER0/7/19	TIMER1/2/3/4	TIMER14	TIMER15/16	TIMER5/6
TYPE		Advanced	General-L0	General-L3	General-L4	Basic
Prescaler		16-bit	16-bit	16-bit	16-bit	16-bit
Counter		16-bit	16-bit(TIMER2/3) 32-bit(TIMER1/4)	16-bit	16-bit	16-bit (TIMER5/6)
Count mode		UP, DOWN, Center-aligned	UP, DOWN, Center-aligned	UP ONLY	UP ONLY	UP ONLY
Repetition		●	×	●	●	×
Channel Capture/ Compare		8	4	3	2	0
Composite PWM mode		●	●	●	×	×
Output match pulse select		●	●	●	×	×
Complementary & Dead-time		●	×	●	●	×
Break function	BREAK0	●	×	●	●	×
	BREAK1	●	×	×	×	×
Locked break function		●	×	●	●	×
Single Pulse		●	●	●	●	●
Delayable single pulse		●	●	●	×	×
Quadrature decoder		●	●	×	×	×
decoder		●	●	×	×	×
Master-slave management		●	●	●	×	×
Inter Connection		● (1)	● (2)	● (3)	×	TRGO TO DAC (5)
Synchronization and initial direction and value refresh		●	×	×	×	×
DMA		●	●	●	●	● (4)
Debug Mode		●	●	●	●	●

	TIMERx	ITI0	ITI1	ITI2	ITI3	ITI4	ITI5	ITI6	ITI7	ITI8	ITI9	ITI10	ITI11	ITI12	ITI13	ITI14
(1)	TIMER0	-	TIMER1_TRGO0	TIMER2_TRGO0	TIMER3_TRGO0	TIMER4_TRGO0	TIMER7_TRGO0	TIMER14_TRGO0	TIMER15_CH0	TIMER16_CH0	TIMER19_TRGO0	HRTIMER_SCOUT (Please refer to ***High-Resolution Timer (HRTIMER) Synchronization input/output*** for more details)	-	-	-	from TRIGSEL
	TIMER7	TIMER0_TRGO0	TIMER1_TRGO0	TIMER2_TRGO0	TIMER3_TRGO0	TIMER4_TRGO0	-	TIMER14_TRGO0	TIMER15_CH0	TIMER16_CH0	TIMER19_TRGO0		-	-	-	
	TIMER19	TIMER0_TRGO0	TIMER1_TRGO0	TIMER2_TRGO0	TIMER3_TRGO0	TIMER4_TRGO0	TIMER7_TRGO0	TIMER14_TRGO0	TIMER15_CH0	TIMER16_CH0	-		-	-	-	
(2)	TIMER1	TIMER0_TRGO0	-	TIMER2_TRGO0	TIMER3_TRGO0	TIMER4_TRGO0	TIMER7_TRGO0	TIMER14_TRGO0	TIMER15_CH0	TIMER16_CH0	TIMER19_TRGO0			-	-	
	TIMER2	TIMER0_TRGO0	TIMER1_TRGO0	-	TIMER3_TRGO0	TIMER4_TRGO0	TIMER7_TRGO0	TIMER14_TRGO0	TIMER15_CH0	TIMER16_CH0	TIMER19_TRGO0		-			
	TIMER3	TIMER0_TRGO0	TIMER1_TRGO0	TIMER2_TRGO0	-	TIMER4_TRGO0	TIMER7_TRGO0	TIMER14_TRGO0	TIMER15_CH0	TIMER16_CH0	TIMER19_TRGO0		-			
	TIMER4	TIMER0_TRGO0	TIMER1_TRGO0	TIMER2_TRGO0	TIMER3_TRGO0	-	TIMER7_TRGO0	TIMER14_TRGO0	TIMER15_CH0	TIMER16_CH0	TIMER19_TRGO0		-			
(3)	TIMER14	TIMER0_TRGO0	TIMER1_TRGO0	TIMER2_TRGO0	TIMER3_TRGO0	TIMER4_TRGO0	TIMER7_TRGO0	-	TIMER15_CH0	TIMER16_CH0	TIMER19_TRGO0		-	-	-	
(4)	Only update events will generate a DMA request. TIMER5 / 6 do not have DMAS bit (DMA request source selection).															
(5)	TIMER5 / 6 TRGO to DAC by TRIGSEL module.															

23.1. Advanced timer (TIMERx, x=0, 7, 19)

23.1.1. Overview

The advanced timer module (TIMER0 / 7 / 19) is an eight-channel timer that supports both input capture and output compare. They can generate PWM signals to control motor or be used for power management applications. The advanced timer has a 16-bit counter that can be used as an unsigned counter.

In addition, the advanced timers can be programmed and be used for counting, their external events can be used to drive other timers.

Timer also includes a dead-time insertion module which is suitable for motor control applications.

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

23.1.2. Characteristics

- Total channel num: 8.
- Counter width: 16 bits.
- Selectable clock source: internal clock, internal trigger, external input, external trigger.
- Multiple counter modes: up counting, down counting and center-aligned counting.
- Quadrature decoder: used for motion tracking and determination of both rotation direction and position.
- Hall sensor function: used for 3-phase motor control.
- Programmable prescaler: 16 bits. The factor can be changed ongoing.
- Each channel is independent and user-configurable: input capture mode, output compare mode, programmable PWM mode, single pulse mode and trigger out.
- Programmable dead time insertion and separated dead time insertion.
- Auto reload function.
- Programmable counter repetition function.
- Break input function: BREAK0 and BREAK1.
- 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, and *Table 23-2. Advanced timer channel description* introduces the input and output of the channels.

Figure 23-1. Advanced timer block diagram

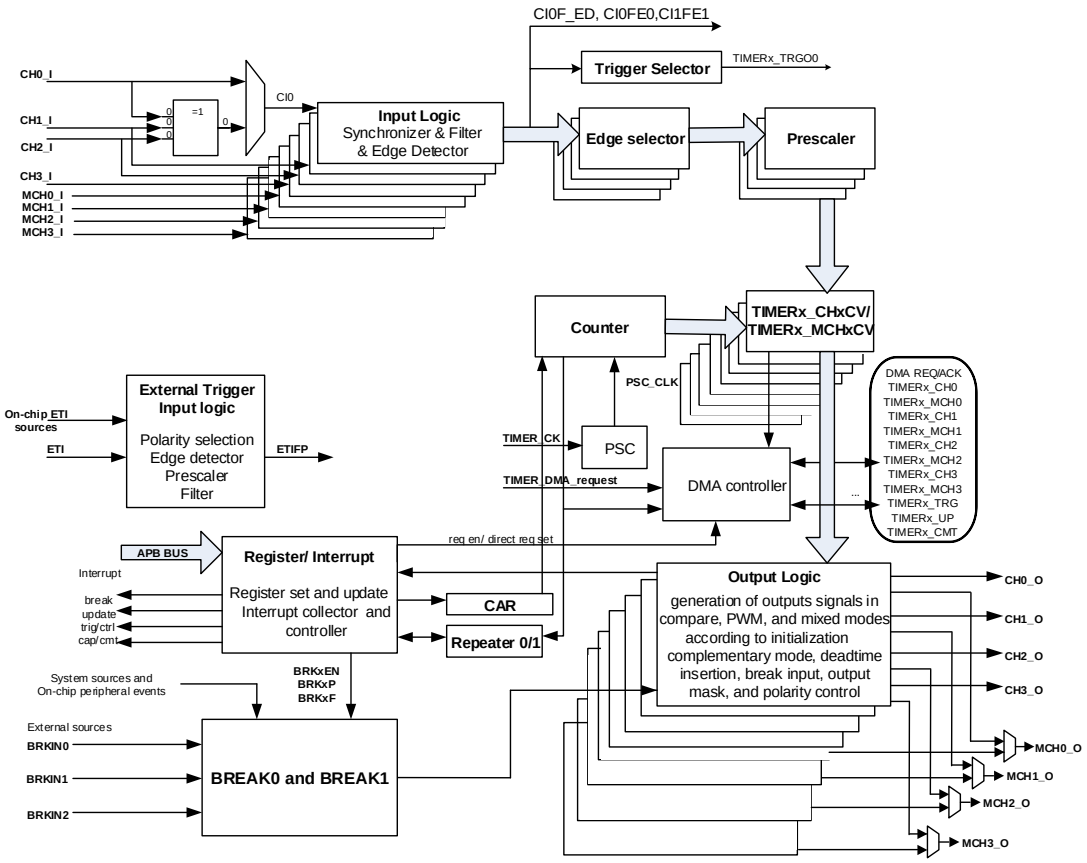


Table 23-2. Advanced timer channel description

Channel name (x=0...3)	MCHxMSEL[1:0]=00 independent mode	MCHxMSEL[1:0]=11 complementary mode
CHx (Channel x)	CHx and MCHx can independently input capture and compare output	only the CHx is valid for input, and the outputs of MCHx and CHx are complementary
MCHx (Multi mode channel x)		

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...15) in SYSCFG_TIMERxCFG(x=0, 7, 19) registers.

- TSCFGy[4:0] (y = 0...15) = 5'b00000 in SYSCFG_TIMERxCFG(x=0, 7, 19) registers.

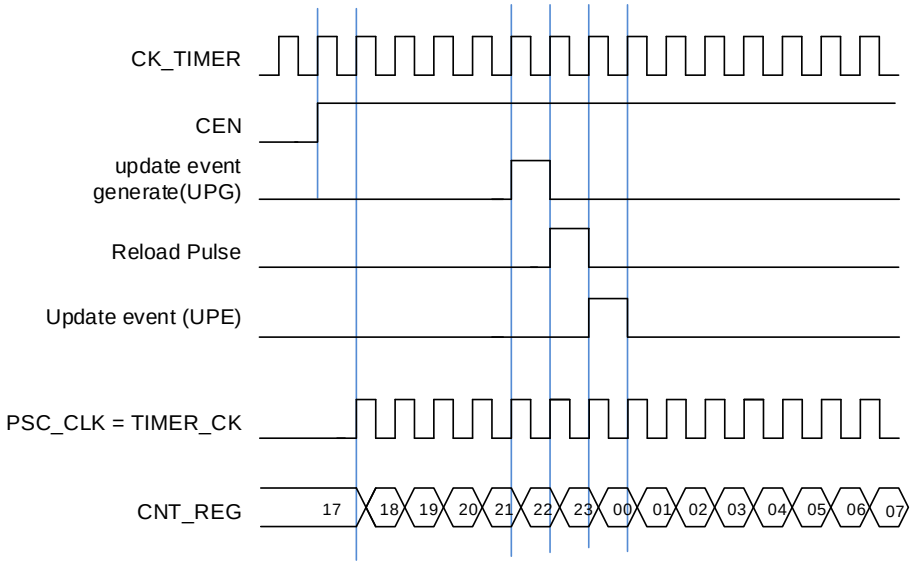
Internal clock CK_TIMER is selected as timer clock source which is from module RCU.

The default clock source is the CK_TIMER for driving the counter prescaler when TSCFGy[4:0] (y = 0...15) = 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...2,6,8,9) 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...2,6,8,9) bit-filed, more details will be introduced later. When the TSCFGy[4:0] (y=3,4,5,7) 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 CI0 / CI1. This mode can be selected by setting TSCFG6[4:0] to 0x5~0x7 and 0x9~0xE.

And, the counter prescaler can also be driven by rising edge on the internal trigger input pin ITI0 ~ 10 / ITI14. This mode can be selected by setting TSCFG6[4:0] to 0x1 ~ 0x4, 0x9 ~ 0xF or 0x13.

- 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

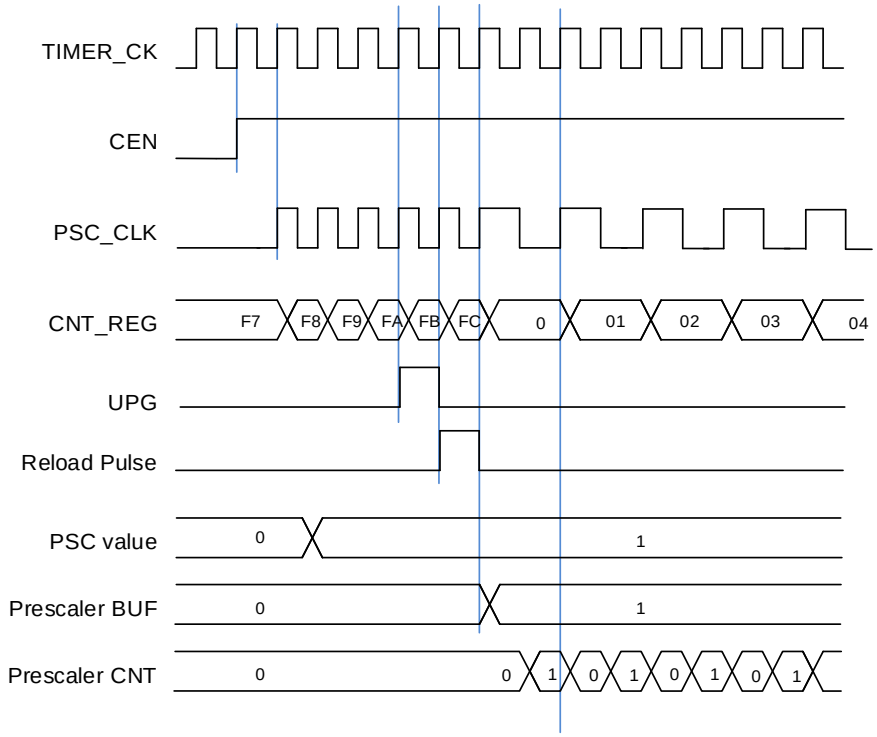
source is setting the TSCFG6[4:0] to 0x8. Note that the ETI signal is derived from the ETI pin sampled by a digital filter. When the ETI signal is selected as the clock source, the trigger controller including the edge detection circuitry will generate a clock pulse on each ETI signal rising edge to clock the counter prescaler.

Note: The ETI signal can be input from an external ETI pin or provide by on-chip peripherals, please refer to ***Trigger selection for TIMER0_ETI register (TRIGSEL_TIMER0ETI)***, ***Trigger selection for TIMER7_ETI register (TRIGSEL_TIMER7ETI)*** and ***Trigger selection for TIMER19_ETI register (TRIGSEL_TIMER19ETI)*** for more details.

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+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

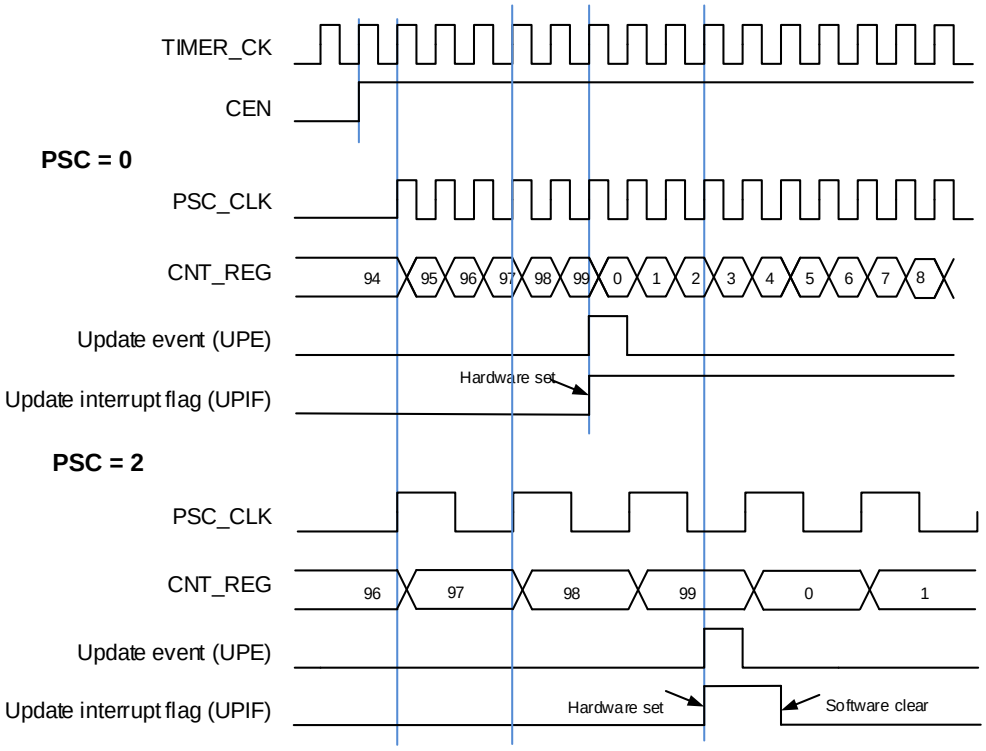
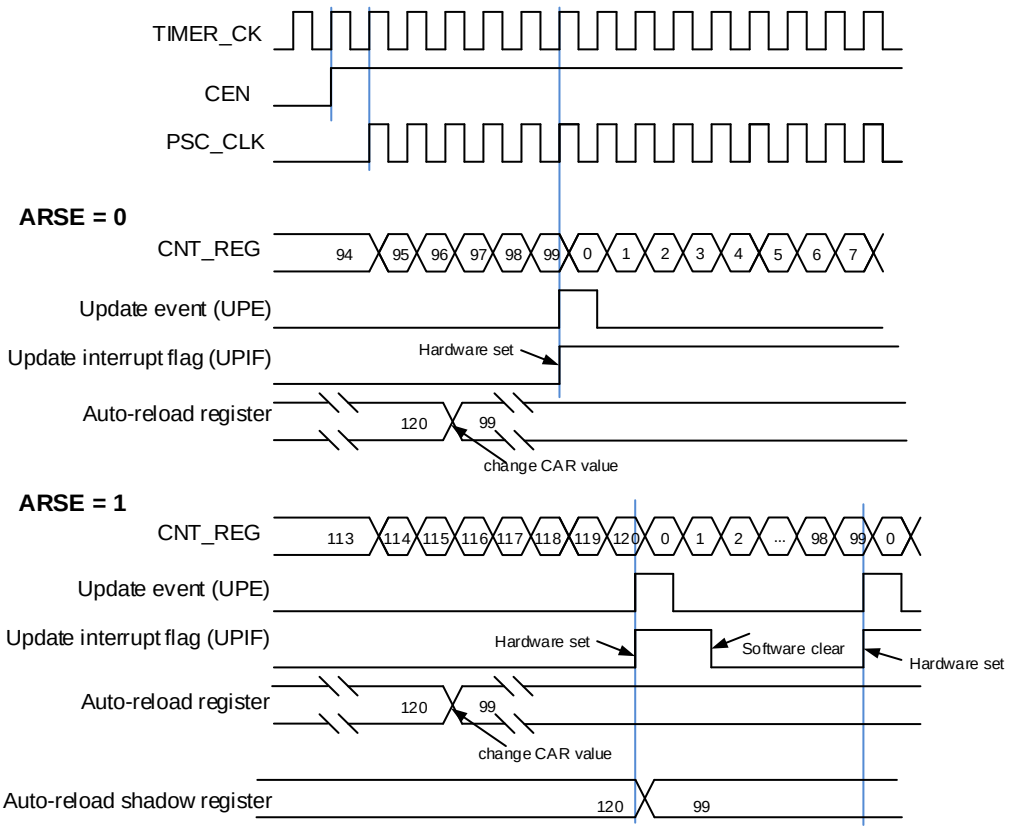


Figure 23-5. Timing diagram of up counting mode, change TIMERx_CAR ongoing



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 + 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

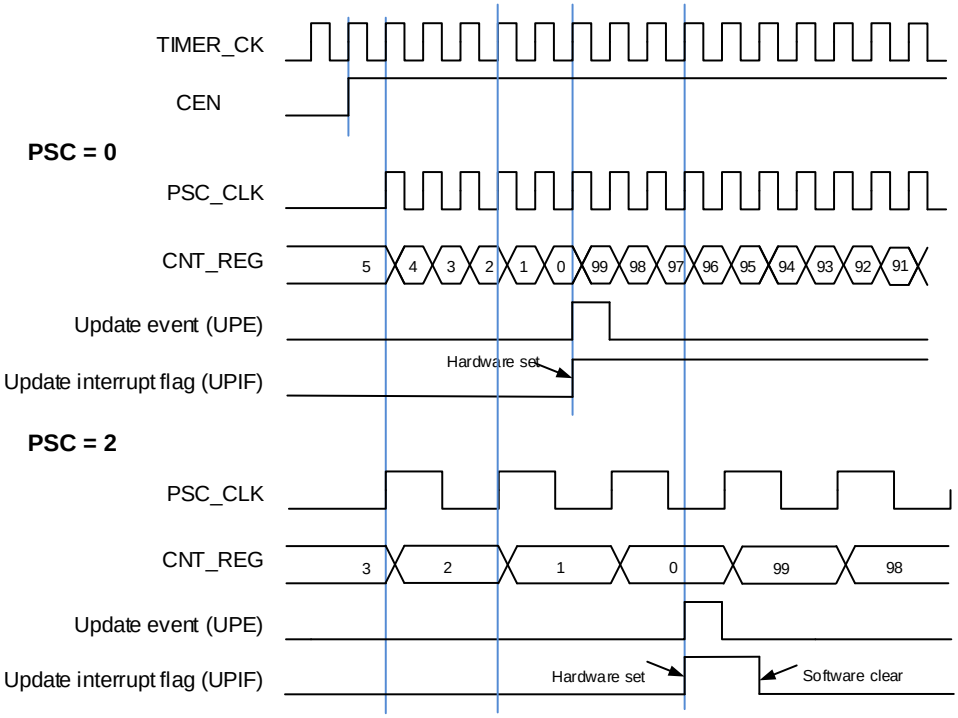
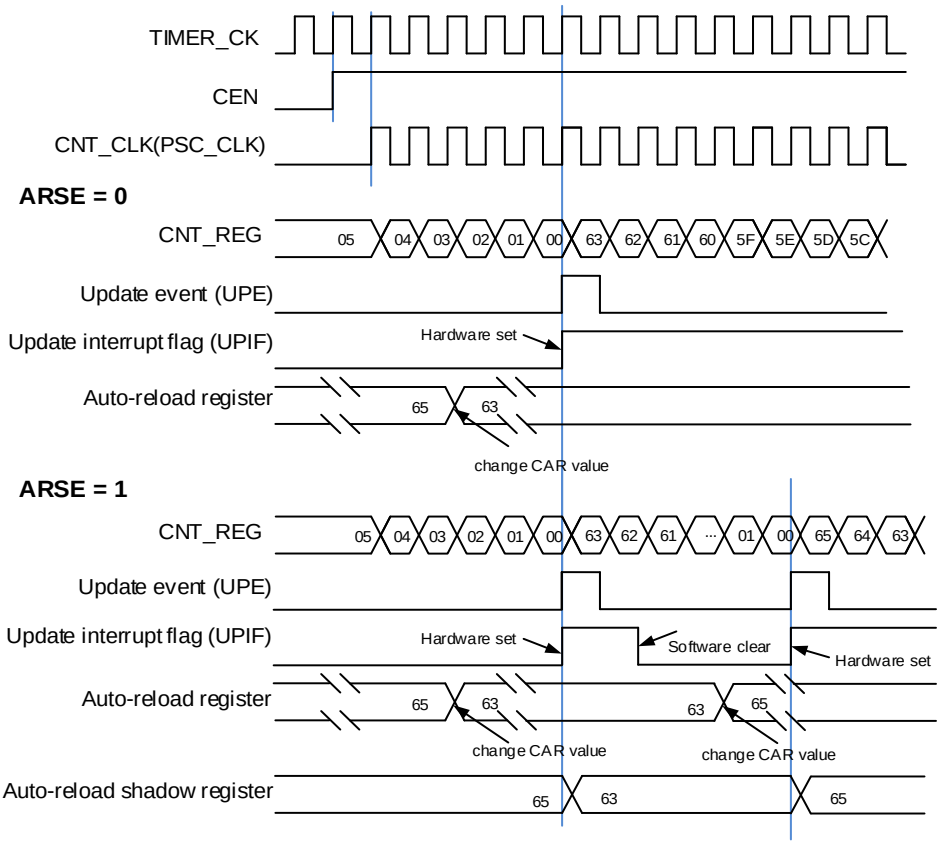


Figure 23-7. Timing diagram of down counting mode, change TIMERx_CAR ongoing



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.

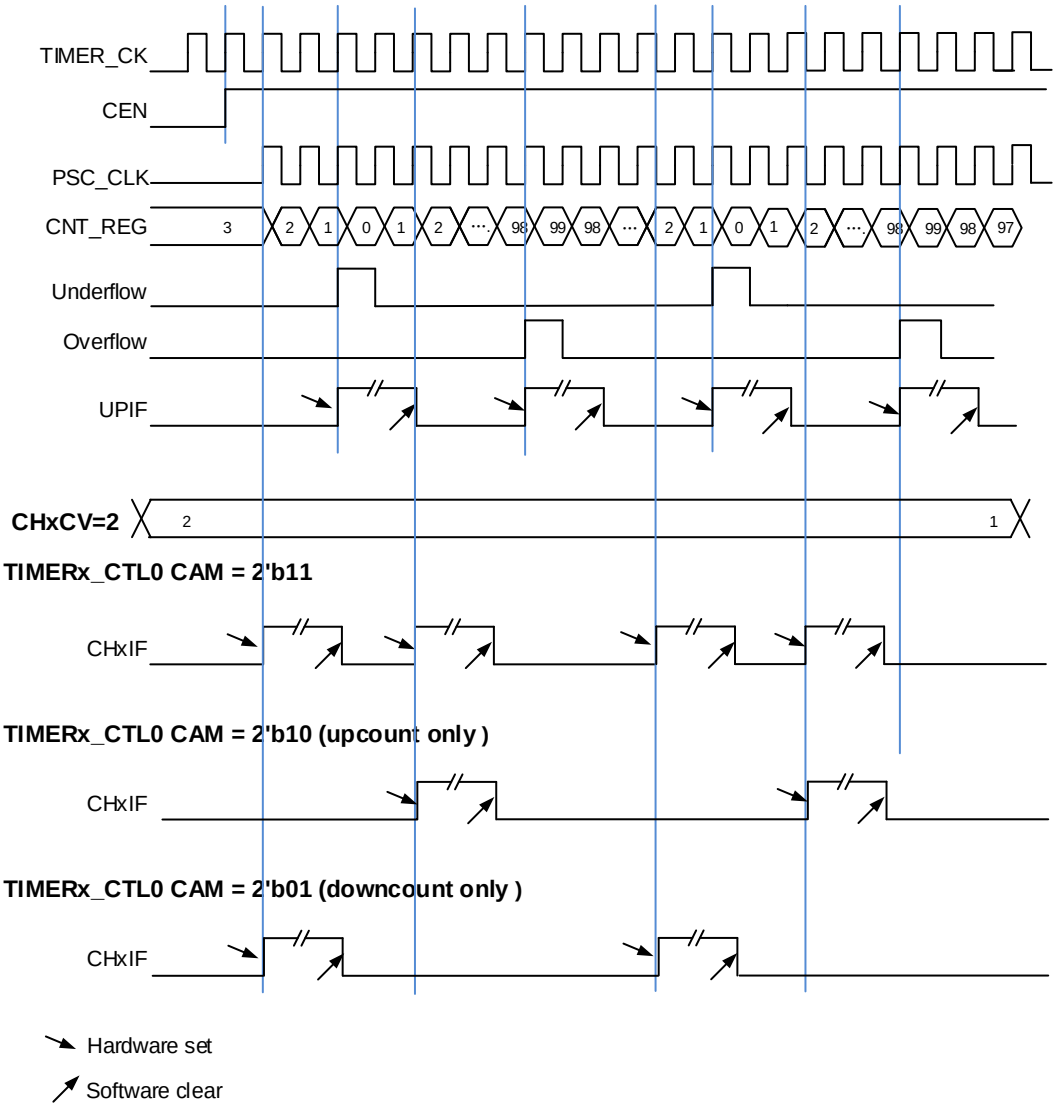
The UPIF bit in the TIMERx_INTF register will be set to 1 either when an underflow event or an overflow event occurs. While the CHxIF bit is associated with the value of CAM[1:0] in TIMERx_CTL0. The details refer to ***Figure 23-8. Timing diagram of center-aligned counting mode***.

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

The advance timer has two repetitions counter TIMERx_CREP0 / 1, which can be selected by configuring the CPERSEL bit in the TIMERx_CFG register. The CPEP[7:0] bit-field is 8bits, the CPEP[31:0] bit-field is 32bits and can be read on the fly.

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 / 1 bit in TIMERx_CREP0 / 1 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 / 1 in TIMERx_CREP0 / 1 register and generate an update event.

The new written CREP0/1 value will not take effect until the next update event. When the

value of CREP0/1 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 CREP0/1 value takes effect. If an update event is generated by software after writing an odd number to CREP0/1, the update events will be generated on the underflow. If the next update event occurs on overflow after writing an odd number to CREP0/1, then the subsequent update events will be generated on the overflow.

Figure 23-9. Repetition counter timing diagram of center-aligned counting mode

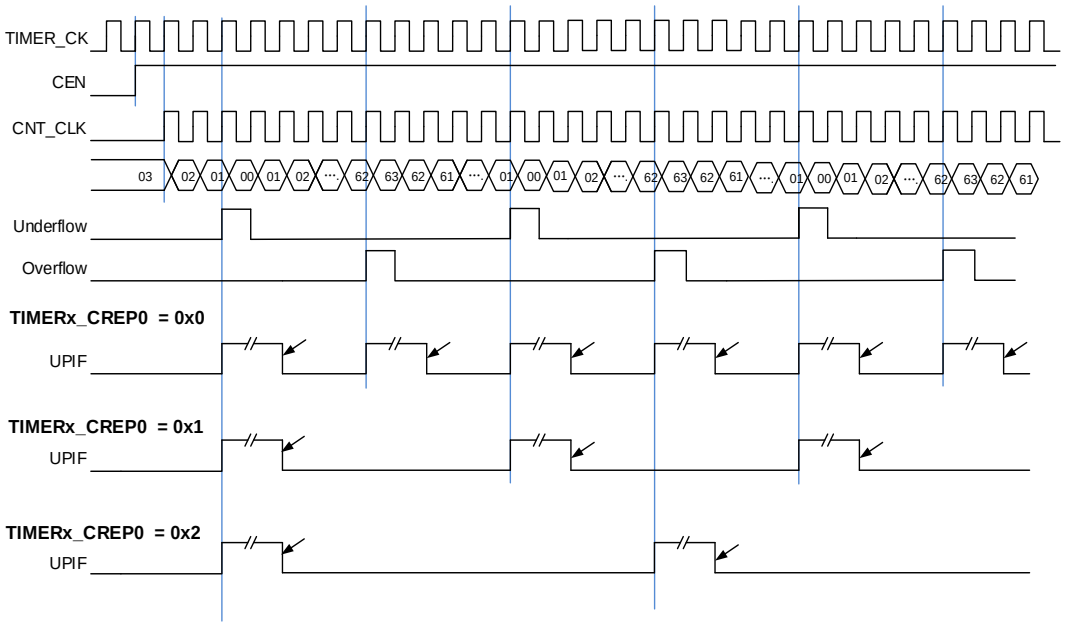


Figure 23-10. Repetition counter timing diagram of up counting mode

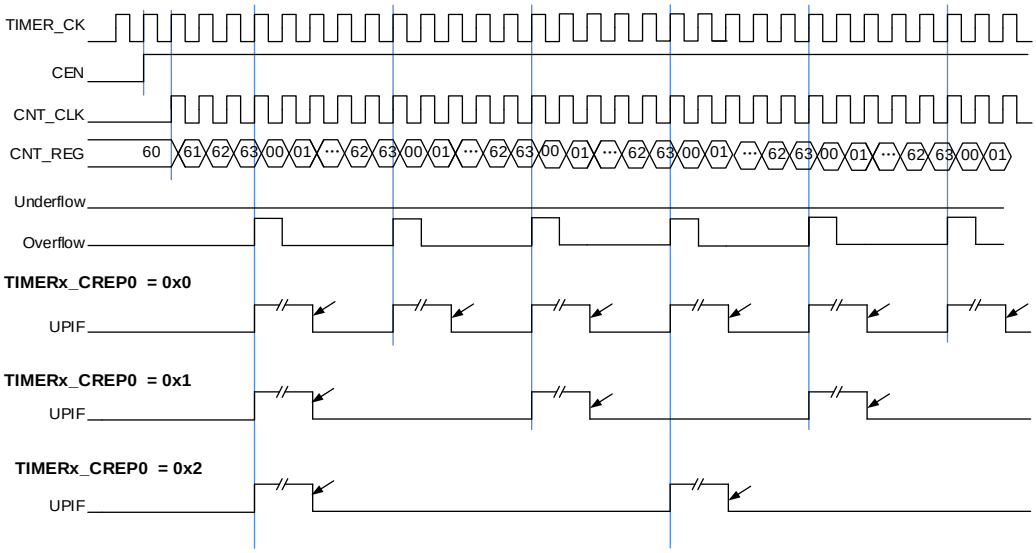
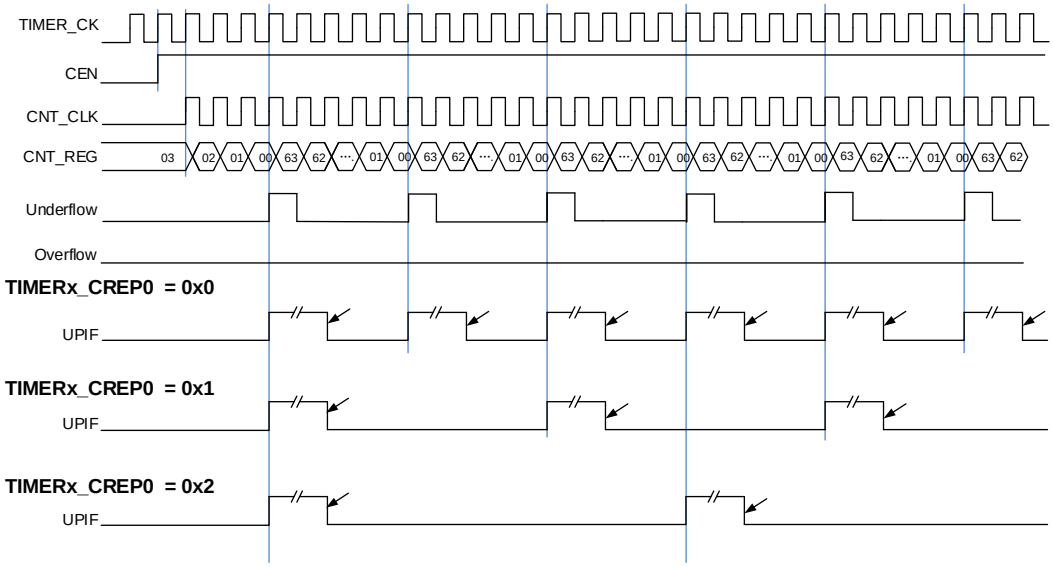


Figure 23-11. Repetition counter timing diagram of down counting mode



Capture / compare channels

The advanced timer has eight independent channels which can be used as capture inputs or compare outputs. Each channel is built around a channel capture compare register including an input stage, a channel controller and an output stage.

When the channels are used for input, channel x and multi mode channel x can perform input capture independently; when the channels are used for comparison output, the channel x and multi mode channel x can output independent 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_MCHxCV(x=0...3) registers, at the same time the CHxIF / MCHxIF (x=0...3) bits are set and the channel interrupt is generated if it is enabled when CHxIE / MCHxIE =1(x=0...3).

Figure 23-12. Input capture logic for channel 0

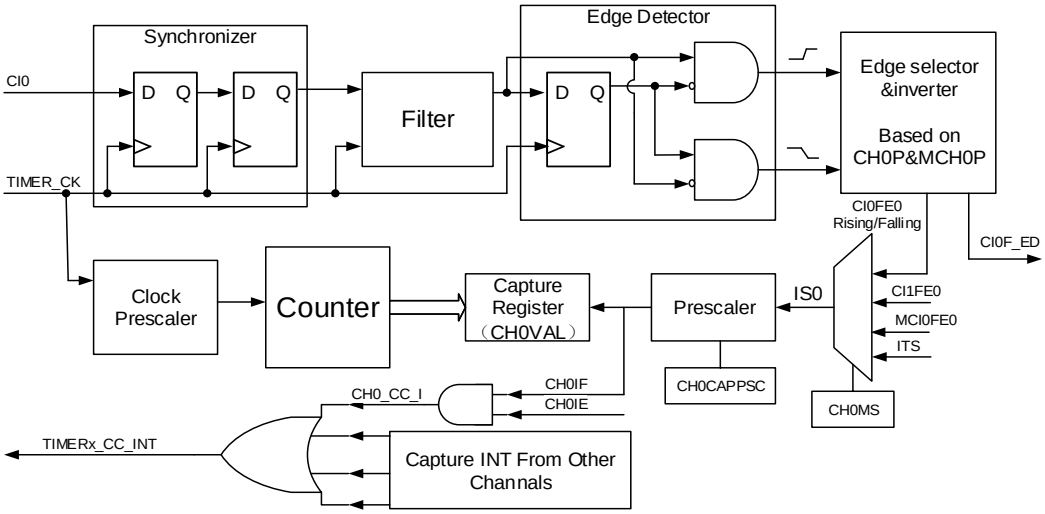
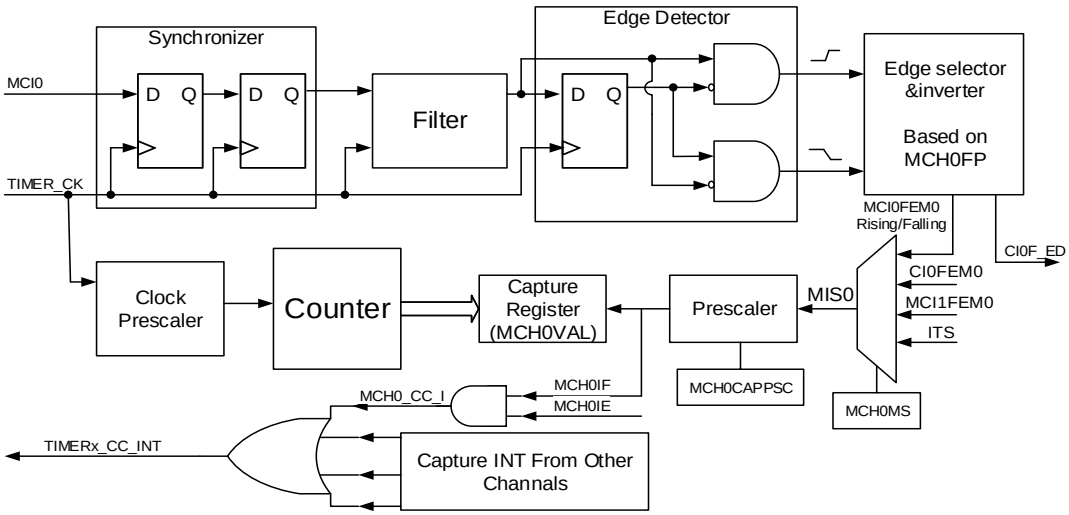


Figure 23-13. Input capture logic for multi mode channel 0



The input signals of channelx (CIx / MCIx) can be the TIMERx_CHx / TIMERx_MCHxCV signal or the XOR signal of the TIMERx_CH0, TIMERx_CH1 and TIMERx_CH2 signals (just for CI0).

First, the input signal of channel (CIx / MCIx) is synchronized to TIMER_CK signal, and then sampled by a digital filter to generate a filtered input signal. Then through the edge detector, the rising or falling edge is detected by configuring CHxP / MCHxP or MCHxFP bits. The input capture signal can also be selected from the input signal of other channel or the internal trigger signal by configuring CHxMS / MCHxMS bits. The IC prescaler makes several input events generate one effective capture event. On the capture event, TIMERx_CHxCV / TIMERx_MCHxCV will store the value of counter.

So, the process can be divided into several steps as below:

Step1: Filter configuration (CHxCAPFLT bit in TIMERx_CHCTL0 register and MCHxCAPFLT bit in TIMERx_MCHCTL0 register).

Based on the input signal and quality of requested signal, configure compatible CHxCAPFLT or MCHxCAPFLT bit.

Step2: Edge selection (CHxP and MCHxP bits in TIMERx_CHCTL2 register, MCHxFP[1:0] bits in TIMERx_MCHCTL2 register).

Rising edge or falling edge, choose one by configuring CHxP and MCHxP bits or MCHxFP[1:0] bits.

Step3: Capture source selection (CHxMS bit in TIMERx_CHCTL0 register, MCHxMS bit in TIMERx_MCHCTL0 register).

As soon as selecting one input capture source by CHxMS, the channel must be set to input mode (CHxMS! =0x000 or MCHxMS !=0x000) and TIMERx_CHxCV / TIMERx_MCHxCV cannot be written any more.

Step4: Interrupt enable (CHxIE and CHxDEN bits, MCHxIE and MCHxDEN bits in TIMERx_DMAINTEN).

Enable the related interrupt to get the interrupt and DMA request.

Step5: Capture enable (CHxEN and MCHxEN bits in TIMERx_CHCTL2).

Result: When the wanted input signal is captured, TIMERx_CHxCV / TIMERx_MCHxCV will be set by counter's value and CHxIF / MCHxIF bit is asserted. If the CHxIF / MCHxIF bit is 1, the CHxOF / MCHxOF bit will also be asserted. The interrupt and DMA request will be asserted or not based on the configuration of CHxIE and CHxDEN bits, MCHxIE and MCHxDEN bits in TIMERx_DMAINTEN.

Direct generation: A DMA 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 23-14. Output compare logic (when MCHxMSEL = 2'b00, x=0, 1, 2, 3) and ***Figure 23-15. Output compare logic (when MCHxMSEL = 2'b11, x=0,1,2,3)*** show the logic circuit of output compare mode.

Figure 23-14. Output compare logic (when MCHxMSEL = 2'b00, x=0, 1, 2, 3)

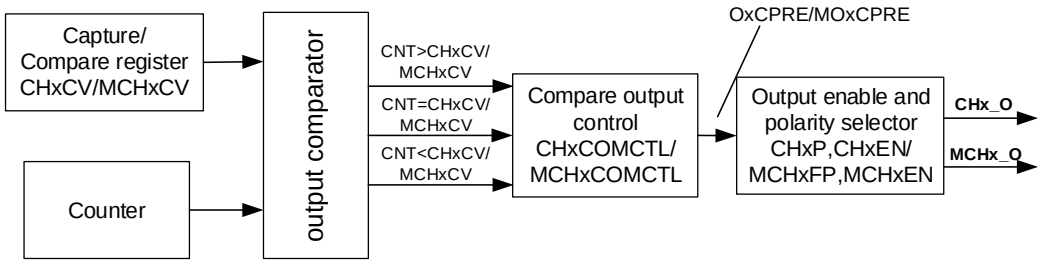
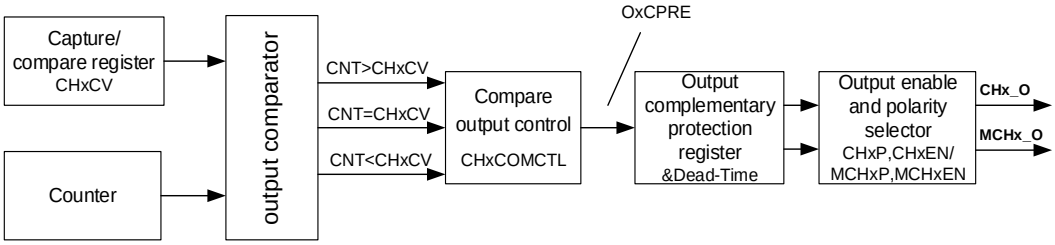


Figure 23-15. Output compare logic (when MCHxMSEL = 2'b11, x=0,1,2,3)



The relationship between the channel output signal CHx_O / MCHx_O and the OxCPRE / MOxCPRE signal (more details refer to ***Channel output prepare signal***) is described as blew(the active level of OxCPRE is high and the active level of MOxCPRE is high).

- When MCHxMSEL=2'b00 (in TIMERx_CTL2 register), the MCHx_O output is independent from the CHx_O output. The output level of CHx_O depends on OxCPRE signal, CHxP bit and CHxEN bit (please refer to the TIMERx_CHCTL2 register for more details). The output level of MCHx_O depends on MOxCPRE signal, MCHxFP[1:0] bits and MCHxEN bit (please refer to the TIMERx_MCHCTL2 and TIMERx_CHCTL2 registers for more details). Please refer to ***Figure 23-14. Output compare logic (when MCHxMSEL = 2'b00, x=0, 1, 2, 3)***.
- When MCHxMSEL=2'b11, the MCHx_O output is the inverse of the CHx_O output. The output level of CHx_O / MCHx_O depends on OxCPRE signal, CHxP / MCHxP bits and CHxEN / MCHxEN bits. Please refer to ***Figure 23-15. Output compare logic (when MCHxMSEL = 2'b11, x=0,1,2,3)***.

For examples (the MCHx_O output is independent from the CHx_O output):

1) Configure CHxP=0 (the active level of CHx_O is high, the same as OxCPRE), CHxEN=1 (the output of CHx_O is enabled):
   If the output of OxCPRE is active(high) level, the output of CHx_O is active(high) level;
   If the output of OxCPRE is inactive(low) level, the output of CHx_O is active(low) level.

2) Configure MCHxP=1 (the active level of MCHx_O is low, contrary to MOxCPRE), MCHxEN=1 (the output of MCHx_O is enabled):

If the output of MOxCPRE is active(high) level, the output of 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_CCHP0 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.

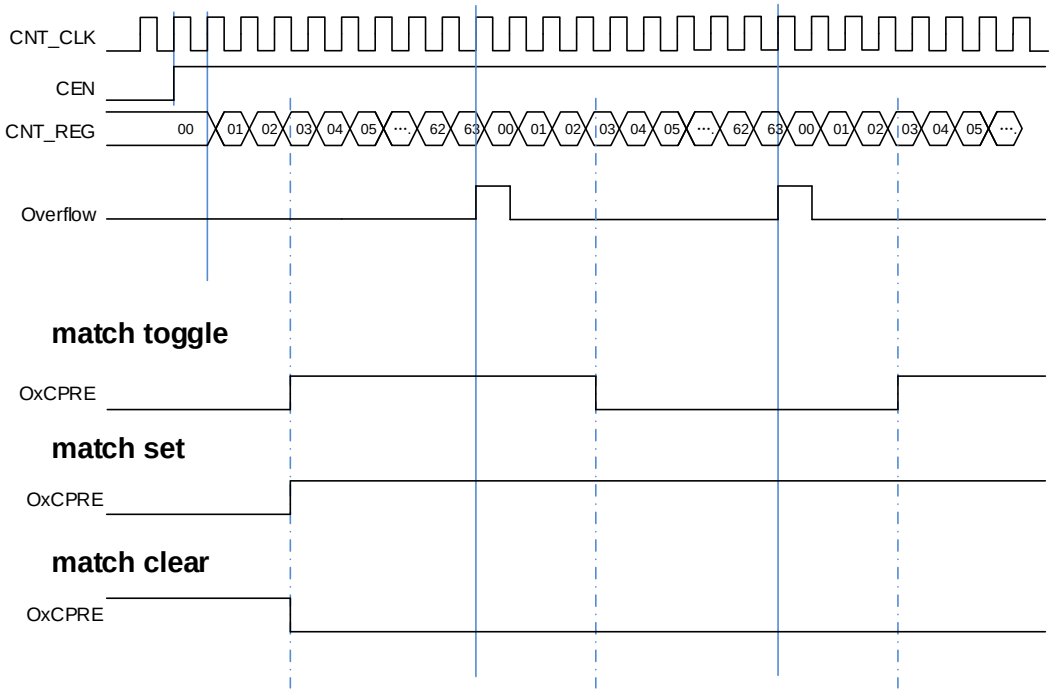
Step4: Compare output timing configuration by TIMERx_CAR and TIMERx_CHxCV / TIMERx_MCHxCV.

The TIMERx_CHxCV / TIMERx_MCHxCV can be changed ongoing to meet the expected waveform.

Step5: Start the counter by configuring CEN to 1.

Figure 23-16. Output-compare in three modes shows the three compare modes: toggle/set/clear. CARL=0x63, CHxVAL=0x3.

Figure 23-16. 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 23-17. 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 23-18. Timing diagram of CAPWM*** shows the CAPWM output and interrupts waveform.

In up counting mode, if the value of TIMERx_CHxCV / TIMERx_MCHxCV is greater than the value of TIMERx_CAR, the output will be always active in PWM mode 0 (CHxCOMCTL / MCHxCOMCTL =4'b0110). And if the value of TIMERx_CHxCV / TIMERx_MCHxCV is greater than the value of TIMERx_CAR, the output will be always inactive in PWM mode 1 (CHxCOMCTL / MCHxCOMCTL =4'b0111).

Figure 23-17. Timing diagram of EAPWM

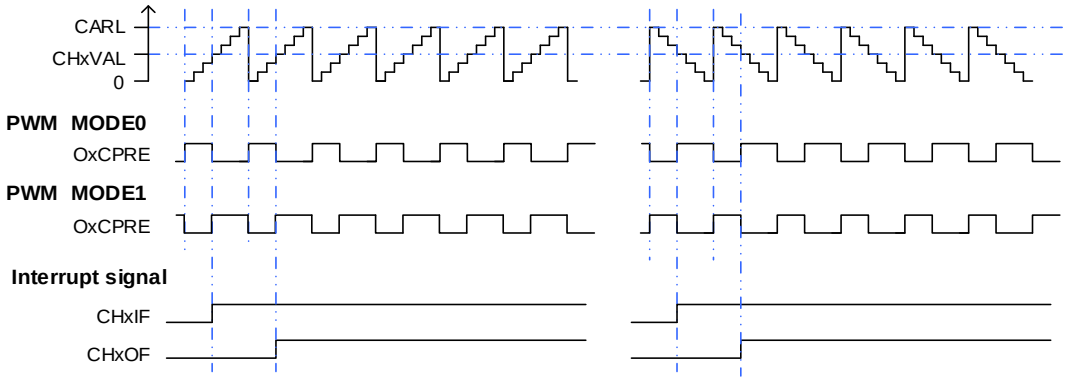
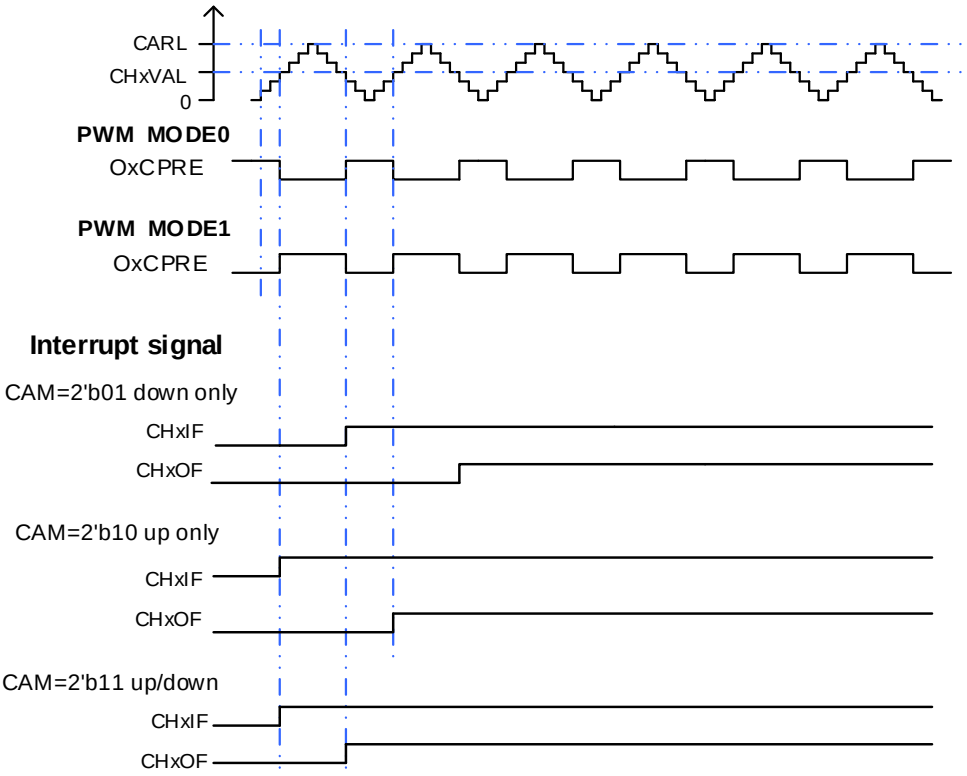


Figure 23-18. Timing diagram of CAPWM

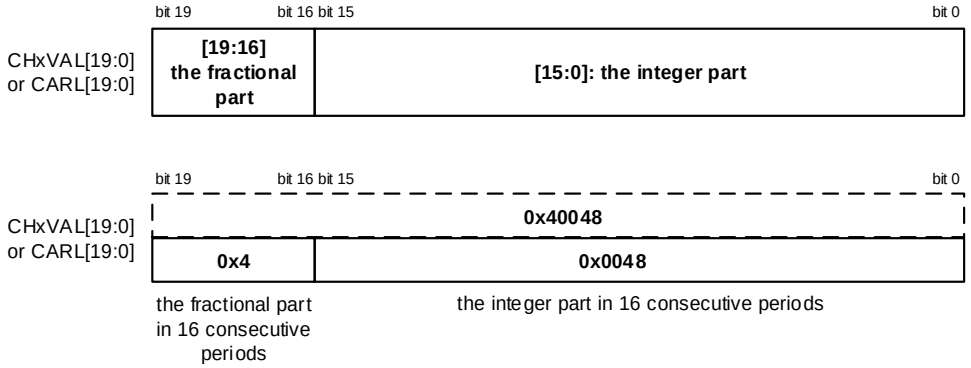


Adjustment mode

Adjustment mode is enabled by setting ADMEN bit in TIMERx_CTL0 register to 1, and this mode can improve the effective resolution of the output PWM wave. The duty cycle resolution can be improved by the CHxVAL[19:0] bit-field in the TIMERx_CHxCV register, and the PWM frequency resolution can be improved by the CARL[19:0] bit-field in the TIMERx_CAR register.

When the adjustment mode is enabled, the low 16 bits CHxVAL[15:0] bit-field and CARL[15:0] bit-field are used for the integer part and the high 4 bits CHxVAL[19:16] and CARL[19:16] are used for the fractional part. By adjust the CHxVAL or CARL values over 16 consecutive periods (no more than one TIMER clock cycle at a time) in a predefined way, can increase 16-fold in resolution.

Figure 23-19. Adjustment mode: Data format and the register bit-field



Depending on the configuration of the ADMEN bit (set or clear), the CHxVAL and CARL bit-field are automatically updated. To clear the ADMEN bit, must follow the following steps:

1. CEN bit and ARSE bits must be cleared.
2. CARL[19:16] bit-field must be cleared.
3. ADMEN bit must be cleared.
4. CHxIF bit must be cleared.
5. Set the CEN bit to 1.

The following formula to calculate the PWM Resolution:

$$\text{Resolution} = f_{PSC_CLK} \ / \ f_{pwm} \quad (23\text{-}1)$$

According to Equation ***(23-1)***, when the adjustment mode is disabled (ADMEN=0), the PWM minimum frequency f_{pwm}:

$$(f_{pwm})_{min} = f_{PSC_CLK} \ / \ 65536 \quad (23\text{-}2)$$

When the adjustment mode is enabled (ADMEN=1),

$$(f_{pwm})_{min} = f_{PSC_CLK} \ / \ (65535 + 15 \ / \ 16) \quad (23\text{-}3)$$

When the adjustment mode is enabled, the max values of the CHxVAL[19:0] bit-field and CARL[19:0] bit-field are 0xFFFFE (the integer part is 0xFFFE, the fractional part is 0xF).

The changes of duty cycle and period within 16 consecutive periods are shown in ***Figure 23-20. PWM adjustment mode schematic diagram*** and ***Table 23-3. CHxVAL and CARL bit-field change in edge-aligned***.

Figure 23-20. PWM adjustment mode schematic diagram

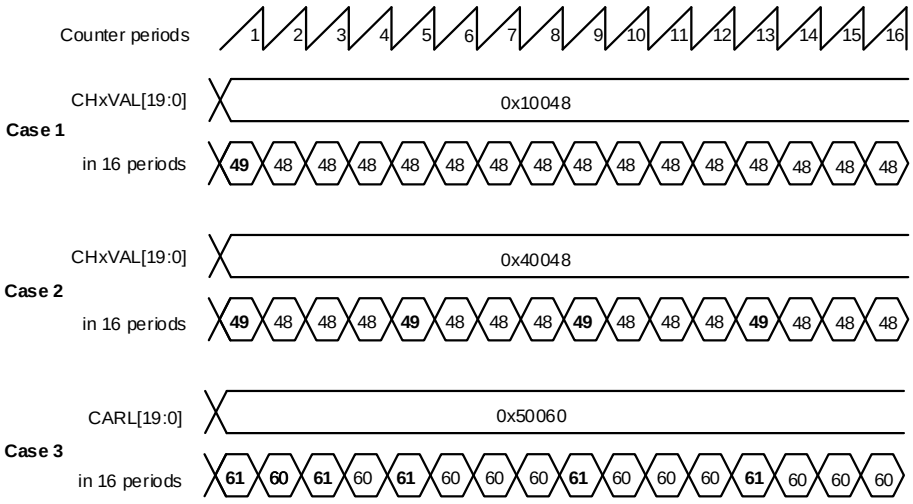


Table 23-3. CHxVAL and CARL bit-field change in edge-aligned

CHxVAL[19:16] / CARL[19:16]	Period															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0000	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0001	+1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0010	+1	-	-	-	-	-	-	-	+1	-	-	-	-	-	-	-
0011	+1	-	-	-	+1	-	-	-	+1	-	-	-	-	-	-	-
0100	+1	-	-	-	+1	-	-	-	+1	-	-	-	+1	-	-	-
0101	+1	-	+1	-	+1	-	-	-	+1	-	-	-	+1	-	-	-
0110	+1	-	+1	-	+1	-	-	-	+1	-	+1	-	+1	-	-	-
0111	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	-	-
1000	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-
1001	+1	+1	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-
1010	+1	+1	+1	-	+1	-	+1	-	+1	+1	+1	-	+1	-	+1	-
1011	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-	+1	-	+1	-
1100	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-
1101	+1	+1	+1	+1	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-
1110	+1	+1	+1	+1	+1	+1	+1	-	+1	+1	+1	+1	+1	+1	+1	-
1111	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	-

The PWM adjustment mode can also use in center-aligned counting mode, the details are shown in ***Table 23-4. CHxVAL and CARL bit-field changes in the center-aligned counting mode***. The adjustment mode is applied over 8 consecutive PWM cycles.

Table 23-4. CHxVAL and CARL bit-field changes in the center-aligned counting mode

CHxVAL [19:16] / CARL [19:16]	Period															
	1		2		3		4		5		6		7		8	
	Up	Down	Up	Down	Up	Down	Up	Down	Up	Down	Up	Down	Up	Down	Up	Down
0000	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0001	+1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0010	+1	-	-	-	-	-	-	-	+1	-	-	-	-	-	-	-
0011	+1	-	-	-	+1	-	-	-	+1	-	-	-	-	-	-	-
0100	+1	-	-	-	+1	-	-	-	+1	-	-	-	+1	-	-	-
0101	+1	-	+1	-	+1	-	-	-	+1	-	-	-	+1	-	-	-
0110	+1	-	+1	-	+1	-	-	-	+1	-	+1	-	+1	-	-	-
0111	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	-	-
1000	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-
1001	+1	+1	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-
1010	+1	+1	+1	-	+1	-	+1	-	+1	+1	+1	-	+1	-	+1	-
1011	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-	+1	-	+1	-
1100	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-
1101	+1	+1	+1	+1	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-
1110	+1	+1	+1	+1	+1	+1	+1	-	+1	+1	+1	+1	+1	+1	+1	-
1111	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	-

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.

The PWM period is determined by (CARL + 0x0001) and the PWM pulse width is determined by the following table.

Table 23-5.The Composite PWM pulse width

Condition	Mode	PWM pulse width
CHxVAL < CHxCOMVAL_ADD ≤ CARL	PWM mode 0	(CARL + 0x0001) + (CHxVAL – CHxCOMVAL_ADD)

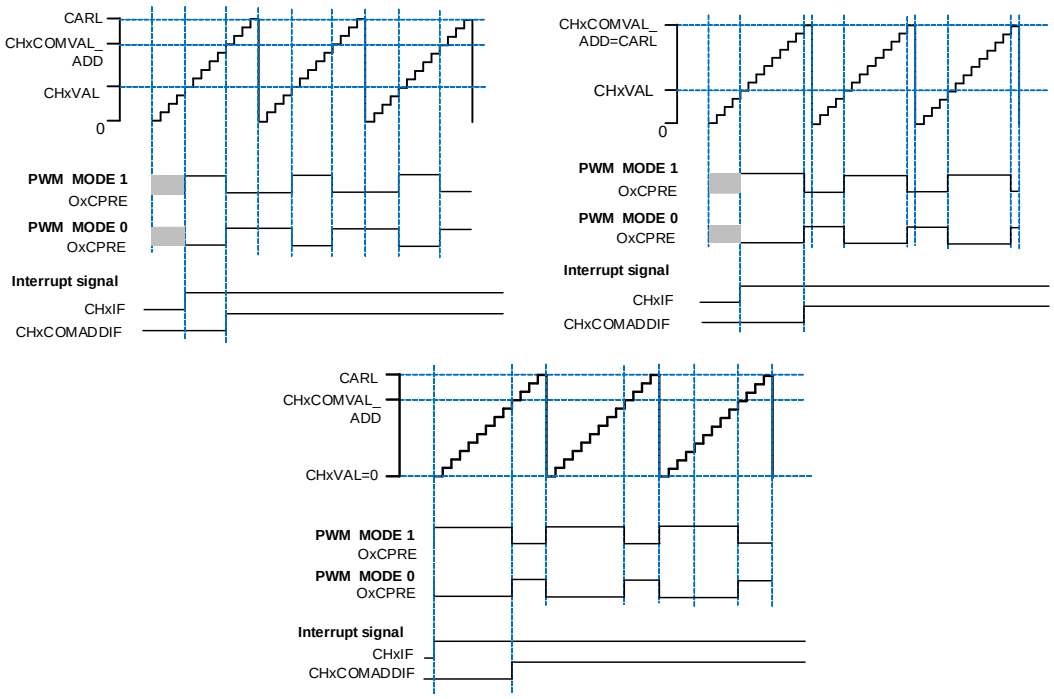
Condition	Mode	PWM pulse width
	PWM mode 1	(CHxCOMVAL_ADD – CHxVAL)
CHxCOMVAL_ADD < CHxVAL ≤ CARL	PWM mode 0	(CHxVAL - CHxCOMVAL_ADD)
	PWM mode 1	(CARL + 0x0001) + (CHxCOMVAL_ADD – CHxVAL)
(CHxVAL = CHxCOMVAL_ADD ≤ CARL) or (CHxVAL > CARL > CHxCOMVAL_ADD)	PWM mode 0 (up counting) or PWM mode 1 (down 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:

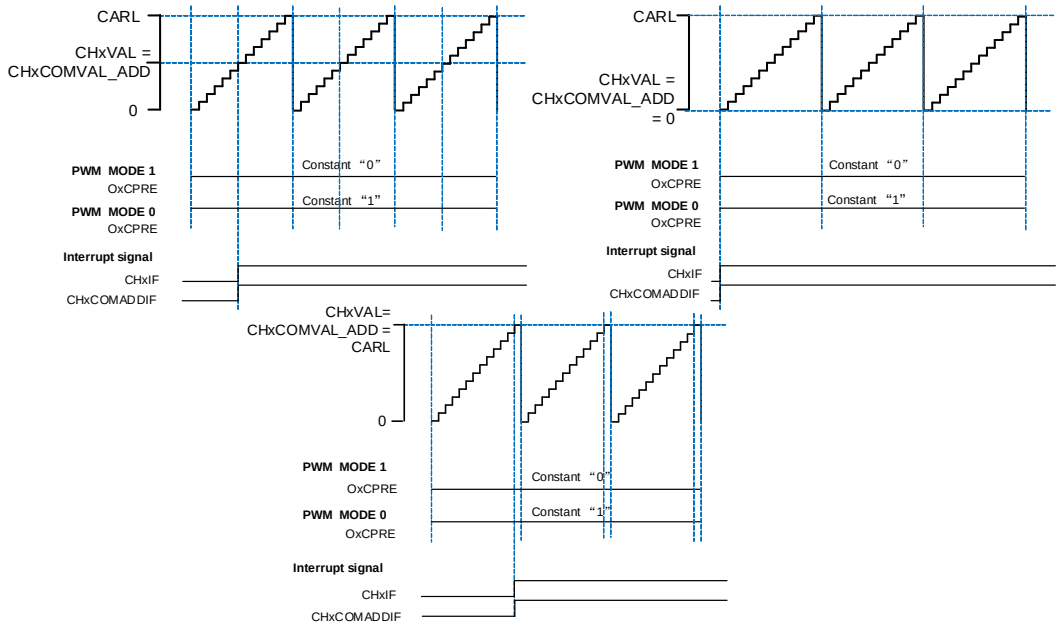
- CHxVAL < CHxCOMVAL_ADD, and the values of CHxVAL and CHxCOMVAL_ADD between 0 and CARL.

Figure 23-21. Channel x output PWM with (CHxVAL < CHxCOMVAL_ADD)



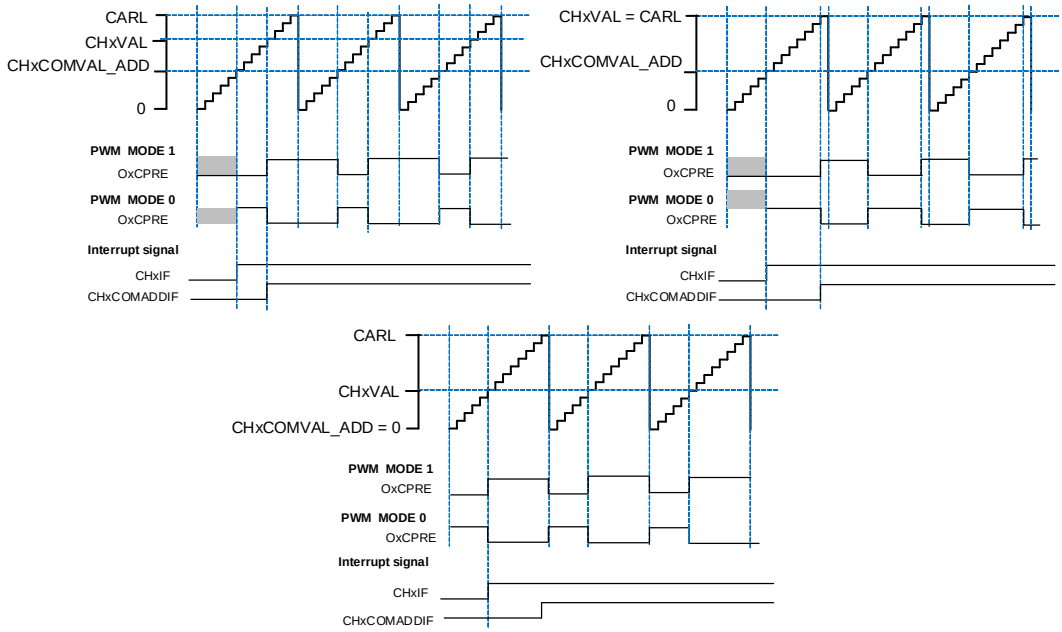
- CHxVAL = CHxCOMVAL_ADD, and the value of CHxVAL and CHxCOMVAL_ADD between 0 and CARL.

Figure 23-22. Channel x output PWM with (CHxVAL = CHxCOMVAL_ADD)



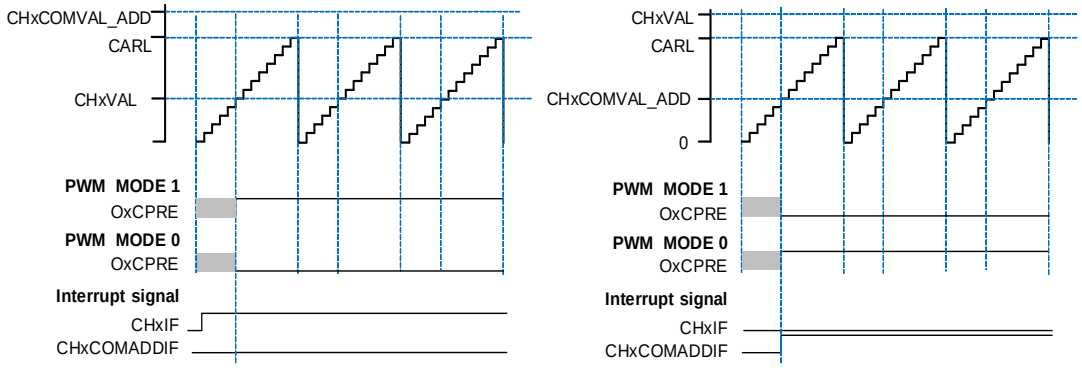
- CHxVAL > CHxCOMVAL_ADD, and the value of CHxVAL and CHxCOMVAL_ADD between 0 and CARL.

Figure 23-23. Channel x output PWM with (CHxVAL > CHxCOMVAL_ADD)



- One of the value of CHxVAL and CHxCOMVAL_ADD exceeds CARL.

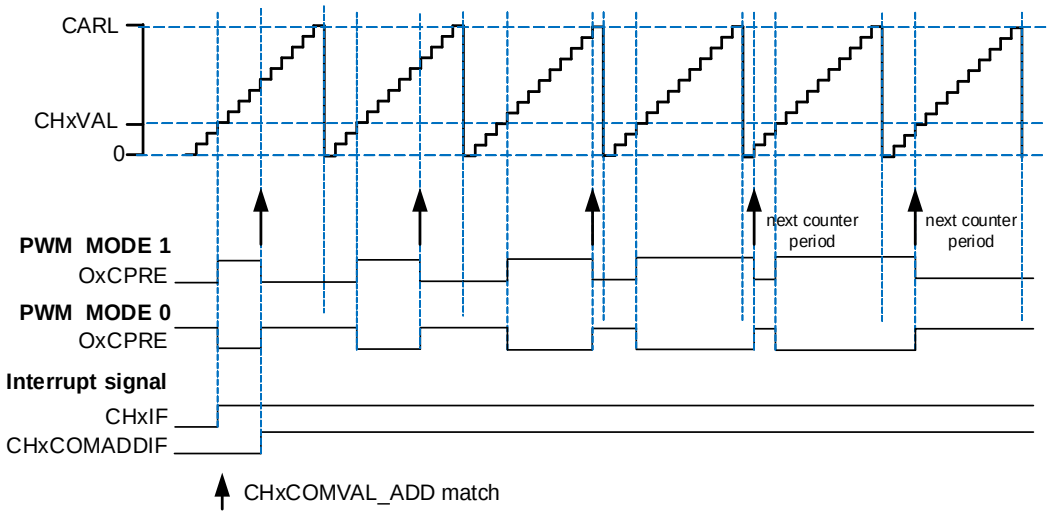
Figure 23-24. Channel x output PWM with CHxVAL or CHxCOMVAL_ADD exceeds CARL



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 23-25. Channel x output PWM duty cycle changing with CHxCOMVAL_ADD*** shows the PWM output and interrupts waveform.

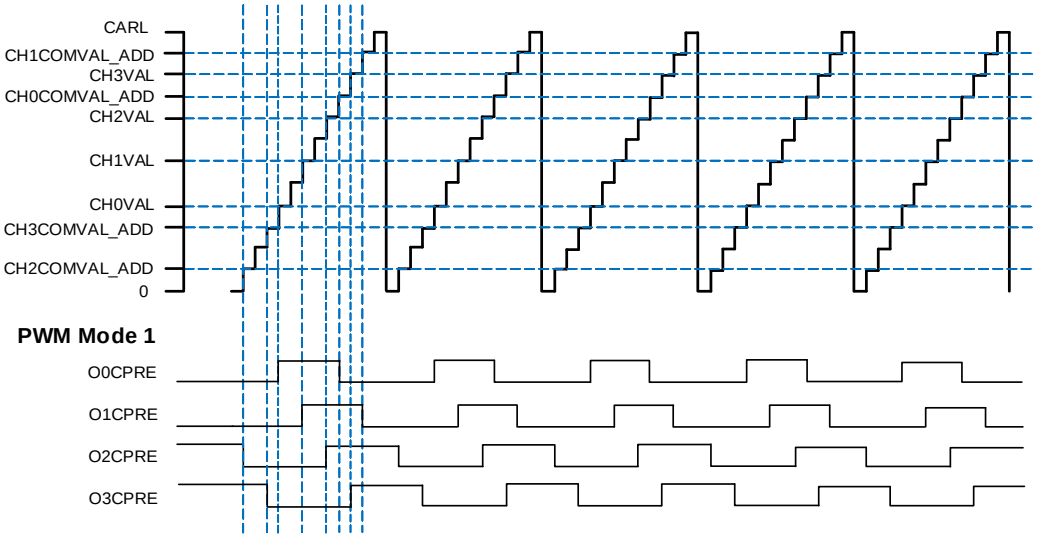
In some cases, the CHxCOMVAL_ADD match can happen on the next counter period (the value of CHxCOMVAL_ADD was written after the counter reaches the value of CHxVAL, and the value of CHxCOMVAL_ADD was less than or equal to the CHxVAL).

Figure 23-25. 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 23-26. Four Channels outputs in Composite PWM mode



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 23-27. CHx_O output with a pulse in edge-aligned mode (CHxOMPSEL≠2'b00)

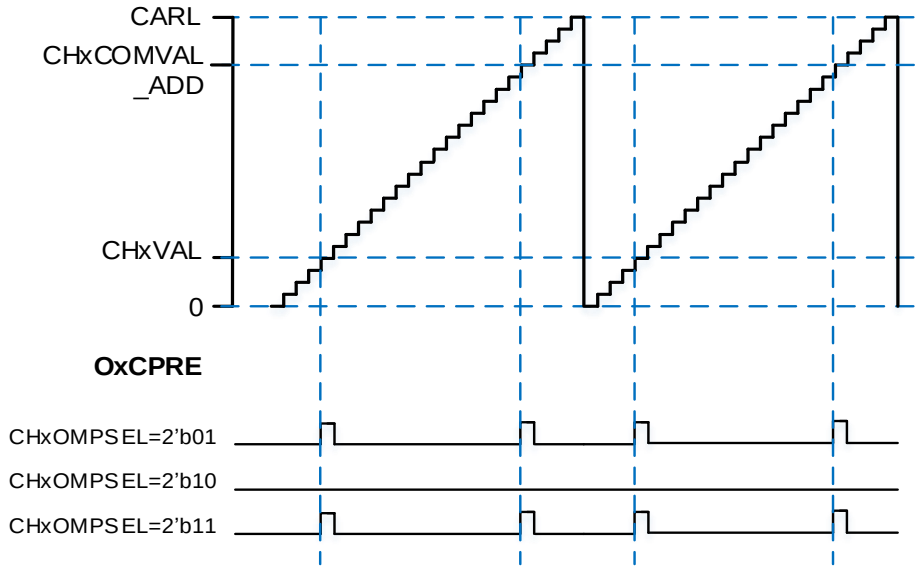
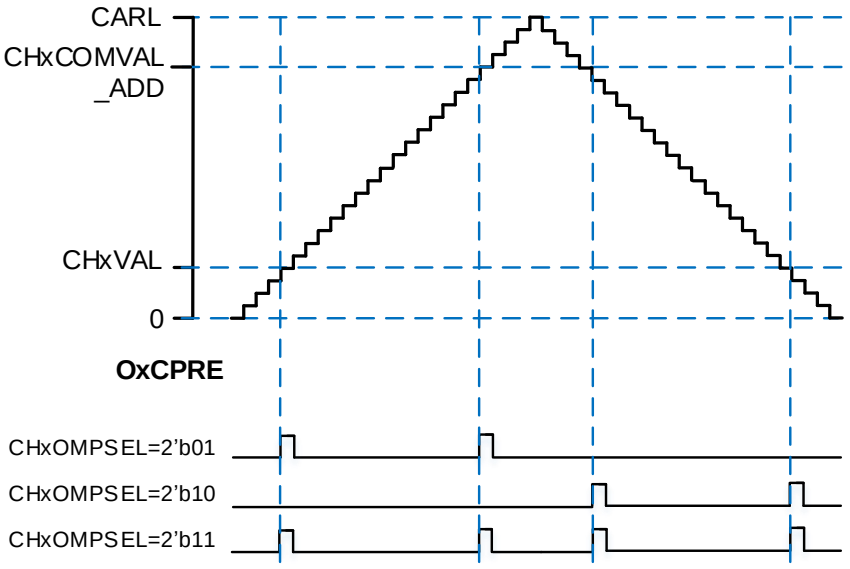


Figure 23-28. CHx_O output with a pulse in center-aligned mode (CHxOMPSEL≠2'b00)



Channel output prepare signal

As is shown in ***Figure 23-14. Output compare logic (when MCHxMSEL = 2'b00, x=0, 1, 2, 3)*** and ***Figure 23-15. Output compare logic (when MCHxMSEL = 2'b11, x=0,1,2,3)***, when TIMERx is configured in compare match output mode, a middle signal named OxCPRE or MOxCPRE (channel x output or multi mode channel x output prepare signal) will be generated before the channel outputs signal.

The OxCPRE and MOxCPRE signal have several types of output function. The OxCPRE signal type is defined by configuring the CHxCOMCTL bit and the MOxCPRE signal type is defined by configuring the MCHxCOMCTL bit.

Take OxCPRE as an example for description below, these include keeping the original level by configuring the CHxCOMCTL field to 0x00, setting to high by configuring the CHxCOMCTL field to 0x01, setting to low by configuring the CHxCOMCTL field to 0x02 or toggling signal by configuring the CHxCOMCTL field to 0x03 when the counter value matches the content of the TIMERx_CHxCV register.

The PWM mode 0 / PWM mode 1 output is another output type of OxCPRE which is setup by configuring the CHxCOMCTL field to 0x06 / 0x07. In these modes, the OxCPRE signal level is changed according to the counting direction and the relationship between the counter value and the TIMERx_CHxCV content. Refer to the definition of relative bit for more details.

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

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

Clear the channel output prepare signal

The OxCPRE and MOxCPRE signal can be cleared by the OCPRE_CLR_INT signal when the CHxCOMCEN or MCHxCOMCEN bit (in TIMERx_CHCTLy / TIMERx_MCHCTLy register) is set. This function can be used in the compare output modes which configured in CHxCOMCTL[3:0] or MCHxCOMCTL[3:0] bit-field (without the value of 4'b0100 and 4'b0101).

The source of the OCPRE_CLR_INT signal can be selected by the OCRC bit in the TIMERx_SMCFG register.

When OCRC is reset, the OCPRE_CLR_INT is connected to the OCPRE_CLR input. The OxCPRE /MOxCPRE signal is cleared by the high level of the OCPRE_CLR_INT signal, and restored until the next update event occurs. The OCPRE_CLR inputs can be selected in OCRINSEL[2:0] bit-field in the TIMERx_AFCTL1 register.

When OCRC is set, the OCPRE_CLR_INT is connected to the ETIF. The OCPRE_CLR_INT

input polarity is configured by ETP bit in TIMERx_SMCFG register. The ETPSC[1:0] bit-field must be set to 2'b00.

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. The TIMERx has 4 pairs of channels, all the four pairs have this function. The complementary signals CHx_O and MCHx_O are controlled by a group of parameters: the CHxEN and MCHxEN bits in the TIMERx_CHCTL2 register, the POEN, ROS and IOS bits in the TIMERx_CCHP0 register(when CHx_O and MCHx_O channels has separated deadtime value and break function, please refer to ***Separated dead time insertion and Break function***), 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 23-6. Complementary outputs controlled by parameters (MCHxMSEL =2'b11)***.

Table 23-6. Complementary outputs controlled by parameters (MCHxMSEL =2'b11)

<table>
<tr><th colspan="5">Complementary Parameters</th><th colspan="2">Output Status</th></tr>
<tr><th>POEN</th><th>ROS</th><th>IOS</th><th>CHxEN</th><th>MCHxEN</th><th>CHx_O</th><th>MCHx_O</th></tr>
<tr><td rowspan="5">0</td><td rowspan="5">0 / 1</td><td rowspan="3">0</td><td rowspan="2">0</td><td>0</td><td colspan="2">CHx_O / MCHx_O = LOW
CHx_O / MCHx_O output disable(1).</td></tr>
<tr><td>1</td><td colspan="2" rowspan="3">CHx_O/ MCHx_O output “off-state” (2):
the CHx_O/ MCHx_O output inactive level firstly: CHx_O = CHxP, MCHx_O = CHxNP; If the clock for deadtime generator is present, after a deadtime: CHx_O = ISOx, MCHx_O = ISOxN. (3)</td></tr>
<tr><td rowspan="2">1</td><td>0</td></tr>
<tr><td>1</td></tr>
<tr><td>1</td><td>x</td><td>x</td><td colspan="2">CHx_O/ MCHx_O output “off-state”:
the CHx_O/ MCHx_O output inactive level firstly: CHx_O = CHxP, MCHx_O = CHxNP; If the clock for deadtime generator is present, after a deadtime: CHx_O = ISOx, MCHx_O = ISOxN.</td></tr>
<tr><td rowspan="3">1</td><td rowspan="3">0</td><td rowspan="3">0 / 1</td><td rowspan="2">0</td><td>0</td><td colspan="2">CHx_O / MCHx_O = LOW
CHx_O / MCHx_O output disable.</td></tr>
<tr><td>1</td><td>CHx_O = LOW
CHx_O output disable.</td><td>MCHx_O=OxCPRE⊕(4)MCHxP
MCHx_O output enable.</td></tr>
<tr><td>1</td><td>0</td><td>CHx_O=OxCPRE⊕CHxP
CHx_O output enable.</td><td>MCHx_O = LOW
MCHx_O output disable.</td></tr>
</table>

Complementary Parameters					Output Status	
POEN	ROS	IOS	CHxEN	MCHxEN	CHx_O	MCHx_O
				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:

(5) 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.

(6) “off-state”: CHx_O / MCHx_O output with inactive state (e.g., CHx_O = 0⊕CHxP = CHxP).

(7) See Break mode section for more details.

(8) ⊕: Xor calculate.

(9) (! OxCPRE)：the complementary output of the OxCPRE signal.

Dead time insertion

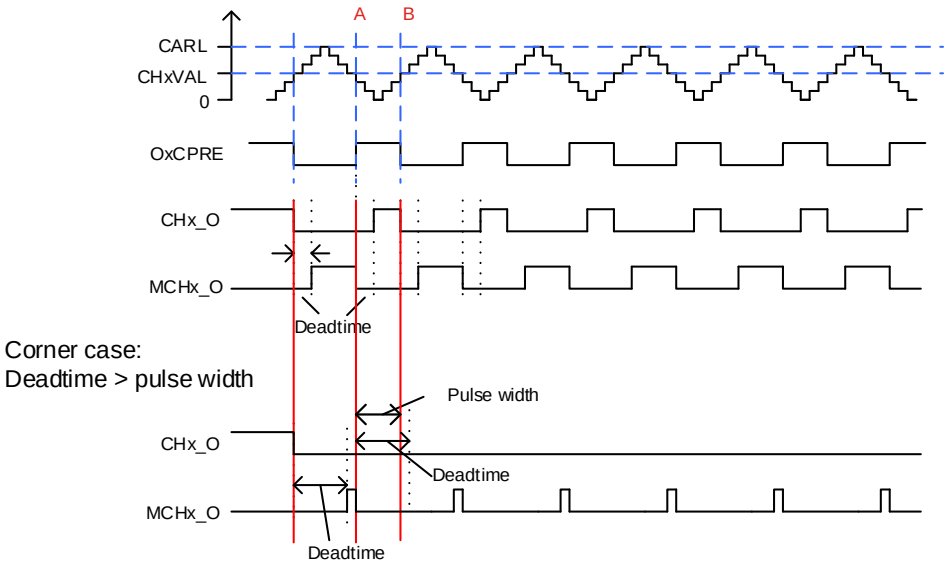
The dead time insertion is enabled when MCHxMSEL=2’b11 and both CHxEN and MCHxEN are configured to 1’b1, it is also necessary to configure POEN to 1. The field named DTCFG defines the dead time delay that can be used for all channels. Refer to the ***Complementary channel protection register 0 (TIMERx_CCHP0)*** 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 23-29. 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 23-29. Complementary output with dead time insertion***).

Figure 23-29. Complementary output with dead time insertion



When CHx_O and MCHx_O channels has separated deadtime value, please refer to ***Separated dead time insertion and Break function***.

By configuring the DTIENCHx (x=0...3) bit in the TIMERx_CTL2 register to realize the independent control of dead-time insertion function for each pair of channels. When the DTIENCHx (x=0...3) bit is “0”, the corresponding channels CHx_O and MCHx_O will not be inserted into the dead-time.

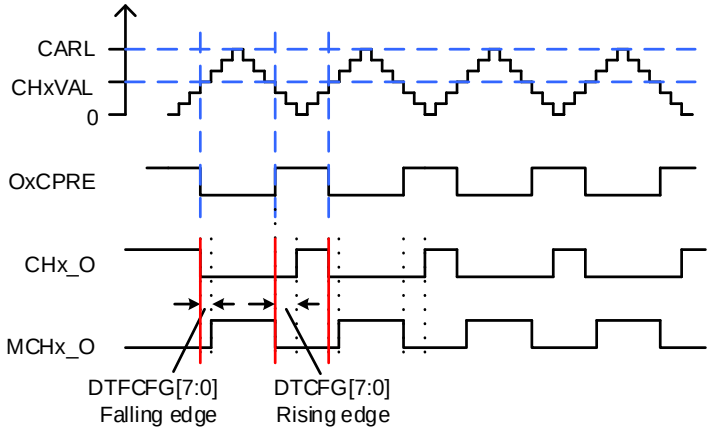
Different dead time insertion

The CHx_O and MCHx_O signal can output with different dead time when the DTDIFEN bit (in the TIMERx_CCHP1register) is set to 1. As shown in ***Figure 23-30. 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 or TIMERx_FCCHPy 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 or TIMERx_FCCHPy register.

The dead time can be modified on-the-fly when the CHx_O and MCHx_O signals are output. When DTMODEN bit in TIMERx_CCHP1 register is set, this function can be enabled. The new value of DTCFG[7:0] bit-field and DTFCFG[7:0] bit-field will be active when the next update event occurs.

Figure 23-30. Complementary output with different dead time(DTDIFEN=1)



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 advanced timers have two kinds of break function: BREAK0 and BREAK1. The break functions can be enabled by setting the BRK0EN and BRK1EN bits in the TIMERx_CCHP0 register. The break input polarities are configured by the BRK0P and BRK1P bits in TIMERx_CCHP0 register, the inputs are active on level.

In break functions, CHx_O and MCHx_O are controlled by the POEN, OAEN, IOS and ROS bits in the TIMERx_CCHP0 register, ISOx and ISOxN bits in the TIMERx_CTL1 register.

The break event is the result of logic ORed of all sources. The break functions can handle three types of event sources:

- External sources: coming from BRKINx (x=0...2) inputs;
- System sources: HXTAL stuck event which is generated by Clock Monitor CKM in RCU, LVD lock event, Cortex®-M33 LOCKUP_LOCK event, SRAM parity error event or flash ECC error event;
- On-chip peripheral events: input by comparator output or HPDF watchdog output.

Break events can also be generated by software using BRK0G or BRK1G bits in the TIMERx_SWEVG register.

Refer to ***Figure 23-31. BREAK0 function logic diagram*** and ***Figure 23-32. BREAK1 function logic diagram***, BRKINx(x=0...2) can select GPIO pins from the TRIGSEL module, which can select by ***Trigger selection for TIMER0 BRKIN register (TRIGSEL_TIMER0BRKIN)***, ***Trigger selection for TIMER7 BRKIN register (TRIGSEL_TIMER7BRKIN)*** and ***Trigger selection for TIMER19 BRKIN register (TRIGSEL_TIMER19BRKIN)***.

Figure 23-31. BREAK0 function logic diagram

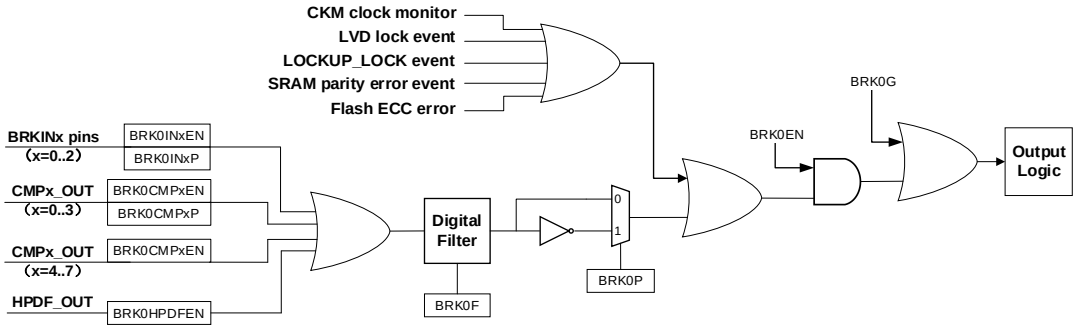
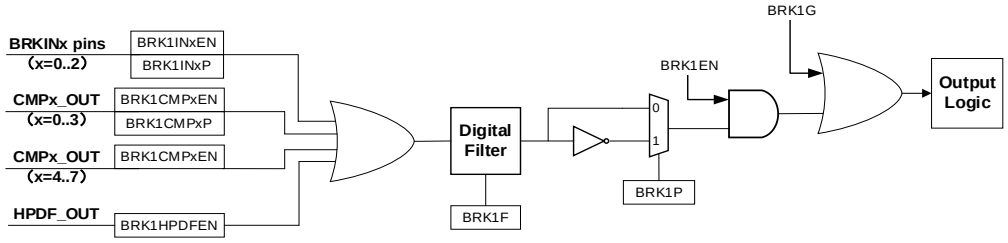


Figure 23-32. BREAK1 function logic diagram

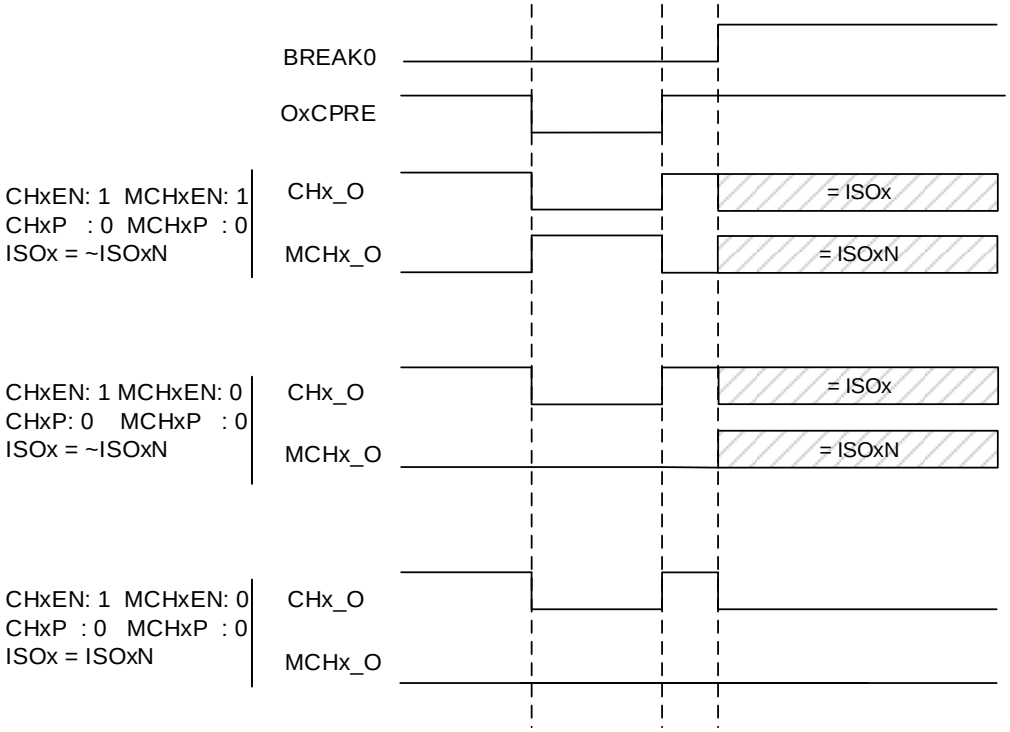


BREAK0 can be used to handle the faults of system sources, on-chip peripheral events and external sources. When a BREAK0 event occurs, the outputs are force at an inactive level, or at a predefined level (either active or inactive) after a deadtime duration. BREAK1 only can be used to handle the faults of on-chip peripheral events and external sources. When a BREAK1 event occurs, the outputs are force at an inactive level.

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 23-33. 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 23-33. Output behavior of the channel in response to BREAK0 (the break input

high active and IOS=1)



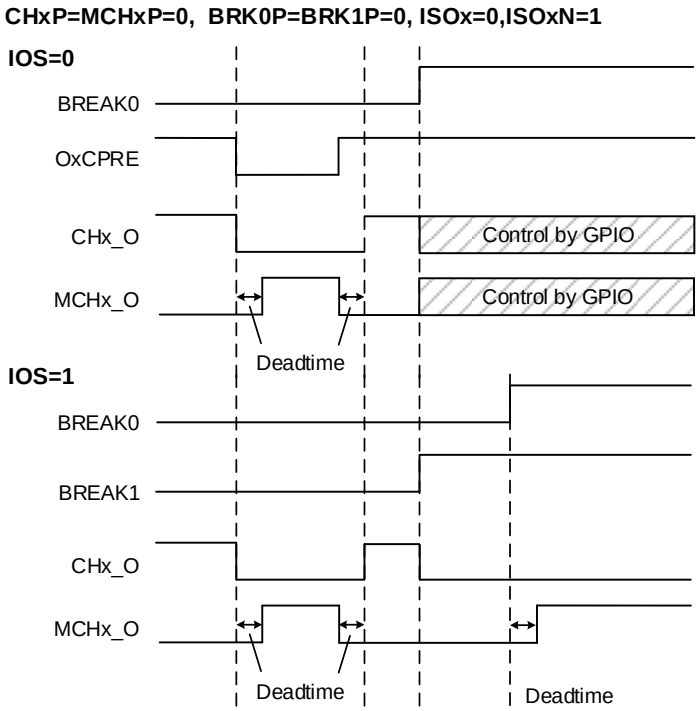
BREAK0 function has a higher priority than BREAK1 function. BREAK1 function only can be used when the IOS =1 and ROS =1.

Table 23-7. Output behavior of the channel in response to a BREAK0 and BREAK1 (the break input is high active)

BREAK0 inputs	BREAK1 inputs	Output Status	
		CHx_O	MCHx_O
High	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: MCHx_O output is inactive and then output to level (by IOSxN bit) after a deadtime time. IOS=0: MCHx_O output disable (inactive).
	Low		
Low	High	CHx_O output disable (inactive).	MCHx_O output disable (inactive).

When a break occurs, the BRKIF bit in the TIMERx_INTF register will be set. If BRKIE is 1, an interrupt will be generated.

Figure 23-34. Output behavior of the channel outputs with the BREAK0 and BREAK1



When CHx_O and MCHx_O channels has separated break function, please refer to ***Separated dead time insertion and Break function***.

By configuring the BRKENCHx (x=0...3) bit in the TIMERx_CTL2 register to realize the independent control of break function for each pair of channels. When the BRKENCHx (x=0...3) bit is “0” and a break event occurs, the corresponding channels CHx_O and MCHx_O will not be changed and the outputs is keeping.

Locked break function

The BRKINx (x=0...2) input pins of advanced timer have the locked break function, this function can be enabled by setting the BRK0LK and BRK1LK bits in the TIMERx_CCHP0 register.

When the locked break function is enabled, the BRKINx (x=0...2) pins need to be configured to open-drain output mode with low level active (BRK0P / BRK1P=0 and BRK0INxP / BRK1INxP=0). When any break source requests occur, the corresponding BRKINx (x=0...2) pin can be forced to low level. If the break input polarity is active high (BRK0P / BRK1P=1 and BRK0INxP / BRK1INxP=1), the locked break function is invalid.

When the break function is enabled (the BRK0EN =1 or BRK1EN = 1), the BRKINx (x=0...2) pin can be forced to low level with the BRK0G or BRK1G bits setting to 1 by software.

When the break function is disabled (the BRK0EN =0 or BRK1EN = 0), setting the BRK0G or BRK1G bits will have no effect on the BRKINx (x=0...2) pin. The BRK0F or BRK1F bits will set and the channel outputs will be in a safe state.

The BRKINx (x=0...2) pin can be released by setting the BRK0REL / BRK1REL bit in the

TIMERx_CCHP0 register. When the break input sources are inactive, the BRK0REL / BRK1REL bit will cleared by hardware and the BRKINx (x=0...2) pin will restore the locked break function.

In the following two cases, the BRKINx (x=0...2) pin cannot be released:

- Break input sources are active: the BRK0REL / BRK1REL bit is set to 1 and the BRKINx (x=0...2) pin locked break function is released. The break events are still active, because the break input sources are still active.
- POEN=1: when the channel outputs are enabled, the BRKINx (x=0...2) pin cannot be released even if the BRK0REL / BRK1REL is set.

Table 23-8. Break function input pins locked / released conditions

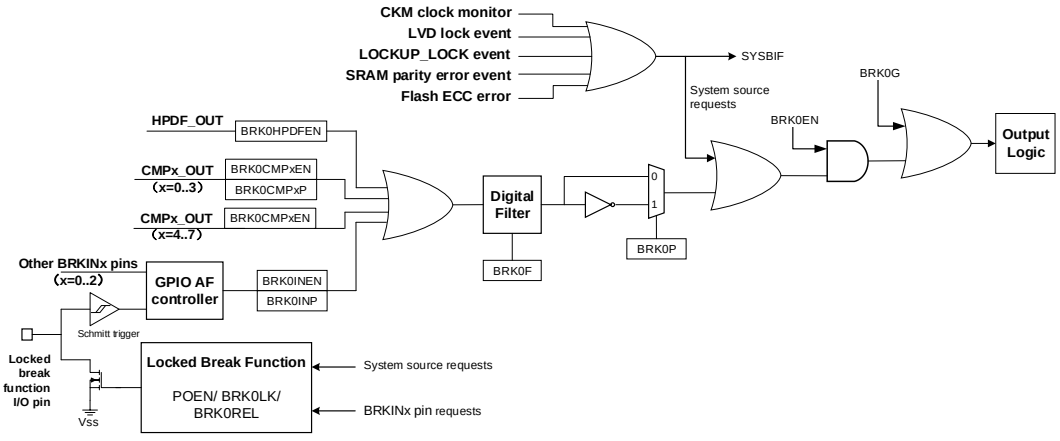
POEN	BRK0LK / BRK1LK	BRK0REL / BRK1REL	Break input pin state
0	1	0	Locked
	1	1	Released

The locked break function of the BREAK0 / BREAK1 input pin BRKINx (x=0...2) is enabled by default (BRK0REL=0 and BRK1REL=0). When the BREAK0 / BREAK1 events occur, the following steps can be used to reconfigure the locked break function:

- Set the BRK0REL or BRK1REL bit to 1 and released the BRKINx (x=0...2) pin;
- The software waits for the system break sources inactive, and then clears the SYSBIF flag;
- The software polls the state of BRK0REL / BRK1REL bits, until the BRK0REL / BRK1REL bits are cleared (cleared by hardware).

Then the locked break function of BREAK0 / BREAK1 input pin is re-enabled, and the channel outputs can be restored by setting the POEN bit to 1 by software.

Figure 23-35. BRKINx (x=0...2) pins logic with BREAK0 function



Separated dead time insertion and Break function

The separated dead time insertion and break function for CHx_O and MCHx_O allows that each pair of channels has its own deadtime value and break function. In this function, CHx_O and MCHx_O are actually controlled by the IOS bit、ROS bit and DTCFG[7:0] bits in

TIMERx_FCCHPy(y=0...3) register.

By configuring the FCCHPyEN (y=0...3) bits in the TIMERx_FCCHPy (y=0...3) registers can select whether each pair of channels uses the separated dead time insertion and break function. When the FCCHPyEN=0, the ROS、IOS and DTCFG[7:0] bits in TIMERx_CCHP0 register is active; When the FCCHPyEN=1, the ROS、IOS and DTCFG[7:0] bits in TIMERx_FCCHP0 register is active.

Quadrature decoder

The quadrature decoder function uses two quadrature inputs 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 23-9. 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 23-9. 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	CI0FE0=1	Down	Up	-	-
TSCFG13[4:0]!= 5'b00000	CI0FE0=0	-	-	-	-
Quadrature decoder mode 4	CI1FE1=1	-	-	Up	Down
TSCFG14[4:0]!= 5'b00000	CI1FE1=0	-	-	-	-

Note: "-" means "no counting"; "X" means impossible. "0" means "low level", "1" means "high level".

Figure 23-36. Example of counter operation in decoder interface mode

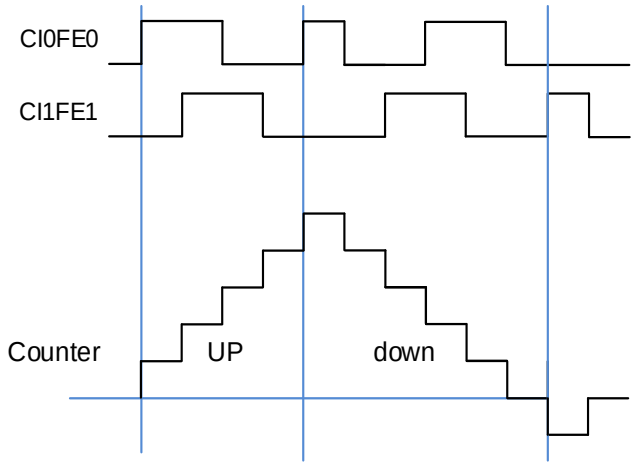
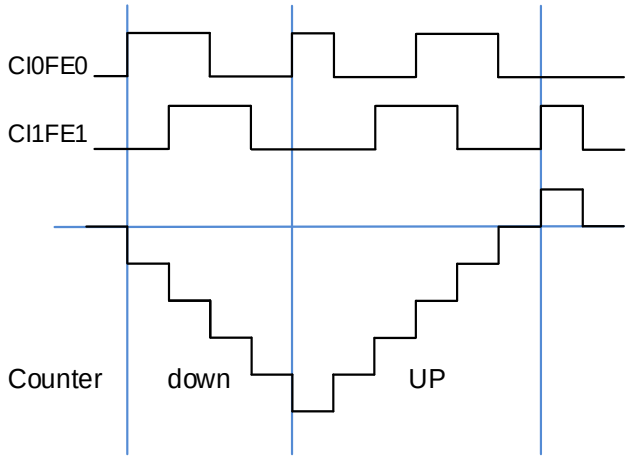


Figure 23-37. Example of decoder interface mode with CI0FE0 polarity inverted



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.

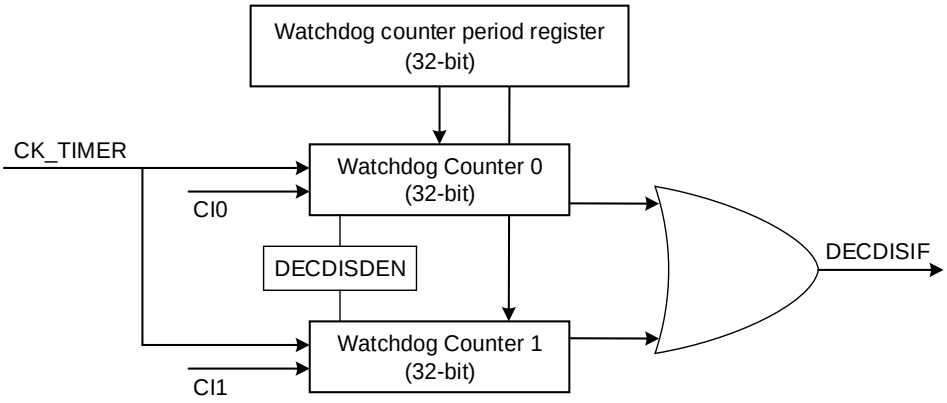
The quadrature decoder signal disconnection detection function can be enabled by setting the DECDISDEN bit (in TIMERx_CTL2 register) to 1, which can be used to detect the signal conditions of the CI0 and CI1 signals.

As shown in ***Figure 23-38. 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 23-38. 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.

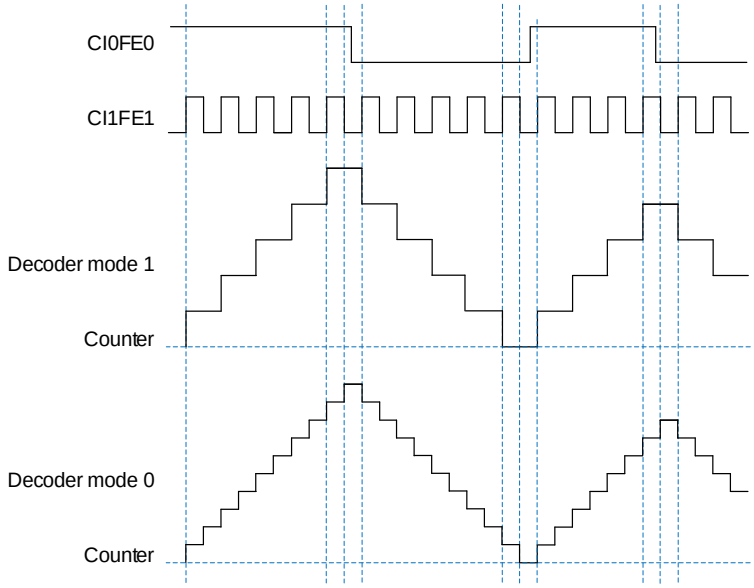
The CH0P is used to configure the count direction selection: When CH0P =0, the counter will count up when the CI0FE0 signal is high and will count down when the CI0FE0 signal is low; When CH0P =1, the counter will count down when the CI0FE0 signal is high and will count up when the CI0FE0 signal is low.

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 23-10. the counter operation in decoder mode 1* and *Figure 23-39. Example of counter operation in decoder mode 0 / 1 with CH1P=0*.

Table 23-10. 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 23-39. 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 23-40. Example of counter operation in decoder mode 2 / 3 (CH0P / CH1P=0)*** and ***Table 23-11. the counter operation in decoder mode 2 / 3***.

Figure 23-40. Example of counter operation in decoder mode 2 / 3 (CH0P / CH1P=0)

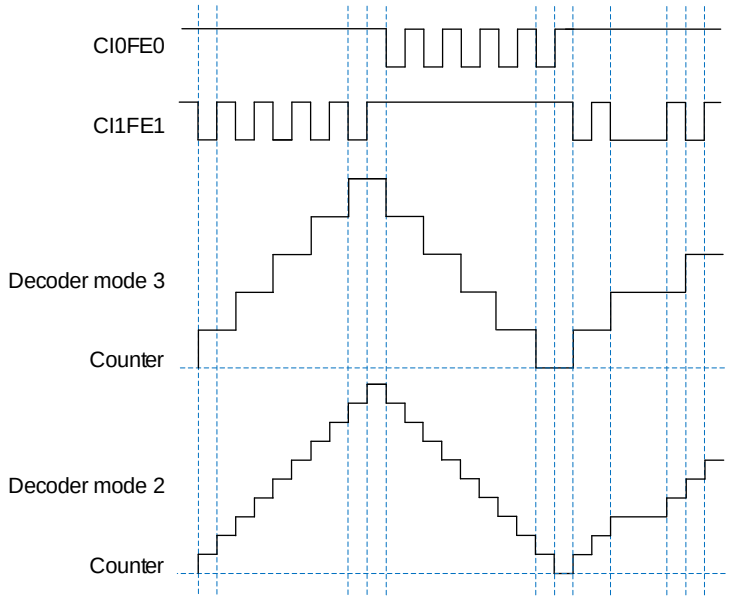


Table 23-11. the counter operation in decoder mode 2 / 3

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	-	-
		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 TRGO0. 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.

Index input function

Index input for quadrature decoder

There are three output signals are commonly used in decoder: A pulse, B pulse and 1 is the location of reference index of the pulse signal. The index pulse signal can reset the counter. When using this function, the index pulse signal must be connected to the TIMER_ETI pin, and can be filtered.

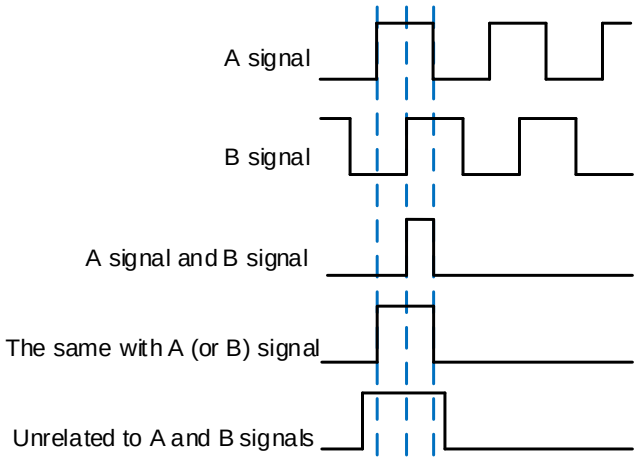
This function can be enabled by setting the INDRSTEN bit in the TIMERx_DECCTL register, and just used in quadrature decoder mode 0~4 and decoder mode 0~3.

The index input function of the TIMER supports the index pulses of the following three orthogonal decoders (no additional configuration required):

1) A signal and B signal: the index signal is a quarter width of A or B signal, and aligned with the edges of A and B signals;
2) The same with A (or B) signal: the index signal is a half width of A or B signal, and aligned with the both edges of A (or B) signal;
3) Unrelated to A and B signals: the index signal exceeds one A / B signal pulse period and is not aligned with any edge of A and B signals.

According to different index signal, the tolerance of the circuit to index signal jitter is also different. In case 3), index signal needs to be less than 2 pulse cycles, otherwise, the counter will be reset multiple times.

Figure 23-41. Three types of index signals



The relationship between the AB signals and the index signal is selected by configuring the INDP[1:0] bit-field in the TIMERx_DECCTL register.

According to different counter count mode, the phenomenon of index input detection event occurrence is different:

- The counter is reset (DIR bit is 0) when counting up;
- The counter is set to the value of theTIMERx CAR register (DIR bit is 1) when counting down.

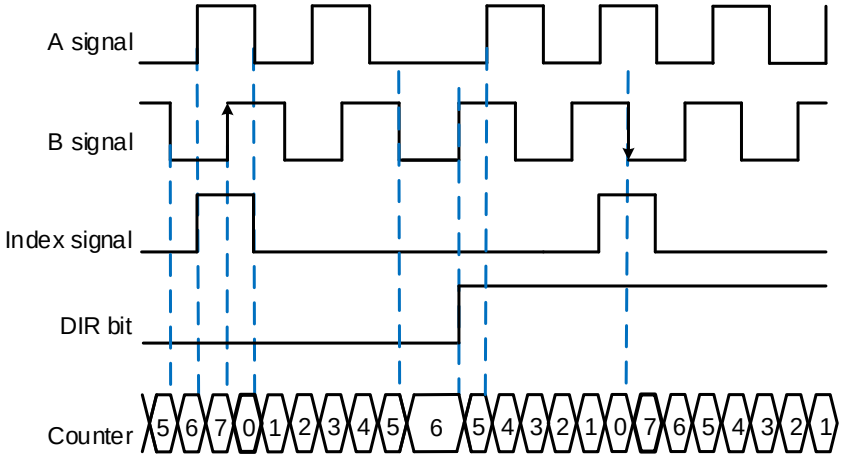
This ensures that the index pulse is always generated at the same mechanical angle regardless of the up counting mode or down counting mode.

When INDP[1:0]=2'b11, the ***Figure 23-42. Counter with the index signal the same as A (INDP[1:0]=2'b11)*** shows the operation of the counter. Depending on the counter direction, the counter resets differently when the index input reset event occurs.

- Counting up: When the AB signal status becomes "11", the counter value is reset to zero;
- Counting down: When the AB signal state leaves "11", the counter value is set to the value of TIMERx CAR register.

The arrows in figure indicate the changes which produced by index events.

Figure 23-42. Counter with the index signal the same as A (INDP[1:0]=2'b11)



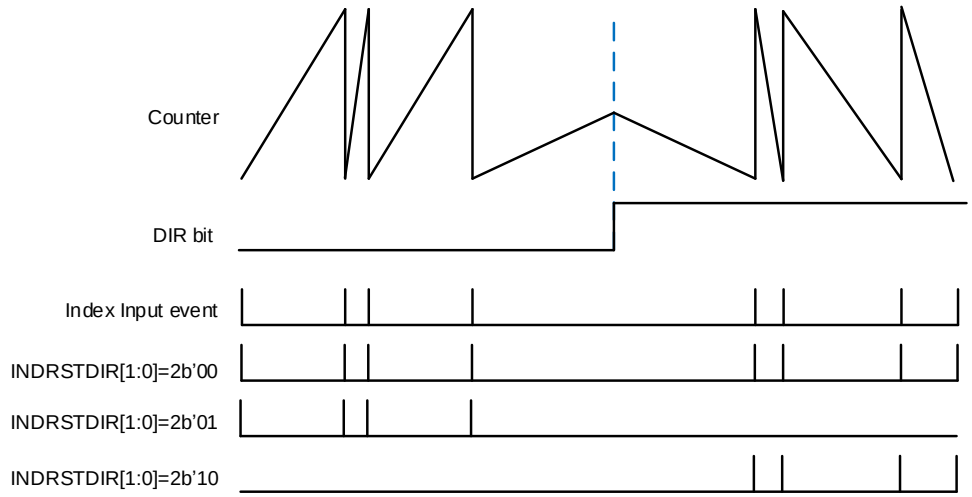
When the index event occurs, the INDIF bit in TIMERx_INTF is set to 1. And the corresponding interrupt is generated if the INDIE bit in TIMERx_DMAINTEN register is set to 1.

Index direction selection

By configuring the INDRSTDIR[1:0] bit-field in the TIMERx_DECCTL register, can select the counting direction in which the index reset event is active. ***Figure 23-43. The relationship between the index signal and counter reset events*** shows the counter reset events with different of INDRSTDIR[1:0] bit-field.

Note: the INDRSTDIR[1:0] bit-field must be 2'b00 in the decoder mode 0 / 1.

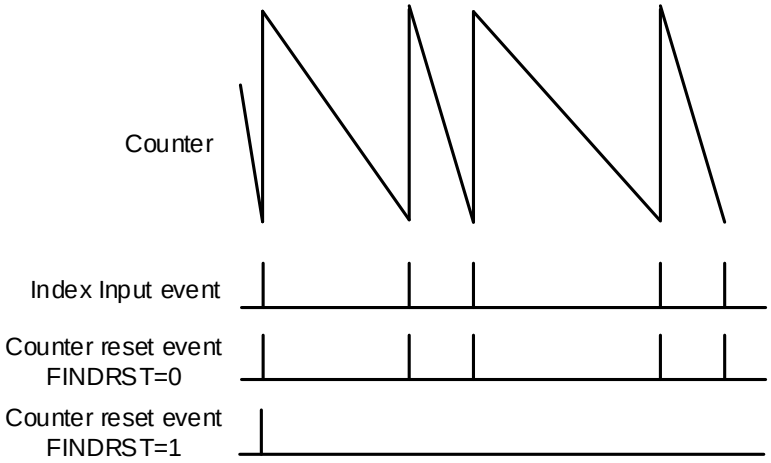
Figure 23-43. The relationship between the index signal and counter reset events



First index signal reset the counter

The first index signal reset counter function can enable by configuring the FINDRST bit to 1 in the TIMERx_DECCTL register. As shown in ***Figure 23-44. The counter reset events with the FINDRST bit***, when FINDRST=1, only the first index pulse resets the counter, and the subsequent pulse signals are invalid.

Figure 23-44. The counter reset events with the FINDRST bit



Index function for decoder modes

By configuring the FINDRST bit to 1 in TIMERx_DECCTL register, can determined the effective time of the index signal reset event when in the decoder mode 0~3.

- INDP[0]=0: The index reset event is active when the count direction signal is low level;
- INDP[0]=1: The index reset event is active when the count direction signal is high level.

Note: INDP[1] bit is not used in the decoder mode 0~3.

When the index event occurs, the INDIF bit in TIMERx_INTF is set to 1. And the

corresponding interrupt is generated if the INDIE bit in TIMERx_DMAINTEN register is set to1.

Index error detection

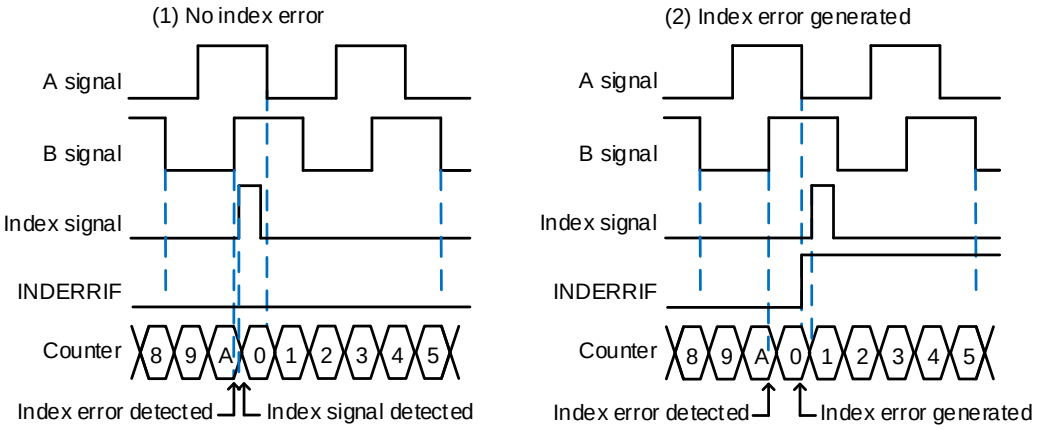
IIf there is no index pulse is detected, an index error is generated, when the counter counts up from the value of TIMERx CAR register to 0 or counts down from 0 to the value of TIMERx CAR register.

When the counter counts up, the index error occurs is delayed until the counter value changes from 0 to 1. The more details is shown in ***Figure 23-45. The Index error detection in up counting mode***. In the waveform on the left, the counter value changes from 0 to 1 without an index signal, so the index error is generated. In the waveform on the right, the counter value changes from 0 to 1 with index signal detected, so there is no index error occurs.

When the counter counts down, the index error will be advanced until the counter value changes from 1 to 0.

When the index error occurs, the INDERRIF bit in TIMERx_INTF is set to 1. And the corresponding interrupt is generated if the INDERRIE bit in TIMERx_DMAINTEN register is set to1.

Figure 23-45. The Index error detection in up counting mode



Quadrature decoder mode and decoder mode modified on-the-fly

The quadrature decoder modes and decoder modes can modify on-the-fly; this function can be enabled by setting the DECMODEN bit to 1 in the TIMERx_SMCFG register.

In quadrature decoder modes, TIMER just can modify from quadrature decoder mode 3 to quadrature decoder mode 0 / 1, or modified from quadrature decoder mode 0/1 to quadrature decoder mode 3 / 4; In decoder modes, TIMER just can modify from decoder mode 0 to decoder mode 1, or modified from decoder mode 2 to decoder mode 3.

The update source for mode modification can be selected by configuring the DECMODS bit in the TIMERx SMCFG register: When DECMODS=0, the decoder mode is updated by the TIMER update event; When DECMODS=1, the decoder mode is updated by the index event.

Hall sensor function

Hall sensor is generally used to control BLDC motor; the advanced timer supports this function.

Figure 23-46. 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 (TRGO0-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 23-46. Hall sensor is used for BLDC motor

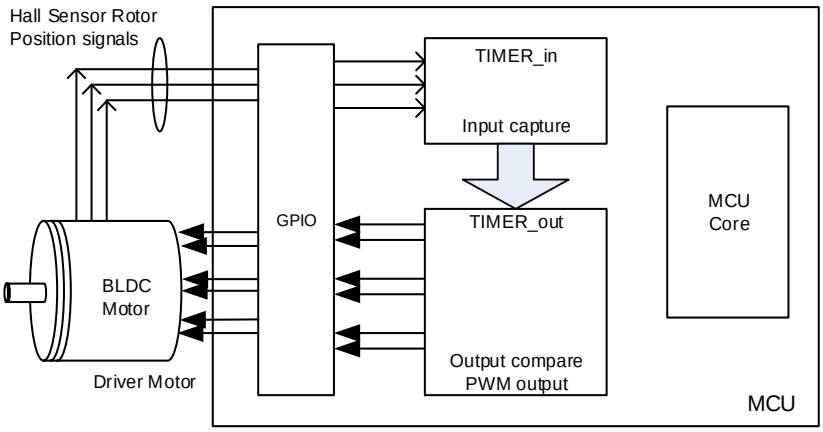
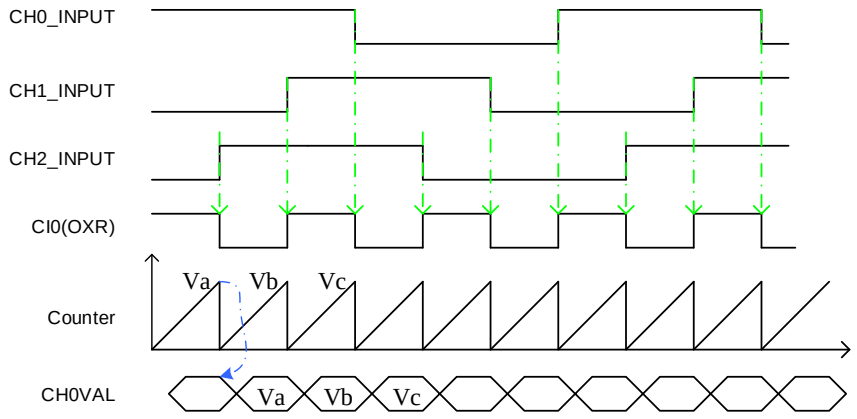
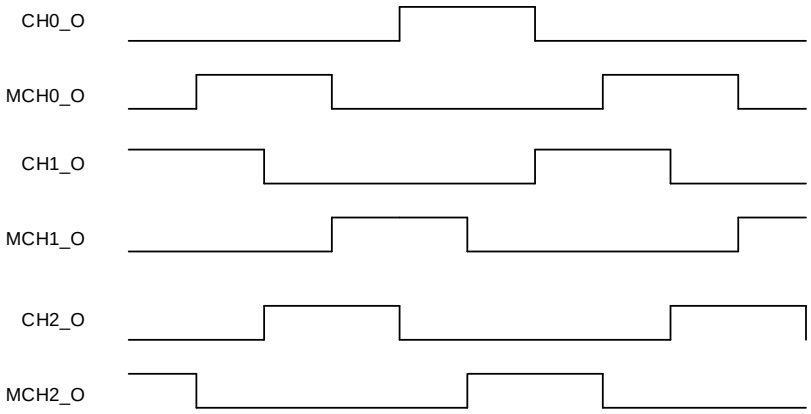


Figure 23-47. Hall sensor timing between two timers

Advanced/General L0 TIMER_in under input capture mode



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

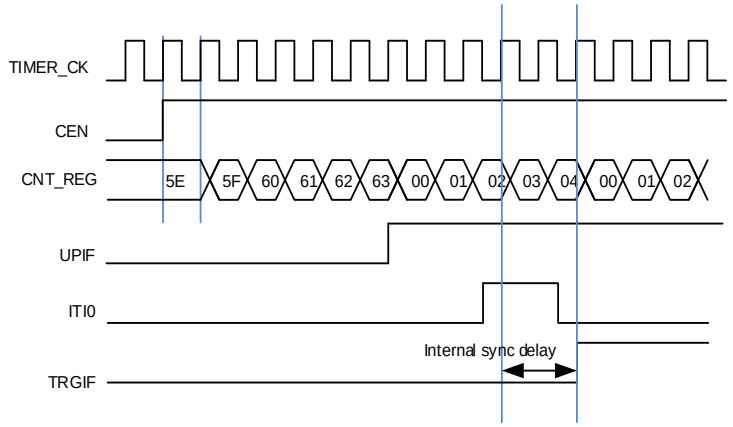


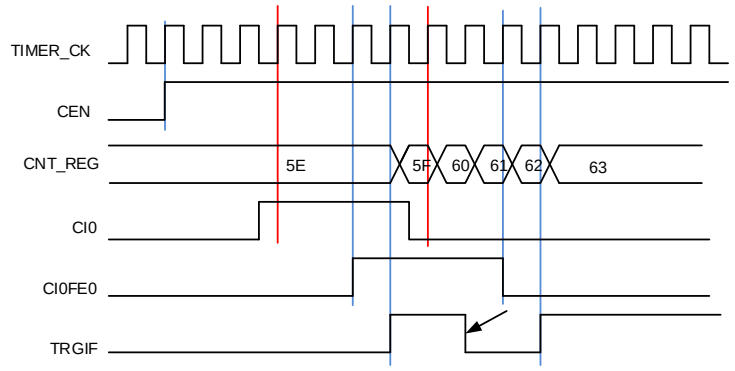
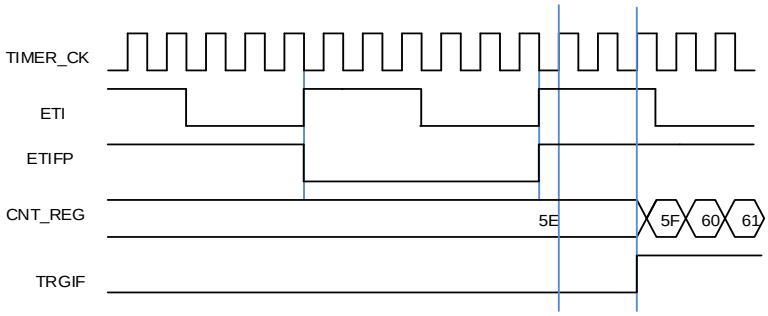
Master-slave management

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..8) in SYSCFG_TIMERxCFG(x=0, 7, 19).

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 y=7: restart + event mode y=8: pause + restart mode	TSCFGy[4:0] 00000:Mode disable 00001: ITI0 00010: ITI1 00011: ITI2 00100: ITI3 00101: CI0F_ED 00110: CI0FE0 00111: CI1FE1 01000: ETIFP(1)	If CIxFEx (x=0...3) or MCIxFEMx (x=0...3) are selected as the trigger source, configure the CHxP, MCHxP and MCHxFP for the polarity selection and inversion. If ETIFP (the filtered	For the ITIx, no filter and prescaler can be used. For the CIx / MCIx, filter can be used by configuring CHxCAPFLT / MCHxCAPFLT, no prescaler can be used. For the ETIFP, filter

	Mode Selection	Source Selection	Polarity Selection	Filter and Prescaler
		01001: ITI4 01010: ITI5 01011: ITI6 01100: ITI7 01101: ITI8 01110: ITI9 01111: ITI10 10000: Reserved 10001: Reserved 10010: Reserved 10011: ITI14	output of external trigger input ETI) is selected as the trigger source, configure the ETP for polarity selection and inversion.	can be used by configuring ETFC and prescaler can be used by configuring ETPSC.
Exam1	**Restart mode** The counter will be cleared and restart when a rising edge of trigger input comes.	TSCFG3[4:0] 5'b00001, ITI0 is selected.	For ITI0, no polarity selector can be used.	For the ITI0, no filter and prescaler can be used.
	Figure 23-48. 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) [MCH0P=0, CH0P=0] CI0FE0 does not invert. The capture event will occur on the rising edge only.	Filter is bypassed 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 23-49. 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 23-50. Event mode
 
</td></tr>
<tr><td>Exam4</td><td colspan="4">Restart + event mode
The counter is reinitialized and started when a rising edge of trigger input comes.</td></tr>
<tr><td>Exam5</td><td colspan="4">Pause + restart mode
The counter will be reset when a rising edge or falling edge (is configured by PRMRPSEL bit in TIMERx_SMCFG register) of trigger input comes. And the counter counts when the trigger input is high, and it will stop when the trigger input is low. In this mode, the start and stop of the counter can be controlled.</td></tr>
</table>

(1) The ETI signal can be input from an external ETI pin or provide by on-chip peripherals, please refer to ***Trigger selection for TIMER0 ETI register (TRIGSEL_TIMER0ETI)***, ***Trigger selection for TIMER7 ETI register (TRIGSEL_TIMER7ETI)*** and ***Trigger selection for TIMER19 ETI register (TRIGSEL_TIMER19ETI)*** for more details.

Single pulse mode

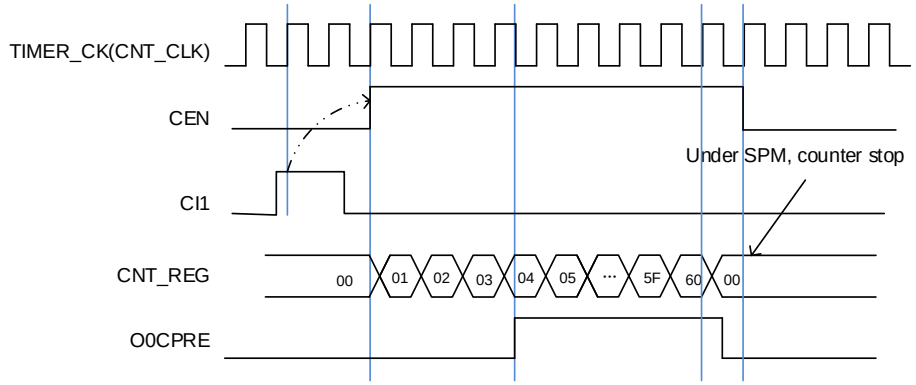
Single pulse mode is opposite to the repetitive mode, which can be enabled by setting SPM in TIMERx_CTL0. If SPM is set, the counter will be cleared and stopped automatically when the next update event occurs. In order to get a pulse waveform, the TIMERx is configured to PWM mode or compare mode by CHxCOMCTL 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 23-51. Single pulse mode TIMERx_CHxCV=0x04, TIMERx_CAR=0x60



Delayable single pulse mode

Delayable single pulse mode is enabled by setting CHxCOMCTL[3:0] / MCHxCOMCTL[3:0] in TIMERx_CHCTLx / TIMERx_MCHCTLx registers. In this mode, the pulse width of OxCPRE / MOxCPRE signal is determined by the TIMERx_CAR register.

Once the timer is set to the delayable single pulse mode, the following configuration is required:

- TIMERx need to work in slave mode and TSCFG7[4:0] != 5'b00000 in SYSCFG_TIMERxCFG(x=0, 7, 19) (restart +event mode);
- The CHxCOMCTL[3:0] / MCHxCOMCTL[3:0] bit-field is setting to 4'b1000 (delayable single pulse mode 0) or 4'b1001 (delayable single pulse mode 1).

In delayable SPM mode 0. The behavior of OxCPRE / MOxCPRE is performed as in PWM mode 0. When counting up, the OxCPRE / MOxCPRE is active. When a trigger event occurs, the OxCPRE / MOxCPRE is inactive. The OxCPRE / MOxCPRE is active again at the next

update event; When counting down, the OxCPRE / MOxCPRE is inactive, when a trigger event occurs, the OxCPRE / MOxCPRE is active. The OxCPRE / MOxCPRE is inactive again at the next update event.

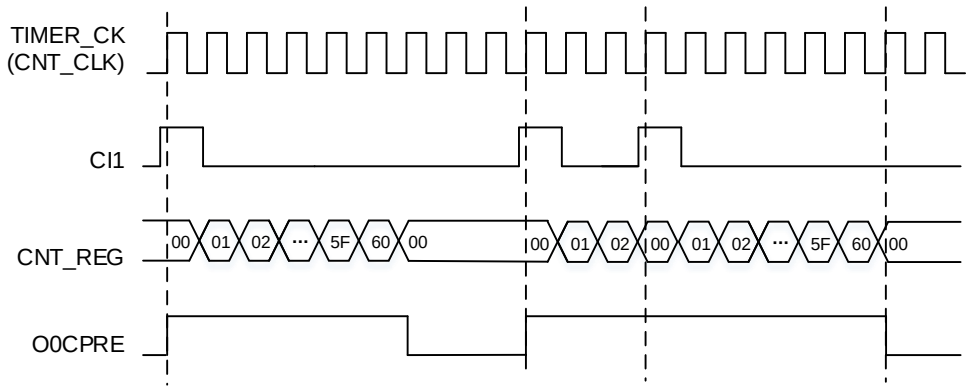
In delayable mode 1. The behavior of OxCPRE / MOxCPRE is performed as in PWM mode 1. When counting up, the OxCPRE / MOxCPRE is inactive, when a trigger event occurs, the OxCPRE / MOxCPRE is active. The OxCPRE / MOxCPRE is inactive again at the next update event; When counting down, the OxCPRE / MOxCPRE is active. When a trigger event occurs, the OxCPRE / MOxCPRE is inactive. The OxCPRE / MOxCPRE is active again at the next update event.

The PWM adjustment mode can also use in delayable SPM modes.

Note:

- The center-aligned counting mode cannot be used in this mode and the CAM[1:0] = 2'b00(in TIMERx_CTL0 register);
- When counter counting up (DIR = 0 in TIMERx_CTL0 register), the value of TIMERx_CHxCV / TIMERx_MCHxCV should be set to 0; When counting down (DIR =1 in TIMERx_CTL0 register), the value of TIMERx_CHxCV / TIMERx_MCHxCV should be greater than or equal to the value of TIMERx_CAR register.

Figure 23-52. delayable single pulse mode with TIMERx_CHxCV=0x00, TIMERx_CAR=0x60

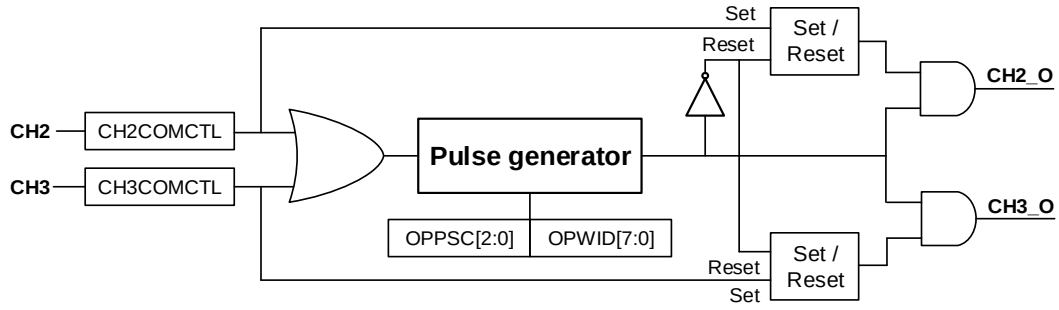


Programmable pulse output

Programmable pulse output mode is enabled by setting CH2COMCTL or CH3COMCTL bit-field in TIMERx_CHCTL1 registers to 4'b1010. When the counter matches the output compare register TIMERx_CH2CV / TIMERx_CH3CV register, CH2_O /CH3_O will output a pulse with the programmable pulse width.

In this mode, the pulse width of the pulse is determined by the OPPSC[2:0] bit-field and OPWID[7:0] bit-field in TIMERx_DECCTL register. The OPPSC[2:0] bit field determines the clock frequency division coefficient of the pulse, and the OPWID[7:0] bit field determines the pulse width.

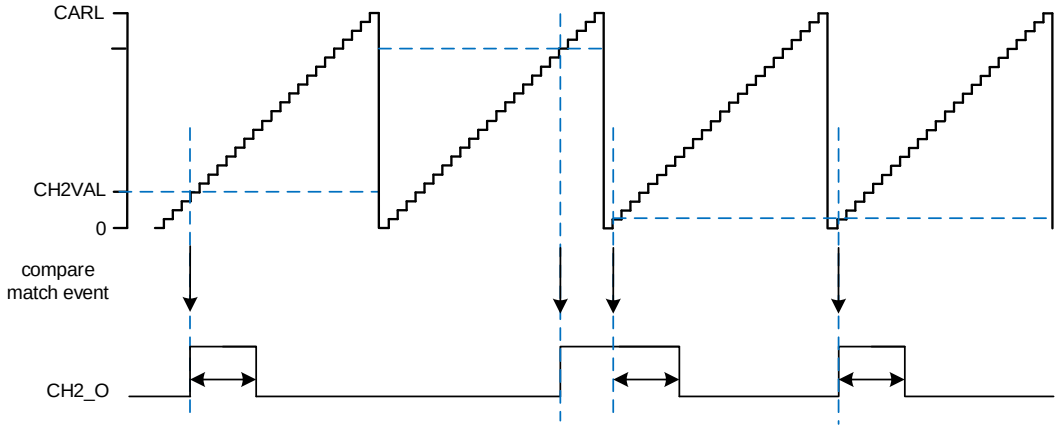
Figure 23-53. Programmable pulse output circuity



This mode can be used in three counter counting modes (up counting, down counting, and center-aligned counting) and all slave modes.

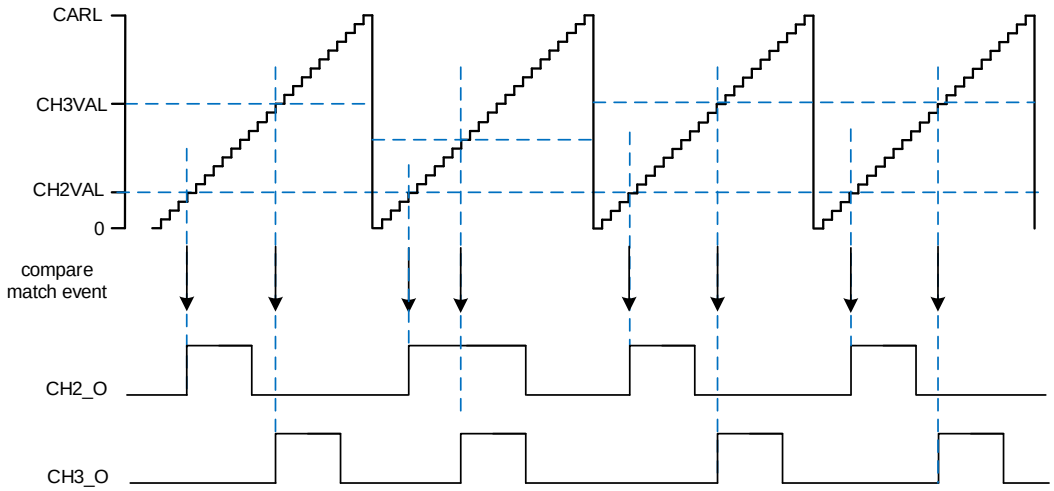
The channel output pulse waveform is retriggerable. When the output pulse is not finished, a new channel match event occurs, the channel output pulse waveform will be extended. The actual output pulse is the high level time before the last match event and the width of one output pulse.

Figure 23-54. Programmable pulse output circuity waveform



When the pulse output function is enabled on CH2_O and CH3_O at the same time, if the compare match event on one channel does not overlap with the pulse waveform on the other channel, the two channels can independently output the pulse waveform. Otherwise, the channel pulse width of the first match event will be extended, and the channel pulse width of the second match event will be the configured value. As shown in ***Figure 23-55. CH2_O and CH3_O output the pulse at the same time***.

Figure 23-55. CH2_O and CH3_O output the pulse at the same time



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 (TRCFG6[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

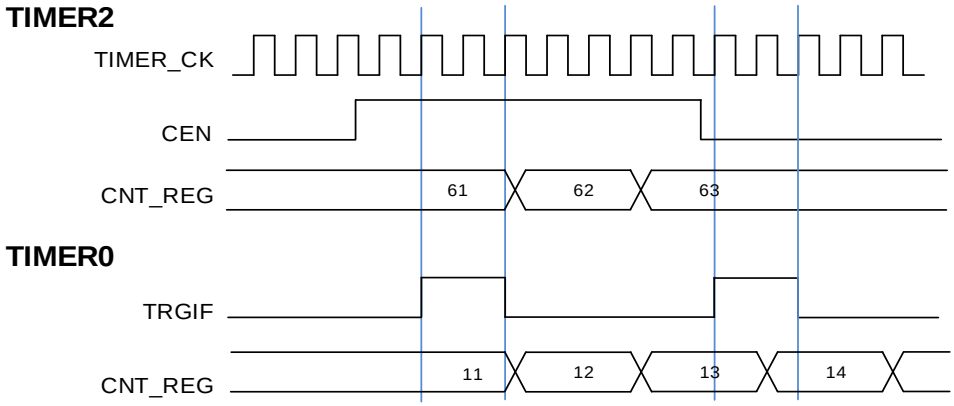
First, enable TIMER0 with the enable signal of TIMER2. Refer to ***Figure 23-56. Trigger mode of TIMER0 controlled by enable signal of TIMER2***. TIMER0 starts counting from its current value with the divided internal clock after being triggered by TIMER2 enable signal output.

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 (TRCFG5[4:0] = 5'b00011 in the_SYSCFG_TIMER0CFG0 register).
3. Start TIMER2 by writing 1 to the CEN bit (TIMER2_CTL0 register).

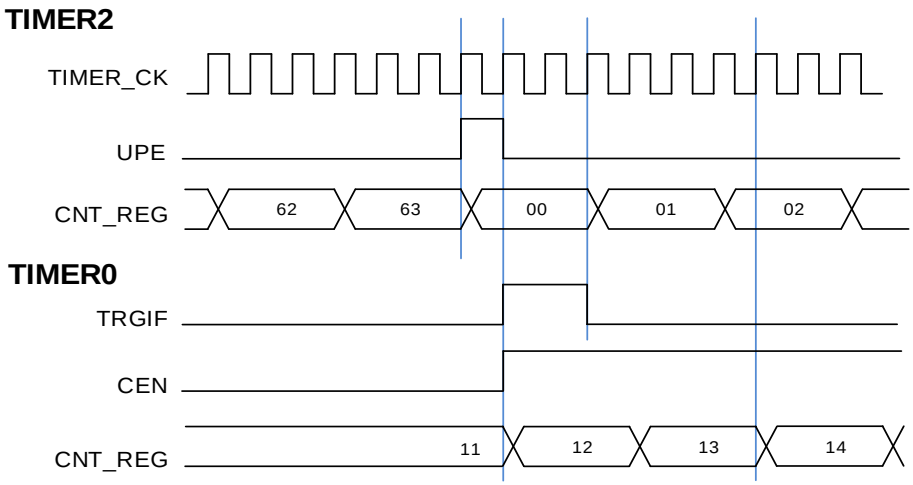
Figure 23-56. Trigger mode of TIMER0 controlled by enable signal of TIMER2



In this example, the update event can also be used as trigger source instead of enable signal. Refer to ***Figure 23-57. 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 (TRCFG5[4:0] = 5'b00011 in the_SYSCFG_TIMER0CFG0 register).
4. Start TIMER2 by writing '1' to the CEN bit (TIMER2_CTL0 register).

Figure 23-57. Trigger mode of TIMER0 controlled by update signal of TIMER2



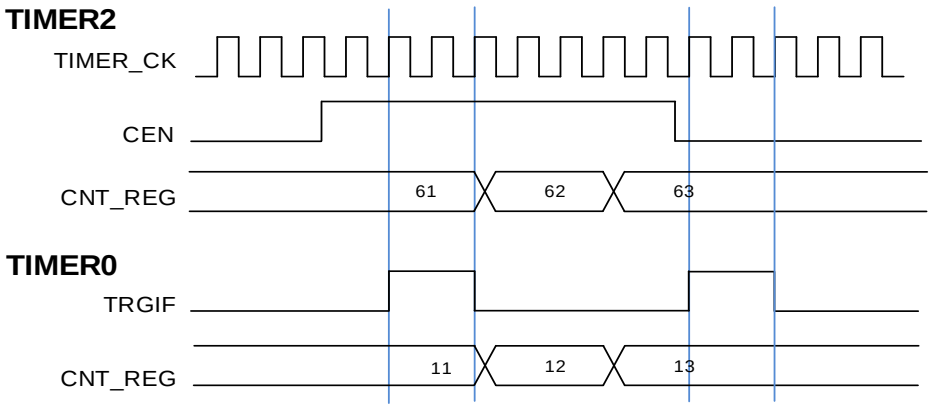
- Enable TIMER0 to count with the enable / O0CPRE signal of TIMER2.

In this example, TIMER0 is enabled with the enable signal of TIMER2. Refer to ***Figure 23-58. 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 (TRCFG4[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).

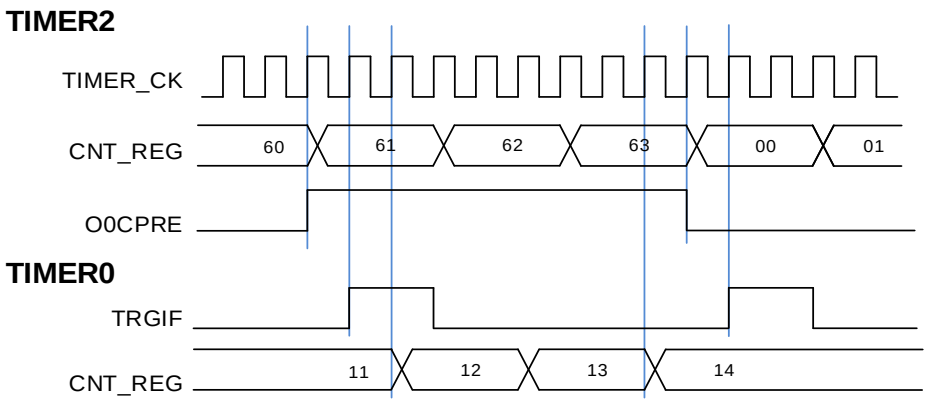
Figure 23-58. Pause mode of TIMER0 controlled by enable signal of TIMER2



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 (TRCFG4[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 23-59. Pause mode of TIMER0 controlled by O0CPRE signal of TIMER2



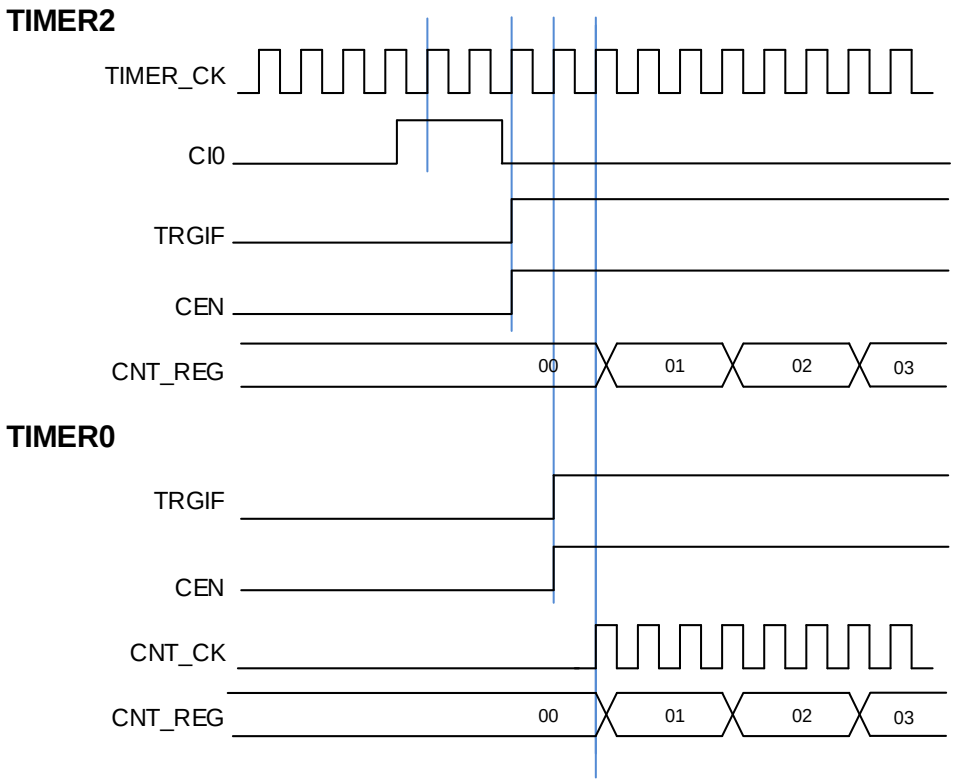
- Using an external trigger to start two timers synchronously.

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 (TRCFG5[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 (TRCFG5[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 23-60. Trigger TIMER0 and TIMER2 by the CI0 signal of TIMER2



Counter synchronization and counter initial direction and value refresh

In some timer daisy chain configurations, multiple timers are triggered and synchronized to start counting at the same time. Due to the use of different clock buses between different timers, phase shifts may occur after prolonged counting. By periodically refreshing the synchronization counter through triggering, the phase shift between multiple timer counters generated on hardware can be eliminated. In addition, the phase shift relationship between multiple synchronous timers can be controlled through the software configurable counter initial value configuration (TIMERx_CINITV).

For instance, three advanced timers are configured in restart + event mode, and enabled the initial value loading function of the counter by setting CINITVEN bit of the TIMERx_CINITCTL register. As is show in ***Figure 23-61. Configurable phase method diagram*** and ***Figure 23-62. Phase shift diagram for three timers***. The three initial values loaded from the counter initial value registers (TIMERx_CINITV) are 0, 20 and 40, respectively, and an equidistant phase shift of (20 x TIMER_CK) was generated in three TIMERs.

The count value 9998 which marked in red in the figure indicates a possible phase shift between timers on different clock buses over a long period of counting. Through the trigger output of the TIMER1 in the master mode, the count values of synchronous slave timers are refreshed to avoid the accumulation of phase shift.

Figure 23-61. Configurable phase method diagram

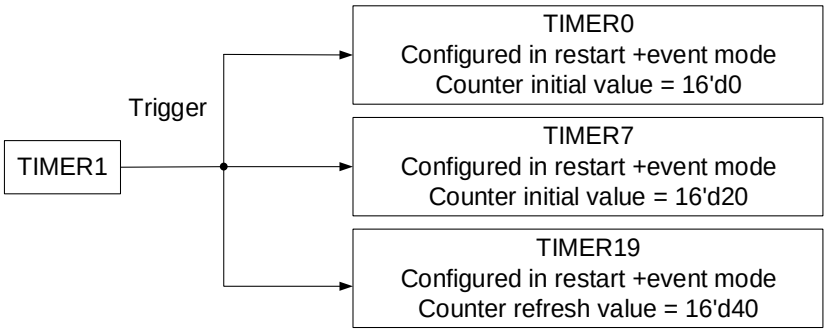
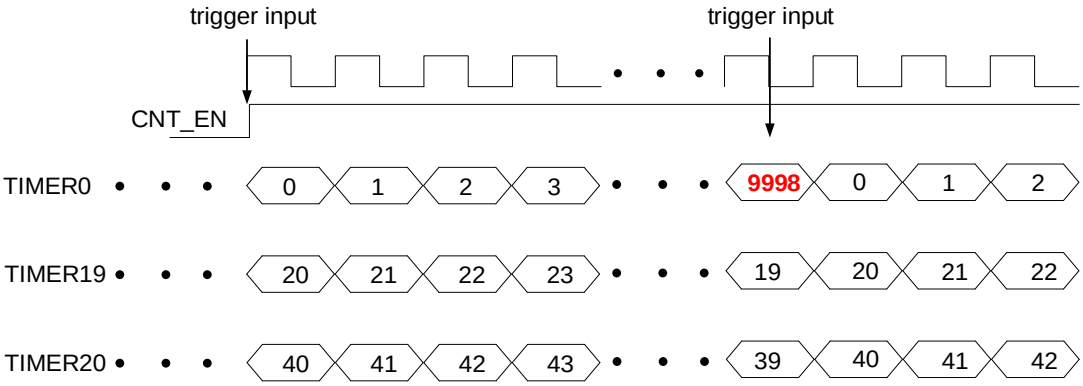


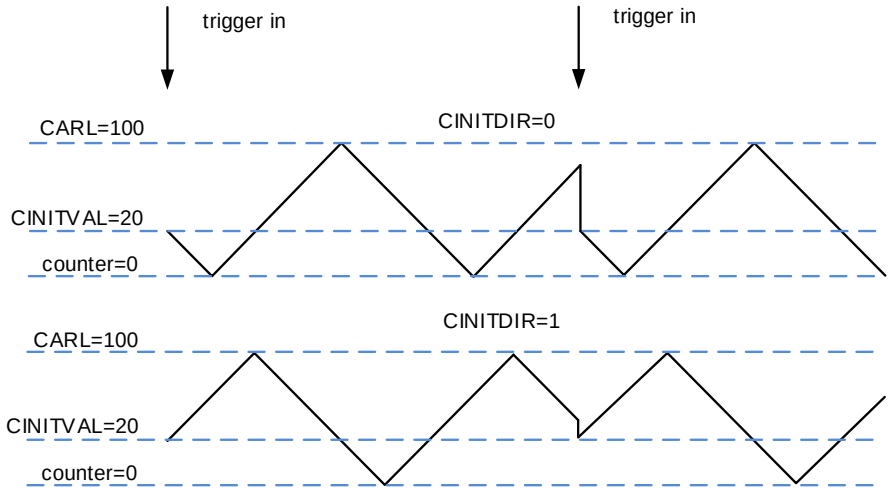
Figure 23-62. Phase shift diagram for three timers



When the counter is in the center-aligned counting mode, the counting direction after resetting the counter can be configured by configuring the CINITDIR bit of the TIMERx_CINITCTL register. CINITDIR bit will only take effect when the CINITVEN bit is set, and the configuration of CINITDIR will not take effect when the counter is in up counting mode and down counting mode. When the counter is in up or down counting mode, the counting direction after resetting the counter can be configured by configuring the DIR bit of the TIMERx_CTL0 register.

When CINITDIR bit value is 0, the counting direction after resetting the counter is down; When CINITDIR bit value is 1, the counting direction after resetting the counter is up. As shown in ***Figure 23-63. Direction of the counter after the reset in center-aligned counting mode***.

Figure 23-63. Direction of the counter after the reset in center-aligned counting mode



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 TRGO0 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.

Output DIR bit

The DIR bit can output on the CH2 and CH3, and this function is enabled by setting CH2COMCTL[3:0] or CH3COMCTL[3:0] bit-field to 4'b1011 in the TIMERx_CHCTL1 register.

When the counter works in center-aligned counting mode, this function can be used to indicates the counter direction of the counter. When the counter works in decoder modes, this function can be used to indicates the rotation direction for the external signal.

Backup UPIF bit

The UPIF bit backup function is enabled by setting UPIFBUEN in the TIMERx_CTL0 register. The UPIF and UPIFBU bits are fully synchronized and without latency.

By using this function, the UPIF bit in the TIMERx_INTF register will be backuped to the UPIFBU bit in the TIMERx_CNT register. This can avoid conflicts when reading the counter and interrupt processing.

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

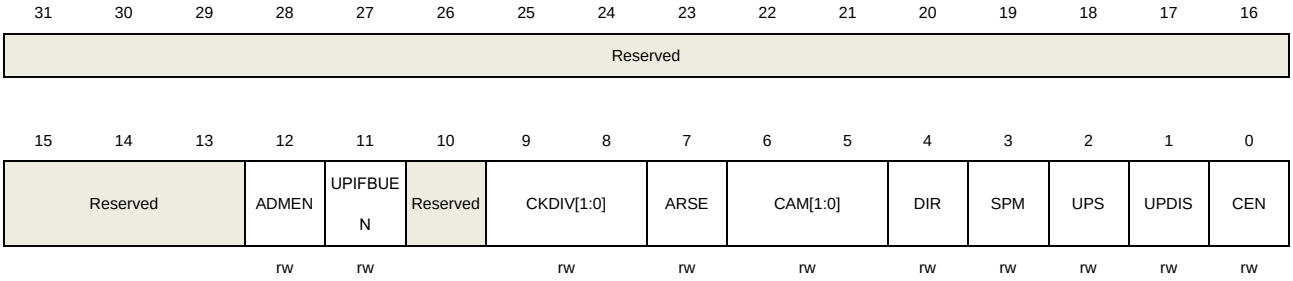
TIMER7 base address: 0x4001 3400

TIMER19 base address: 0x4001 5000

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
31:13	Reserved	Must be kept at reset value.
12	ADMEN	Adjustment mode enable 0: Adjustment mode disabled 1: Adjustment mode enabled Note: This bit can be modified only when the CEN bit is 0.
11	UPIFBUEN	UPIF bit backup enable 0: Backup disable. UPIF bit is not backuped to UPIFBU bit in TIMERx_CNT register. 1: Backup enabled. UPIF bit is backuped to UPIFBU bit in TIMERx_CNT register.
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).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCUC[2:1]		Reserved				MMC0[3]	Reserved		MMC1[2:0]			Reserved			
rw						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.
24:23	Reserved	Must be kept at reset value.
22:20	MMC1[2:0]	Master mode control 1 These bits control the selection of TRGO1 signal. 000: Reset. When the UPG bit in the TIMERx_SWEVG register is set or a reset is generated by the slave mode controller, a TRGO1 pulse occurs. And in the latter case, the signal on TRGO1 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 TRGO1. 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 TRGO1 output, except if the master-slave mode is selected.

		010: Update. In this mode, the master mode controller selects the update event as TRGO1. 011: Capture / compare pulse. In this mode, the master mode controller generates a TRGO1 pulse when a capture or a compare match occurs in channel 0. 100: Compare. In this mode, the master mode controller selects the O0CPRE signal as TRGO1. 101: Compare. In this mode, the master mode controller selects the O1CPRE signal as TRGO1. 110: Compare. In this mode, the master mode controller selects the O2CPRE signal as TRGO1. 111: Compare. In this mode, the master mode controller selects the O3CPRE signal as TRGO1. **Note**: The clock of the slave timer or ADC must be enabled prior to receive events from the master timer, and must not be changed on-the-fly while triggers are received from the master timer.
19:16	Reserved	Must be kept at reset value.
15	ISO3N	Idle state of multi mode channel 3 complementary output. Refer to ISO0N bit.
14	ISO3	Idle state of channel 3 output. Refer to ISO0 bit.
13	ISO2N	Idle state of multi mode channel 2 complementary output. Refer to ISO0N bit
12	ISO2	Idle state of channel 2 output Refer to ISO0 bit
11	ISO1N	Idle state of multi mode channel 1 complementary output Refer to ISO0N bit
10	ISO1	Idle state of channel 1 output Refer to ISO0 bit
9	ISO0N	Idle state of multi mode channel 0 complementary output 0: When POEN bit is reset, MCH0_O is set low. 1: When POEN bit is reset, MCH0_O is set high. This bit can be modified only when PROT[1:0] bits in TIMERx_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 MCH0_O 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 TRGO0 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 TRGO0 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 TRGO0. 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 TRGO0 output, except if the master-slave mode is selected. 0010: Update. In this mode, the master mode controller selects the update event as TRGO0. 011: Capture / compare pulse. In this mode, the master mode controller generates a TRGO0 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 TRGO0. 0101: Compare. In this mode, the master mode controller selects the O1CPRE signal as TRGO0. 0110: Compare. In this mode, the master mode controller selects the O2CPRE signal as TRGO0. 0111: Compare. In this mode, the master mode controller selects the O3CPRE signal as TRGO0. 1000: Decoder clock output. In this mode, the master mode controller selects the decoder clock signal as TRGO0. This value just used in quadrature decoder mode 0~4 and decoder mode 0~3. 1001: In this mode, the master mode controller selects the soft synchronization event signal (generated by setting the SWSYNCG to 1) as TRGO0. 1010~1111: 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. 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, 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					PRMRPSEL	DECMODS	DECMODEN	Reserved							
					rw	rw	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			OCRC	Reserved		
rw	rw	rw		rw				rw				rw			

Bits	Fields	Descriptions
31:27	Reserved	Must be kept at reset value.
26	PRMRPSEL	Pause + restart mode reset polarity selection 0: Counter is reset at falling edge 1: Counter is reset at rising edge
25	DECMODS	Decoder mode update source This bit used to select the decoder mode update source.

0: Decoder mode is updated by the TIMER update event after the modification.
1: Decoder mode is updated by the index event after the modification.

24	DECMODEN	Decoder mode modified on-the-fly enable 0: Decoder mode modified on-the-fly disable. 1: Decoder mode modified on-the-fly enable. When this bit is set to 1, the decoder mode can be modified from one mode to another mode.
23:16	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'b 01000 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
6:4	Reserved	Must be kept at reset value.
3	OCRC	OxCPRE / MOxCPRE clear source selection This bit is used to select the OxCPRE / MOxCPRE clear source. 0: OCPRE_CLR_INT is connected to the OCPRE_CLR input 1: OCPRE_CLR_INT is connected to ETIF
2: 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	MCH3DEN	MCH2DEN	MCH1DEN	MCH0DEN	MCH3IE	MCH2IE	MCH1IE	MCH0IE	INDERRIE	DIRTRANIE	DECDISIE	DECJIE
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
INDIE	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	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	MCH3DEN	Multi mode channel 3 capture / compare DMA request enable 0: Disabled 1: Enabled **Note:** This bit just used for channel input and output independent mode (when MCH3MSEL[1:0] = 2'b00).
26	MCH2DEN	Multi mode channel 2 capture / compare DMA request enable 0: Disabled 1: Enabled **Note:** This bit just used for channel input and output independent mode (when MCH2MSEL[1:0] = 2'b00).
25	MCH1DEN	Multi mode channel 1 capture / compare DMA request enable 0: Disabled 1: Enabled **Note:** This bit just used for channel input and output independent mode (when MMCH1SEL[1:0] = 2'b00).
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	MCH3IE	Multi mode channel 3 capture / compare interrupt enable 0: Disabled 1: Enabled **Note:** This bit just used for channel input and output independent mode (when MCH3MSEL[1:0] = 2'b00).

22	MCH2IE	Multi mode channel 2 capture / compare interrupt enable 0: Disabled 1: Enabled **Note:** This bit just used for channel input and output independent mode (when MCH2MSEL[1:0] = 2'b00).
21	MCH1IE	Multi mode channel 1 capture / compare interrupt enable 0: Disabled 1: Enabled **Note:** This bit just used for channel input and output independent mode (when MMCH1SEL[1:0] = 2'b00).
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	INDERRIE	Index error interrupt enable 0: Index error interrupt disabled 1: Index error interrupt enabled
18	DIRTRANIE	Direction transform interrupt enable 0: Direction transform interrupt disabled 1: Direction transform interrupt enabled
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	INDIE	Index interrupt enable 0: Index interrupt disabled 1: Index interrupt enabled
14	TRGDEN	Trigger DMA request enable 0: Disabled 1: Enabled
13	CMTDEN	Commutation DMA request enable

		0: Disabled 1: Enabled
12	CH3DEN	Channel 3 capture / compare DMA request enable 0: Disabled 1: Enabled
11	CH2DEN	Channel 2 capture / compare DMA request enable 0: Disabled 1: Enabled
10	CH1DEN	Channel 1 capture / compare DMA request enable 0: Disabled 1: Enabled
9	CH0DEN	Channel 0 capture / compare DMA request enable 0: Disabled 1: Enabled
8	UPDEN	Update DMA request enable 0: Disabled 1: Enabled
7	BRKIE	Break interrupt enable 0: Disabled 1: Enabled
6	TRGIE	Trigger interrupt enable 0: Disabled 1: Enabled
5	CMTIE	Commutation interrupt enable 0: Disabled 1: Enabled
4	CH3IE	Channel 3 capture / compare interrupt enable 0: Disabled 1: Enabled
3	CH2IE	Channel 2 capture / compare interrupt enable 0: Disabled 1: Enabled
2	CH1IE	Channel 1 capture / compare interrupt enable 0: Disabled 1: Enabled
1	CH0IE	Channel 0 capture / compare interrupt enable 0: Disabled

		1: Enabled
0	UPIE	Update interrupt enable 0: Disabled 1: Enabled

Interrupt flag register (TIMERx_INTF)

Address offset: 0x10

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH3COMADDIF	CH2COMADDIF	CH1COMADDIF	CH0COMADDIF	MCH3OF	MCH2OF	MCH1OF	MCH0OF	MCH3IF	MCH2IF	MCH1IF	MCH0IF	INDERRIF	DIRTRANIF	DECDISIF	DECJIF
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	rc_w0	rc_w0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INDIF	Reserved	SYSBIF	CH3OF	CH2OF	CH1OF	CH0OF	BRK1IF	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	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	MCH3OF	Multi mode channel 3 over capture flag Refer to MCH0OF description.
26	MCH2OF	Multi mode channel 2 over capture flag Refer to MCH0OF description.
25	MCH1OF	Multi mode channel 1 over capture flag Refer to MCH0OF description.
24	MCH0OF	Multi mode channel 0 over capture flag When multi mode channel 0 is configured in input mode, this flag is set by hardware

		when a capture event occurs while MCH0IF flag has already been set. This flag is cleared by software. 0: No over capture interrupt occurred 1: Over capture interrupt occurred
23	MCH3IF	Multi mode channel 3 capture / compare interrupt flag Refer to MCH0IF description
22	MCH2IF	Multi mode channel 2 capture / compare interrupt flag Refer to MCH0IF description
21	MCH1IF	Multi mode channel 1 capture / compare interrupt flag Refer to MCH0IF description
20	MCH0IF	Multi mode channel 0 capture / compare interrupt flag This flag is set by hardware and cleared by software. If multi mode channel 0 is in input mode, this flag is set when a capture event occurs. If multi mode channel 0 is in output mode, this flag is set when a compare event occurs. If multi mode channel 0 is set to input mode, this bit will be reset by reading TIMERx_MCH0CV. 0: No multi mode channel 0 capture / compare interrupt occurred 1: Multi mode channel 0 capture / compare interrupt occurred
19	INDERRIF	Index error interrupt flag This flag is set by hardware and cleared by software. 0: No Index error interrupt occurred 1: Index error interrupt occurred
18	DIRTRANIF	Direction transform interrupt flag This flag is set by hardware when the DIR bit in TIMERx_CTL0 is changed in decoder modes, and cleared by software. 0: No direction transform interrupt occurred 1: Direction transform interrupt occurred
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	INDIF	Index interrupt flag This flag is set by hardware and cleared by software. 0: No Index interrupt occurred 1: Index interrupt occurred
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	BRK1IF	BREAK1 interrupt flag This flag is set by hardware as soon as the BREAK1 input is active, and cleared by software if the BREAK1 input is not at active level. 0: No active level on BREAK1 inputs has been detected. 1: An active level on BREAK1 inputs has been detected. An interrupt is generated if BRKIE=1 in the TIMERx_DMAINTEN register.
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				MCH3G	MCH2G	MCH1G	MCH0G	Reserved			
w	w	w	w					w	w	w	w				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							BRK1G	BRK0G	TRGG	CMTG	CH3G	CH2G	CH1G	CH0G	UPG
							w	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.
27:24	Reserved	Must be kept at reset value.
23	MCH3G	Multi mode channel 3 capture or compare event generation. Refer to MCH0G description.
22	MCH2G	Multi mode channel 2 capture or compare event generation. Refer to MCH0G description.
21	MCH1G	Multi mode channel 1 capture or compare event generation. Refer to MCH0G description.
20	MCH0G	Multi mode channel 0 capture or compare event generation. This bit is set by software to generate a capture or compare event in multi mode channel 0, it is automatically cleared by hardware. When this bit is set, the MCH0IF flag will be set, and the corresponding interrupt or DMA request will be sent if enabled. In addition, if multi mode channel 0 is configured in input mode, the current value of the counter is captured to TIMERx_MCH0CV register, and the MCH0OF flag is set if the MCH0IF flag has been set. 0: No generate a multi mode channel 0 capture or compare event 1: Generate a multi mode channel 0 capture or compare event
19:9	Reserved	Must be kept at reset value.
8	BRK1G	BREAK1 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 BRK1IF flag will be set. 0: No generate a BREAK1 event 1: Generate a BREAK1 event
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, MCHxEN and CHxCOMCTL bits) are updated based on the value of CCSE (in the TIMERx_CTL1). 0: No affect 1: Generate channel commutation update event
4	CH3G	Channel 3 capture or compare event generation Refer to CH0G description
3	CH2G	Channel 2 capture or compare event generation Refer to CH0G description
2	CH1G	Channel 1 capture or compare event generation Refer to CH0G description
1	CH0G	Channel 0 capture or compare event generation This bit is set by software to generate a capture or compare event in channel 0, it is automatically cleared by hardware. When this bit is set, the CH0IF flag will be set, and the corresponding interrupt or DMA request will be sent if enabled. In addition, if channel 0 is configured in input mode, the current value of the counter is captured to TIMERx_CH0CV register, and the CH0OF flag is set if the CH0IF flag has been set. 0: No generate a channel 0 capture or compare event 1: Generate a channel 0 capture or compare event
0	UPG	Update event generation This bit can be set by software, and automatically cleared by hardware. When this bit is set, the counter is cleared if the center-aligned or up counting mode is selected, while in down counting mode it takes the auto-reload value. The prescaler counter is cleared at the same time. 0: No generate an update event

1: Generate an update event

Channel control register 0 (TIMERx_CHCTL0)

Address offset: 0x18

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH1MS [2]	CH0MS [2]	CH1COMADDSEN	CH0COMADDSEN	CH1ADDUPS	CH0ADDUPS	Reserved	CH1COMCTL[3]	Reserved							CH0COMCTL[3]
		Reserved	Reserved	Reserved	Reserved		Reserved								Reserved
rw	rw	rw	rw	rw	rw		rw								rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1COMCEN	CH1COMCTL[2:0]			CH1COMSEN	CH1COMFEN	CH1MS[1:0]		CH0COMCEN	CH0COMCTL[2:0]			CH0COMSEN	CH0COMFEN	CH0MS[1:0]	
CH1CAPFLT[3:0]				CH1CAPPSC[1:0]				CH0CAPFLT[3:0]				CH0CAPPSC[1:0]			
rw				rw		rw		rw				rw		rw	

Output compare mode:

Bits	Fields	Descriptions
31	CH1MS[2]	Channel 1 I / O mode selection Refer to CH1MS[1:0]description
30	CH0MS[2]	Channel 0 I / O mode selection Refer to CH0MS[1:0] description
29	CH1COMADDSEN	Channel 1 additional compare output shadow enable Refer to CH0COMADDSEN description.
28	CH0COMADDSEN	Channel 0 additional compare output shadow enable When this bit is set, the shadow register of TIMERx_CH0COMV_ADD register which updates at each update event will be enabled. 0: Channel 0 additional compare output shadow disabled 1: Channel 0 additional compare output shadow enabled The PWM mode can be used without validating the shadow register only in single pulse mode (SPM bit in TIMERx_CTL0 register is set). This bit cannot be modified when PROT[1:0] bit-field in TIMERx_CCHP0 register is 11 and CH0MS bit-field is 000.
27	CH1ADDUPS	Channel 1 additional update source 0: TIMERx_CH1COMV_ADD register is updated when an update event occurs. 1: TIMERx_CH1COMV_ADD register is updated when the counter matches the value of CH1VAL.
26	CH0ADDUPS	Channel 0 additional update source 0: TIMERx_CH0COMV_ADD register is updated when an update event occurs.

		1: TIMERx_CH0COMV_ADD register is updated when the counter matches the value of CH0VAL.
25	Reserved	Must be kept at reset value.
24	CH1COMCTL[3]	Channel 1 compare output control Refer to CH0COMCTL[2:0] description
23:17	Reserved	Must be kept at reset value.
16	CH0COMCTL[3]	Channel 0 compare output control Refer to CH0COMCTL[2:0] description
15	CH1COMCEN	Channel 1 output compare clear enable Refer to CH0COMCEN description
14:12	CH1COMCTL[2:0]	Channel 1 compare output control Refer to CH0COMCTL[2:0] description
11	CH1COMSEN	Channel 1 output compare shadow enable Refer to CH0COMSEN description
10	CH1COMFEN	Channel 1 output compare fast enable Refer to CH0COMFEN description
9:8	CH1MS[1:0]	Channel 1 mode selection This bit-field specifies the direction of the channel and the input signal selection. The CH1MS[2:0] bit-field is writable only when the channel is not active (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=0, 7, 19) register). 100: Channel 1 is configured as input, IS1 is connected to MCI1FE1. 101~111: 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.

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: Delayable SPM mode 0. The behavior of O0CPRE is performed as in PWM mode 0. When counting up, the O0CPRE is active. When a trigger event occurs, the O0CPRE is inactive. The O0CPRE is active again at the next update event; When counting down, the O0CPRE is inactive, when a trigger event occurs, the O0CPRE is active. The O0CPRE is inactive again at the next update event.

1001: Delayable SPM mode 1. The behavior of O0CPRE is performed as in PWM mode 1. When counting up, the O0CPRE is inactive, when a trigger event occurs, the O0CPRE is active. The O0CPRE is inactive again at the next update event; When counting down, the O0CPRE is active. When a trigger event occurs, the O0CPRE is inactive. The O0CPRE is active again at the next update event.

1010 / 1011 / 1100 / 1101 / 1110 / 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. The minimum delay from an edge on the trigger input to activate CH0_O output is 5 clock cycles. 1: Channel 0 output quickly compare enable. The minimum delay from an edge on the trigger input to activate CH0_O output is 3 clock cycles.
1:0	CH0MS[1:0]	Channel 0 I / O mode selection This bit-field specifies the work mode of the channel and the input signal selection. The CH0MS[2:0] bit-field is writable only when the channel is not active (When MCH0MSEL[1:0] = 2'b00, the CH1EN bit in TIMERx_CHCTL2 register is reset; when MCH0MSEL[1:0] = 2'b11, the CH0EN and MCH0EN bits in TIMERx_CHCTL2 register are reset). 000: Channel 0 is configured as output. 001: Channel 0 is configured as input, IS0 is connected to CI0FE0. 010: Channel 0 is configured as input, IS0 is connected to CI1FE0. 011: Channel 0 is configured as input, IS0 is connected to ITS, this mode is working only if an internal trigger input is selected (through TSCFG15[4:0] bit-field in SYSCFG_TIMERxCFG2(0, 7, 19) 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 1 (TIMERx_CHCTL1)

Address offset: 0x1C

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH3MS[2]	CH2MS[2]	CH3COMADDSEN	CH2COMADDSEN	CH3ADDUPS	CH2ADDUPS	Reserved	CH3COMCTL[3]								CH2COMCTL[3]
		Reserved	Reserved	Reserved	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 CH0COMFEN 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 (When MCH3MSEL[1:0] = 2'b00, the CH3EN bit in TIMERx_CHCTL2 register is reset; when MCH3MSEL[1:0] = 2'b11, the CH3EN and MCH3EN bits in TIMERx_CHCTL2 register are 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(0, 7, 19) register). 100: Channel 3 is configured as input, IS3 is connected to MCI3FE3. 101~111: 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. **Note**: When multi mode channel 2 is configured in output mode, and the MCH2MSEL[1:0] = 2'b11, This bit-field controls the behavior of O2CPRE which

drives CH2_O and MCH2_O. The active level of O2CPRE is high, while the active level of CH2_O and MCH2_O depends on CH2P and MCH2P bits.

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: Delayable SPM mode 0. The behavior of O2CPRE is performed as in PWM mode 0. When counting up, the O2CPRE is active. When a trigger event occurs, the O2CPRE is inactive. The O2CPRE is active again at the next update event; When counting down, the O2CPRE is inactive, when a trigger event occurs, the O2CPRE is active. The O2CPRE is inactive again at the next update event.

1001: Delayable SPM mode 1. The behavior of O2CPRE is performed as in PWM mode 1. When counting up, the O2CPRE is inactive, when a trigger event occurs, the O2CPRE is active. The O2CPRE is inactive again at the next update event; When counting down, the O2CPRE is active. When a trigger event occurs, the O2CPRE is inactive. The O2CPRE is active again at the next update event.

1010: Programmable pulse output. A programmable pulse output on the CH2_O when the counter matches the output compare register TIMERx_CH2CV. The pulse is configured by OPPSC[2:0] and OPWID[7:0] bit-field in TIMERx_DECCTL register.

1011: Output the DIR bit. The O2CPRE signal indicates the value of the DIR bit (in TIMERx_CTL0 register).

1100 / 1101 / 1110 /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 and MCH2 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 PWM0 or PWM1 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. The minimum delay from an edge on the trigger input to activate CH2_O output is 5 clock cycles. 1: Channel 2 output quickly compare enable. The minimum delay from an edge on the trigger input to activate CH2_O output is 3 clock cycles.
1:0	CH2MS[1:0]	Channel 2 I / O mode selection This bit-field specifies the work mode of the channel and the input signal selection. The CH2MS[2:0] bit-field is writable only when the channel is not active (When MCH2MSEL[1:0] = 2’b00, the CH2EN bit in TIMERx_CHCTL2 register is reset; when MCH2MSEL[1:0] = 2’b11, the CH2EN and MCH2EN bits in TIMERx_CHCTL2 register are 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(0, 7, 19) register). 100: Channel 2 is configured as input, IS2 is connected to MCI2FE2. 101~111: Reserved.

Input capture mode:

Bits	Fields	Descriptions
31	CH3MS[2]	Channel 3 I / O mode selection Same as output compare mode.
30	CH2MS[2]	Channel 2 I / O mode selection Same as output compare mode.
29:16	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
MCH3P	MCH3EN	CH3P	CH3EN	MCH2P	MCH2EN	CH2P	CH2EN	MCH1P	MCH1EN	CH1P	CH1EN	MCH0P	MCH0EN	CH0P	CH0EN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	MCH3P	Multi mode channel 3 capture / compare polarity Refer to MCH0P description.
14	MCH3EN	Multi mode channel 3 capture / compare enable Refer to MCH0EN description.
13	CH3P	Channel 3 capture / compare polarity Refer to CH0P description.
12	CH3EN	Channel 3 capture / compare enable Refer to CH0EN description.
11	MCH2P	Multi mode channel 2 output polarity Refer to MCH0P description.
10	MCH2EN	Multi mode channel 2 output enable Refer to MCH0EN description.
9	CH2P	Channel 2 capture / compare polarity Refer to CH0P description.
8	CH2EN	Channel 2 capture / compare enable Refer to CH0EN description.
7	MCH1P	Multi mode channel 1 output polarity Refer to MCH0P description.
6	MCH1EN	Multi mode channel 1 output enable Refer to MCH0EN description.

5	CH1P	Channel 1 capture / compare polarity Refer to CH0P description.
4	CH1EN	Channel 1 capture / compare enable Refer to CH0EN description.
3	MCH0P	Multi mode channel 0 output polarity When Multi mode channel 0 is configured in output mode, and the MCH0MSEL[1:0] = 2'b11, this bit specifies the MCH0_O output signal polarity. 0: Multi mode channel 0 output active high 1: Multi mode channel 0 output active low When CH0 is configured in input mode, in conjunction with CH0P, this bit is used to define the polarity of CH0. This bit cannot be modified when PROT[1:0] bit-field in TIMERx_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 MCH0_O signal in active state. When multi mode channel 0 is configured in input mode, setting this bit enables the capture event in multi mode channel 0. 0: Multi mode channel 0 disabled 1: Multi mode channel 0 enabled
1	CH0P	Channel 0 capture / compare polarity When channel 0 is configured in output mode, this bit specifies the output signal polarity. 0: Channel 0 active high 1: Channel 0 active low When channel 0 is configured in input mode, these bits specific the channel 0 input signal's polarity. [MCH0P, CH0P] will select the active trigger or capture polarity for channel 0 input signals. 00: channel 0 input signal's rising edge is the active signal for capture or trigger operation in slave mode. And channel 0 input signal will not be inverted. 01: channel 0 input signal's falling edge is the active signal for capture or trigger operation in slave mode. And channel 0 input signal will be inverted. 10: Reserved. 11: Noninverted / both channel 0 input signal's edges. This bit cannot be modified when PROT[1:0] bit-field in TIMERx_CCHP0 register is 11 or 10.
0	CH0EN	Channel 0 capture / compare enable When channel 0 is configured in output mode, setting this bit enables CH0_O signal in active state. When channel 0 is configured in input mode, setting this bit enables the capture event in channel 0. 0: Channel 0 disabled

1: Channel 0 enabled

Counter register (TIMERx_CNT)

Address offset: 0x24
Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31	UPIFBU	UPIF bit backup This bit is a backup of UPIF bit in TIMERx_INTF register and read-only. This bit is only valid when UPIFBUEN = 1. If the UPIFBUEN =0, this bit is reserved and read the result is 0.
30:16	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. When the PWMADMEN =0, this bit-field indicates the current counter value. Writing to this bit-field can change the value of the counter. When the PWMADMEN =1, this bit-field just indicates the integer part of the counter value, and without the fractional part.

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															

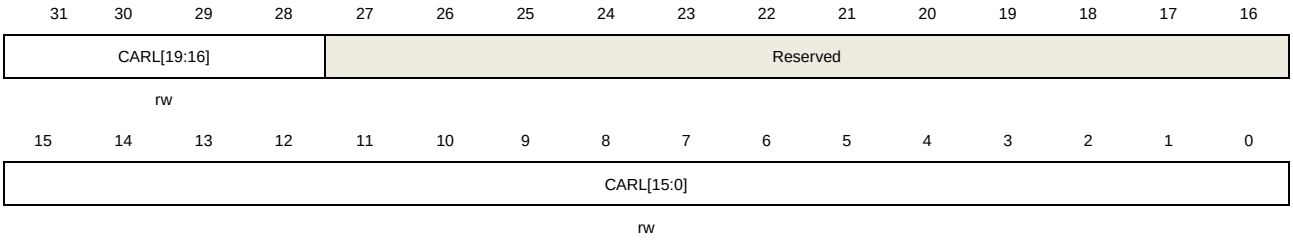
Bits	Fields	Descriptions

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).

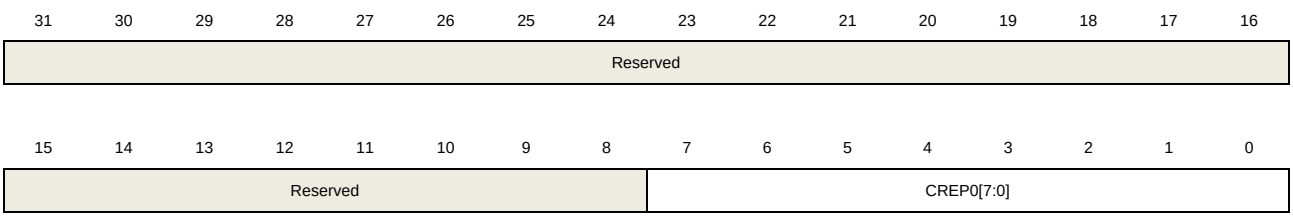


Bits	Fields	Descriptions
31:28	CARL[19:16]	Counter auto reload value (bit 16 to 19) When the PWMADMEN =0, CARL[19:16] bit-field is 0000. When the PWMADMEN =1, CARL[19:16] bit-field specifies fractional part of the auto reload value.
27:16	Reserved	Must be kept at reset value.
15:0	CARL[15:0]	Counter auto reload value (bit 0 to 15) When the PWMADMEN =0, CARL[15:0] bit-field specifies the auto reload value of the counter. When the PWMADMEN =1, CARL[15:0] bit-field specifies integer part of the auto reload value.

Counter repetition register 0 (TIMERx_CREP0)

Address offset: 0x30
Reset value: 0x0000 0000

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



rw

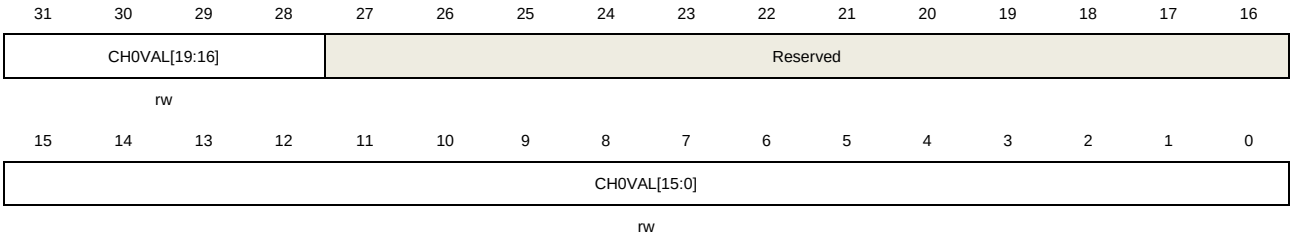
Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	CREP0[7:0]	Counter repetition value 0 This bit-field specifies the update event generation rate. Each time the repetition counter counts down to zero, an update event will be generated. The update rate of the shadow registers is also affected by this bit-field when these shadow registers are enabled. **Note**: This bit-field just used with CREPSEL =0 (in TIMERx_CFG register).

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
31:28	CH0VAL[19:16]	Capture / compare value of channel 0 (bit 16 to 19) When channel 0 is configured in input mode, CH0VAL[19:16] bit-field is 0000. 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. When the PWMADMEN = 0, the CH0VAL[19:16] bit-field is 0000. When the PWMADMEN =1, CH0VAL[19:16] bit-field specifies the fractional part.
27:16	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, CH0VAL[15:0] 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. When the PWMADMEN = 0, the CH0VAL[15:0] bit-field specifies the compare value. When the PWMADMEN =1, CH0VAL[15:0] bit-field specifies integer part of the

compare value.

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[19:16]				Reserved											
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1VAL[15:0]															
rw															

Bits	Fields	Descriptions
31:28	CH1VAL[19:16]	Capture / compare value of channel 1 (bit 16 to 19) When channel 1 is configured in input mode, CH1VAL[19:16] bit-field is 0000. 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. When the PWMADMEN = 0, the CH1VAL[19:16] bit-field is 0000. When the PWMADMEN =1, CH1VAL[19:16] bit-field specifies the fractional part.
27:16	Reserved	Must be kept at reset value.
15:0	CH1VAL[15:0]	Capture / compare value of channel 1 (bit 0 to 15) When channel 1 is configured in input mode, CH1VAL[15:0] 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. When the PWMADMEN = 0, the CH1VAL[15:0] bit-field specifies the compare value. When the PWMADMEN =1, CH1VAL[15:0] bit-field specifies integer part of the compare value.

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[19:16]				Reserved											

rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH2VAL[15:0]															

rw

Bits	Fields	Descriptions
31:28	CH2VAL[19:16]	Capture / compare value of channel 2 (bit 16 to 19) When channel 2 is configured in input mode, CH2VAL[19:16] bit-field is 0000. 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. When the PWMADMEN = 0, the CH2VAL[19:16] bit-field is 0000. When the PWMADMEN =1, CH2VAL[19:16] bit-field specifies the fractional part.
27:16	Reserved	Must be kept at reset value.
15:0	CH2VAL[15:0]	Capture / compare value of channel 2 (bit 0 to 15) When channel 2 is configured in input mode, CH2VAL[15:0] 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. When the PWMADMEN = 0, the CH2VAL[15:0] bit-field specifies the compare value. When the PWMADMEN =1, CH2VAL[15:0] bit-field specifies integer part of the compare value.

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[19:16]				Reserved											

rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3VAL[15:0]															

rw

Bits	Fields	Descriptions
31:28	CH3VAL[19:16]	Capture / compare value of channel 3 (bit 16 to 19) When channel 3 is configured in input mode, CH3VAL[19:16] bit-field is 0000. 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. When the PWMADMEN = 0, the CH3VAL[19:16] bit-field is 0000. When the PWMADMEN =1, CH3VAL[19:16] bit-field specifies the fractional part.
27:16	Reserved	Must be kept at reset value.
15:0	CH3VAL[15:0]	Capture / compare value of channel 3 (bit 0 to 15) When channel 3 is configured in input mode, CH3VAL[15:0] 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. When the PWMADMEN = 0, the CH3VAL[15:0] bit-field specifies the compare value. When the PWMADMEN =1, CH3VAL[15:0] bit-field specifies integer part of the compare value.

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		BRK1LK	BRK0LK	BRK1REL	BRK0REL	BRK1P	BRK1EN	BRK1F[3:0]				BRK0F[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
POEN	OAEN	BRK0P	BRK0EN	ROS	IOS	PROT[1:0]		DTCFG[7:0]							
rw	rw	rw	rw	rw	rw	rw		rw							

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29	BRK1LK	BREAK1 input locked Refer to BRK0LK description
28	BRK0LK	BREAK0 input locked 0: BREAK0 input in input mode 1: BREAK0 input in locked mode When the BRK0LK is set to 1, the BREAK0 input is configured in open drain output mode. Any active BREAK0 event asserts a low logic level on the BREAK0 input to indicate an internal BREAK0 event to external devices. This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.

Note: Every write operation to this bit needs a delay of 1 APB clock to active.

27 BRK1REL

BREAK1 input released

Refer to BRK0REL description.

26 BRK0REL

BREAK0 input released

This bit is cleared by hardware when the BREAK0 input is invalid.

0: BREAK0 input is unreleased

1: BREAK0 input is released

The locked output control (open drain mode in Hi-z state) is released by setting this bit with software. And when the fault is disappeared, this bit will reset by hardware.

Note: Every write operation to this bit needs a delay of 1 APB clock to active.

25 BRK1P

BREAK1 input signal polarity

This bit specifies the polarity of the BREAK1 input signal.

0: BREAK1 input active low

1: BREAK1 input active high

This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.

Note: Every write operation to this bit needs a delay of 1 APB clock to active.

24 BRK1EN

BREAK1 input signal enable

This bit can be set to enable the BREAK1 input signal.

0: BREAK1 input disabled

1: BREAK1 input enabled

This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.

Note:

1) Every write operation to this bit needs a delay of 1 APB clock to active.
2) This bit is only used with ROS=1 and IOS=1.

23:20 BRK1F[3:0]

BREAK1 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 BREAK1 input signal and the length of the digital filter applied to BREAK1.

0000: Filter disabled. BREAK1 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.

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 23-6. Complementary outputs controlled by parameters (MCHxMSEL =2’b11)***.
0: “off-state” disabled. If the CHxEN or CHxNEN bit is reset, the corresponding channel is output disabled.
1: “off-state” enabled. If the CHxEN or CHxNEN bit is reset, the corresponding channel is “off-state”.
This bit cannot be modified when PROT [1:0] bit-filed in TIMERx_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 23-6. Complementary outputs controlled by parameters (MCHxMSEL =2’b11)***.
0: “off-state” disabled. If the CHxEN / CHxNEN bits are both reset, the channels are output disabled.
1: “off-state” enabled. No matter the CHxEN / CHxNEN bits, the channels are “off-state”.
This bit cannot be modified when PROT [1:0] bit-filed in TIMERx_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 / BRK1EN / BRK1P / OAEN / DTCFG bits in TIMERx_CCHP0 register, the DTCFG bits in TIMERx_FCCHPx (x = 0...3) register, are writing protected.
10: PROT mode 1. In addition to the registers in PROT mode 0, the CHxP / MCHxP

bits in TIMERx_CHCTL2 register (if related channel is configured in output mode), the ROS / IOS bits in TIMERx_CCHP0 register and the ROS / IOS bits in TIMERx_FCCHPx (x = 0...3) register are writing protected.

11: PROT mode 2. In addition to the registers in PROT mode 1, the CHxCOMCTL / CHxCOMSEN / CHxCOMADDSEN / MCHxCOMCTL / MCHxCOMSE bits in TIMERx_CHCTL0 / 1 and TIMERx_MCHCTL0 / 1 registers (if the related channel is configured in output) are writing protected.

This bit-field can be written only once after the system reset. Once the TIMERx_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).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MCH1MS[2]	MCH0MS[2]	Reserved					MCH1COMCTL[3]	Reserved							MCH0COMCTL[3]	Reserved
rw	rw						rw								rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCH1COMCEN	MCH1COMCTL[2:0]			MCH1COMSEN	MCH1COMFEN	MCH1MS[1:0]		MCH0COMCEN	MCH0COMCTL[2:0]			MCH0COMSEN	MCH0COMFEN	MCH0MS[1:0]	
MCH1CAPFLT[3:0]				MCH1CAPPSC[1:0]				MCH0CAPFLT[3:0]				MCH0CAPPSC[1:0]			
rw				rw		rw		rw				rw		rw	

Output compare mode:

Bits	Fields	Descriptions
31	MCH1MS[2]	Multi mode channel 1 I / O mode selection Refer to MCH1MS[1:0]description.
30	MCH0MS[2]	Multi mode channel 0 I / O mode selection Refer to MCH0MS[1:0] description.

29:25	Reserved	Must be kept at reset value.
24	MCH1COMCTL[3]	Multi mode channel 1 compare output control. Refer to MCH0COMCTL[2:0] description.
23:17	Reserved	Must be kept at reset value.
16	MCH0COMCTL[3]	Multi mode channel 0 compare output control. Refer to MCH0COMCTL[2:0] description.
15	MCH1COMCEN	Multi mode channel 1 output compare clear enable. Refer to MCH0COMCEN description.
14:12	MCH1COMCTL[2:0]	Multi mode channel 1 compare output control. Refer to MCH0COMCTL description.
11	MCH1COMSEN	Multi mode channel 1 output compare shadow enable Refer to MCH0COMSEN description.
10	MCH1COMFEN	Multi mode channel 1 output compare fast enable Refer to MCH0COMSEN description
9:8	MCH1MS[1:0]	Multi mode channel 1 I / O mode selection This bit-field specifies the direction of the channel and the input signal selection. This bit-field is writable only when the channel is not active (the MCH1EN bit in TIMERx_CHCTL2 register is reset) 000: Multi mode channel 1 is configured as output. 001: Multi mode channel 1 is configured as input, MIS1 is connected to MCI1FEM1. 010: Multi mode channel 1 is configured as input, MIS1 is connected to MCI0FEM1. 011: Multi mode channel 1 is configured as input, MIS1 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, 19) register). 100: Multi mode channel 1 is configured as input, MIS1 is connected to CI1FEM1. 101~111: Reserved.
7	MCH0COMCEN	Multi mode channel 0 output compare clear enable. When this bit is set, the MO0CPRE signal is cleared when high level is detected on ETIFP input. 0: Multi mode channel 0 output compare clear disabled. 1: Multi mode channel 0 output compare clear enabled.
6:4	MCH0COMCTL[2:0]	Multi mode channel 0 output compare control When multi mode channel 0 is configured in output mode, and the MCH0MSEL[1:0] = 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: Delayable SPM mode 0. The behavior of MO0CPRE is performed as in PWM mode 0. When counting up, the MO0CPRE is active. When a trigger event occurs, the MO0CPRE is inactive. The MO0CPRE is active again at the next update event; When counting down, the MO0CPRE is inactive, when a trigger event occurs, the O0CPR MO0CPRE E is active. The MO0CPRE is inactive again at the next update event.

1001: Delayable SPM mode 1. The behavior of MO0CPRE is performed as in PWM mode 1. When counting up, the MO0CPRE is inactive, when a trigger event occurs, the MO0CPRE is active. The MO0CPRE is inactive again at the next update event; When counting down, the MO0CPRE is active. When a trigger event occurs, the MO0CPRE is inactive. The MO0CPRE is active again at the next update event.

1010~1111: 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 CH0NMS bit-field is 00(compare mode).

3	MCH0COMSEN	Multi mode channel 0 output compare shadow enable When this bit is set, the shadow register of TIMERx_MCH0CV register which

		updates at each update event will be enabled. 0: Multi mode channel 0 output compare shadow disabled 1: Multi mode channel 0 output compare shadow enabled The PWM mode can be used without 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 PWM0 or PWM1 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. The minimum delay from an edge on the trigger input to activate MCH0_O output is 5 clock cycles. 1: Multi mode channel 0 output quickly compare enable. The minimum delay from an edge on the trigger input to activate MCH0_O output is 3 clock cycles.
1:0	MCH0MS[1:0]	Multi mode channel 0 I / O mode selection This bit-field specifies the work mode of the channel and the input signal selection. This bit-field is writable only when the channel is not active (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: Multi mode channel 0 is configured as input, MIS0 is connected to MCI1FEM0. 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=0, 7, 19) 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	MCH1MS[2]	Multi mode channel 1 I / O mode selection Refer to MCH1MS[1:0]description.
30	MCH0MS[2]	Multi mode channel 0 I / O mode selection Refer to MCH0MS[1:0] description.
29:16	Reserved	Must be kept at reset value.
15:12	MCH1CAPFLT[3:0]	Multi mode channel 1 input capture filter control. Refer to MCH0CAPFLT description.
11:10	MCH1CAPPSC[1:0]	Multi mode channel 1 input capture prescaler.

Refer to MCH0CAPPSC description.

9:8	MCH1MS[1:0]	Multi mode channel 1 I / O mode selection. Same as output compare mode.
7:4	MCH0CAPFLT[3:0]	Multi mode channel 0 input capture filter control. 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

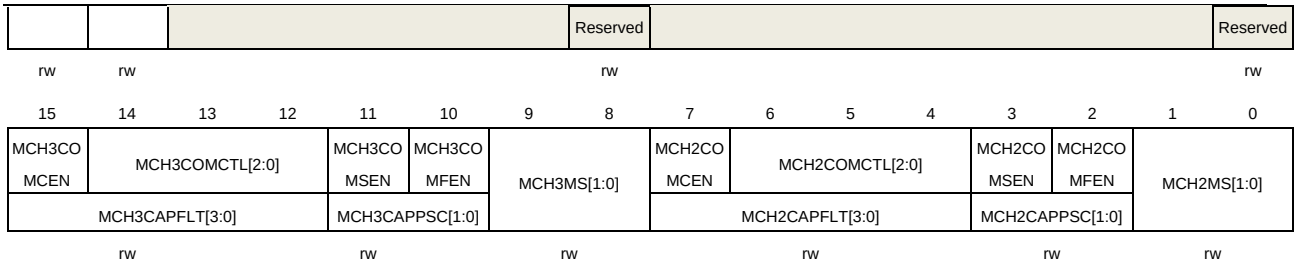
Multi mode channel control register 1 (TIMERx_MCHCTL1)

Address offset: 0x4C

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MCH3MS[2]	MCH2MS[2]	Reserved					MCH3COMCTL[3]	Reserved							MCH3COMCTL[3]



Output compare mode:

Bits	Fields	Descriptions
31	MCH3MS[2]	Multi mode channel 1 I / O mode selection Refer to MCH3MS[1:0]description.
30	MCH2MS[2]	Multi mode channel 0 I / O mode selection Refer to MCH2MS[1:0] description.
29:25	Reserved	Must be kept at reset value.
24	MCH3COMCTL[3]	Multi mode channel 3 compare output control. Refer to MCH2COMCTL[2:0] description.
23:17	Reserved	Must be kept at reset value.
16	MCH2COMCTL[3]	Multi mode channel 2 compare output control. Refer to MCH2COMCTL[2:0] description.
15	MCH3COMCEN	Multi mode channel 3 output compare clear enable Refer to MCH2COMCEN description
14:12	MCH3COMCTL[2:0]	Multi mode channel 3 compare output control Refer to MCH2COMCTL description
11	MCH3COMSEN	Multi mode channel 3 output compare shadow enable Refer to MCH2COMSEN description
10	MCH3COMFEN	Multi mode channel 3 output compare fast enable Refer to MCH2COMSEN description
9:8	MCH3MS[1:0]	Multi mode channel 3 I / O mode selection This bit-field specifies the direction of the channel and the input signal selection. This bit-field is writable only when the channel is not active (MCH3EN bit in TIMERx_CHCTL2 register is reset). 000: Multi mode channel 3 is configured as output. 01: Multi mode channel 3 is configured as input, MIS3 is connected to MCI3FEM3. 010: Multi mode channel 3 is configured as input, MIS3 is connected to MCI2FEM3. 011: Multi mode channel 3 is configured as input, MIS3 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, 19) register). 100: Multi mode channel 3 is configured as input, MIS3 is connected to CI3FEM3.

		101~111: Reserved.
7	MCH2COMCEN	Multi mode channel 2 output compare clear enable. When this bit is set, the MO2CPRE signal is cleared when high level is detected on ETIFP input. 0: Multi mode channel 2 output compare clear disabled 1: Multi mode channel 2 output compare clear enabled
6:4	MCH2COMCTL[2:0]	Multi mode channel 2 compare output control When multi mode channel 2 is configured in output mode, and the MCH2MSEL[1:0] = 2'b00, MCH2COMCTL[3] and MCH2COMCTL[2:0] bit-field control the behavior of MO2CPRE which drives MCH2_O. The active level of MO2CPRE is high, while the active level of MCH2_O depends on MCH2FP[1:0] bits. **Note**: When multi mode channel 2 is configured in output mode, and the MCH2MSEL[1:0] = 2'b11, the CH2COMCTL[2:0] bit-field controls the behavior of O2CPRE which drives CH2_O and MCH2_O, while the active level of CH2_O and MCH2_O depends on CH2P and MCH2P bits. 0000: Timing mode. The MO2CPRE signal keeps stable, independent of the comparison between the output compare register TIMERx_MCH2CV and the counter TIMERx_CNT. 0001: Set the channel output on match. MO2CPRE signal is forced high when the counter matches the output compare register TIMERx_MCH2CV. 0010: Clear the channel output on match. MO2CPRE signal is forced low when the counter matches the output compare register TIMERx_MCH2CV. 0011: Toggle on match. MO2CPRE toggles when the counter matches the output compare register TIMERx_MCH2CV. 0100: Force low. MO2CPRE is forced low level. 0101: Force high. MO2CPRE is forced high level. 0110: PWM mode 0. When counting up, MO2CPRE is active as long as the counter is smaller than TIMERx_MCH2CV, otherwise it is inactive. When counting down, MO2CPRE is inactive as long as the counter is larger than TIMERx_MCH2CV, otherwise it is active. 0111: PWM mode 1. When counting up, MO2CPRE is inactive as long as the counter is smaller than TIMERx_MCH2CV, otherwise it is active. When counting down, MO2CPRE is active as long as the counter is larger than TIMERx_MCH2CV, otherwise it is inactive. 1000: Delayable SPM mode 0. The behavior of MO2CPRE is performed as in PWM mode 0. When counting up, the MO2CPRE is active. When a trigger event occurs, the MO2CPRE is inactive. The MO2CPRE is active again at the next update event; When counting down, the MO2CPRE is inactive, when a trigger event occurs, the O0CPR MO2CPRE E is active. The MO2CPRE is inactive again at the next update event. 1001: Delayable SPM mode 1. The behavior of MO2CPRE is performed as in PWM mode 1. When counting up, the MO2CPRE is inactive, when a trigger event occurs,

		the MO2CPRE is active. The MO2CPRE is inactive again at the next update event; When counting down, the MO2CPRE is active. When a trigger event occurs, the MO2CPRE is inactive. The MO2CPRE is active again at the next update event. 1010~1111: Reserved. If configured in PWM mode, the MO2CPRE 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 and MCH2 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 CH2NMS bit-field is 00(compare mode).
3	MCH2COMSEN	Multi mode channel 2 output compare shadow enable When this bit is set, the shadow register of TIMERx_MCH2CV register, which updates at each update event will be enabled. 0: Multi mode channel 2 output compare shadow disabled 1: Multi mode channel 2 output compare shadow enabled The PWM mode can be used without 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 CH2NMS bit-field is 00.
2	MCH2COMFEN	Multi mode 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 PWM0 or PWM1 mode. The output channel will treat an active edge on the trigger input as a compare match, and MCH2_O is set to the compare level independently from the result of the comparison. 0: Multi mode channel 2 output quickly compare disable. The minimum delay from an edge on the trigger input to activate MCH2_O output is 5 clock cycles. 1: Multi mode channel 2 output quickly compare enable. The minimum delay from an edge on the trigger input to activate MCH2_O output is 3 clock cycles.
1:0	MCH2MS[1:0]	Multi mode channel 2 I / O mode selection This bit-field specifies the work mode of the channel and the input signal selection. This bit-field is writable only when the channel is not active (MCH2EN bit in TIMERx_CHCTL2 register is reset). 000: Multi mode channel 2 is configured as output. 001: Multi mode channel 2 is configured as input, MIS2 is connected to MCI2FEM2. 010: Multi mode channel 2 is configured as input, MIS2 is connected to MCI3FEM2. 011: Multi mode channel 2 is configured as input, MIS2 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, 19) register). 100: Multi mode channel 2 is configured as input, MIS2 is connected to CI2FEM2.

101~111: Reserved.

Input capture mode:

Bits	Fields	Descriptions
31	MCH3MS[2]	Multi mode channel 1 I / O mode selection Refer to MCH3MS[1:0]description.
30	MCH2MS[2]	Multi mode channel 0 I / O mode selection Refer to MCH2MS[1:0] description.
29:16	Reserved	Must be kept at reset value.
15:12	MCH3CAPFLT[3:0]	Multi mode channel 3 input capture filter control. Refer to MCH2CAPFLT description.
11:10	MCH3CAPPSC[1:0]	Multi mode channel 3 input capture prescaler. Refer to MCH2CAPPSC description.
9:8	MCH3MS[1:0]	Multi mode channel 3 I / O mode selection. Same as output compare mode.
7:4	MCH2CAPFLT[3:0]	Multi mode channel 2 input capture filter control. 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 MCI2 input signal and the length of the digital filter applied to MCI2. 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	MCH2CAPPSC[1:0]	Multi mode channel 2 input capture prescaler. This bit-field specifies the factor of the prescaler on channel 2 input. The prescaler is reset when MCH2EN bit in TIMERx_CHCTL2 register is cleared. 00: Prescaler 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	MCH2MS[1:0]	Multi mode channel 2 I / O mode selection Same as output compare mode.

Multi mode channel control register 2 (TIMERx_MCHCTL2)

Address offset: 0x50

Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								MCH3FP[1:0]		MCH2FP[1:0]		MCH1FP[1:0]		MCH0FP[1:0]	
								rw		rw		rw		rw	

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:6	MCH3FP[1:0]	Multi mode channel 3 capture / compare free polarity Refer to MCH0FP[1:0] description.
5:4	MCH2FP[1:0]	Multi mode channel 2 capture / compare free polarity Refer to MCH0FP[1:0] description.
3:2	MCH1FP[1:0]	Multi mode channel 1 capture / compare free polarity Refer to MCH0FP[1:0] description.
1:0	MCH0FP[1:0]	Multi mode channel 0 capture / compare free polarity When multi mode channel 0 is configured in output mode, and the MCH0MSEL[1:0] = 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.

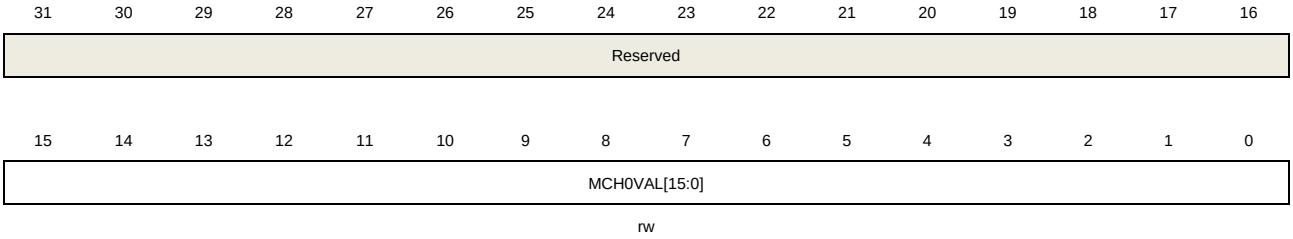
This bit cannot be modified when PROT[1:0] bit-field in TIMERx_CCHP0 register is 11 or 10.

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).



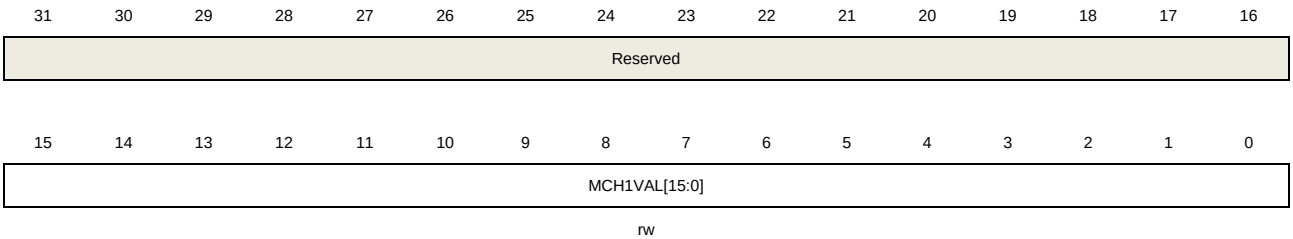
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	MCH0VAL[15:0]	Capture / compare value of multi mode channel 0. When multi mode channel 0 is configured in input mode, this bit-field indicates the counter value at the last capture event. And this bit-field is read-only. When multi mode channel 0 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event.

Multi mode channel 1 capture / compare value register (TIMERx_MCH1CV)

Address offset: 0x58

Reset value: 0x0000 0000

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



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

15:0	MCH1VAL[15:0]	Capture / compare value of multi mode channel 1. When multi mode channel 1 is configured in input mode, this bit-field indicates the counter value at the last capture event. And this bit-field is read-only. When multi mode channel 1 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event.

Multi mode channel 2 capture / compare value register (TIMERx_MCH2CV)

Address offset: 0x5C
Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCH2VAL[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	MCH2VAL[15:0]	Capture / compare value of multi mode channel 2. When multi mode channel 2 is configured in input mode, this bit-field indicates the counter value at the last capture event. And this bit-field is read-only. When multi mode channel 2 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event.

Multi mode channel 3 capture / compare value register (TIMERx_MCH3CV)

Address offset: 0x60
Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCH3VAL[15:0]															
rw															

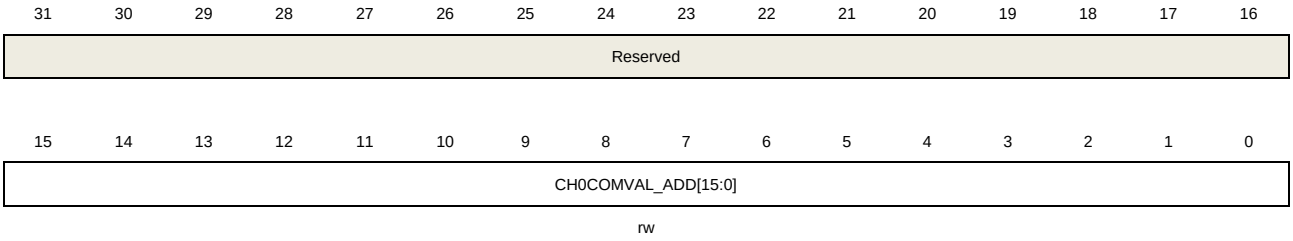
Bits	Fields	Descriptions

31:16	Reserved	Must be kept at reset value.
15:0	MCH3VAL[15:0]	Capture / compare value of channel 3. When multi mode channel 3 is configured in input mode, this bit-field indicates the counter value at the last capture event. And this bit-field is read-only. When multi mode channel 3 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event.

Channel 0 additional compare value register (TIMERx_CH0COMV_ADD)

Address offset: 0x64
Reset value: 0x0000 0000

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

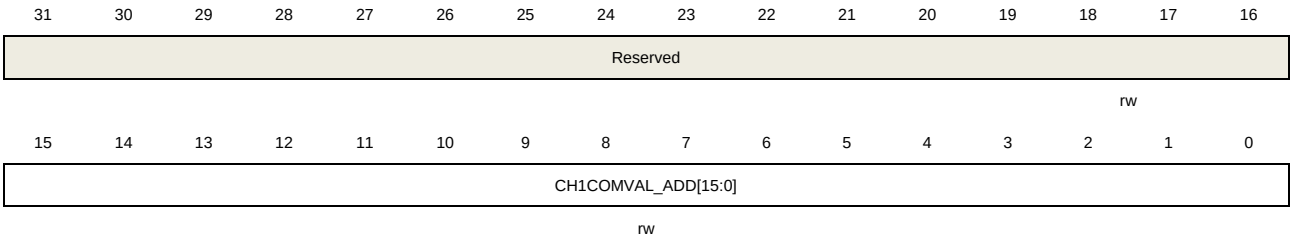


Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH0COMVAL_ADD[15:0]	Additional compare value of channel 0 When channel 0 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event. **Note:** This register just used in composite PWM mode(when CH0CPWMEN=1).

Channel 1 additional compare value register (TIMERx_CH1COMV_ADD)

Address offset: 0x68
Reset value: 0x0000 0000

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



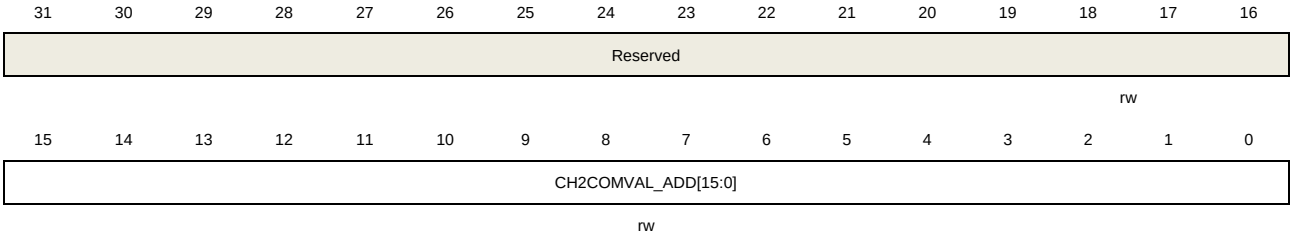
Bits	Fields	Descriptions

31:16	Reserved	Must be kept at reset value.
15:0	CH1COMVAL_ADD [15:0]	Additional compare value of channel 1 When channel 1 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event. **Note:** This register just used in composite PWM mode(when CH0CPWMEN=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).

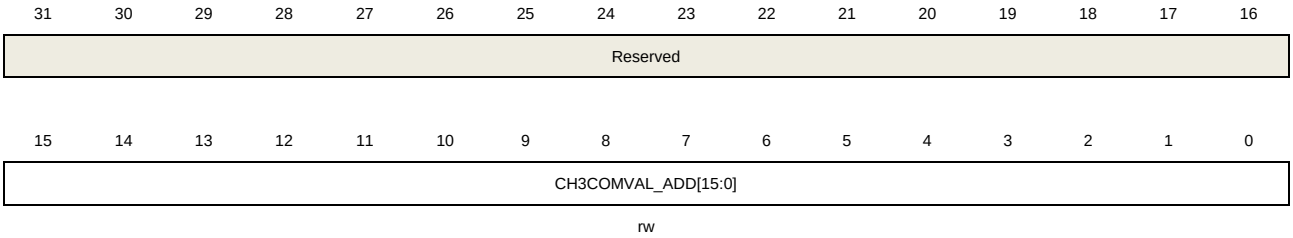


Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH2COMVAL_ADD [15:0]	Additional compare value of channel 2 When channel 2 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event. **Note:** This register just used in composite PWM mode(when CH0CPWMEN=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).



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 CH0CPWMEN=1).

Control register 2 (TIMERx_CTL2)

Address offset: 0x74

Reset value: 0x0FF0 00FF

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH3CPWMEN	CH2CPWMEN	CH1CPWMEN	CH0CPWMEN	MCH3MSEL[1:0]		MCH2MSEL[1:0]		MCH1MSEL[1:0]		MCH0MSEL[1:0]		DECDISDEN	DECJDEN	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
CH3OMPSEL[1:0]		CH2OMPSEL[1:0]		CH1OMPSEL[1:0]		CH0OMPSEL[1:0]		BRKENCH3	BRKENCH2	BRKENCH1	BRKENCH0	DTIENCH3	DTIENCH2	DTIENCH1	DTIENCH0
rw		rw		rw		rw		rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31	CH3CPWMEN	Channel 3 composite PWM mode enable 0: Disabled 1: Enabled
30	CH2CPWMEN	Channel 2 composite PWM mode enable 0: Disabled 1: Enabled
29	CH1CPWMEN	Channel 1 composite PWM mode enable 0: Disabled 1: Enabled
28	CH0CPWMEN	Channel 0 composite PWM mode enable 0: Disabled 1: Enabled
27:26	MCH3MSEL[1:0]	Multi mode channel 3 mode select 00: Independent mode, MCH3 is independent of CH3 01: Reserved 10: Reserved 11: Complementary mode, only the CH3 is valid for input, and the outputs of MCH3 and CH3 are complementary

25:24	MCH2MSEL[1:0]	Multi mode channel 2 mode select 00: Independent mode, MCH2 is independent of CH2 01: Reserved 10: Reserved 11: Complementary mode, only the CH2 is valid for input, and the outputs of MCH2 and CH2 are complementary
23:22	MCH1MSEL[1:0]	Multi mode channel 1 mode select 00: Independent mode, MCH1 is independent of CH1 01: Reserved 10: Reserved 11: Complementary mode, only the CH1 is valid for input, and the outputs of MCH1 and CH1 are complementary
21:20	MCH0MSEL[1:0]	Multi mode channel 0 mode select 00: Independent mode, MCH0 is independent of CH0 01: Reserved 10: Reserved 11: Complementary mode, only the CH0 is valid for input, and the outputs of MCH0 and CH0 are complementary
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
17:16	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 MCH3_O and CH3_O. 0: Disabled 1: Enabled
2	DTIENCH2	Dead time inserted enable for channel 2 Enables the deadtime insertion in the outputs of MCH2_O and CH2_O. 0: Disabled 1: Enabled
1	DTIENCH1	Dead time inserted enable for channel 1 Enables the deadtime insertion in the outputs of MCH1_O and CH1_O. 0: Disabled 1: Enabled
0	DTIENCH0	Dead time inserted enable for channel 0 Enables the deadtime insertion in the outputs of MCH0_O and CH0_O. 0: Disabled 1: Enabled

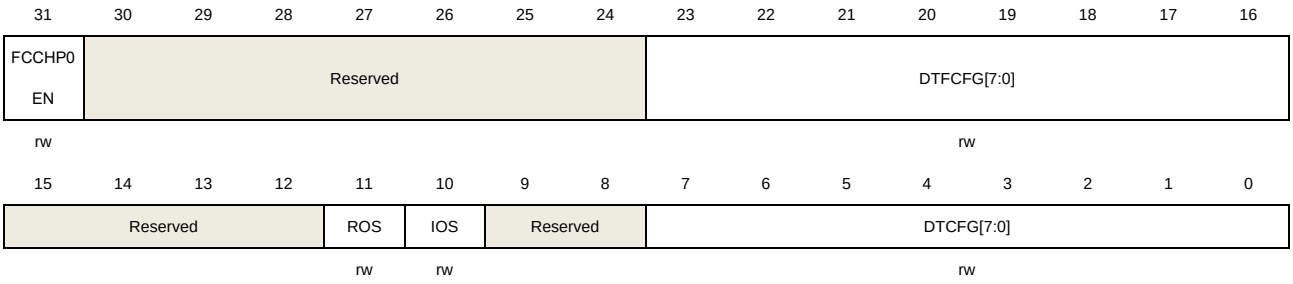
Free complementary channel protection register 0 (TIMERx_FCCHP0)

Address offset: 0x7C

Reset value: 0x0000 0000

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

This register is used to configure the outputs of CH0_O / MCH0_O.



Bits	Fields	Descriptions
31	FCCHP0EN	Free complementary channel protection register 0 enable

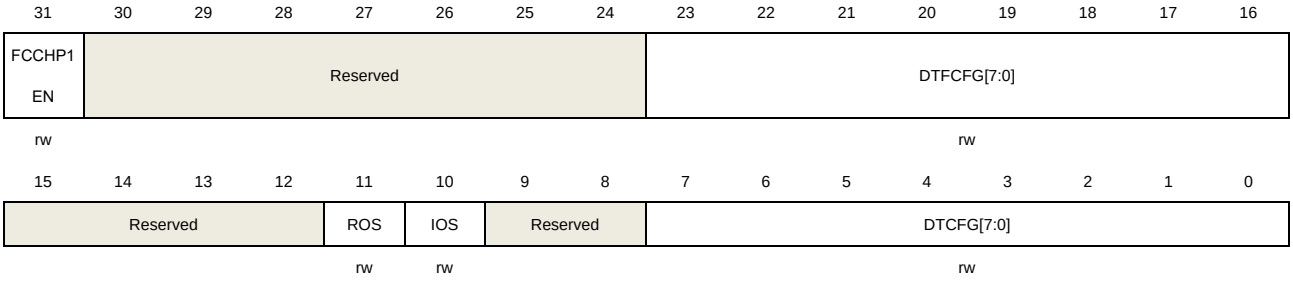
		0: the ROS、IOS and DTCFG[7:0] bits in TIMERx_CCHP0 register is active 1: the ROS、IOS and DTCFG[7:0] bits in TIMERx_FCCHP0 register is active This bit can be modified only when PROT [1:0] bit-filed in TIMERx_CCHP0 register is 00.
30:24	Reserved	Must be kept at reset value.
23:16	DTFCFG[7:0]	Dead time 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.
15:12	Reserved	Must be kept at reset value.
11	ROS	Run mode off-state configure When POEN bit is set, this bit specifies the output state for the channels which has a complementary output and has been configured in output mode. 0: When POEN bit is set, the channel output signals (CH0_O / MCH0_O) are disabled. 1: When POEN bit is set, the channel output signals (CH0_O / MCH0_O) are enabled, with relationship to CH0EN / MCH0EN bits in TIMERx_CHCTL2 register. This bit cannot be modified when PROT [1:0] bit-filed in TIMERx_CCHP0 register is 10 or 11.
10	IOS	Idle mode off-state configure When POEN bit is reset, this bit specifies the output state for the channels which has been configured in output mode. 0: When POEN bit is reset, the channel output signals (CH0_O / MCH0_O) are disabled. 1: When POEN bit is reset, the channel output signals (CH0_O / MCH0_O) are enabled, with relationship to CH0EN / MCH0EN bits in TIMERx_CHCTL2 register. This bit cannot be modified when PROT [1:0] bit-filed in TIMERx_CCHP0 register is 10 or 11.
9:8	Reserved	Must be kept at reset value.
7:0	DTCFG[7:0]	Dead time configure This bit-field controls the value of the dead-time, which is inserted before the output transitions. The relationship between DTCFG value and the duration of dead-time is as follow: DTCFG [7:5] =3'b0xx: DTvalue = DTCFG [7:0]x t_{DT}, t_{DT}=t_{DTS}.

DTCFG [7:5] =3'b10x: DTvalue = (64+DTCFG [5:0])xt_{DT}, t_{DT} =t_{DTS}*2.
DTCFG [7:5] =3'b110: DTvalue = (32+DTCFG [4:0])xt_{DT}, t_{DT}=t_{DTS}*8.
DTCFG [7:5] =3'b111: DTvalue = (32+DTCFG [4:0])xt_{DT}, t_{DT} =t_{DTS}*16.
This bit can be modified only when PROT [1:0] bit-filed in TIMERx_CCHP0 register is 00.

Free complementary channel protection register 1 (TIMERx_FCCHP1)

Address offset: 0x80
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).
This register is used to configure the outputs of CH1_O / MCH1_O.



Bits	Fields	Descriptions
31	FCCHP1EN	Free complementary channel protection register 1 enable 0: the ROS、IOS and DTCFG[7:0] bits in TIMERx_CCHP0 register is active 1: the ROS、IOS and DTCFG[7:0] bits in TIMERx_FCCHP1 register is active This bit can be modified only when PROT [1:0] bit-filed in TIMERx_CCHP0 register is 00.
30:24	Reserved	Must be kept at reset value.
23:16	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.
15:12	Reserved	Must be kept at reset value.
11	ROS	Run mode off-state configure When POEN bit is set, this bit specifies the output state for the channels which has

		a complementary output and has been configured in output mode. 0: When POEN bit is set, the channel output signals (CH1_O / MCH1_O) are disabled. 1: When POEN bit is set, the channel output signals (CH1_O / MCH1_O) are enabled, with relationship to CH1EN / MCH1EN bits in TIMERx_CHCTL2 register. This bit cannot be modified when PROT [1:0] bit-filed in TIMERx_CCHP0 register is 10 or 11.
10	IOS	Idle mode off-state configure When POEN bit is reset, this bit specifies the output state for the channels which has been configured in output mode. 0: When POEN bit is reset, the channel output signals (CH1_O / MCH1_O) are disabled. 1: When POEN bit is reset, the channel output signals (CH1_O / MCH1_O) are enabled, with relationship to CH1EN / MCH1EN bits in TIMERx_CHCTL2 register. This bit cannot be modified when PROT [1:0] bit-filed in TIMERx_CCHP0 register is 10 or 11.
9:8	Reserved	Must be kept at reset value.
7:0	DTCFG[7:0]	Dead time configure This bit-field controls the value of the dead-time, which is inserted before the output transitions. The relationship between DTCFG value and the duration of dead-time is as follow: DTCFG [7:5] =3'b0xx: DTvalue = DTCFG [7:0]x t_{DT}, t_{DT}=t_{DTS}. DTCFG [7:5] =3'b10x: DTvalue = (64+DTCFG [5:0])xt_{DT}, t_{DT} =t_{DTS}*2. DTCFG [7:5] =3'b110: DTvalue = (32+DTCFG [4:0])xt_{DT}, t_{DT}=t_{DTS}*8. DTCFG [7:5] =3'b111: DTvalue = (32+DTCFG [4:0])xt_{DT}, t_{DT} =t_{DTS}*16. This bit can be modified only when PROT [1:0] bit-filed in TIMERx_CCHP0 register is 00.

Free complementary channel protection register 2 (TIMERx_FCCHP2)

Address offset: 0x84
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).
This register is used to configure the outputs of CH2_O / MCH2_O.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FCCHP2EN	Reserved							DTFCFG[7:0]							
rw								rw							

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

Bits	Fields	Descriptions
31	FCCHP2EN	Free complementary channel protection register 2 enable 0: the ROS、IOS and DTCFG[7:0] bits in TIMERx_CCHP0 register is active 1: the ROS、IOS and DTCFG[7:0] bits in TIMERx_FCCHP2 register is active This bit can be modified only when PROT [1:0] bit-filed in TIMERx_CCHP0 register is 00.
30:24	Reserved	Must be kept at reset value.
23:16	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.
15:12	Reserved	Must be kept at reset value.
11	ROS	Run mode off-state configure When POEN bit is set, this bit specifies the output state for the channels which has a complementary output and has been configured in output mode. 0: When POEN bit is set, the channel output signals (CH2_O / MCH2_O) are disabled. 1: When POEN bit is set, the channel output signals (CH2_O / MCH2_O) are enabled, with relationship to CH2EN / MCH2EN bits in TIMERx_CHCTL2 register. This bit cannot be modified when PROT [1:0] bit-filed in TIMERx_CCHP0 register is 10 or 11.
10	IOS	Idle mode off-state configure When POEN bit is reset, this bit specifies the output state for the channels which has been configured in output mode. 0: When POEN bit is reset, the channel output signals (CH2_O / MCH2_O) are disabled. 1: When POEN bit is reset, the channel output signals (CH2_O / MCH2_O) are enabled, with relationship to CH2EN / MCH2EN bits in TIMERx_CHCTL2 register. This bit cannot be modified when PROT [1:0] bit-filed in TIMERx_CCHP0 register is 10 or 11.
9:8	Reserved	Must be kept at reset value.
7:0	DTCFG[7:0]	Dead time configure This bit-field controls the value of the dead-time, which is inserted before the output

transitions. The relationship between DTCFG value and the duration of dead-time is as follow:

DTCFG [7:5] =3'b0xx: DTvalue = DTCFG [7:0]x t_{DT}, t_{DT}=t_{DTS}.

DTCFG [7:5] =3'b10x: DTvalue = (64+DTCFG [5:0])xt_{DT}, t_{DT} =t_{DTS}*2.

DTCFG [7:5] =3'b110: DTvalue = (32+DTCFG [4:0])xt_{DT}, t_{DT}=t_{DTS}*8.

DTCFG [7:5] =3'b111: DTvalue = (32+DTCFG [4:0])xt_{DT}, t_{DT} =t_{DTS}*16.

This bit can be modified only when PROT [1:0] bit-filed in TIMERx_CCHP0 register is 00.

Free complementary channel protection register 3 (TIMERx_FCCHP3)

Address offset: 0x88

Reset value: 0x0000 0000

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

This register is used to configure the outputs of CH3_O / MCH3_O.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FCCHP3EN	Reserved							DTFCFG[7:0]							
rw								rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				ROS	IOS	Reserved		DTCFG[7:0]							
				rw	rw			rw							

Bits	Fields	Descriptions
31	FCCHP3EN	Free complementary channel protection register 0 enable 0: the ROS、IOS and DTCFG[7:0] bits in TIMERx_CCHP0 register is active 1: the ROS、IOS and DTCFG[7:0] bits in TIMERx_FCCHP3 register is active This bit can be modified only when PROT [1:0] bit-filed in TIMERx_CCHP0 register is 00.
30:24	Reserved	Must be kept at reset value.
23:16	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.

15:12	Reserved	Must be kept at reset value.
11	ROS	Run mode off-state configure When POEN bit is set, this bit specifies the output state for the channels which has a complementary output and has been configured in output mode. 0: When POEN bit is set, the channel output signals (CH3_O / MCH3_O) are disabled. 1: When POEN bit is set, the channel output signals (CH3_O / MCH3_O) are enabled, with relationship to CH3EN / MCH3EN bits in TIMERx_CHCTL2 register. This bit cannot be modified when PROT [1:0] bit-filed in TIMERx_CCHP0 register is 10 or 11.
10	IOS	Idle mode off-state configure When POEN bit is reset, this bit specifies the output state for the channels which has been configured in output mode. 0: When POEN bit is reset, the channel output signals (CH3_O / MCH3_O) are disabled. 1: When POEN bit is reset, the channel output signals (CH3_O / MCH3_O) are enabled, with relationship to CH3EN / MCH3EN bits in TIMERx_CHCTL2 register. This bit cannot be modified when PROT [1:0] bit-filed in TIMERx_CCHP0 register is 10 or 11.
9:8	Reserved	Must be kept at reset value.
7:0	DTCFG[7:0]	Dead time configure This bit-field controls the value of the dead-time, which is inserted before the output transitions. The relationship between DTCFG value and the duration of dead-time is as follow: DTCFG [7:5] =3'b0xx: DTvalue = DTCFG [7:0]x t_{DT}, t_{DT}=t_{DTS}. DTCFG [7:5] =3'b10x: DTvalue = (64+DTCFG [5:0])xt_{DT}, t_{DT} =t_{DTS}*2. DTCFG [7:5] =3'b110: DTvalue = (32+DTCFG [4:0])xt_{DT}, t_{DT}=t_{DTS}*8. DTCFG [7:5] =3'b111: DTvalue = (32+DTCFG [4:0])xt_{DT}, t_{DT} =t_{DTS}*16. This bit can be modified only when PROT [1:0] bit-filed in TIMERx_CCHP0 register is 00.

Alternate function control register 0 (TIMERx_AFCTL0)

Address offset: 0x8C

Reset value: 0x0000 0007

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			BRK0CMP3P	BRK0CMP2P	BRK0CMP1P	BRK0CMP0P	Reserved						BRK0IN2P	BRK0IN1P	BRK0IN0P
			rw	rw	rw	rw							rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

BRK0CMP6EN	BRK0CMP5EN	BRK0CMP4EN	BRK0CMP3EN	BRK0CMP2EN	BRK0CMP1EN	BRK0CMP0EN	BRK0HPDFEN	BRK0CMP7EN	Reserved	BRK0IN2EN	BRK0IN1EN	BRK0IN0EN
rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw

Bits	Fields	Descriptions
31:29	Reserved	Must be kept at reset value.
28	BRK0CMP3P	BREAK0 CMP3 input polarity This bit is used to configure the CMP3 input polarity, and the specific polarity is determined by this bit and the BRK0P bit. 0: CMP3 input signal will not be inverted (BRK0P =0, the input signal is active low; BRK0P =1, the input signal is active high) 1: CMP3 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.
27	BRK0CMP2P	BREAK0 CMP2 input polarity This bit is used to configure the CMP2 input polarity, and the specific polarity is determined by this bit and the BRK0P bit. 0: CMP2 input signal will not be inverted (BRK0P =0, the input signal is active low; BRK0P =1, the input signal is active high) 1: CMP2 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.
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.
24:19	Reserved	Must be kept at reset value.
18	BRK0IN2P	BREAK0 BRKIN2 alternate function input polarity This bit is used to configure the BRKIN2 input polarity, and the specific polarity is determined by this bit and the BRK0P bit. 0: BRKIN2 input signal will not be inverted (BRK0P =0, the input signal is active low; BRK0P =1, the input signal is active high) 1: BRKIN2 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.
17	BRK0IN1P	BREAK0 BRKIN1 alternate function input polarity This bit is used to configure the BRKIN1 input polarity, and the specific polarity is determined by this bit and the BRK0P bit. 0: BRKIN1 input signal will not be inverted (BRK0P =0, the input signal is active low; BRK0P =1, the input signal is active high) 1: BRKIN1 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.
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	BRK0CMP6EN	BREAK0 CMP6 enable 0: CMP6 input disabled 1: CMP6 input enabled This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.
14	BRK0CMP5EN	BREAK0 CMP5 enable 0: CMP5 input disabled 1: CMP5 input enabled This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.

13	BRK0CMP4EN	BREAK0 CMP4 enable 0: CMP4 input disabled 1: CMP4 input enabled This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.
12	BRK0CMP3EN	BREAK0 CMP3 enable 0: CMP3 input disabled 1: CMP3 input enabled This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.
11	BRK0CMP2EN	BREAK0 CMP2 enable 0: CMP2 input disabled 1: CMP2 input enabled This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.
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	BRK0HPDFEN	BREAK0 HPDF input enable 0: HPDF input disabled 1: HPDF input enabled **Note**: HPDF inputs are different for TIMER0 (HPDF_BREAK[0]), TIMER7 (HPDF_BREAK[2]) and TIMER19 (HPDF_BREAK[0]). This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.
7	BRK0CMP7EN	BREAK0 CMP7 enable 0: CMP7 input disabled 1: CMP7 input enabled This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.
6:3	Reserved	Must be kept at reset value.
2	BRK0IN2EN	BREAK0 BRKIN2 alternate function input enable

		0: BRKIN2 alternate function input disabled 1: BRKIN2 alternate function input enabled This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.
1	BRK0IN1EN	BREAK0 BRKIN1 alternate function input enable 0: BRKIN1 alternate function input disabled 1: BRKIN1 alternate function input enabled This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.
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.

Alternate function control register 1 (TIMERx_AFCTL1)

Address offset: 0x90
Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			BRK1CMP3P	BRK1CMP2P	BRK1CMP1P	BRK1CMP0P	Reserved			OCRINSEL[2:0]			BRK1IN2P	BRK1IN1P	BRK1IN0P
			rw	rw	rw	rw					rw		rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BRK1CMP6EN	BRK1CMP5EN	BRK1CMP4EN	BRK1CMP3EN	BRK1CMP2EN	BRK1CMP1EN	BRK1CMP0EN	BRK1HPDFEN	BRK1CMP7EN	Reserved				BRK1IN2EN	BRK1IN1EN	BRK1IN0EN
rw	rw	rw	rw	rw	rw	rw	rw	rw					rw	rw	rw

Bits	Fields	Descriptions
31:29	Reserved	Must be kept at reset value.
28	BRK1CMP3P	BREAK1 CMP3 input polarity This bit is used to configure the CMP3 input polarity, and the specific polarity is determined by this bit and the BRK1P bit. 0: CMP3 input signal will not be inverted (BRK1P =0, the input signal is active low; BRK1P =1, the input signal is active high) 1: CMP3 input signal will be inverted (BRK1P=0, the input signal is active high; BRK1P =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.

27	BRK1CMP2P	BREAK1 CMP2 input polarity This bit is used to configure the CMP2 input polarity, and the specific polarity is determined by this bit and the BRK1P bit. 0: CMP2 input signal will not be inverted (BRK1P =0, the input signal is active low; BRK1P =1, the input signal is active high) 1: CMP2 input signal will be inverted (BRK1P=0, the input signal is active high; BRK1P =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.
26	BRK1CMP1P	BREAK1 CMP1 input polarity This bit is used to configure the CMP1 input polarity, and the specific polarity is determined by this bit and the BRK1P bit. 0: CMP1 input signal will not be inverted (BRK1P =0, the input signal is active low; BRK1P =1, the input signal is active high) 1: CMP1 input signal will be inverted (BRK1P =0, the input signal is active high; BRK1P =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	BRK1CMP0P	BREAK1 CMP0 input polarity This bit is used to configure the CMP0 input polarity, and the specific polarity is determined by this bit and the BRK1P bit. 0: CMP0 input signal will not be inverted (BRK1P =0, the input signal is active low; BRK1P =1, the input signal is active high) 1: CMP0 input signal will be inverted (BRK1P =0, the input signal is active high; BRK1P =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.
24:22	Reserved	Must be kept at reset value.
21:19	OCRINSEL[2:0]	OCPRE_CLR inputs selection 000: OCPRE_CLR0 001: OCPRE_CLR1 … 111: OCPRE_CLR7

OCPRE_CLR inputs selection	**TIMER0 / 7 / 19**
OCPRE_CLR0	CMP0_OUT
OCPRE_CLR1	CMP1_OUT
OCPRE_CLR2	CMP2_OUT
OCPRE_CLR3	CMP3_OUT
OCPRE_CLR4	CMP4_OUT
OCPRE_CLR5	CMP5_OUT
OCPRE_CLR6	CMP6_OUT

OCPRE_CLR7	CMP7_OUT

This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.

18 BRK1IN2P

BREAK1 BRKIN2 alternate function input polarity

This bit is used to configure the BRKIN2 input polarity, and the specific polarity is determined by this bit and the BRK1P bit.

0: BRKIN2 input signal will not be inverted (BRK1P =0, the input signal is active low; BRKP =1, the input signal is active high)

1: BRKIN2 input signal will be inverted (BRK1P =0, the input signal is active high; BRK1P =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.

17 BRK1IN1P

BREAK1 BRKIN1 alternate function input polarity

This bit is used to configure the BRKIN1 input polarity, and the specific polarity is determined by this bit and the BRK1P bit.

0: BRKIN1 input signal will not be inverted (BRK1P =0, the input signal is active low; BRKP =1, the input signal is active high)

1: BRKIN1 input signal will be inverted (BRK1P =0, the input signal is active high; BRK1P =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.

16 BRK1IN0P

BREAK1 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 BRK1P bit.

0: BRKIN0 input signal will not be inverted (BRK1P =0, the input signal is active low; BRKP =1, the input signal is active high)

1: BRKIN0 input signal will be inverted (BRK1P =0, the input signal is active high; BRK1P =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 BRK1CMP6EN

BREAK1 CMP6 enable

0: CMP6 input disabled

1: CMP6 input enabled

This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.

14 BRK1CMP5EN

BREAK1 CMP5 enable

0: CMP5 input disabled

1: CMP5 input enabled

This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.

13	BRK1CMP4EN	BREAK1 CMP4 enable 0: CMP4 input disabled 1: CMP4 input enabled This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.
12	BRK1CMP3EN	BREAK1 CMP3 enable 0: CMP3 input disabled 1: CMP3 input enabled This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.
11	BRK1CMP2EN	BREAK1 CMP2 enable 0: CMP2 input disabled 1: CMP2 input enabled This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.
10	BRK1CMP1EN	BREAK1 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	BRK1CMP0EN	BREAK1 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	BRK1HPDFEN	BREAK1 HPDF input enable 0: HPDF input disabled 1: HPDF input enabled **Note**: HPDF inputs are different for TIMER0 (HPDF_BREAK[1]), TIMER7 (HPDF_BREAK[3]) and TIMER19 (HPDF_BREAK[1]). This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.
7	BRK1CMP7EN	BREAK1 CMP7 enable 0: CMP7 input disabled 1: CMP7 input enabled This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.
6:3	Reserved	Must be kept at reset value.
2	BRK1IN2EN	BREAK1 BRKIN2 alternate function input enable 0: BRKIN2 alternate function input disabled

		1: BRKIN2 alternate function input enabled This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.
1	BRK1IN1EN	BREAK1 BRKIN1 alternate function input enable 0: BRKIN1 alternate function input disabled 1: BRKIN1 alternate function input enabled This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.
0	BRK1IN0EN	BREAK1 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.

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).

Counter repetition register 1 (TIMERx_CREP1)

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
CREP1[31:16]															

rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CREP1[15:0]															

rw

Bits	Fields	Descriptions
31:0	CREP1[31:0]	Counter repetition value 1 This bit-field is 32 bits and can be read on the fly. 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. **Note**: This bit-field just used with CREPSEL =1(in TIMERx_CFG register).

Complementary channel protection register 1 (TIMERx_CCHP1)

Address offset: 0x09C

Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:18	Reserved	Must be kept at reset value.
17	DTMODEN	Dead time modified on-the-fly enable 0: Dead time value modified on-the-fly disable 1: Dead time value modified on-the-fly enable This bit can be modified only when PROT [1:0] bit-filed in TIMERx_CCHP register is 00. **Note**: If this bit is set while the CEN=1, any value (in DTCFG[7:0] bit-field and DTFCFG[7:0] bit-field) written after the last update event is invalid, and only the previous value is used.
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_CCHP1 register or TIMERx_FCCHPx register. 1: The dead time on rising edge is defined in DTCFG[7:0] bit-field in TIMERx_CCHP1 register or TIMERx_FCCHPx register, and the dead time on falling

		edge is defined in DTFCFG [7:0] bit-field in TIMERx_CCHP2 register or TIMERx_FCCHPx 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.

Decoder control register (TIMERx_DECCTL)

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
Reserved					OPPSC[2:0]			OPWID[7:0]							
					rw			rw							

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

Bits	Fields	Descriptions
31:27	Reserved	Must be kept at reset value.
26:24	OPPSC[2:0]	Output pulse prescaler This bit-field specifies the clock prescaler for the pulse generator. t_{opclk} = ($2^{(OPPSC\ [2:0])}$) x t_{CK_TIMER}
23:16	OPWID[7:0]	Output pulse width This bit-field specifies the pulse width. t_{opw} = OPWID [7:0] x t_{opclk}
15:8	Reserved	Must be kept at reset value.
7:6	INDP[1:0]	Index positioning This bit-field indicates in which type of the AB inputs the index event will reset the

		counter, in quadrature decoder modes 0~4. 00: When AB inputs are 00, index event will reset the counter 01: When AB inputs are 01, index event will reset the counter 10: When AB inputs are 10, index event will reset the counter 11: When AB inputs are 11, index event will reset the counter This bit-field indicates in which type of the AB inputs the index event will reset the counter, in decoder modes 0~3. x0: When clock is low, index event will reset the counter x1: When clock is high, index event will reset the counter **Note**: In decoder modes 0~3, INDP[1] bit is no use.
5	FINDRST	First index signal reset the counter 0: All the index signals are can reset the counter 1: Only the first index signal is active and can reset the counter
4:3	Reserved	Must be kept at reset value.
2:1	INDRSTDIR[1:0]	Index signal reset counter direction This bit-field indicates in which counter direction the index signal can reset the counter. 00: Index signal resets the counter when it counts up and count down 01: Index signal resets the counter only when it counts up 10: Index signal resets the counter only when it counts down 11: Reserved **Note**: The INDRSTDIR[1:0] bit-field can be modified only when INDRSTEN =0.
0	INDRSTEN	Index signal reset enable 0: Index signal resets the counter disabled 1: Index signal resets the counter enabled

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).

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

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

Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value.
13:8	DMATC[5:0]	DMA transfer count This field defines the times of accessing (R / W) the TIMERx_DMATB register by DMA. 6'b000000: transfer 1 time 6'b000001: transfer 2 times … 6'b111000: transfer 57 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'b111000: TIMERx_CINITV 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	CREPSEL	CHVSEL	OUTSEL
												rw	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, 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	CREPSEL	The counter repetition register select This bit is used to select the counter repetition register. 0: The update event rate is depended to TIMERx_CREP0 register 1: The update event rate is depended to TIMERx_CREP1 register
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.

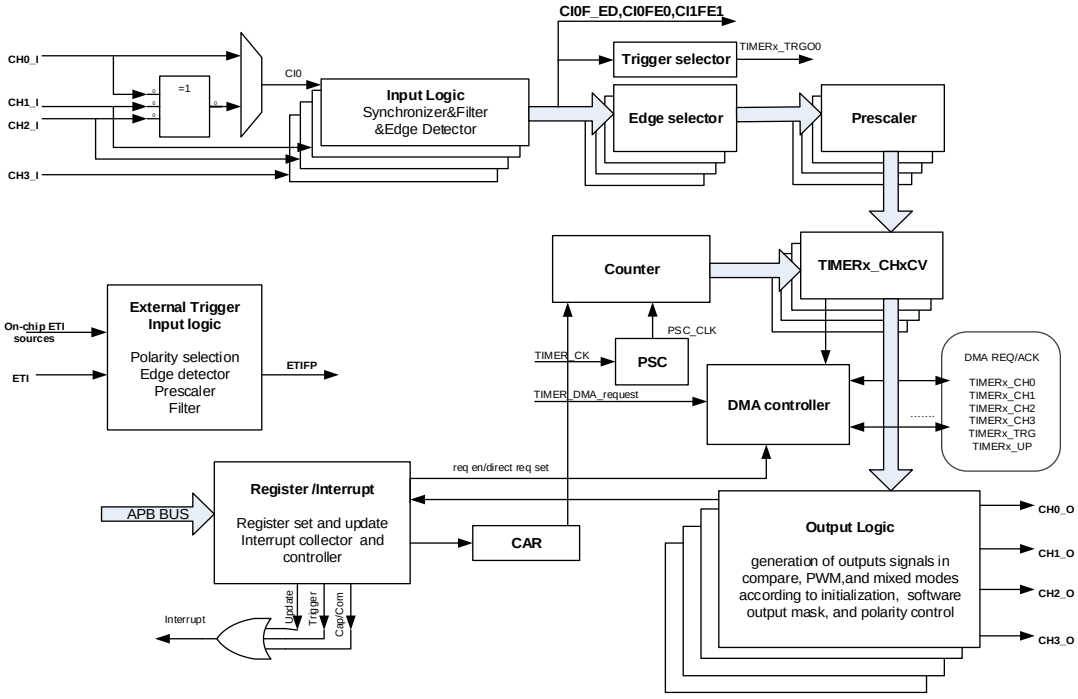
23.2.2. Characteristics

- Total channel num: 4.
- Counter width: 16 bits (TIMER2 / 3) or 32 bits (TIMER1 / 4).
- 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 23-64. General Level 0 timer block diagram provides details on the internal configuration of the general level 0 timer.

Figure 23-64. 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...15) in SYSCFG_TIMERxCFG (x=1..4) registers.

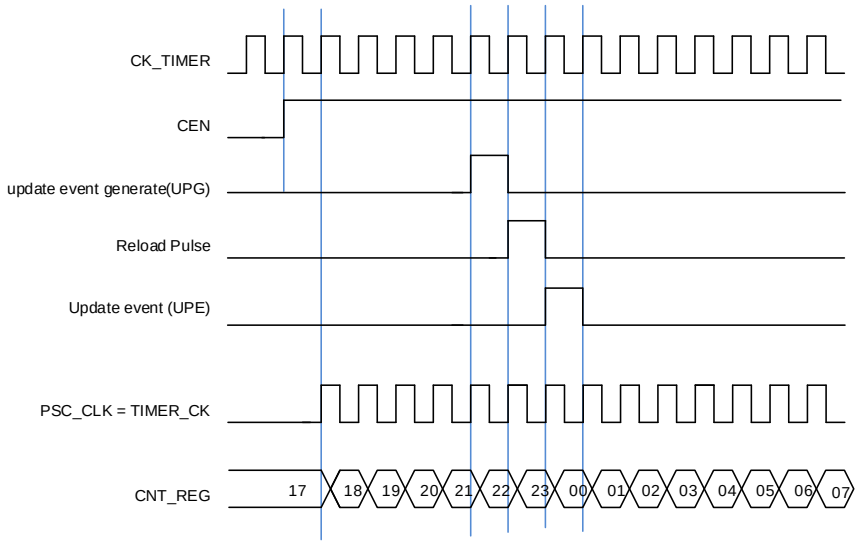
1) TSCFGy[4:0] (y = 0...15) in SYSCFG_TIMERxCFG (x=1..4) registers. Internal clock CK_TIMER is selected as timer clock source which is from module RCU.

The default clock source is the CK_TIMER for driving the counter prescaler when TSCFGy[4:0] (y = 0...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...2,6,8,9) 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...2,6,8,9) bit-filed, more details will be introduced later. When the TSCFGy[4:0] (y=3,4,5,7) are setting to an available value, the internal clock TIMER_CK is the counter prescaler driving clock source.

Figure 23-65. Normal mode, internal clock divided by 1



2) 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 CI0 / CI1. 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 ~ 10 / ITI14. This mode can be selected by setting TSCFG6[4:0] to 0x1~0x4, 0x9 ~ 0xF or 0x13.

3) 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.

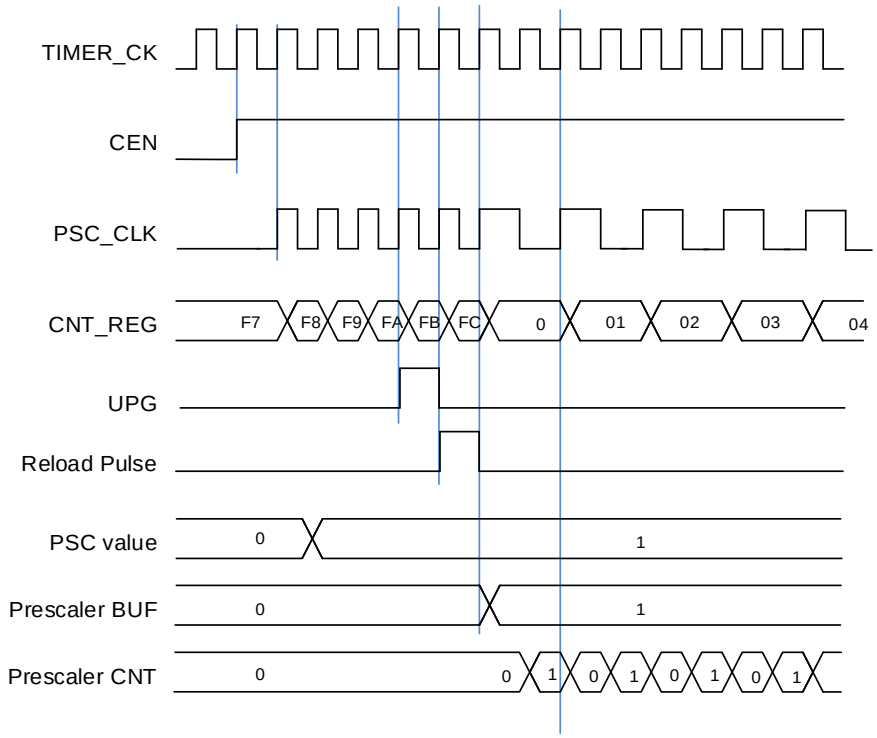
Note: The ETI signal can be input from an external ETI pin or provide by on-chip peripherals, please refer to ***Trigger selection for TIMER1_ETI register (TRIGSEL_TIMER1ETI)***, ***Trigger selection for TIMER2_ETI register (TRIGSEL_TIMER2ETI)***, ***Trigger selection for TIMER3_ETI register (TRIGSEL_TIMER3ETI)*** and ***Trigger selection for TIMER4_ETI register (TRIGSEL_TIMER4ETI)*** for more details.

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 23-66. 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 23-67. Timing chart of up counting mode, PSC=0 / 2 and ***Figure 23-68. 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 23-67. Timing chart of up counting mode, PSC=0 / 2

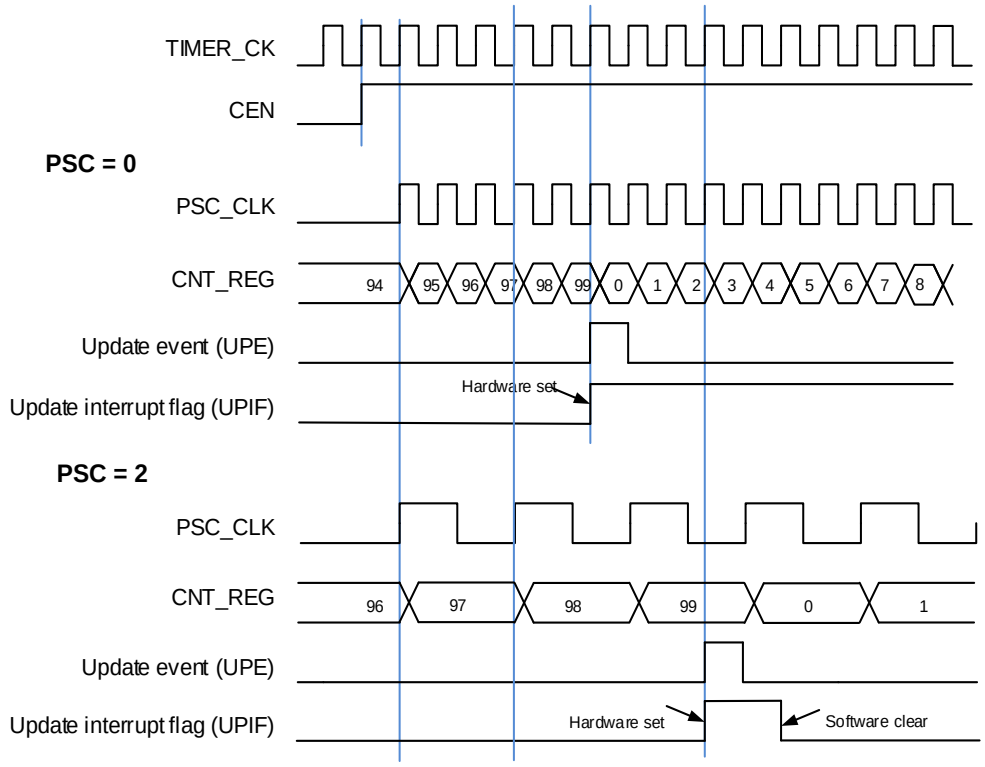
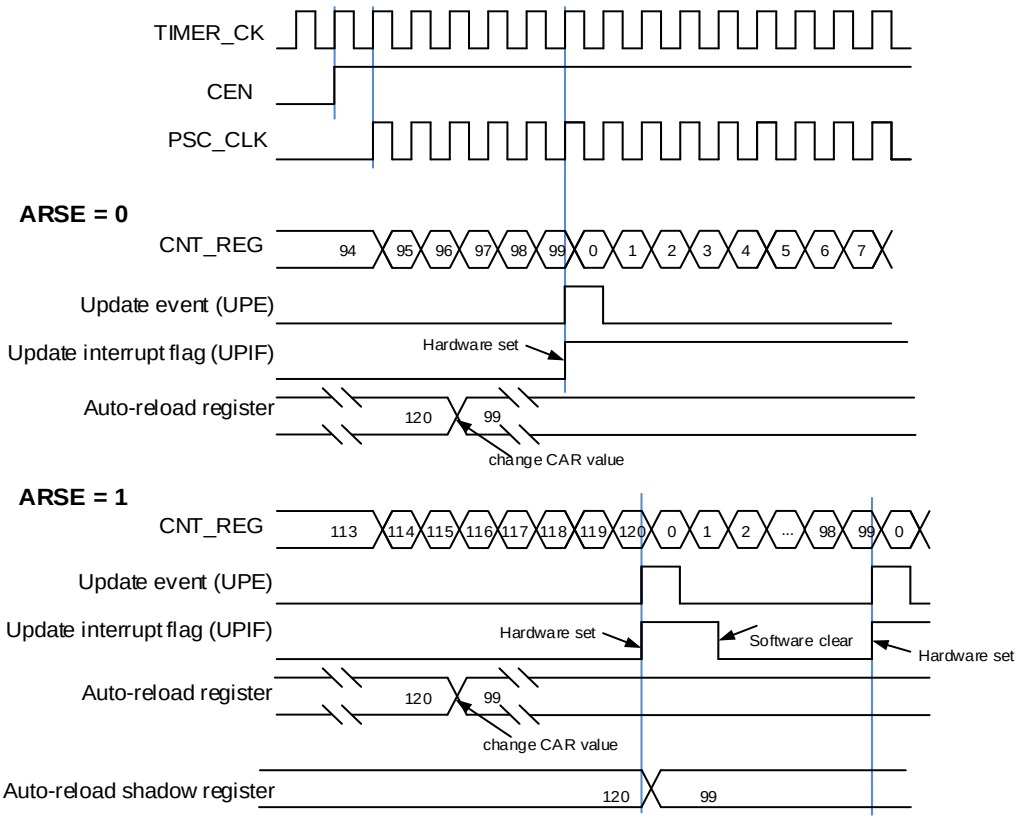


Figure 23-68. Timing chart of up counting, change TIMERx_CAR ongoing



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 23-69. Timing chart of down counting mode, PSC=0 / 2 and *Figure 23-70. 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 23-69. Timing chart of down counting mode, PSC=0 / 2

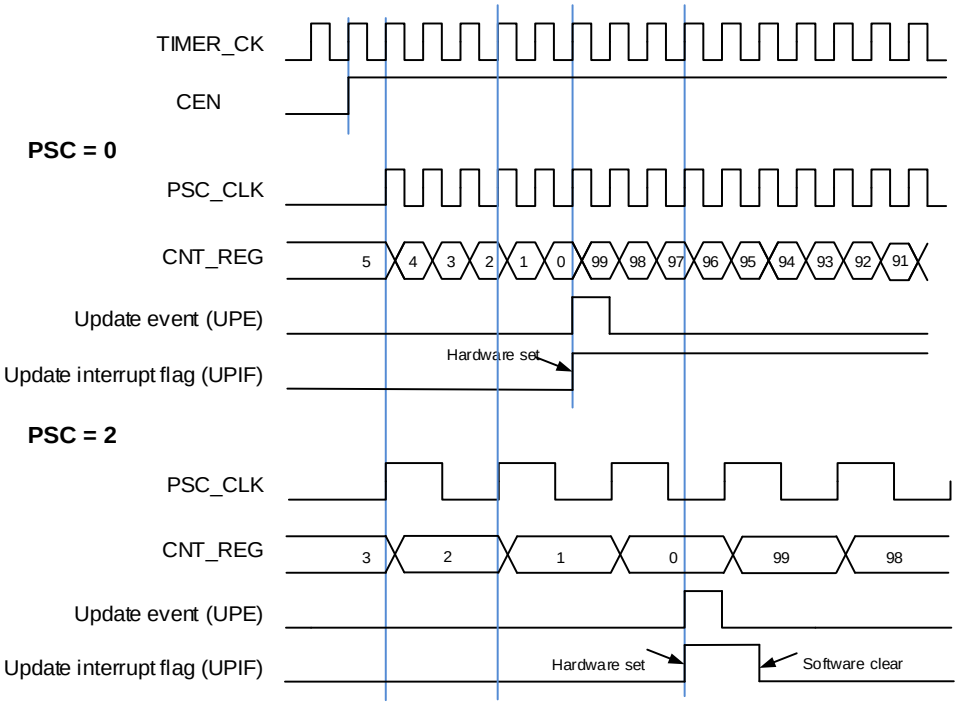
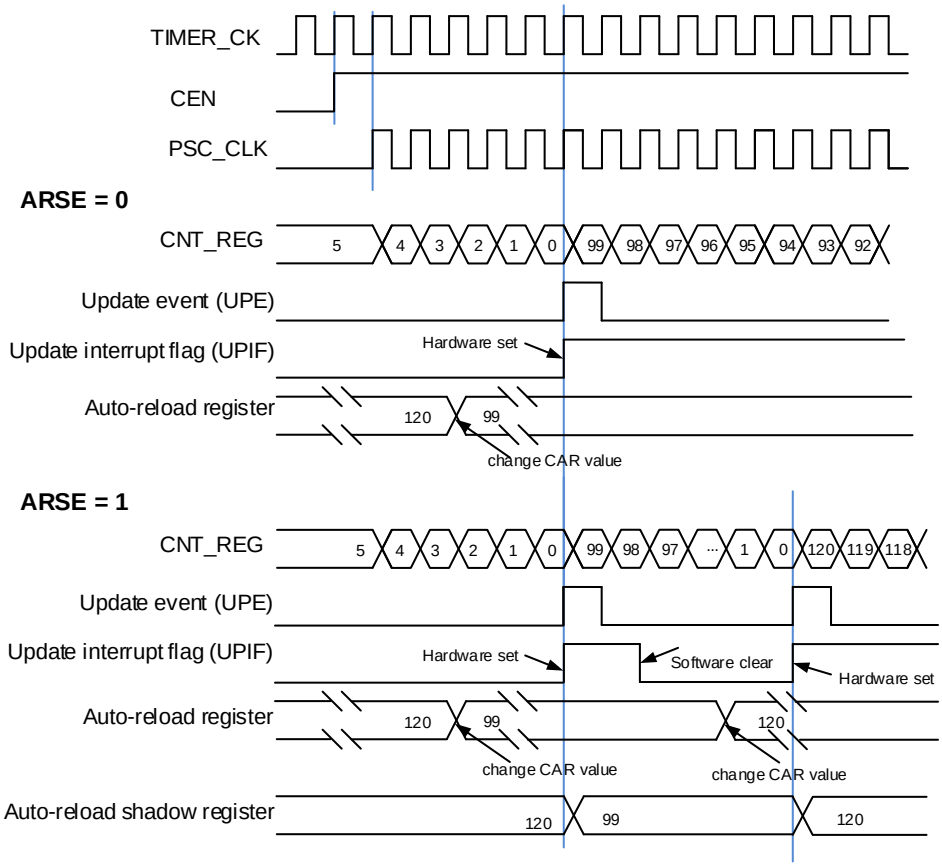


Figure 23-70. Timing chart of down counting mode, change TIMERx_CAR ongoing



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.

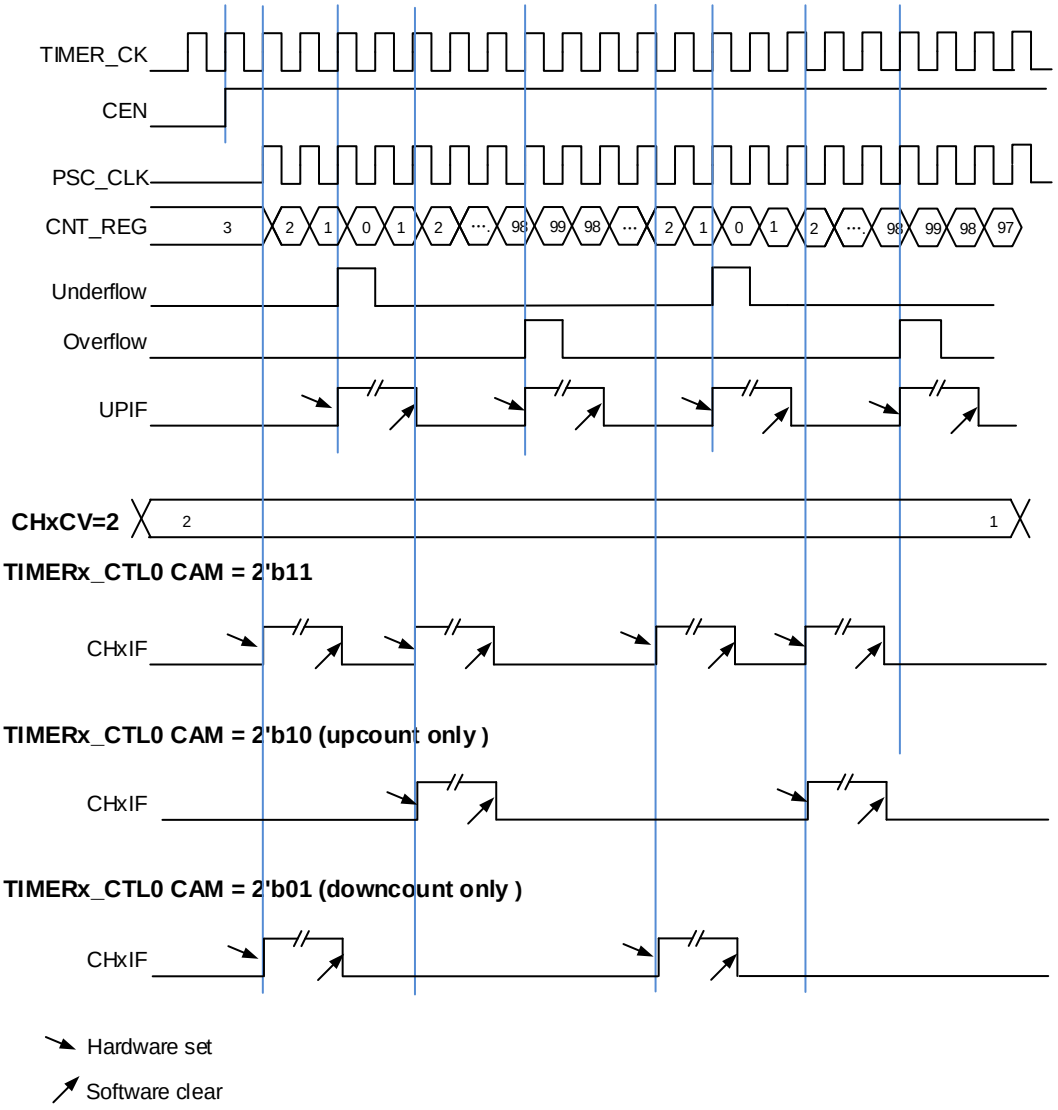
The UPIF bit in the TIMERx_INTF register will be set to 1 either when an underflow event or an overflow event occurs. While the CHxIF bit is associated with the value of CAM in TIMERx_CTL0. The details refer to ***Figure 23-71. Timing chart of center-aligned counting mode***.

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 23-71. Timing chart of center-aligned counting mode*** shows the example of the counter behavior when TIMERx_CAR=0x99, TIMERx_PSC=0x0.

Figure 23-71. Timing chart of center-aligned counting mode



Capture / compare channels

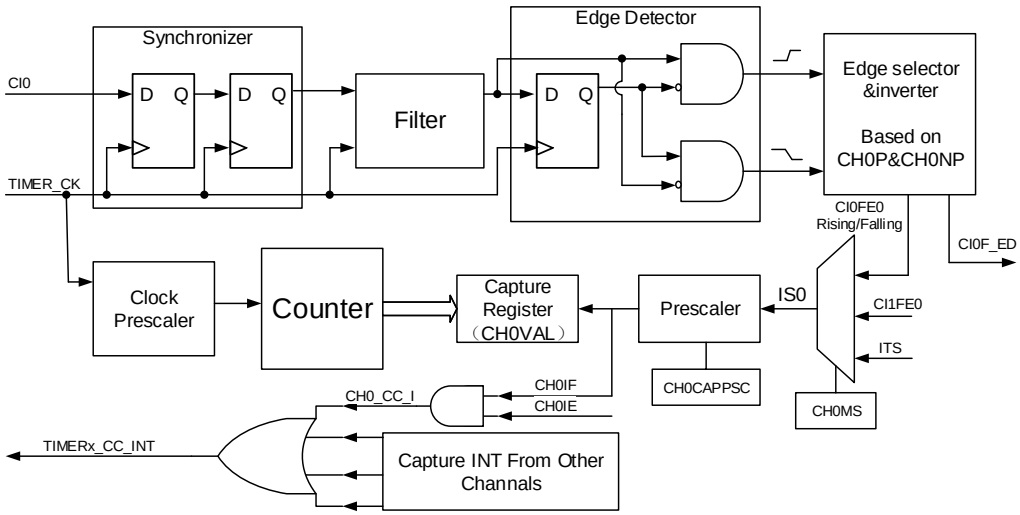
The general level0 timer has four independent channels which can be used as capture inputs or compare match outputs. Each channel is built around a channel capture compare register including an input stage, channel controller and an output stage.

4) **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 23-72. 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.

5) Output compare mode

Figure 23-73. Output compare logic (x=0,1,2,3)

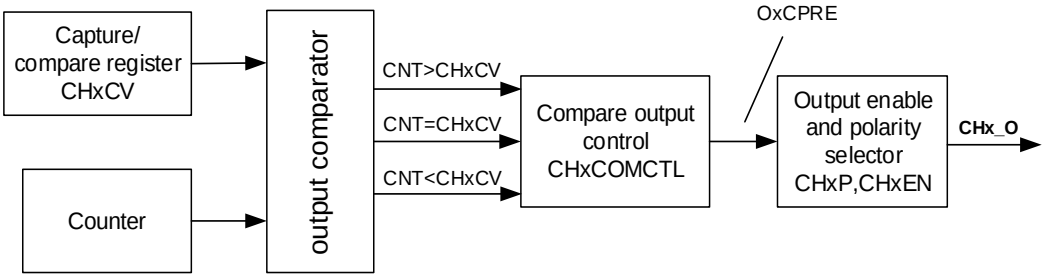


Figure 23-73. Output compare logic (x=0,1,2,3) shows the logic circuit of output compare mode. The relationship between the channel output signal CHx_O and the OxCPRE signal (more details refer to *Clear the channel output prepare signal*) is described as blew: The active level of O0CPRE is high, the 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.

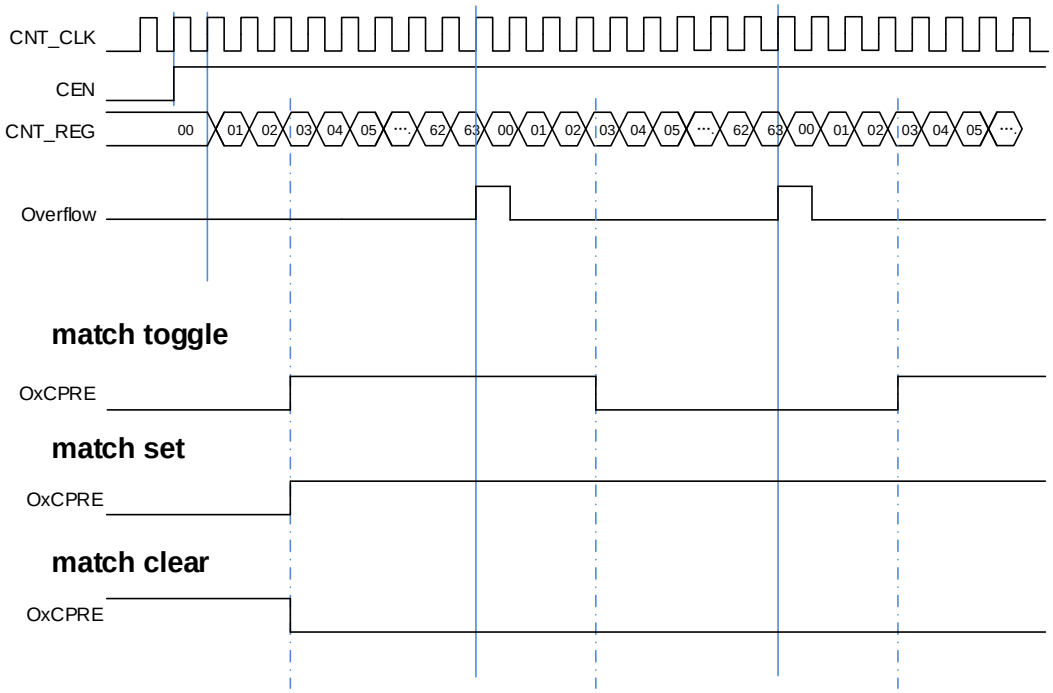
Step3: Interrupt / DMA-request enables configuration by CHxIE / CHxDEN.

Step4: Compare output timing configuration by TIMERx_CAR and TIMERx_CHxCV.
The TIMERx_CHxCV can be changed ongoing to meet the expected waveform.

Step5: Start the counter by configuring CEN to 1.

Figure 23-74. Output-compare under three modes shows the three compare modes toggle / set / clear. CAR=0x63, CHxVAL=0x3

Figure 23-74. 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 23-75. 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 23-76. Timing chart of CAPWM*** shows the CAPWM output and interrupts waveform.

In up counting mode, if the value of TIMERx_CHxCV is greater than the value of TIMERx_CAR, the output will be always active in PWM mode 0 (CHxCOMCTL=4'b0110). And if the value of TIMERx_CHxCV is greater than the value of TIMERx_CAR, the output will

be always inactive in PWM mode 1 (CHxCOMCTL=4'b0111).

Figure 23-75. Timing chart of EAPWM

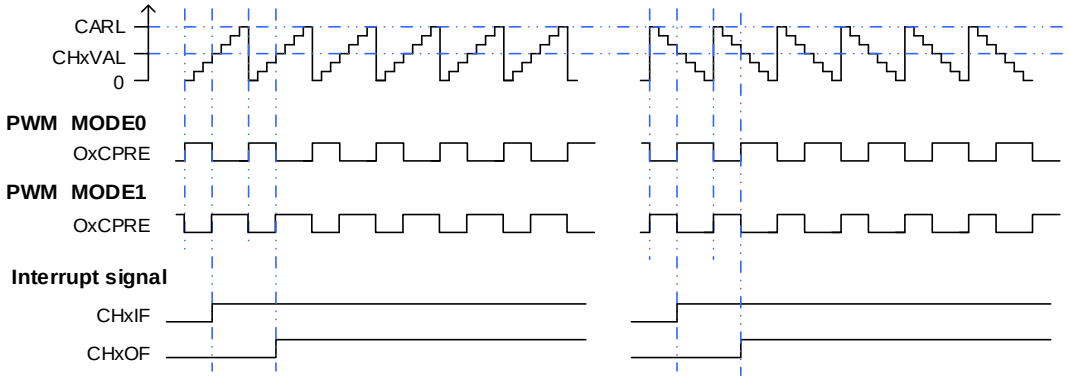
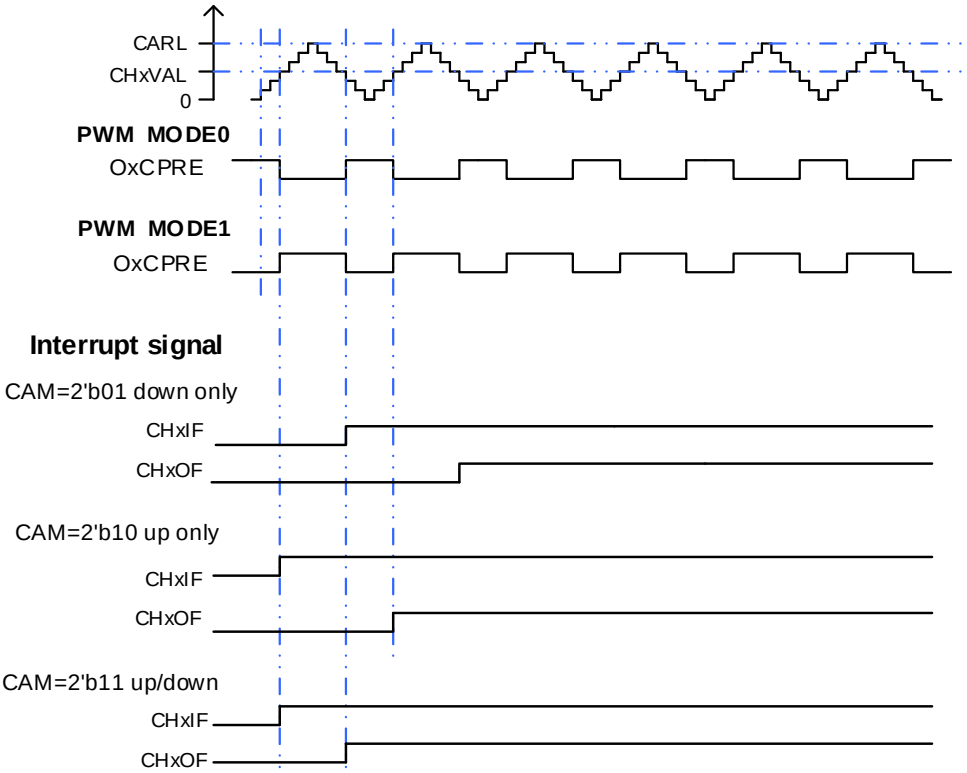


Figure 23-76. Timing chart of CAPWM



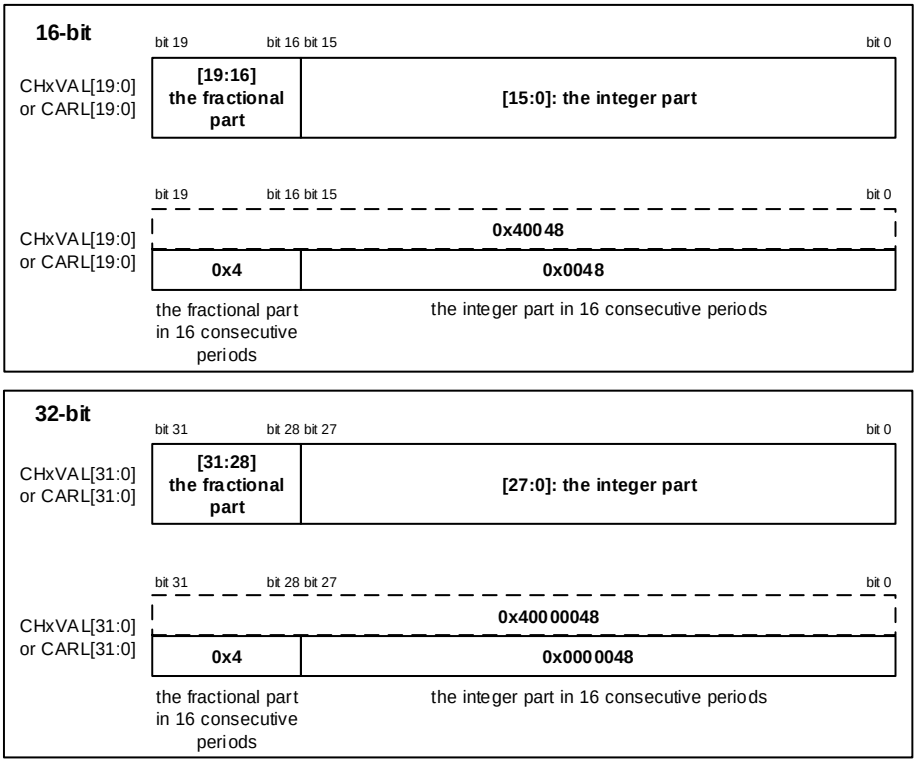
Adjustment mode

Adjustment mode is enabled by setting ADMEN bit in TIMERx_CTL0 register to 1, and this mode can improve the effective resolution of the output PWM wave. The duty cycle resolution can be improved by the CHxVAL[19:0] bit-field (TIMER2 / 3) or CHxVAL[31:0] bit-field (TIMER1 / 4) in the TIMERx_CHxCV register, and the PWM frequency resolution can be improved by the CARL[19:0] bit-field (TIMER2 / 3) or CARL [31:0] bit-field (TIMER1 / 4) in the TIMERx CAR register.

When the adjustment mode is enabled, the low 16 bits [15:0] (TIMER2 / 3) or the low 28 bits [27:0] bit-field (TIMER1 / 4) for CHxVAL bit-field and CARL bit-field are used for the integer

part and the high 4 bits [19:16] (TIMER2 / 3) or [31:28] (TIMER1 / 4) for CHxVAL bit-field and CARL bit-field are used for the fractional part. By adjust the CHxVAL or CARL values over 16 consecutive periods (no more than one TIMER clock cycle at a time) in a predefined way, can increase 16-fold in resolution.

Figure 23-77. Adjustment mode: Data format and the register bit-field



Depending on the configuration of the ADMEN bit (set or clear), the CHxVAL and CARL bit-field are automatically updated. To clear the ADMEN bit, must follow the following steps:

1. CEN bit and ARSE bits must be cleared;
2. CARL[19:16] bit-field (TIMER2 / 3) or CARL[31:28] bit-field (TIMER1 / 4) must be cleared;
3. ADMEN bit must be cleared;
4. CHxIF bit must be cleared;
5. Set the CEN bit to 1.

The following formula to calculate the PWM Resolution:

$$\text{Resolution} = f_{PSC_CLK} / f_{pwm} \quad (23\text{-}4)$$

According to Equation ***(23-4)***, when the adjustment mode is disabled (ADMEN=0), the PWM minimum frequency f_{pwm}:

For 16-bit (TIMER2 / 3),

$$(f_{pwm})_{min} = f_{PSC_CLK} / 65536 \quad (23\text{-}5)$$

For 32-bit (TIMER1 / 4),

$$(f_{pwm})_{min} = f_{PSC_CLK}/2^{28} \quad (23\text{-}6)$$

When the adjustment mode is enabled (ADMEN=1),

For 16-bit (TIMER2 / 3),

$$(f_{pwm})_{min} = f_{PSC_CLK}/(65535 + 15/16) \quad (23\text{-}7)$$

For 32-bit (TIMER1 / 4),

$$(f_{pwm})_{min} = f_{PSC_CLK}/[(2^{28}-1) + 15/16] \quad (23\text{-}8)$$

When the adjustment mode is enabled, the max values of the CHxVAL[19:0] bit-field and CARL[19:0] bit-field are 0xFFFFE (the integer part is 0xFFFE, the fractional part is 0xF), and the max values of the CHxVAL[31:0] bit-field and CARL[31:0] bit-field are 0xFFFFFFFE (the integer part is 0xFFFFFFE, the fractional part is 0xF)。

The changes of duty cycle and period within 16 consecutive periods are shown in ***Figure 23-78. PWM adjustment mode schematic diagram*** and ***Table 23-13. CHxVAL and CARL bit-field change in edge-aligned***.

Figure 23-78. PWM adjustment mode schematic diagram

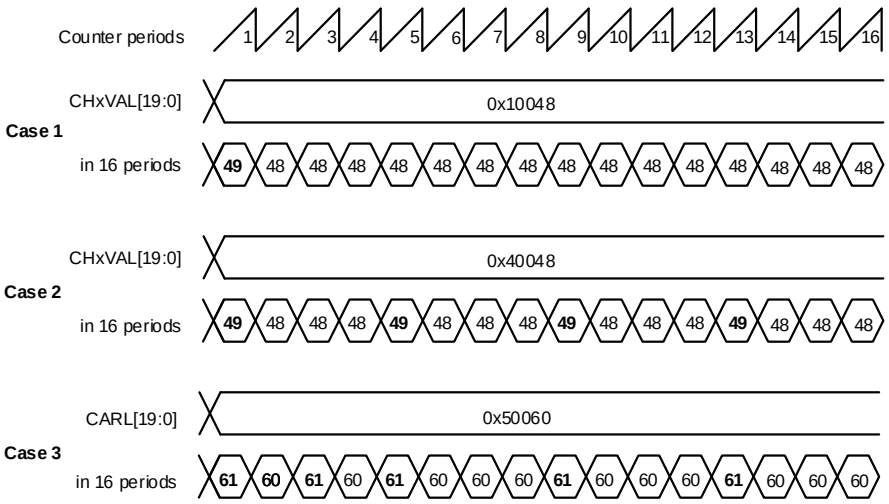


Table 23-13. CHxVAL and CARL bit-field change in edge-aligned

CHxVAL[19:16] / CARL[19:16]	Period															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0000	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0001	+1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0010	+1	-	-	-	-	-	-	-	+1	-	-	-	-	-	-	-
0011	+1	-	-	-	+1	-	-	-	+1	-	-	-	-	-	-	-
0100	+1	-	-	-	+1	-	-	-	+1	-	-	-	+1	-	-	-
0101	+1	-	+1	-	+1	-	-	-	+1	-	-	-	+1	-	-	-

CHxVAL[19:16] / CARL[19:16]	Period															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0110	+1	-	+1	-	+1	-	-	-	+1	-	+1	-	+1	-	-	-
0111	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	-	-
1000	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-
1001	+1	+1	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-
1010	+1	+1	+1	-	+1	-	+1	-	+1	+1	+1	-	+1	-	+1	-
1011	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-	+1	-	+1	-
1100	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-
1101	+1	+1	+1	+1	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-
1110	+1	+1	+1	+1	+1	+1	+1	-	+1	+1	+1	+1	+1	+1	+1	-
1111	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	-

The PWM adjustment mode can also use in center-aligned counting mode, the details are shown in ***Table 23-4. CHxVAL and CARL bit-field changes in the center-aligned counting mode***. The adjustment mode is applied over 8 consecutive PWM cycles.

Table 23-14. CHxVAL bit-field changes in the center-aligned counting mode

CHxVAL [19:16] / CARL [19:16]	Period															
	1		2		3		4		5		6		7		8	
	Up	Down	Up	Down	Up	Down	Up	Down	Up	Down	Up	Down	Up	Down	Up	Down
0000	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0001	+1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0010	+1	-	-	-	-	-	-	-	+1	-	-	-	-	-	-	-
0011	+1	-	-	-	+1	-	-	-	+1	-	-	-	-	-	-	-
0100	+1	-	-	-	+1	-	-	-	+1	-	-	-	+1	-	-	-
0101	+1	-	+1	-	+1	-	-	-	+1	-	-	-	+1	-	-	-
0110	+1	-	+1	-	+1	-	-	-	+1	-	+1	-	+1	-	-	-
0111	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	-	-
1000	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-
1001	+1	+1	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-
1010	+1	+1	+1	-	+1	-	+1	-	+1	+1	+1	-	+1	-	+1	-
1011	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-	+1	-	+1	-
1100	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-
1101	+1	+1	+1	+1	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-
1110	+1	+1	+1	+1	+1	+1	+1	-	+1	+1	+1	+1	+1	+1	+1	-
1111	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	-

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.

The PWM period is determined by (CARL + 0x0001) and the PWM pulse width is determined by the following table.

Table 23-15.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

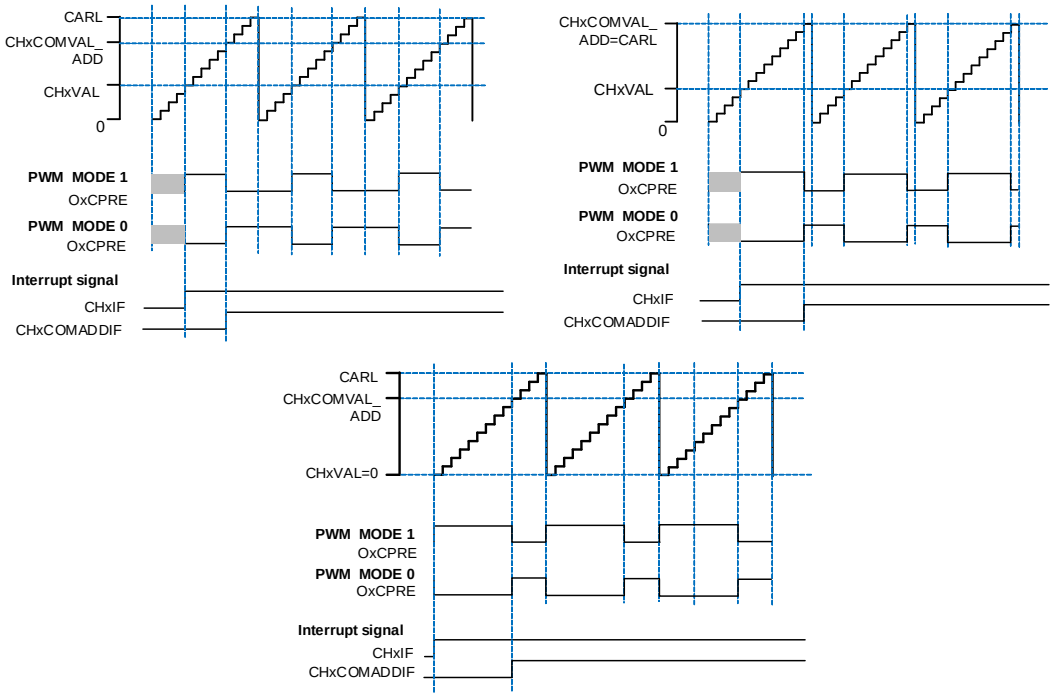
When the counter reaches the value of CHxVAL, the CHxIF bit is set and the channel x interrupt is generated if CHxIE = 1, and the DMA request will be asserted, if CHxDEN=1. When the counter reaches the value of CHxCOMVAL_ADD, the CHxCOMADDIF bit is set (this flag just used in composite PWM mode, when 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:

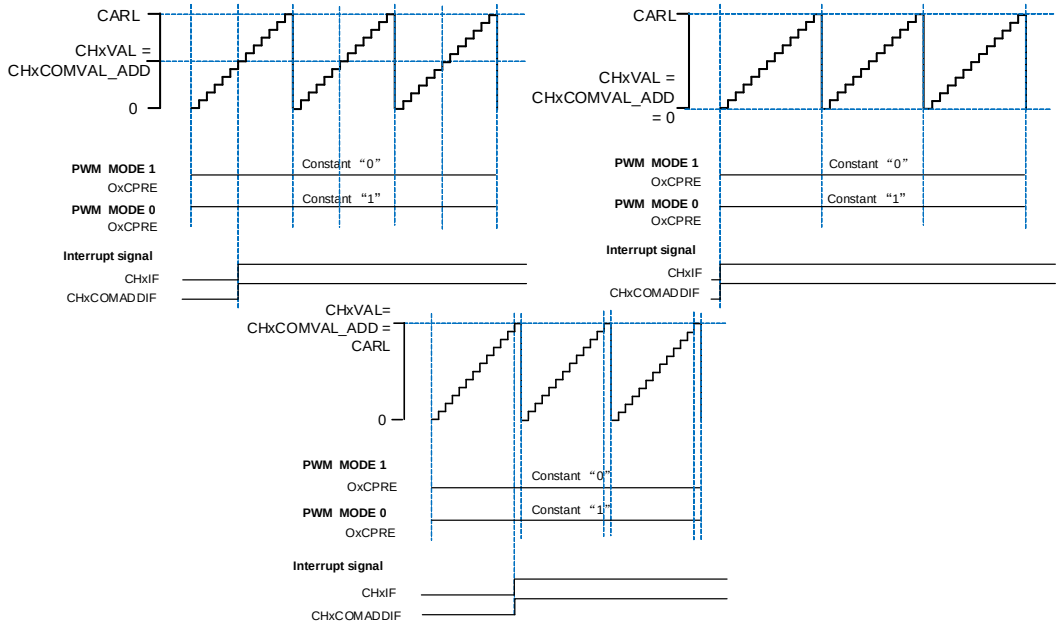
- CHxVAL < CHxCOMVAL_ADD, and the values of CHxVAL and CHxCOMVAL_ADD between 0 and CARL.

Figure 23-79. Channel x output PWM with (CHxVAL < CHxCOMVAL_ADD)



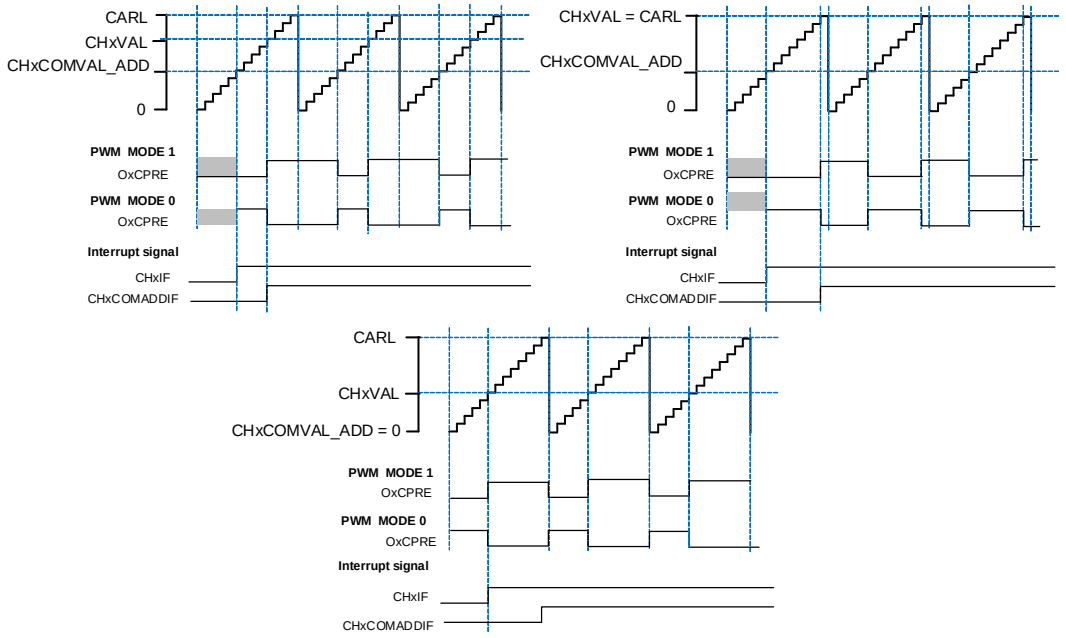
- CHxVAL = CHxCOMVAL_ADD, and the value of CHxVAL and CHxCOMVAL_ADD between 0 and CARL.

Figure 23-80. Channel x output PWM with (CHxVAL = CHxCOMVAL_ADD)



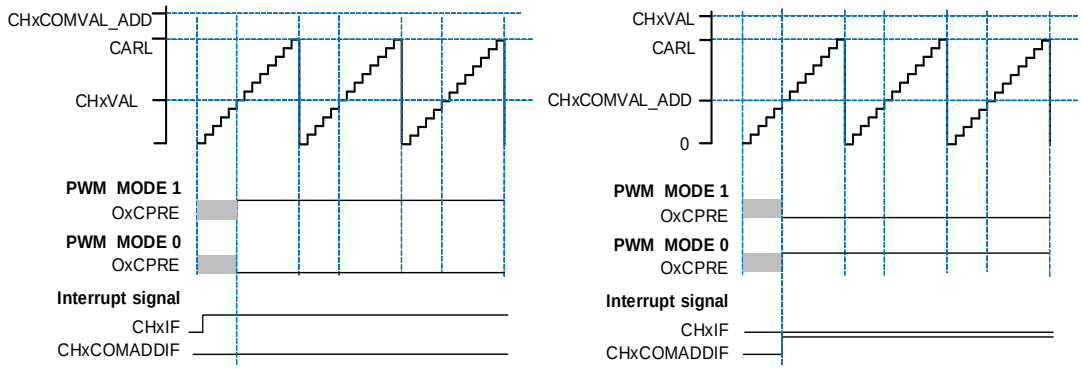
- CHxVAL > CHxCOMVAL_ADD, and the value of CHxVAL and CHxCOMVAL_ADD between 0 and CARL.

Figure 23-81. Channel x output PWM with (CHxVAL > CHxCOMVAL_ADD)



- One of the value of CHxVAL and CHxCOMVAL_ADD exceeds CARL.

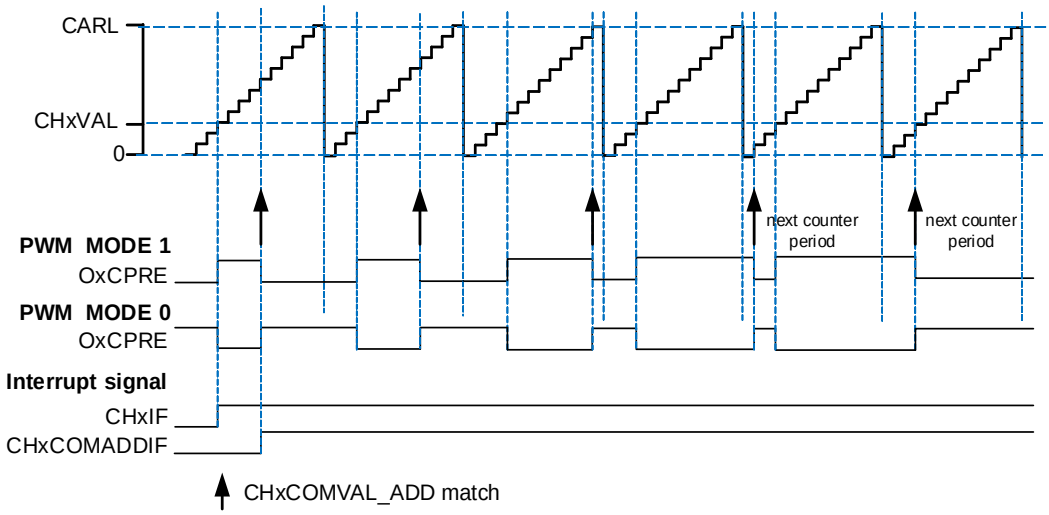
Figure 23-82. Channel x output PWM with CHxVAL or CHxCOMVAL_ADD exceeds CARL



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 23-83. Channel x output PWM duty cycle changing with CHxCOMVAL_ADD* shows the PWM output and interrupts waveform.

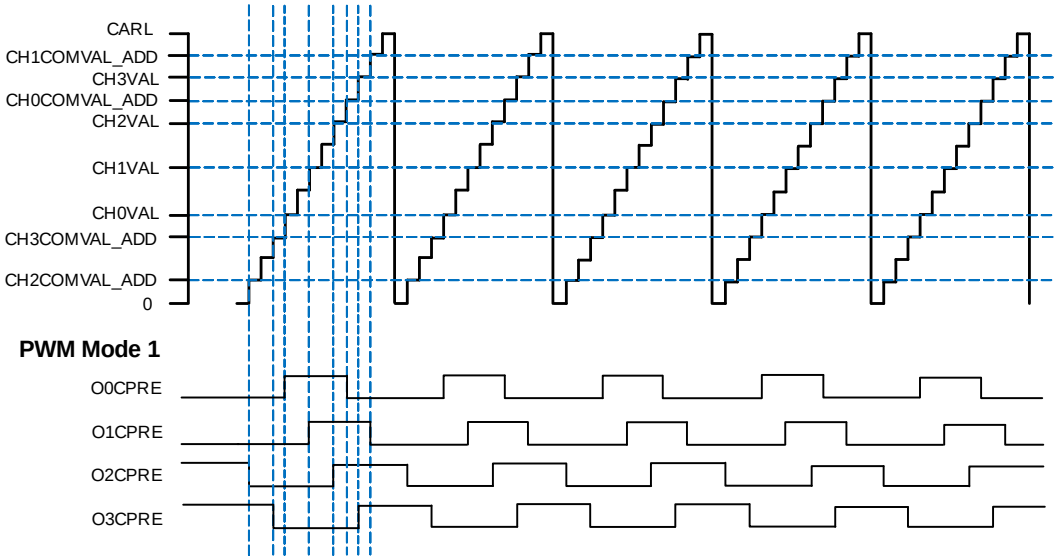
In some cases, the CHxCOMVAL_ADD match can happen on the next counter period (the value of CHxCOMVAL_ADD was written after the counter reaches the value of CHxVAL, and the value of CHxCOMVAL_ADD was less than or equal to the CHxVAL).

Figure 23-83. 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 23-84. Four Channels outputs in Composite PWM mode



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 23-85. CHx_O output with a pulse in edge-aligned mode (CHxOMPSEL≠2'b00)

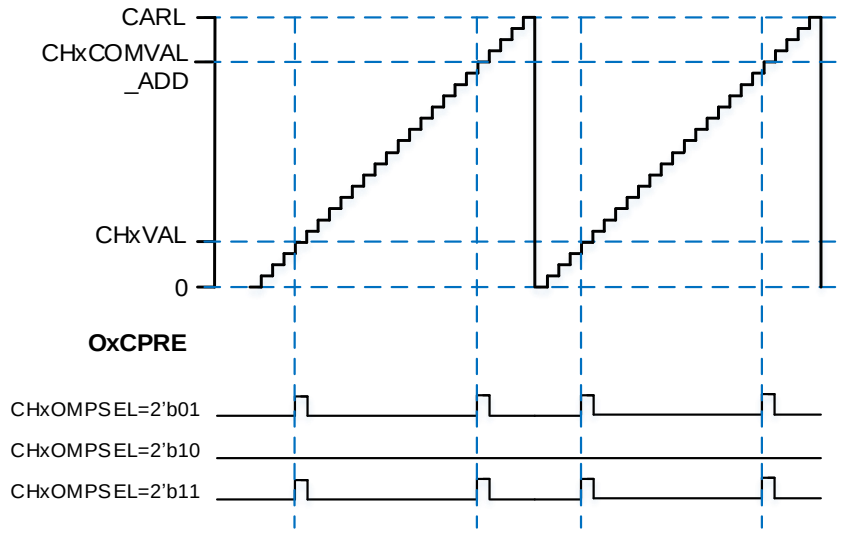
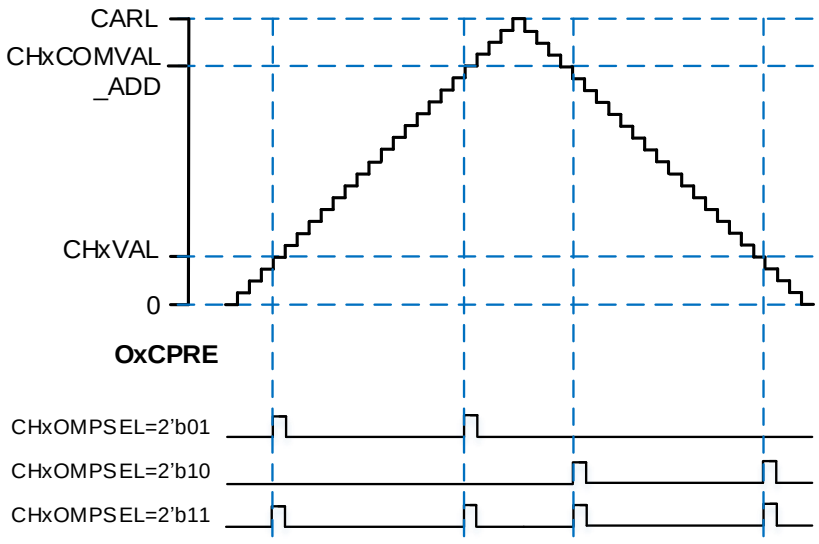


Figure 23-86. CHx_O output with a pulse in center-aligned mode (CHxOMPSEL≠2'b00)



Channel output prepare signal

As is shown in ***Figure 23-73. 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. The OxCPRE signal will not return to its active level until the next update event occurs.

Clear the channel output prepare signal

The OxCPRE signal can be cleared by the OCPRE_CLR_INT signal when the CHxCOMCEN bit (in TIMERx_CHCTLy register) is set. This function can be used in the compare output modes which configured in CHxCOMCTL[3:0] bit-field (without the value of 4’b0100 and 4’b0101).

The source of the OCPRE_CLR_INT signal can be selected by the OCRC bit in the TIMERx_SMCFG register.

When OCRC is reset, the OCPRE_CLR_INT is connected to the OCPRE_CLR input. The OxCPRE signal is cleared by the high level of the OCPRE_CLR_INT signal, and restored until the next update event occurs. The OCPRE_CLR inputs can be selected in OCRINSEL[2:0] bit-field in the TIMERx_AFCTL1 register.

When OCRC is set, the OCPRE_CLR_INT is connected to the ETIF. The OCPRE_CLR_INT input polarity is configured by ETP bit in TIMERx_SMCFG register. The ETPSC[1:0] bit-field must be set to 2’b00.

Quadrature decoder

Please refer to ***Advanced timer (TIMERx, x=0, 7,19) Quadrature decoder***.

Decoder

Please refer to ***Advanced timer (TIMERx, x=0, 7,19) Decoder***.

Quadrature decoder and decoder clock output

Please refer to ***Advanced timer (TIMERx, x=0, 7,19) Quadrature decoder and decoder clock output***.

Index input function

Please refer to ***Advanced timer (TIMERx, x=0, 7,19) Index input function***.

Hall sensor function

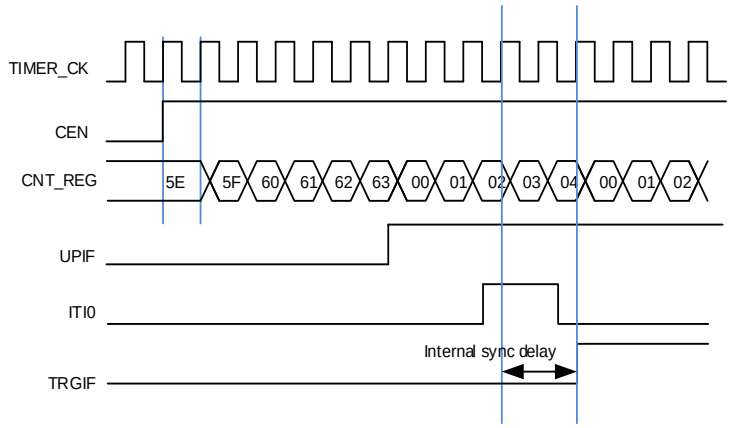
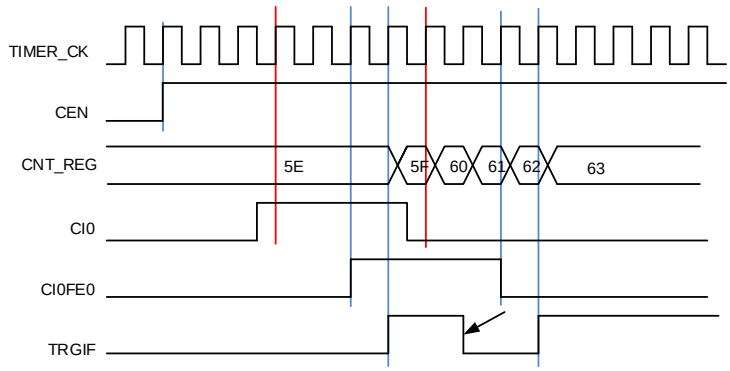
Refer to ***Advanced timer (TIMERx, x=0, 7,19) Hall sensor function***.

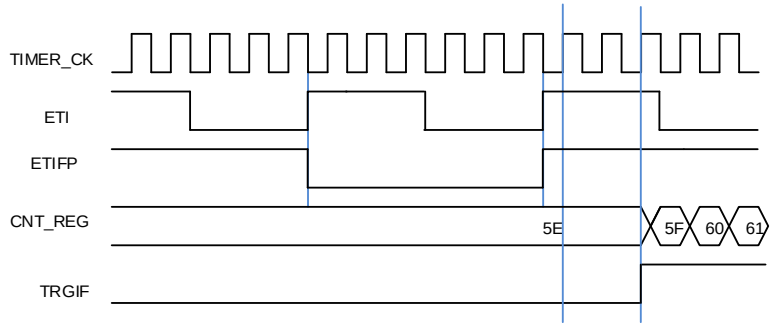
Master-slave management

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..8) in SYSCFG_TIMERxCFG(x=1..4).

Table 23-16. 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 y=7: restart + event mode y=8: pause + restart mode	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: ITI4 01010: ITI5 01011: ITI6 01100: ITI7 01101: ITI8 01110: ITI9 01111: ITI10 10000: Reserved 10001: Reserved 10010: Reserved 10011: ITI14	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 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 23-87. 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 23-88. 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 23-89. Event mode			

	Mode Selection	Source Selection	Polarity Selection	Filter and Prescaler
				
Exam4	**Restart + event mode** The counter is reinitialized and started when a rising edge of trigger input comes.			
Exam5	**Pause + restart mode** The counter will be reset when a rising edge or falling edge (is configured by PRMRPSEL bit in TIMERx_SMCFG register) of trigger input comes. And the counter counts when the trigger input is high, and it will stop when the trigger input is low. In this mode, the start and stop of the counter can be controlled.			

(1) The ETI signal can be input from an external ETI pin or provide by on-chip peripherals, please refer to ***Trigger selection for TIMER1 ETI register (TRIGSEL_TIMER1ETI)***, ***Trigger selection for TIMER2 ETI register (TRIGSEL_TIMER2ETI)***, ***Trigger selection for TIMER3 ETI register (TRIGSEL_TIMER3ETI)*** and ***Trigger selection for TIMER4 ETI register (TRIGSEL_TIMER4ETI)*** for more details.

Single pulse mode

Single pulse mode is opposite to the repetitive mode, which can be enabled by setting SPM in TIMERx_CTL0. 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.

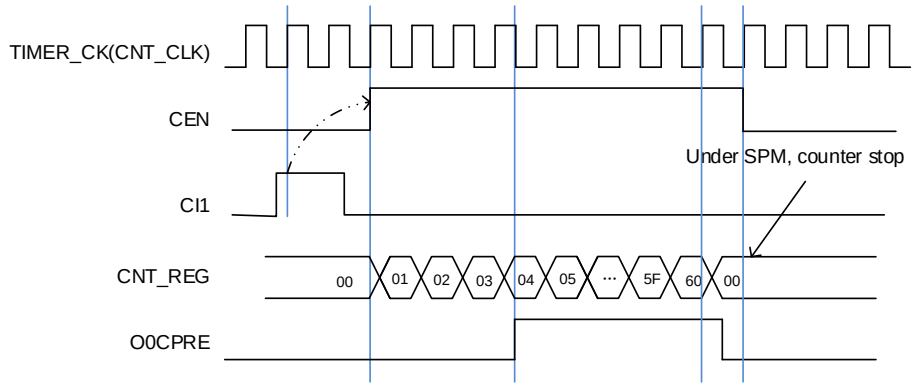
Once the timer is set to the single pulse mode, it is not necessary to configure the timer enable bit CEN in the TIMERx_CTL0 register to 1 to enable the counter. Setting the CEN bit to 1 or a trigger signal edge can generate a pulse and then keep the CEN bit at a high state until the update event occurs 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.

Single pulse mode is also applicable to composite PWM mode (CHxCPWMEN = 1’b1 and

CHxMS[2:0] = 3’b000).

Figure 23-90. Single pulse mode TIMERx_CHxCV = 0x04, TIMERx_CAR=0x60



Delayable single pulse mode

Delayable single pulse mode is enabled by setting CHxCOMCTL[3:0] in TIMERx_CHCTLx registers. In this mode, the pulse width of OxCPRE signal is determined by the TIMERx_CAR register.

Once the timer is set to the delayable single pulse mode, the following configuration is required:

- TIMERx need to work in slave mode and TSCFG7[4:0] != 5’b00000 in SYSCFG_TIMERxCFG(x=1..4) (restart +event mode);
- The CHxCOMCTL[3:0] bit-field is setting to 4’b1000 (delayable single pulse mode 0) or 4’b1001 (delayable single pulse mode 1).

In delayable SPM mode 0. The behavior of OxCPREis performed as in PWM mode 0. When counting up, the OxCPRE is active. When a trigger event occurs, the OxCPRE is inactive. The OxCPRE is active again at the next update event; When counting down, the OxCPRE is inactive, when a trigger event occurs, the OxCPRE is active. The OxCPRE is inactive again at the next update event.

In delayable mode 1. The behavior of OxCPRE is performed as in PWM mode 1. When counting up, the OxCPRE is inactive, when a trigger event occurs, the OxCPRE is active. The OxCPRE is inactive again at the next update event; When counting down, the OxCPRE is active. When a trigger event occurs, the OxCPRE is inactive. The OxCPRE is active again at the next update event.

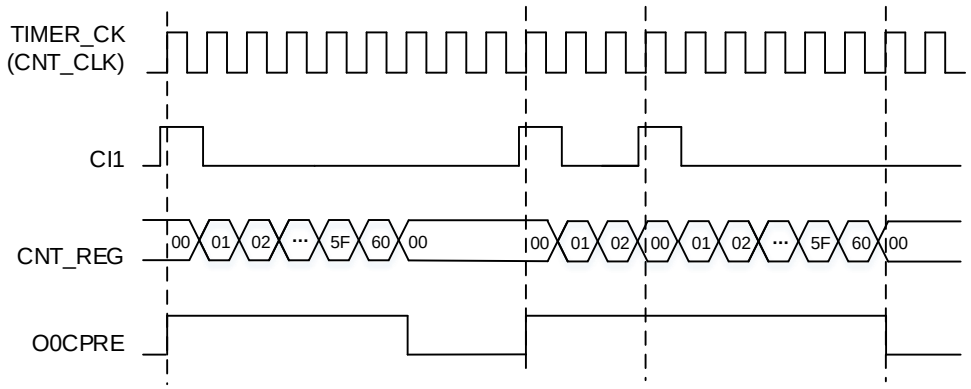
The PWM adjustment mode can also use in delayable SPM modes.

Note:

- The center-aligned counting mode cannot be used in this mode and the CAM[1:0] = 2’b00(in TIMERx_CTL0 register);
- When counter counting up (DIR = 0 in TIMERx_CTL0 register), the value of

TIMERx_CHxCV should be set to 0; When counting down (DIR =1 in TIMERx_CTL0 register), the value of TIMERx_CHxCV should be greater than or equal to the value of TIMERx_CAR register.

Figure 23-91. delayable single pulse mode TIMERx_CHxCV=0x00, TIMERx_CAR=0x60



Programmable pulse output

Please refer to ***Advanced timer (TIMERx, x=0, 7,19) Programmable pulse output***.

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.

Output DIR bit

The DIR bit can output on the CH2 and CH3, and this function is enabled by setting CH2COMCTL[3:0] or CH3COMCTL[3:0] bit-field to 4'b1011 in the TIMERx_CHCTL1 register.

When the counter works in center-aligned counting mode, this function can be used to indicates the counter direction of the counter. When the counter works in decoder modes, this function can be used to indicates the rotation direction for the external signal.

UPIF bit backup

The UPIF bit backup function is enabled by setting UPIFBUEN in the TIMERx_CTL0 register. The UPIF and UPIFBU bits are fully synchronized and without latency.

By using this function, the UPIF bit in the TIMERx_INTF register will be backuped to the UPIFBU bit in the TIMERx_CNT register. This can avoid conflicts when reading the counter and interrupt processing.

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			ADMEN	UPIFBUEN	Reserved	CKDIV[1:0]		ARSE	CAM[1:0]		DIR	SPM	UPS	UPDIS	CEN
			rw	rw		rw		rw	rw		rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:13	Reserved	Must be kept at reset value.
12	ADMEN	Adjustment mode enable 0: Adjustment mode disabled 1: Adjustment mode enabled Note: This bit can be modified only when the CEN bit is 0.
11	UPIFBUEN	UPIF bit backup enable 0: Backup disable. UPIF bit is not backuped to UPIFBU bit in TIMERx_CNT register. 1: Backup enabled. UPIF bit is backuped to UPIFBU bit in TIMERx_CNT register.
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.

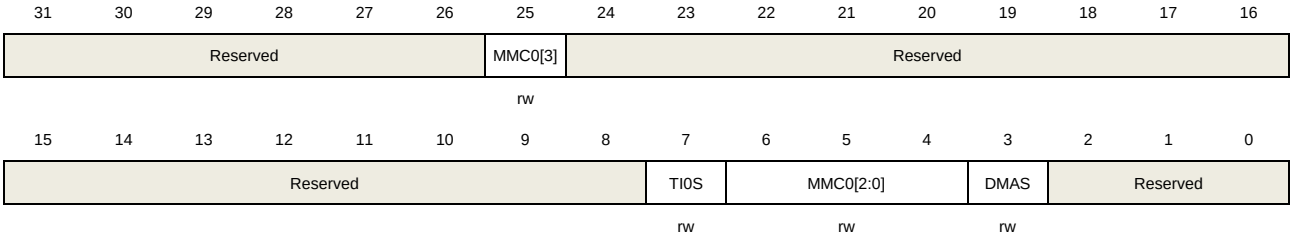
0	CEN	Counter enable 0: Counter disable 1: Counter enable The CEN bit must be set by software when timer works in external clock mode, pause mode or decoder mode. While in event mode, the hardware can set the CEN bit automatically.

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 TRGO0 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 TRGO0 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 TRGO0. 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 TRGO0 output, except if the master-slave mode is selected. 0010: Update. In this mode, the master mode controller selects the update event as TRGO0. 011: Capture / compare pulse. In this mode, the master mode controller generates a TRGO0 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 TRGO0. 0101: Compare. In this mode, the master mode controller selects the O1CPRE signal as TRGO0. 0110: Compare. In this mode, the master mode controller selects the O2CPRE signal as TRGO0. 0111: Compare. In this mode, the master mode controller selects the O3CPRE signal as TRGO0. 1000: Decoder clock output. In this mode, the master mode controller selects the Decoder clock output signal as TRGO0. 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					PRMRPSEL	DECMODS	DECMODEN	Reserved							
					rw	rw	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			OCRC	Reserved		
rw	rw	rw		rw				rw				rw			

Bits	Fields	Descriptions
31:27	Reserved	Must be kept at reset value.
26	PRMRPSEL	Pause + restart mode reset polarity selection

		0: Counter is reset at falling edge 1: Counter is reset at rising edge
25	DECMODS	Decoder mode update source This bit used to select the decoder mode update source 0: Decoder mode is updated by the TIMER update event after the modification. (Just used when the index signal is disabled.) 1: Decoder mode is updated by the index event after the modification.
24	DECMODEN	Decoder mode modified on-the-fly enable 0: Decoder mode modified on-the-fly disable 1: Decoder mode modified on-the-fly enable. When this bit is set to 1, the decoder mode can be modified from one mode to another mode.
23:16	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
6:4	Reserved	Must be kept at reset value.
3	OCRC	OxCPRE clear source selection This bit is used to select the OxCREF clear source. 0: OCPRE_CLR_INT is connected to the OCPRE_CLR input 1: OCPRE_CLR_INT is connected to ETIF
2:0	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								INDERRIE	DIRTRANIE	DECDISIE	DECJIE
rw	rw	rw	rw									rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INDIE	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 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.
27:20	Reserved	Must be kept at reset value.
19	INDERRIE	Index error interrupt enable 0: Index error interrupt disabled 1: Index error interrupt enabled
18	DIRTRANIE	Direction transform interrupt enable 0: Direction transform interrupt disabled 1: Direction transform interrupt enabled
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	INDIE	Index interrupt enable 0: Index interrupt disabled

		1: Index interrupt enabled
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
CH3COMADDIF	CH2COMADDIF	CH1COMADDIF	CH0COMADDIF	Reserved								INDERRIF	DIRTRANIF	DECDISIF	DECJIF
rc_w0	rc_w0	rc_w0	rc_w0									rc_w0	rc_w0	rc_w0	rc_w0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INDIF	Reserved		CH3OF	CH2OF	CH1OF	CH0OF	Reserved		TRGIF	Reserved	CH3IF	CH2IF	CH1IF	CH0IF	UPIF
rc_w0			rc_w0	rc_w0	rc_w0	rc_w0			rc_w0		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

Bits	Fields	Descriptions
31	CH3COMADDIF	Channel 3 additional compare interrupt flag. Refer to CH0COMADDIF description.
30	CH2COMADDIF	Channel 2 additional compare interrupt flag. Refer to CH0COMADDIF description.
29	CH1COMADDIF	Channel 1 additional compare interrupt flag. Refer to CH0COMADDIF description.
28	CH0COMADDIF	Channel 0 additional compare interrupt flag. This flag is set by hardware and cleared by software. If channel 0 is in output mode, this flag is set when a compare event occurs. 0: No channel 0 output compare interrupt occurred 1: Channel 0 output compare interrupt occurred **Note:** This flag just used in composite PWM mode (when CH0CPWMEN = 1'b1, CH0MS[2:0] = 3'b000 and CH0COMCTL = 4'b0110 or 4'b0111).
27:20	Reserved	Must be kept at reset value.
19	INDERRIF	Index error interrupt flag This flag is set by hardware and cleared by software. 0: No Index error interrupt occurred 1: Index error interrupt occurred
18	DIRTRANIF	Direction transform interrupt flag This flag is set by hardware when the DIR bit in TIMERx_CTL0 is changed in

		decoder modes, and cleared by software. 0: No direction transform interrupt occurred 1: Direction transform interrupt occurred
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	INDIF	Index interrupt flag This flag is set by hardware and cleared by software. 0: No Index interrupt occurred 1: Index interrupt occurred
14:13	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
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									TRGG	Reserved	CH3G	CH2G	CH1G	CH0G	UPG
									w		w	w	w	w	w

Bits	Fields	Descriptions
31	CH3COMADDG	Channel 3 additional compare event generation. Refer to CH0COMADDG description.
30	CH2COMADDG	Channel 2 additional compare event generation. Refer to CH0COMADDG description.
29	CH1COMADDG	Channel 1 additional compare event generation.

		Refer to CH0COMADDG description.
28	CH0COMADDG	Channel 0 additional compare event generation. This bit is set by software to generate a compare event in channel 0 additional, it is automatically cleared by hardware. When this bit is set, the CH0COMADDIF flag will be set, and the corresponding interrupt will be sent if enabled. 0: No generate a channel 0 additional compare event 1: Generate a channel 0 additional compare event **Note:** This bit just used in composite PWM mode(when CH0CPWMEN=1).
27:7	Reserved	Must be kept at reset value.
6	TRGG	Trigger event generation This bit is set by software and cleared by hardware automatically. When this bit is set, the TRGIF flag in TIMERx_INTF register will be set, related interrupt or DMA transfer can occur if enabled. 0: No generate a trigger event 1: Generate a trigger event
5	Reserved	Must be kept at reset value.
4	CH3G	Channel 3 capture or compare event generation Refer to CH0G description
3	CH2G	Channel 2 capture or compare event generation Refer to CH0G description
2	CH1G	Channel 1 capture or compare event generation Refer to CH0G description
1	CH0G	Channel 0 capture or compare event generation This bit is set by software to generate a capture or compare event in channel 0, it is automatically cleared by hardware. When this bit is set, the CH0IF flag will be set, and the corresponding interrupt or DMA request will be sent if enabled. In addition, if channel 0 is configured in input mode, the current value of the counter is captured to TIMERx_CH0CV register, and the CH0OF flag is set if the CH0IF flag has been set. 0: No generate a channel 0 capture or compare event 1: Generate a channel 0 capture or compare event
0	UPG	Update event generation This bit can be set by software, and automatically cleared by hardware. When this bit is set, the counter is cleared if the center-aligned or up counting mode is selected, while in down counting mode it takes the auto-reload value. The prescaler counter is cleared at the same time. 0: No generate an update event

1: Generate an update event

Channel control register 0 (TIMERx_CHCTL0)

Address offset: 0x18

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH1MS[2]	CH0MS[2]	CH1COMADDSEN	CH0COMADDSEN	CH1ADDUPS	CH0ADDUPS	Reserved	CH1COMCTL[3]	Reserved							CH0COMCTL[3]
		Reserved	Reserved	Reserved	Reserved		Reserved								Reserved
rw	rw	rw	rw	rw	rw		rw								rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1COMCEN	CH1COMCTL[2:0]			CH1COMSEN	CH1COMFEN	CH1MS[1:0]		CH0COMCEN	CH0COMCTL[2:0]			CH0COMSEN	CH0COMFEN	CH0MS[1:0]	
CH1CAPFLT[3:0]				CH1CAPPSC[1:0]				CH0CAPFLT[3:0]				CH0CAPPSC[1:0]			
rw				rw		rw		rw				rw		rw	

Output compare mode:

Bits	Fields	Descriptions
31	CH1MS[2]	Channel 1 I / O mode selection Refer to CH1MS[1:0]description
30	CH0MS[2]	Channel 0 I / O mode selection Refer to CH0MS[1:0] description
29	CH1COMADDSEN	Channel 1 additional compare output shadow enable Refer to CH0COMADDSEN description.
28	CH0COMADDSEN	Channel 0 additional compare output shadow enable When this bit is set, the shadow register of TIMERx_CH0COMV_ADD register which updates at each update event will be enabled. 0: Channel 0 additional compare output shadow disabled 1: Channel 0 additional compare output shadow enabled The PWM mode can be used without validating the shadow register only in single pulse mode (SPM bit in TIMERx_CTL0 register is set).
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 CH0COMSEN 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: Delayable SPM mode 0. The behavior of O0CPRE is performed as in PWM mode 0. When counting up, the O0CPRE is active. When a trigger event occurs, the O0CPRE is inactive. The O0CPRE is active again at the next update event; When counting down, the O0CPRE is inactive, when a trigger event occurs, the O0CPRE is active. The O0CPRE is inactive again at the next update event.
1001: Delayable SPM mode 1. The behavior of O0CPRE is performed as in PWM mode 1. When counting up, the O0CPRE is inactive, when a trigger event occurs, the O0CPRE is active. The O0CPRE is inactive again at the next update event; When counting down, the O0CPRE is active. When a trigger event occurs, the O0CPRE is inactive. The O0CPRE is active again at the next update event.
1010 / 1011 / 1100 / 1101 / 1110 / 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.

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. The minimum delay from an edge on the trigger input to activate CH0_O output is 5 clock cycles. 1: Channel 0 output quickly compare enable. The minimum delay from an edge on the trigger input to activate CH0_O output is 3 clock cycles.
1:0	CH0MS[1:0]	Channel 0 I / O mode selection This bit-field specifies the work mode of the channel and the input signal selection. The CH0MS[2:0] bit-field is writable only when the channel is not active (the CH0EN bit in TIMERx_CHCTL2 register is reset). 000: Channel 0 is configured as output. 001: Channel 0 is configured as input, IS0 is connected to CI0FE0. 010: Channel 0 is configured as input, IS0 is connected to CI1FE0. 011: Channel 0 is configured as input, IS0 is connected to ITS, this mode is working only if an internal trigger input is selected (through TSCFG15[4:0] bit-field in SYSCFG_TIMERxCFG2(x=1..4) register). 100~111: Reserved.

Input capture mode:

Bits	Fields	Descriptions
31	CH1MS[2]	Channel 1 I / O mode selection Same as output compare mode.
30	CH0MS[2]	Channel 0 I / O mode selection Same as output compare mode.
29:16	Reserved	Must be kept at reset value.
15:12	CH1CAPFLT[3:0]	Channel 1 input capture filter control Refer to CH0CAPFLT description.
11:10	CH1CAPPSC[1:0]	Channel 1 input capture prescaler Refer to CH0CAPPSC description.
9:8	CH1MS[1:0]	Channel 1 I / O mode selection Same as output compare mode.
7:4	CH0CAPFLT[3:0]	Channel 0 input capture filter control An event counter is used in the digital filter, in which a transition on the output occurs after N input events. This bit-field specifies the frequency used to sample CI0 input signal and the length of the digital filter applied to CI0. 0000: Filter disabled, $f_{SAMP}=f_{DTS}$, N=1. 0001: $f_{SAMP}=f_{CK_TIMER}$, N=2. 0010: $f_{SAMP}= f_{CK_TIMER}$, N=4. 0011: $f_{SAMP}= f_{CK_TIMER}$, N=8.

0100: $f_{SAMP}=f_{DTS}$ / 2, N=6.
0101: $f_{SAMP}=f_{DTS}$ / 2, N=8.
0110: $f_{SAMP}=f_{DTS}$ / 4, N=6.
0111: $f_{SAMP}=f_{DTS}$ / 4, N=8.
1000: $f_{SAMP}=f_{DTS}$ / 8, N=6.
1001: $f_{SAMP}=f_{DTS}$ / 8, N=8.
1010: $f_{SAMP}=f_{DTS}$ / 16, N=5.
1011: $f_{SAMP}=f_{DTS}$ / 16, N=6.
1100: $f_{SAMP}=f_{DTS}$ / 16, N=8.
1101: $f_{SAMP}=f_{DTS}$ / 32, N=5.
1110: $f_{SAMP}=f_{DTS}$ / 32, N=6.
1111: $f_{SAMP}=f_{DTS}$ / 32, N=8.

3:2	CH0CAPPSC[1:0]	Channel 0 input capture prescaler This bit-field specifies the factor of the prescaler on channel 0 input. The prescaler is reset when CH0EN bit in TIMERx_CHCTL2 register is cleared. 00: Prescaler disabled, capture is done on each channel input edge. 01: Capture is done every 2 channel input edges. 10: Capture is done every 4 channel input edges. 11: Capture is done every 8 channel input edges.
1:0	CH0MS[1:0]	Channel 0 I / O mode selection Same as output compare mode.

Channel control register 1 (TIMERx_CHCTL1)

Address offset: 0x1C
Reset value: 0x0000 0000

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

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	Reserved		Reserved								Reserved
rw	rw	rw	rw	rw	rw		rw								rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3COM CEN	CH3COMCTL[2:0]			CH3COM SEN	CH3COM FEN	CH3MS[1:0]		CH2COM CEN	CH2COMCTL[2:0]			CH2COM SEN	CH2COM FEN	CH2MS[1:0]	
CH3CAPFLT[3:0]				CH3CAPPSC[1:0]				CH2CAPFLT[3:0]				CH2CAPPSC[1:0]			
rw				rw		rw		rw				rw		rw	

Output compare mode:

Bits	Fields	Descriptions
31	CH3MS[2]	Channel 3 I / O mode selection Refer to CH3MS[1:0]description.

30	CH2MS[2]	Channel 2 I / O mode selection Refer to CH2MS[1:0] description.
29	CH3COMADDSEN	Channel 3 additional compare output shadow enable Refer to CH2COMADDSEN description.
28	CH2COMADDSEN	Channel 2 additional compare output shadow enable When this bit is set, the shadow register of TIMERx_CH2COMV_ADD register which updates at each update event will be enabled. 0: Channel 2 additional compare shadow disabled 1: Channel 2 additional compare shadow enabled The PWM mode can be used without validating the shadow register only in single pulse mode (SPM bit in TIMERx_CTL0 register is set).
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 CH2COMSEN 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: Delayable SPM mode 0. The behavior of O2CPRE is performed as in PWM mode 0. When counting up, the O2CPRE is active. When a trigger event occurs, the O2CPRE is inactive. The O2CPRE is active again at the next update event; When counting down, the O2CPRE is inactive, when a trigger event occurs, the O2CPRE is active. The O2CPRE is inactive again at the next update event. 1001: Delayable SPM mode 1. The behavior of O2CPRE is performed as in PWM mode 1. When counting up, the O2CPRE is inactive, when a trigger event occurs, the O2CPRE is active. The O2CPRE is inactive again at the next update event;

When counting down, the O2CPRE is active. When a trigger event occurs, the O2CPRE is inactive. The O2CPRE is active again at the next update event.
1010: Programmable pulse output. A programmable pulse output on the CH2_O when the counter matches the output compare register TIMERx_CH2CV. The pulse is configured by OPPSC[2:0] and OPWID[7:0] bit-field in TIMERx_DECCTL register.
1011: Output the DIR bit. The O2CPRE signal indicates the value of the DIR bit (in TIMERx_CTL0 register).
1100 / 1101 / 1110 / 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.

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 PWM0 or PWM1 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. The minimum delay from an edge on the trigger input to activate CH2_O output is 5 clock cycles. 1: Channel 2 output quickly compare enable. The minimum delay from an edge on the trigger input to activate CH2_O output is 3 clock cycles.
1:0	CH2MS[1:0]	Channel 2 I / O mode selection This bit-field specifies the work mode of the channel and the input signal selection. The CH2MS[2:0] bit-field is writable only when the channel is not active (the CH2EN bit in TIMERx_CHCTL2 register is reset). 00: Channel 2 is configured as output. 01: Channel 2 is configured as input, IS2 is connected to CI2FE2. 10: Channel 2 is configured as input, IS2 is connected to CI3FE2. 11: Channel 2 is configured as input, IS2 is connected to ITS. This mode is working only if an internal trigger input is selected (through TSCFG15[4:0] bit-field in

SYSCFG_TIMERxCFG2(x=1...4) register).

100~111: Reserved.

Input capture mode:

Bits	Fields	Descriptions
31	CH3MS[2]	Channel 3 I / O mode selection Same as output compare mode.
30	CH2MS[2]	Channel 2 I / O mode selection Same as output compare mode.
29:16	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. This bit cannot be modified when PROT[1:0] bit-field in TIMERx_CCHP0 register is 11 or 10.
2	Reserved	Must be kept at reset value.
1	CH0P	Channel 0 capture / compare function polarity When channel 0 is configured in output mode, this bit specifies the output signal polarity. 0: Channel 0 active high 1: Channel 0 active low When channel 0 is configured in input mode, this bit specifies the 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. This bit cannot be modified when PROT [1:0] bit-filed in TIMERx_CCHP0 register is 11 or 10.
0	CH0EN	Channel 0 capture / compare function enable When channel 0 is configured in output mode, setting this bit enables CH0_O signal in active state. When channel 0 is configured in input mode, setting this bit enables the capture event in channel0. 0: Channel 0 disabled 1: Channel 0 enabled

Counter register (TIMERx_CNT) (TIMERx, x= 2,3)

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
UPIFBU	Reserved														
r															

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

Bits	Fields	Descriptions
31	UPIFBU	UPIF bit backup This bit is a backup of UPIF bit in TIMERx_INTF register and read-only. This bit is only valid when UPIFBUEN = 1. If the UPIFBUEN =0, this bit is reserved and read the result is 0.
30:16	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. When the PWMADMEN =0, this bit-field indicates the current counter value. Writing to this bit-field can change the value of the counter. When the PWMADMEN =1, this bit-field just indicates the integer part of the counter value, and without the fractional part.

Counter register (TIMERx_CNT) (TIMERx, x= 1,4)

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]	CNT[30:16]														
UPIFBU															
rw / r	rw														

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

UPIFBUEN = 0:

Bits	Fields	Descriptions
31:0	CNT[31:0]	This bit-field indicates the current counter value. Writing to this bit-field can change the value of the counter. When the PWMADMEN =0, this bit-field indicates the current counter value. Writing to this bit-field can change the value of the counter. When the PWMADMEN =1, this bit-field just indicates the integer part of the counter value, and without the fractional part.

UPIFBUEN = 1:

Bits	Fields	Descriptions
31	UPIFBU	UPIF bit backup This bit is a backup of UPIF bit in TIMERx_INTF register, and read-only. This bit is only valid when UPIFBUEN = 1. If the UPIFBUEN =0, this bit is reserved and read the result is 0.
30	CNT[30:0]	This bit-field indicates the current counter value. Writing to this bit-field can change the value of the counter. When the PWMADMEN =0, this bit-field indicates the current counter value. Writing to this bit-field can change the value of the counter. When the PWMADMEN =1, this bit-field just indicates the integer part of the counter value, and without the fractional part.

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) (TIMERx, x= 2,3)

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
CARL[19:16]				Reserved											
rw															

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

CARL[15:0]
rw

Bits	Fields	Descriptions
31:28	CARL[19:16]	Counter auto reload value (bit 16 to 19) When the PWMADMEN =0, CARL[19:16] bit-field is 0000. When the PWMADMEN =1, CARL[19:16] bit-field specifies fractional part of the auto reload value.
27:16	Reserved	Must be kept at reset value.
15:0	CARL[15:0]	Counter auto reload value (bit 0 to 15) When the PWMADMEN =0, CARL[15:0] bit-field specifies the auto reload value of the counter. When the PWMADMEN =1, CARL[15:0] bit-field specifies integer part of the auto reload value.

Counter auto reload register (TIMERx_CAR) (TIMERx, x= 1,4)

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
31:0	CARL[31:0]	Counter auto reload value When the PWMADMEN =0, CARL[31:0] bit-field specifies the auto reload value of the counter. When the PWMADMEN =1, CARL[27:0] bit-field specifies integer part of the auto reload value, CARL[31:28] bit-field specifies fractional part of the auto reload value.

Channel 0 capture / compare value register (TIMERx_CH0CV) (TIMERx, x= 2,3)

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[19:16]				Reserved											

rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH0VAL[15:0]															

rw

Bits	Fields	Descriptions
31:28	CH0VAL[19:16]	Capture / compare value of channel 0 (bit 16 to 19) When channel 0 is configured in input mode, CH0VAL[19:16] bit-field is 0000. 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. When the PWMADMEN = 0, CH0VAL[19:16] bit-field is 0000. When the PWMADMEN =1, CH0VAL[19:16] bit-field specifies the fractional part.
27:16	Reserved	Must be kept at reset value.
15:0	CH0VAL[15:0]	Capture / compare value of channel 0 (bit 0 to 15) When channel 0 is configured in input mode, CH0VAL[15:0] 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. When the PWMADMEN = 0, CH0VAL[15:0] bit-field specifies the compare value. When the PWMADMEN =1, CH0VAL[15:0] bit-field specifies integer part of the compare value.

Channel 0 capture / compare value register (TIMERx_CH0CV) (TIMERx, x= 1,4)

Address offset: 0x34

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH0VAL[31:16]															

rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH0VAL[15:0]															

rw

Bits	Fields	Descriptions
31:0	CH0VAL[31:0]	Capture / compare value of channel 0 When channel 0 is configured in input mode, CH0VAL[31:28] bit-field is 0000, CH0VAL[27:0] 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.

When the PWMADMEN =0, CH0VAL [31:0] bit-field specifies the compare value of the counter.

When the PWMADMEN =1, CH0VAL [27:0] bit-field specifies integer part of the compare value, CH0VAL [31:28] bit-field specifies fractional part of the compare value.

Channel 1 capture / compare value register (TIMERx_CH1CV) (TIMERx, x= 2,3)

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[19:16]				Reserved											
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1VAL[15:0]															
rw															

Bits	Fields	Descriptions
31:28	CH1VAL[19:16]	Capture / compare value of channel 1 (bit 16 to 19) When channel 1 is configured in input mode, CH1VAL[19:16] bit-field is 0000. 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. When the PWMADMEN = 0, CH1VAL[19:16] bit-field is 0000. When the PWMADMEN =1, CH1VAL[19:16] bit-field specifies the fractional part.
27:16	Reserved	Must be kept at reset value.
15:0	CH1VAL[15:0]	Capture / compare value of channel 1 (bit 0 to 15) When channel 1 is configured in input mode, CH1VAL[15:0] 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. When the PWMADMEN = 0, CH1VAL[15:0] bit-field specifies the compare value. When the PWMADMEN =1, CH1VAL[15:0] bit-field specifies integer part of the compare value.

Channel 1 capture / compare value register (TIMERx_CH1CV) (TIMERx, x= 1,4)

Address offset: 0x38

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH1VAL[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1VAL[15:0]															
rw															

Bits	Fields	Descriptions
31:0	CH1VAL[31:0]	Capture / compare value of channel 1 When channel 1 is configured in input mode, CH1VAL[31:28] bit-field is 0000, CH1VAL[27:0] 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. When the PWMADMEN =0, CH1VAL [31:0] bit-field specifies the compare value of the counter. When the PWMADMEN =1, CH1VAL [27:0] bit-field specifies integer part of the compare value, CH1VAL [31:28] bit-field specifies fractional part of the compare value.

Channel 2 capture / compare value register (TIMERx_CH2CV) (TIMERx, x= 2,3)

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[19:16]				Reserved											
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH2VAL[15:0]															
rw															

Bits	Fields	Descriptions
31:28	CH2VAL[19:16]	Capture / compare value of channel 2 (bit 16 to 19) When channel 2 is configured in input mode, CH2VAL[19:16] bit-field is 0000. 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. When the PWMADMEN = 0, CH2VAL[19:16] bit-field is 0000.

		When the PWMADMEN =1, CH2VAL[19:16] bit-field specifies the fractional part.
27:16	Reserved	Must be kept at reset value.
15:0	CH2VAL[15:0]	Capture / compare value of channel 2 (bit 0 to 15) When channel 2 is configured in input mode, CH2VAL[15:0] 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. When the PWMADMEN = 0, CH2VAL[15:0] bit-field specifies the compare value. When the PWMADMEN =1, CH2VAL[15:0] bit-field specifies integer part of the compare value.

Channel 2 capture / compare value register (TIMERx_CH2CV) (TIMERx, x= 1,4)

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
31:0	CH2VAL[31:0]	Capture / compare value of channel 2 When channel 2 is configured in input mode, CH2VAL[31:28] bit-field is 0000, CH2VAL[27:0] 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. When the PWMADMEN =0, CH2VAL [31:0] bit-field specifies the compare value of the counter. When the PWMADMEN =1, CH2VAL [27:0] bit-field specifies integer part of the compare value, CH2VAL [31:28] bit-field specifies fractional part of the compare value.

Channel 3 capture / compare value register (TIMERx_CH3CV) (TIMERx, x= 2,3)

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[19:16]				Reserved											
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3VAL[15:0]															
rw															

Bits	Fields	Descriptions
31:28	CH3VAL[19:16]	Capture / compare value of channel 3 (bit 16 to 19) When channel 3 is configured in input mode, CH3VAL[19:16] bit-field is 0000. 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. When the PWMADMEN = 0, CH3VAL[19:16] bit-field is 0000. When the PWMADMEN =1, CH3VAL[19:16] bit-field specifies the fractional part.
27:16	Reserved	Must be kept at reset value.
15:0	CH3VAL[15:0]	Capture / compare value of channel 3 (bit 0 to 15) When channel 3 is configured in input mode, CH3VAL[15:0] 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. When the PWMADMEN = 0, CH3VAL[15:0] bit-field specifies the compare value. When the PWMADMEN =1, CH3VAL[15:0] bit-field specifies integer part of the compare value.

Channel 3 capture / compare value register (TIMERx_CH3CV) (TIMERx, x= 1,4)

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
31:0	CH3VAL[31:0]	Capture / compare value of channel 3

When channel 0 is configured in input mode, CH3VAL[31:28] bit-field is 0000, CH3VAL[27:0] 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.

When the PWMADMEN =0, CH3VAL [31:0] bit-field specifies the compare value of the counter.

When the PWMADMEN =1, CH3VAL [27:0] bit-field specifies integer part of the compare value, CH3VAL [31:28] bit-field specifies fractional part of the compare value.

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
CH0COMVAL_ADD[31:16]															
rw															

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	CH0COMVAL_ADD [31:16]	Additional compare value of channel 0 (bit 16 to 31) This bit-field only for TIMER1 / 4.
15:0	CH0COMVAL_ADD [15:0]	Additional compare value of channel 0 (bit 0 to 15) 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
CH1COMVAL_ADD[31:16]															
rw															

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	CH1COMVAL_ADD [31:16]	Additional compare value of channel 1 (bit 16 to 31) This bit-field only for TIMER1 / 4.
15:0	CH1COMVAL_ADD [15:0]	Additional compare value of channel 1 (bit 0 to 15) 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).

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
CH2COMVAL_ADD[31:16]															
rw															

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	CH2COMVAL_ADD [31:16]	Additional compare value of channel 2 (bit 16 to 31) This bit-field only for TIMER1 / 4.
15:0	CH2COMVAL_ADD [15:0]	Additional compare value of channel 2 (bit 0 to 15) 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 CH0CPWMEN=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
CH3COMVAL_ADD[31:16]															

rw															
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	CH3COMVAL_ADD [31:16]	Additional compare value of channel 3 (bit 16 to 31) This bit-field only for TIMER1 / 4.
15:0	CH3COMVAL_ADD [15:0]	Additional compare value of channel 3 (bit 0 to 15) 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 CH0CPWMEN=1).

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
CH3C PWMEN	CH2C PWMEN	CH1C PWMEN	CH0C PWMEN	Reserved								DECDISD EN	DECJDEN	Reserved	
rw	rw	rw	rw									rw	rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3OMPSEL[1:0]		CH2OMPSEL[1:0]		CH1OMPSEL[1:0]		CH0OMPSEL[1:0]		Reserved							
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

27:20	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
17:16	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.

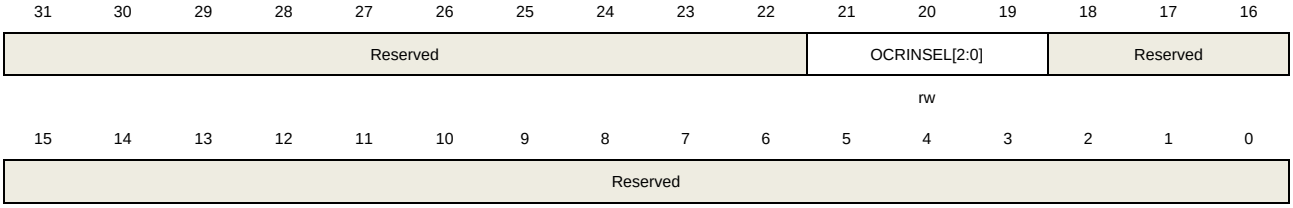
		10: Only when the counter is counting down, the O1CPRE signal is output a pulse when the match events occurs, and the pulse width is one CK_TIMER clock cycle. 11: Both when the counter is counting up and counting down, the O1CPRE signal is output a pulse when the match events occurs, and the pulse width 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.
7:0	Reserved	Must be kept at reset value.

TIMER alternate function control register 1 (TIMERx_AFCTL1)

Address offset: 0x90

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:22	Reserved	must be kept at reset value
21:19	OCRINSEL[2:0]	OCPRE_CLR inputs selection 000: OCPRE_CLR0 001: OCPRE_CLR1 … 111: OCPRE_CLR7

OCPRE_CLR inputs selection	TIMER1 / 2	TIMER3 / 4
OCPRE_CLR0	CMP0_OUT	Reserved
OCPRE_CLR1	CMP1_OUT	Reserved

OCPRE_CLR2	CMP2_OUT	Reserved
OCPRE_CLR3	CMP3_OUT	Reserved
OCPRE_CLR4	CMP4_OUT	Reserved
OCPRE_CLR5	CMP5_OUT	Reserved
OCPRE_CLR6	CMP6_OUT	Reserved
OCPRE_CLR7	CMP7_OUT	Reserved

18:0	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).

Decoder control register (TIMERx_DECCTL)

Address offset: 0x0A0

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved					OPPSC[2:0]			OPWID[7:0]							
					rw			rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								INDP[1:0]		FINDRST	Reserved		INDRSTDIR[1:0]		INDRSTEN
								rw		rw			rw		rw

Bits	Fields	Descriptions

31:27	Reserved	Must be kept at reset value.
26:24	OPPSC[2:0]	Output pulse prescaler This bit-field specifies the clock prescaler for the pulse generator. $t_{opclk} = (2^{(OPPSC\ [2:0])}) \times t_{CK_TIMER}$
23:16	OPWID[7:0]	Output pulse width This bit-field specifies the pulse width. t_{opw} = OPWID [7:0] x t_{opclk}
15:8	Reserved	Must be kept at reset value.
7:6	INDP[1:0]	Index positioning This bit-field indicates in which type of the AB inputs the index event will reset the counter, in quadrature decoder modes 0~4. 00: When AB inputs are 00, index event will reset the counter 01: When AB inputs are 01, index event will reset the counter 10: When AB inputs are 10, index event will reset the counter 11: When AB inputs are 11, index event will reset the counter This bit-field indicates in which type of the AB inputs the index event will reset the counter, in decoder modes 0~3. x0: When clock is low, index event will reset the counter x1: When clock is high, index event will reset the counter **Note**: In decoder modes 0~3, INDP[1] bit is no use.
5	FINDRST	First index signal reset the counter 0: All the index signals are can reset the counter 1: Only the first index signal is active and can reset the counter
4:3	Reserved	Must be kept at reset value.
2:1	INDRSTDIR[1:0]	Index signal reset counter direction This bit-field indicates in which counter direction the index signal can reset the counter. 00: Index signal resets the counter when it counts up and count down 01: Index signal resets the counter only when it counts up 10: Index signal resets the counter only when it counts down 11: Reserved **Note**: The INDRSTDIR[1:0] bit-field can be modified only when INDRSTEN =0.
0	INDRSTEN	Index signal reset enable 0: Index signal resets the counter disabled 1: Index signal resets the counter enabled

DMA configuration register (TIMERx_DMACFG)

Address offset: 0xE0

Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value.
13:8	DMATC[5:0]	DMA transfer count This field defines the times of accessing (R / W) the TIMERx_DMATB register by DMA. 6’b000000: transfer 1 time 6’b000001: transfer 2 times … 6’b111000: transfer 57 times
7:6	Reserved	Must be kept at reset value.
5:0	DMATA[5:0]	DMA transfer access start address This field defines the start address of accessing the TIMERx_DMATB register by DMA. When the first access to the TIMERx_DMATB register is done, this bit-field specifies the address just accessed. And then the address of the second access to the TIMERx_DMATB register will be (start address + 0x4). 6’b000000: TIMERx_CTL0 6’b000001: TIMERx_CTL1 … In a word: start address = TIMERx_CTL0 + DMATA*4

DMA transfer buffer register (TIMERx_DMATB)

Address offset: 0xE4

Reset value: 0x0000 0000

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

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

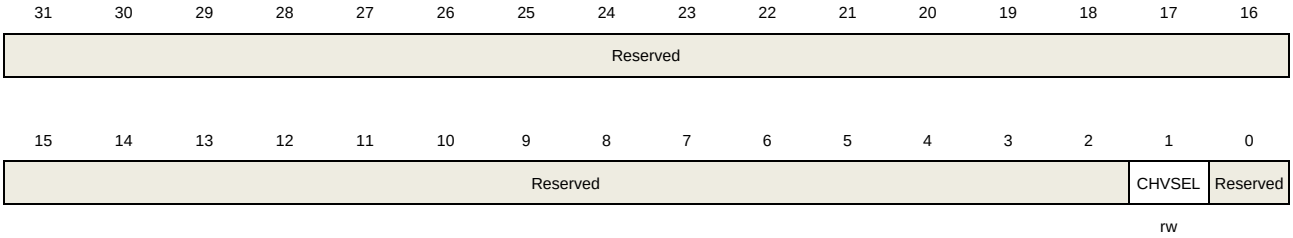
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMATB[15:0]															
rw															

Bits	Fields	Descriptions
31:0	DMATB[31:0]	DMA transfer buffer When a read or write operation is assigned to this register, the register located at the address ranges from (start address) to (start address + transfer count * 4) will be accessed. The transfer count is calculated by hardware, and ranges from 0 to DMATC.

Configuration register (TIMERx_CFG)

Address offset: 0xFC
Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	CHVSEL	Write CHxVAL register selection bit This bit-field is set and reset by software. 1: If the value to be written to the CHxVAL register is the same as the value of CHxVAL register, the write access is ignored. 0: No effect.
0	Reserved	Must be kept at reset value.

23.3. General level3 timer (TIMERx, x=14)

23.3.1. Overview

The general level3 timer module (TIMER14) 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.

23.3.2. Characteristics

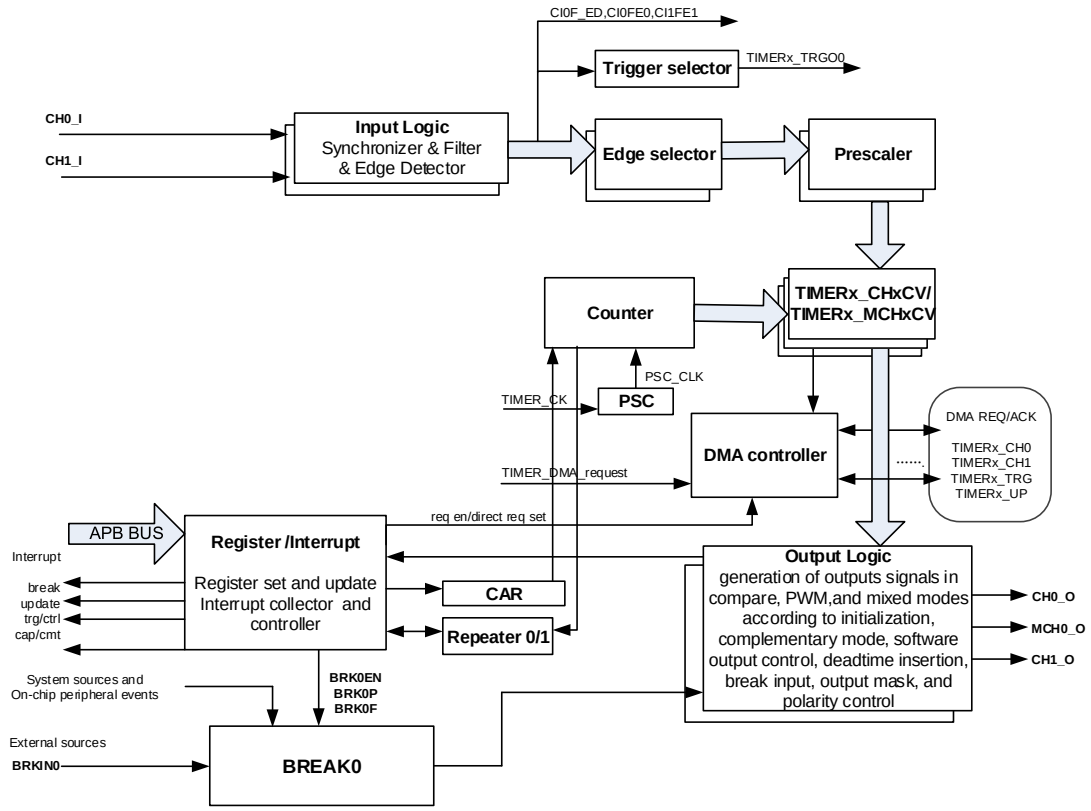
- Total channel num: 3.
- Counter width: 16-bit.
- Source of counter clock is selectable: internal clock, internal trigger, external input.
- Counter modes: count up only.
- Programmable prescaler: 16-bit. The factor can be changed on the go.
- Each channel is user-configurable: input capture mode, output compare mode, programmable PWM mode, single pulse mode.
- Programmable dead time insertion.
- Auto reload function.
- Programmable counter repetition function.
- Break input 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.

23.3.3. Block diagram

Figure 23-92. General level3 timer block diagram provides details of the internal

configuration of the general level3 timer.

Figure 23-92. General level3 timer block diagram



23.3.4. Function overview

Clock selection

The general level3 timer has the capability of being clocked by either the TIMER_CK or an alternate clock source controlled by TSCFGy[4:0] (y=3..7,15) in SYSCFG_TIMERxCFG(x=14) registers.

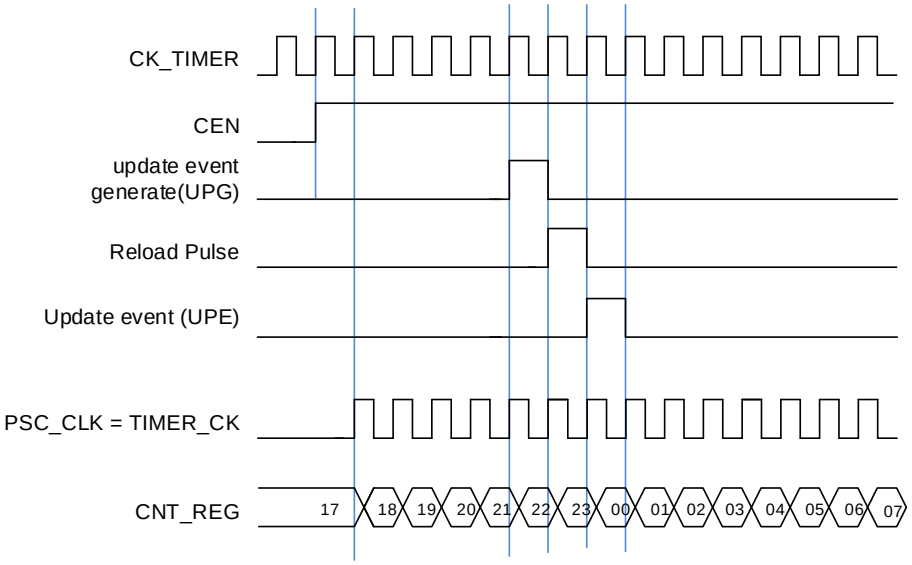
6) TSCFGy[4:0] (y=3..7,15) = 5'b00000 in SYSCFG_TIMERxCFG(x=14) registers.
Internal clock CK_TIMER is selected as timer clock source which is from module RCU.

The default clock source is the CK_TIMER for driving the counter prescaler when TSCFGy[4:0] (y=3..7,15) = 5'b00000 in SYSCFG_TIMERxCFG(x=14) 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=14) 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=3,4,5,7) are setting to an available value, the internal clock TIMER_CK is the counter prescaler driving clock source.

Figure 23-93. Normal mode, internal clock divided by 1



7) TSCFG6[4:0] are setting to a nonzero value (external clock mode 0). External input pin is selected as timer clock source

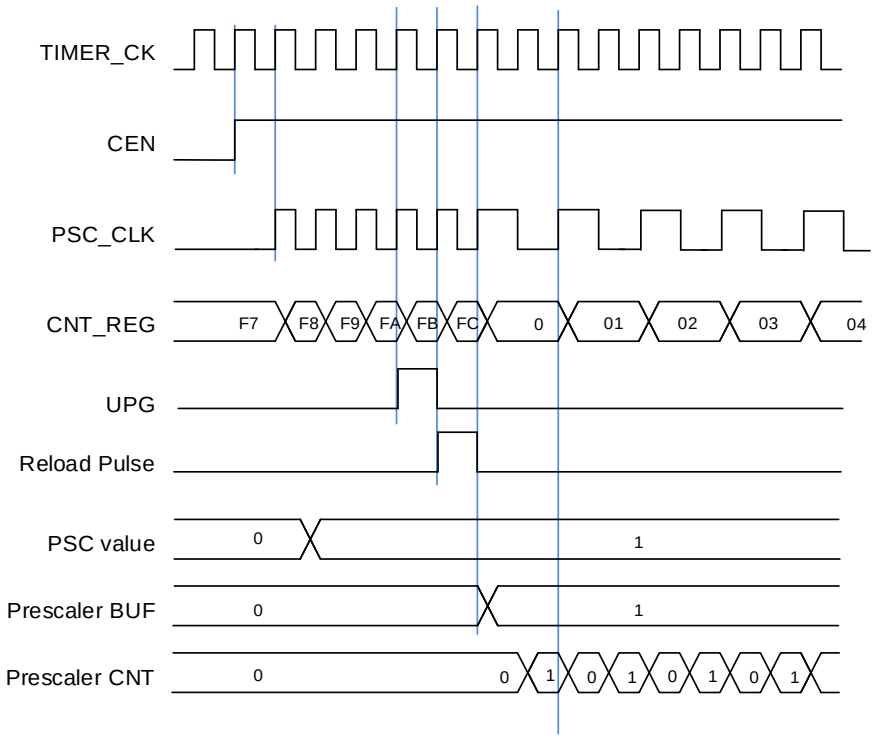
The TIMER_CK, which drives counter's prescaler to count, can be triggered by the event of rising or falling edge on the external pin CI0 / CI1. This mode can be selected by setting TSCFG6[4:0] to 0x5~0x7.

And, the counter prescaler can also be driven by rising edge on the internal trigger input pin ITI0 ~ 10 / ITI14. This mode can be selected by setting TSCFG6[4:0] to 0x9 ~ 0xF or 0x13.

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 23-94. 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+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 23-95. Timing diagram of up counting mode, PSC=0 / 2 and *Figure 23-96. 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 23-95. Timing diagram of up counting mode, PSC=0 / 2

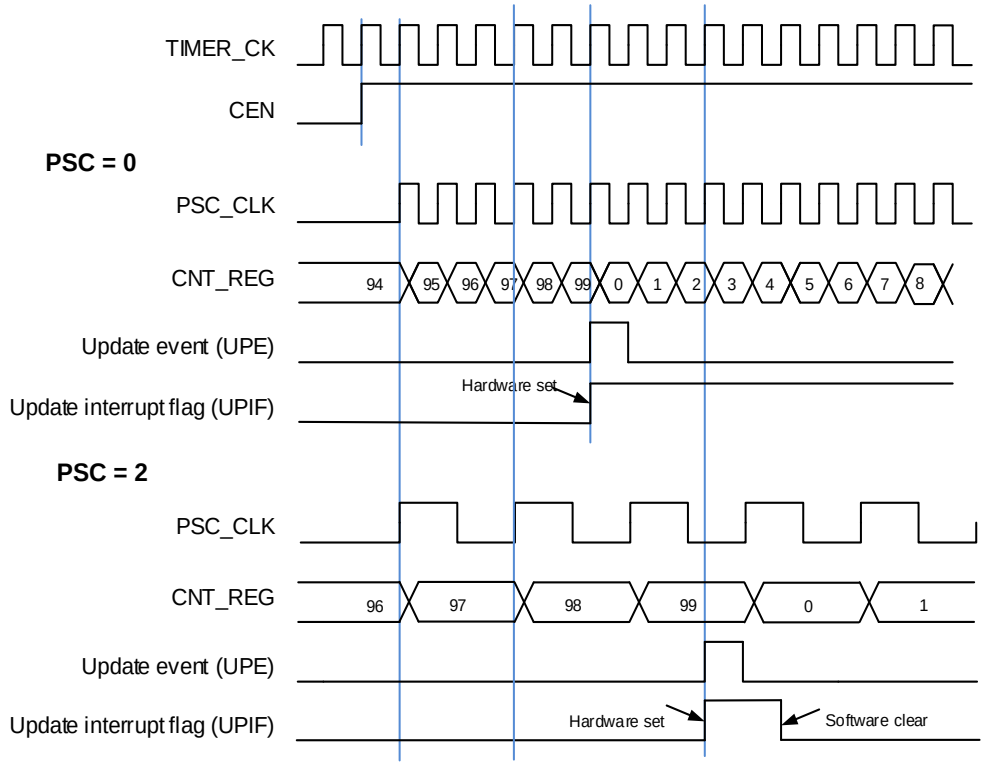
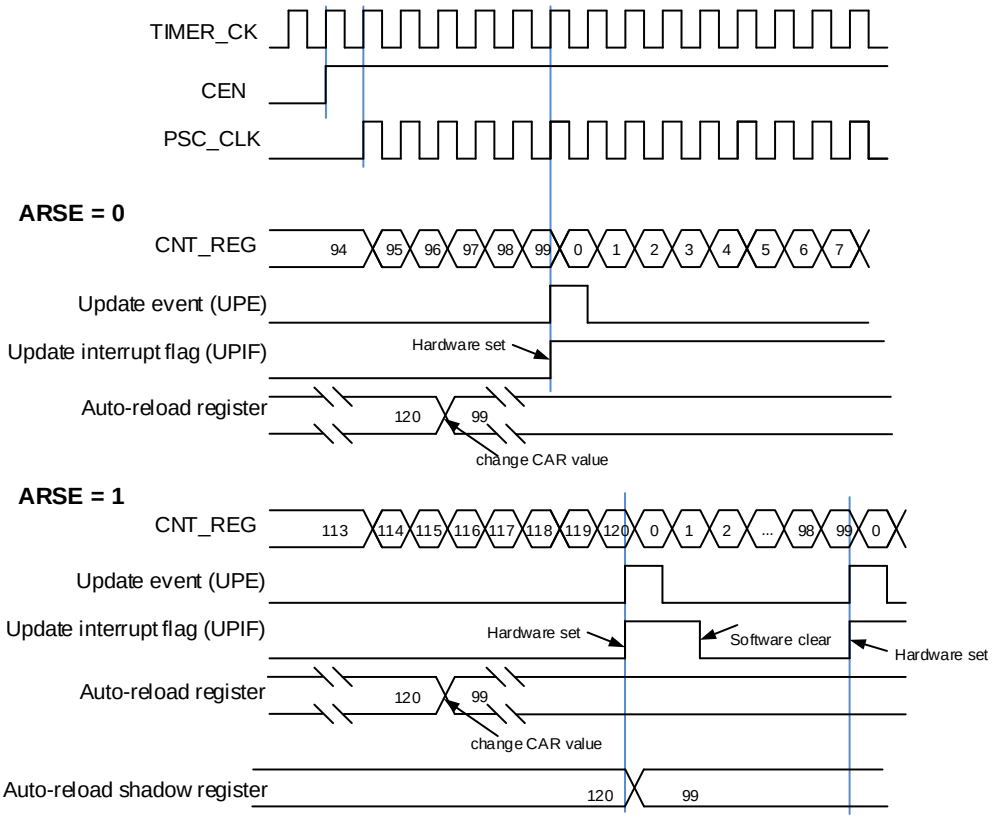


Figure 23-96. Timing diagram of up counting mode, change TIMERx_CAR on the go



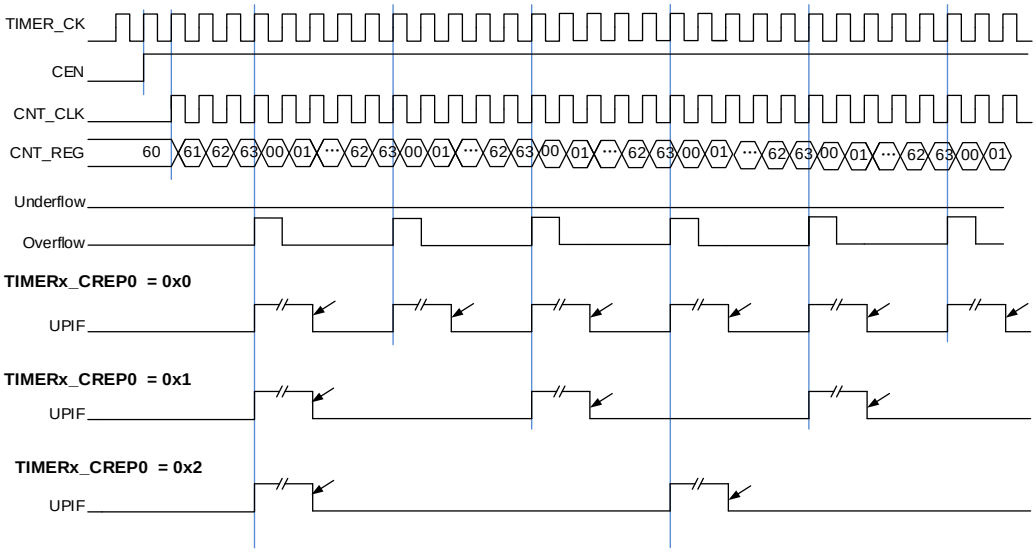
Counter repetition

The general timer has two repetitions counter TIMERx_CREP0 / 1, which can be selected by configuring the CPERSEL bit in the TIMERx_CFG register. The CPEP[7:0] bit-field is 8bits, the CPEP[31:0] bit-field is 32bits and can be read on the fly.

Counter repetition is used to generator update event or updates the timer registers only after a given number (N+1) of cycles of the counter, where N is CREP0 / 1 in TIMERx_CREP0 / 1 register. The repetition counter is decremented at each counter overflow in up-counting mode.

Setting the UPG bit in the TIMERx_SWEVG register will reload the content of CREP0 / 1 in TIMERx_CREP0 / 1 register and generator an update event.

Figure 23-97. Repetition timechart for up-counter



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.

8) 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_MCHxCV(x=0, 1) registers, at the same time the CHxIF / MCHxIF(x=0, 1) bits are set and the channel interrupt is generated if it is enabled when CHxIE / MCHxIE =1(x=0, 1).

Figure 23-98. Input capture logic for channel 0

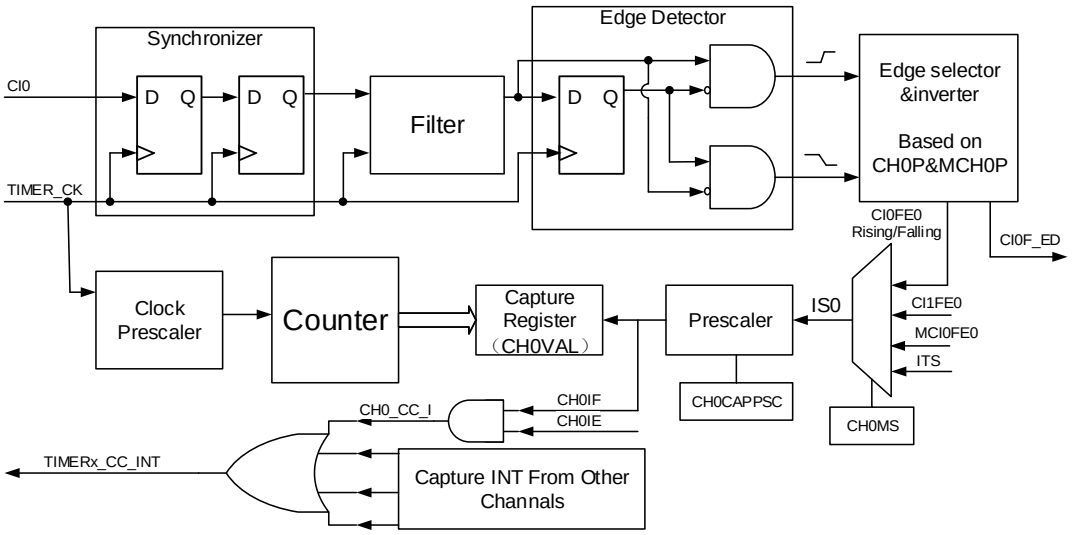
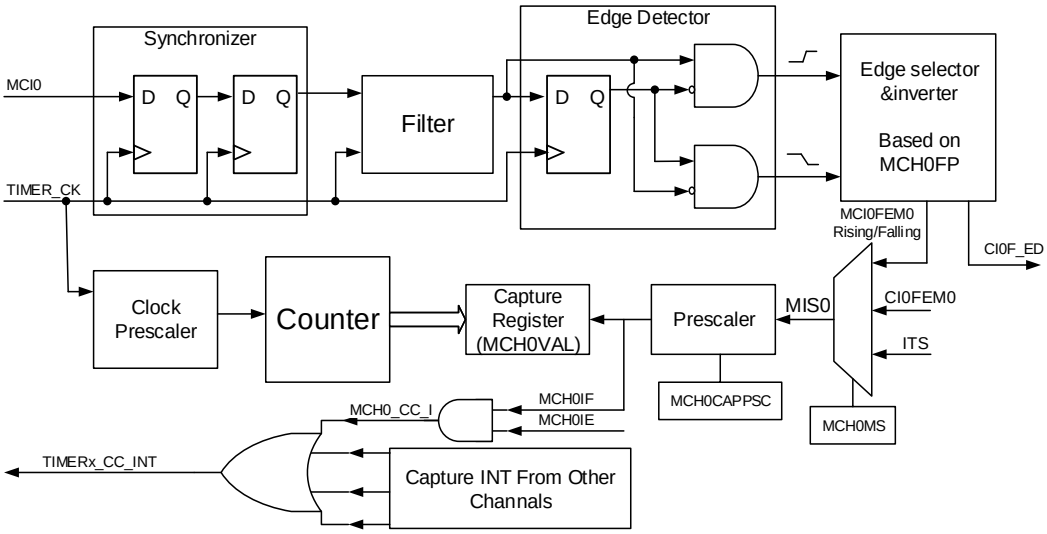


Figure 23-99. Input capture logic for multi mode channel 0



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.

9) Output compare mode

Figure 23-100. Output compare logic (when MCHxMSEL = 2'b00, x=0), *Figure 23-101. Output compare logic (when MCHxMSEL = 2'b11, x=0)* and *Figure 23-102. Output compare logic (x=1)* show the logic circuit of output compare mode.

Figure 23-100. Output compare logic (when MCHxMSEL = 2'b00, x=0)

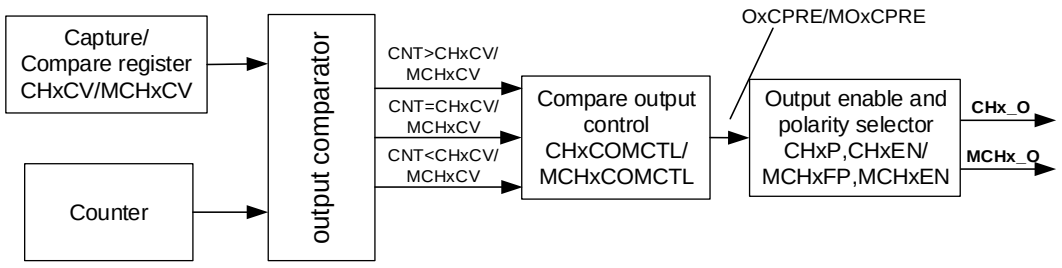


Figure 23-101. Output compare logic (when MCHxMSEL = 2'b11, x=0)

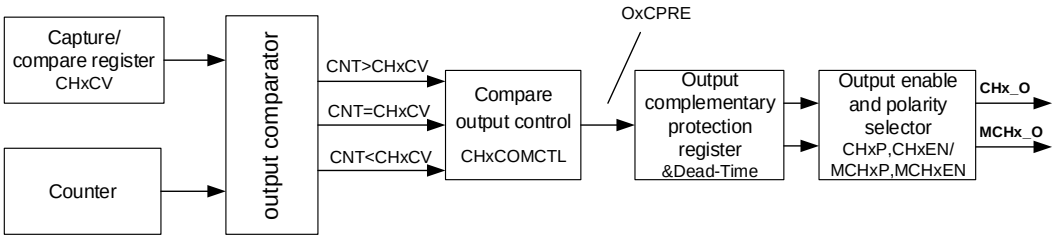
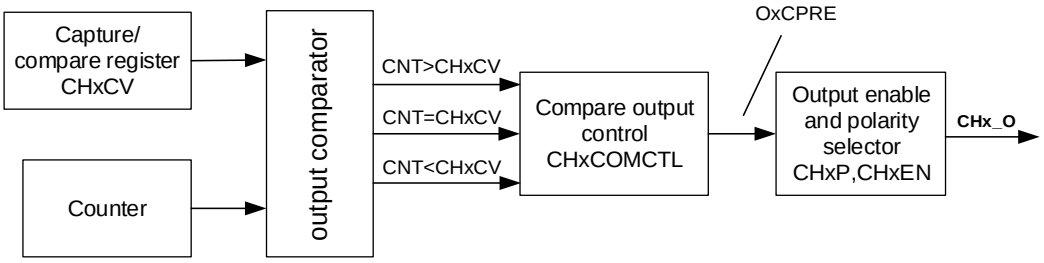


Figure 23-102. Output compare logic (x=1)



The relationship between the channel output signal CHx_O / MCHx_O and the OxCPRE / MOxCPRE signal (more details refer to ***Clear the 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 23-100. Output compare logic (when MCHxMSEL = 2'b00, 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 23-101. Output compare logic (when MCHxMSEL = 2'b11, x=0)***.

For examples (the MCHx_O output is independent from the CHx_O output):

3) 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.

4) 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_CCHP0 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.

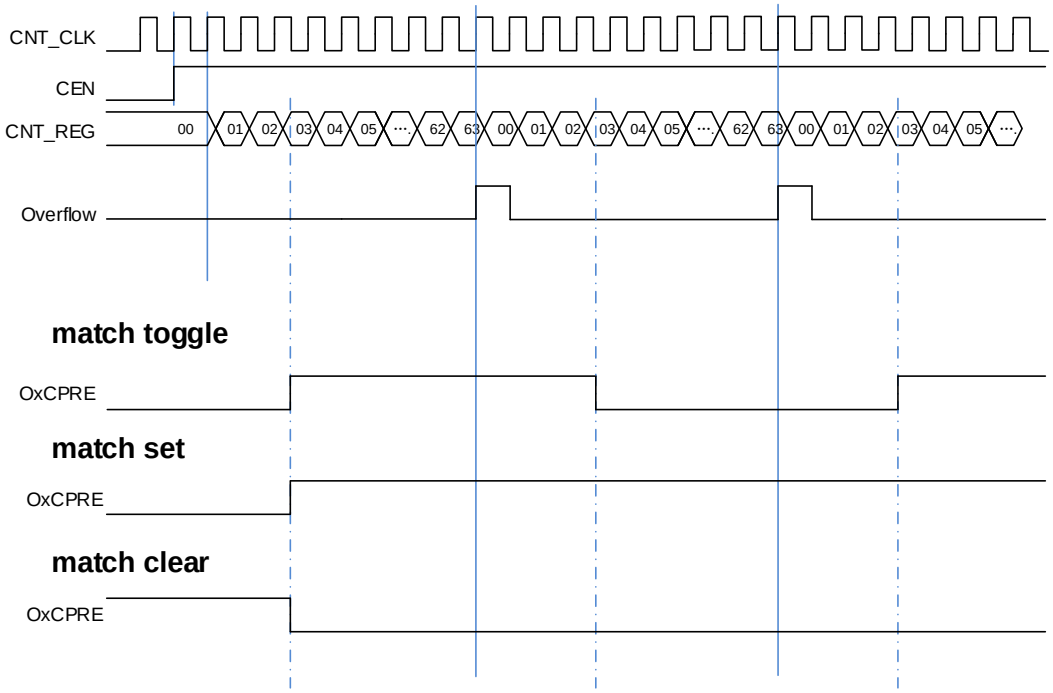
So, the process can be divided into several steps as below:

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 ongoing to meet the expected waveform.
Step5: Start the counter by configuring CEN to 1.

Figure 23-103. Output-compare in three modes shows the three compare modes: toggle/set/clear. CARL=0x63, CHxVAL=0x3.

Figure 23-103. Output-compare in three modes



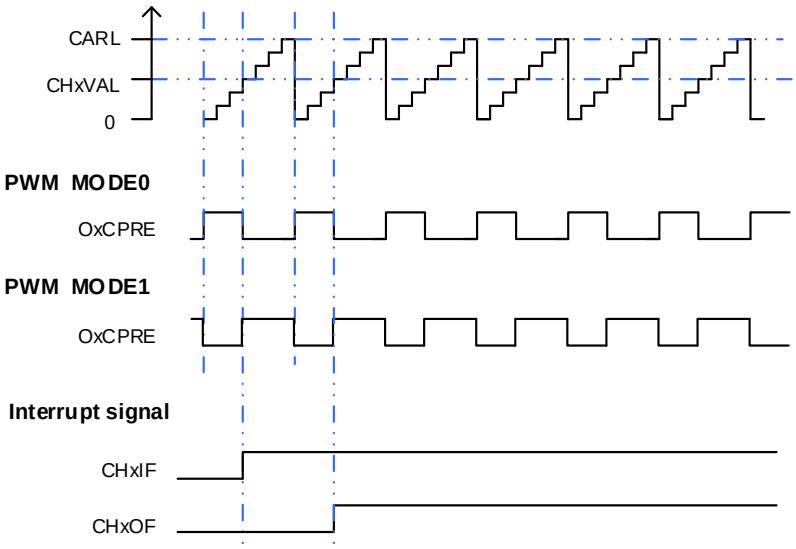
PWM mode

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.

The EAPWM's period is determined by TIMERx_CAR and the duty cycle is determined by TIMERx_CHxCV / TIMERx_MCHxCV. ***Figure 23-104. PWM mode timechart*** shows the EAPWM output and interrupts waveform.

In up counting mode, if the value of TIMERx_CHxCV / TIMERx_MCHxCV is greater than the value of TIMERx_CAR, the output will be always active in PWM mode 0 (CHxCOMCTL / MCHxCOMCTL =4'b0110). And if the value of TIMERx_CHxCV / TIMERx_MCHxCV is greater than the value of TIMERx_CAR, the output will be always inactive in PWM mode 1 (CHxCOMCTL / MCHxCOMCTL =4'b0111).

Figure 23-104. PWM mode timechart

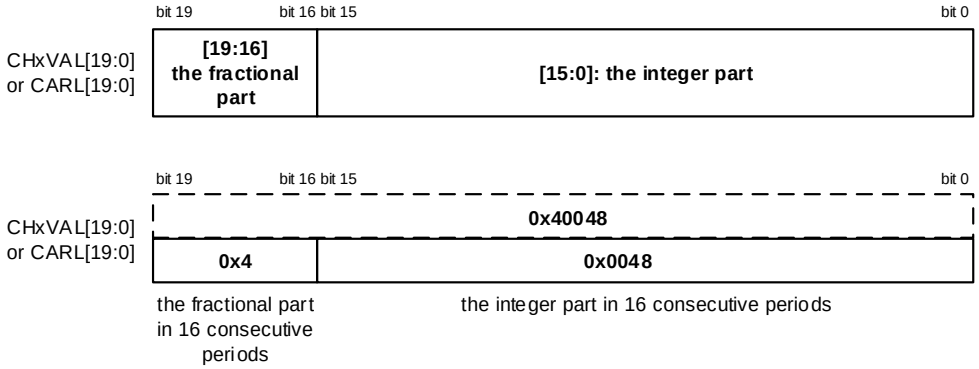


Adjustment mode

Adjustment mode is enabled by setting ADMEN bit in TIMERx_CTL0 register to 1, and this mode can improve the effective resolution of the output PWM wave. The duty cycle resolution can be improved by the CHxVAL[19:0] bit-field in the TIMERx_CHxCV register, and the PWM frequency resolution can be improved by the CARL[19:0] bit-field in the TIMERx CAR register.

When the adjustment mode is enabled, the low 16 bits CHxVAL[15:0] bit-field and CARL[15:0] bit-field are used for the integer part and the high 4 bits CHxVAL[19:16] and CARL[19:16] are used for the fractional part. By adjust the CHxVAL or CARL values over 16 consecutive periods (no more than one TIMER clock cycle at a time) in a predefined way, can increase 16-fold in resolution.

Figure 23-105. Adjustment mode: Data format and the register bit-field



Depending on the configuration of the ADMEN bit (set or clear), the CHxVAL and CARL bit-field are automatically updated. To clear the ADMEN bit, must follow the following steps:

6. CEN bit and ARSE bits must be cleared;
7. CARL[19:16] bit-field must be cleared;

8. ADMEN bit must be cleared;
9. CHxIF bit must be cleared;
10. Set the CEN bit to 1.

The following formula to calculate the PWM Resolution:

$$Resolution = f_{PSC_CLK}/f_{pwm} \quad (23\text{-}9)$$

According to Equation ***(23-9)***, when the adjustment mode is disabled (ADMEN=0), the PWM minimum frequency f_{pwm}:

$$(f_{pwm})_{min} = f_{PSC_CLK}/65536 \quad (23\text{-}10)$$

When the adjustment mode is enabled (ADMEN=1),

$$(f_{pwm})_{min} = f_{PSC_CLK}/(65535+15/16) \quad (23\text{-}11)$$

When the adjustment mode is enabled, the max values of the CHxVAL[19:0] bit-field and CARL[19:0] bit-field are 0xFFFFE (the integer part is 0xFFFE, the fractional part is 0xF)。

The changes of duty cycle and period within 16 consecutive periods are shown in ***Figure 23-106. PWM adjustment mode schematic diagram*** and ***Table 23-17. CHxVAL and CARL bit-field change in edge-aligned***.

Figure 23-106. PWM adjustment mode schematic diagram

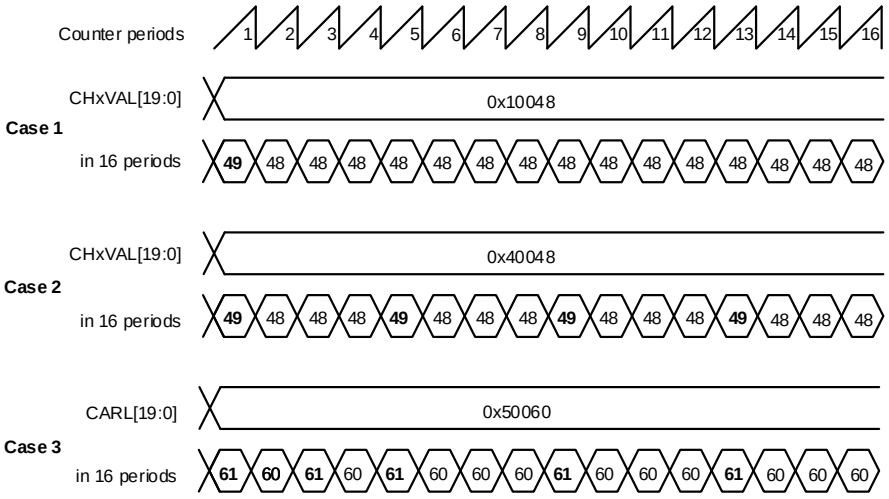


Table 23-17. CHxVAL and CARL bit-field change in edge-aligned

CHxVAL[19:16] / CARL[19:16]	Period															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0000	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0001	+1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0010	+1	-	-	-	-	-	-	-	+1	-	-	-	-	-	-	-
0011	+1	-	-	-	+1	-	-	-	+1	-	-	-	-	-	-	-
0100	+1	-	-	-	+1	-	-	-	+1	-	-	-	+1	-	-	-

CHxVAL[19:16] / CARL[19:16]	Period															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0101	+1	-	+1	-	+1	-	-	-	+1	-	-	-	+1	-	-	-
0110	+1	-	+1	-	+1	-	-	-	+1	-	+1	-	+1	-	-	-
0111	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	-	-
1000	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-
1001	+1	+1	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-
1010	+1	+1	+1	-	+1	-	+1	-	+1	+1	+1	-	+1	-	+1	-
1011	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-	+1	-	+1	-
1100	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-
1101	+1	+1	+1	+1	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-
1110	+1	+1	+1	+1	+1	+1	+1	-	+1	+1	+1	+1	+1	+1	+1	-
1111	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	-

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.

The PWM period is determined by (CARL + 0x0001) and the PWM pulse width is determined by the following table.

Table 23-18.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	0%

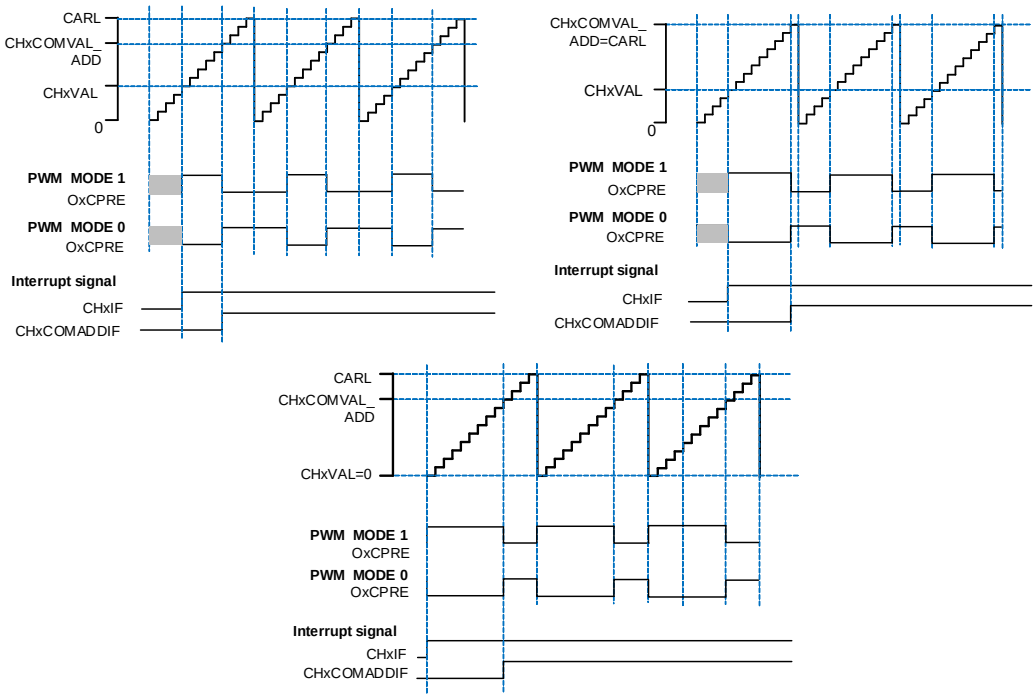
Condition	Mode	PWM pulse width
	PWM mode 1 (up counting)	
CHxCOMVAL_ADD > CARL > CHxVAL	PWM mode 0(up counting) or PWM mode 1(down counting)	0%
	PWM mode 0(down counting) or PWM mode 1(up counting)	100%
(CHxVAL>CARL) and (CHxCOMVAL_ADD > CARL)	-	The output of CHx_O is keeping

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:

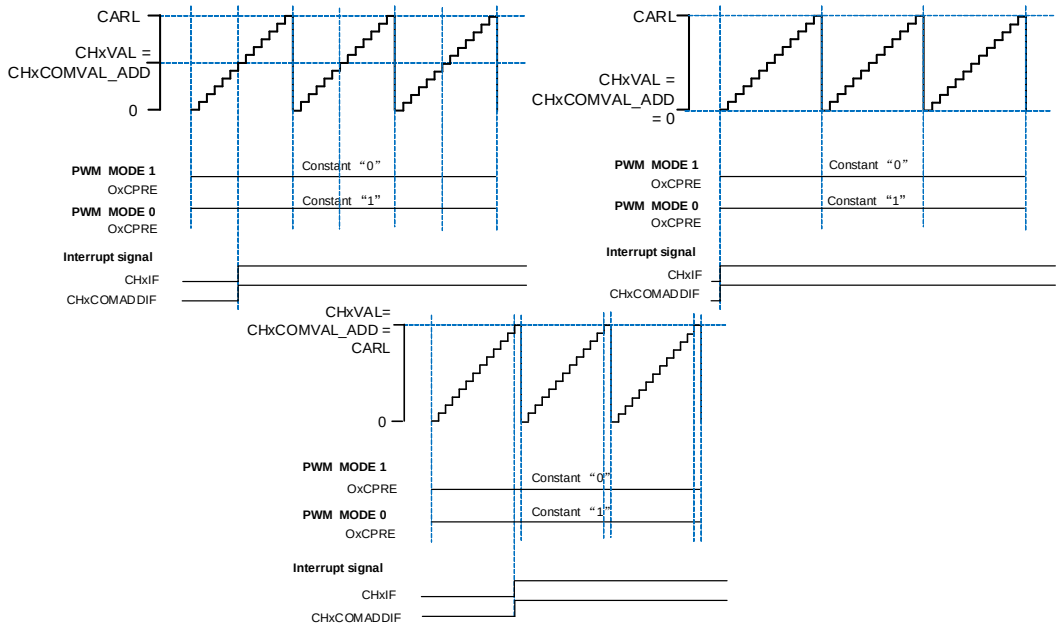
- CHxVAL < CHxCOMVAL_ADD, and the values of CHxVAL and CHxCOMVAL_ADD between 0 and CARL.

Figure 23-107. Channel x output PWM with (CHxVAL < CHxCOMVAL_ADD)



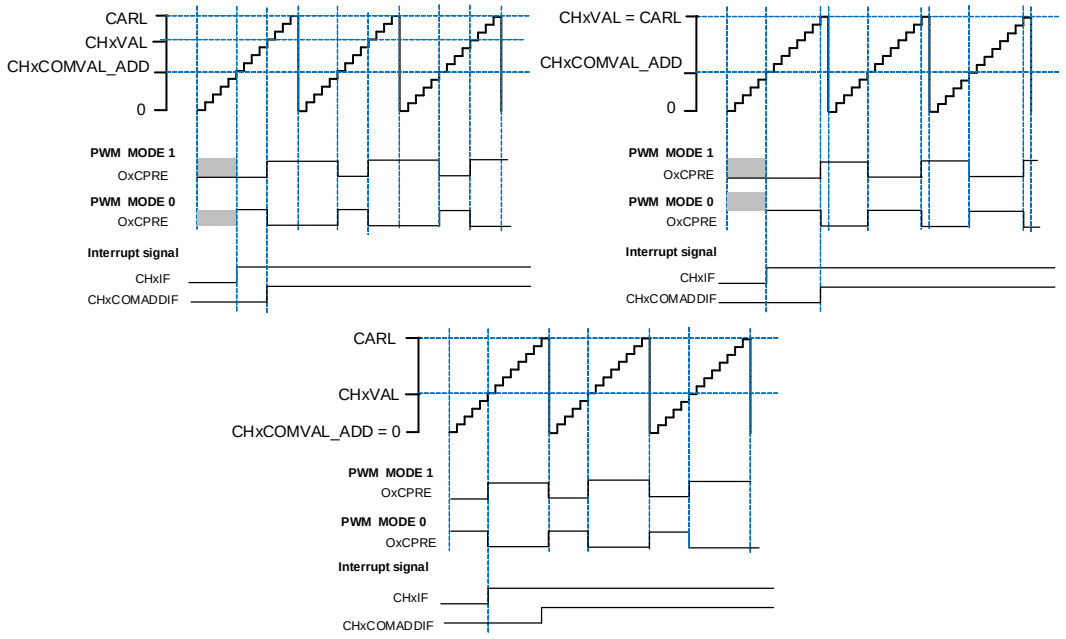
- CHxVAL = CHxCOMVAL_ADD, and the value of CHxVAL and CHxCOMVAL_ADD between 0 and CARL.

Figure 23-108. Channel x output PWM with (CHxVAL = CHxCOMVAL_ADD)



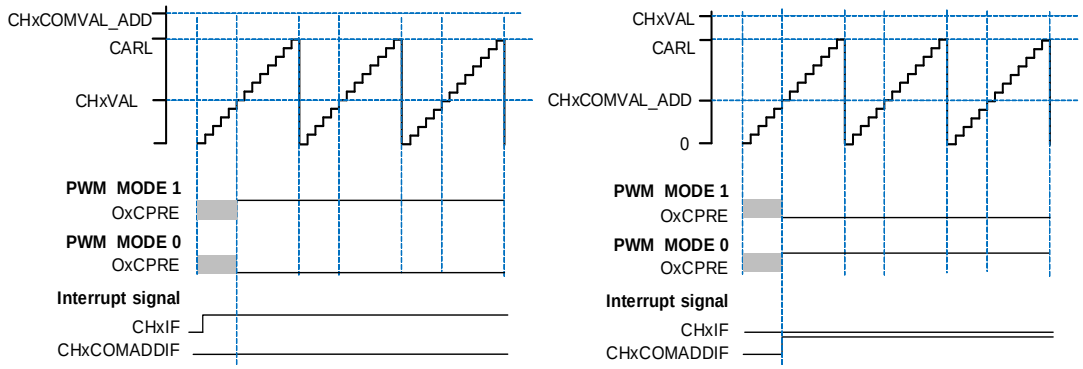
- CHxVAL > CHxCOMVAL_ADD, and the value of CHxVAL and CHxCOMVAL_ADD between 0 and CARL.

Figure 23-109. Channel x output PWM with (CHxVAL > CHxCOMVAL_ADD)



- One of the value of CHxVAL and CHxCOMVAL_ADD exceeds CARL.

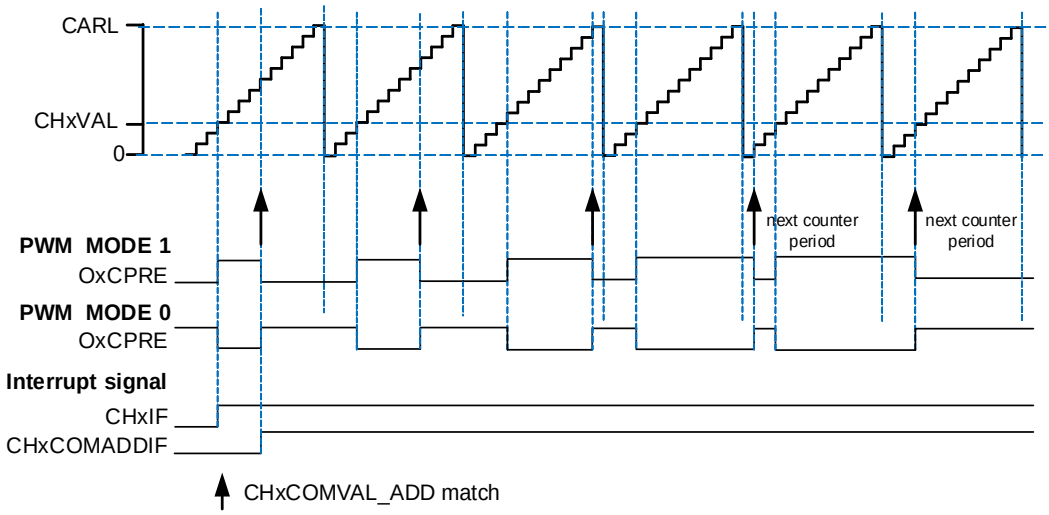
Figure 23-110. Channel x output PWM with CHxVAL or CHxCOMVAL_ADD exceeds CARL



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 23-111. 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 23-111. 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.

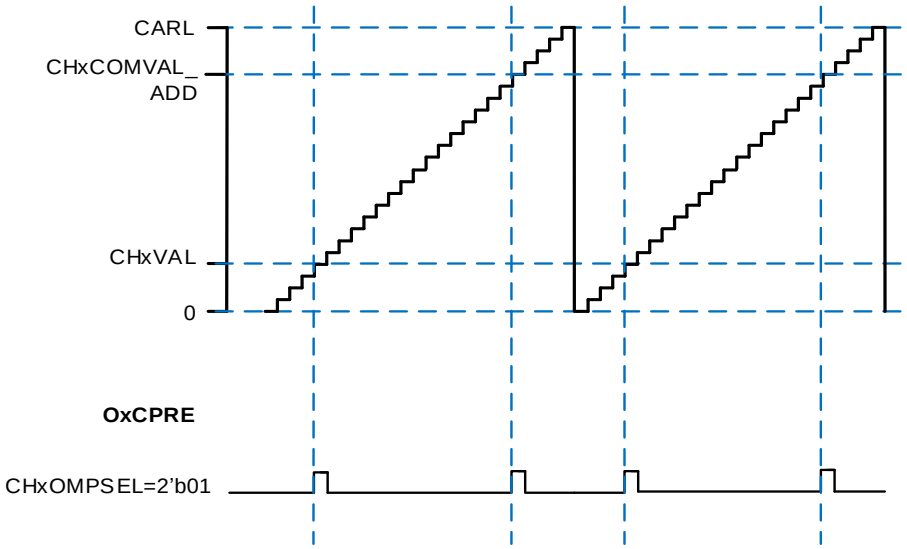
Output match pulse select

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) bits 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.

Figure 23-112. CHx_O output with a pulse in edge-aligned mode (CHxOMPSEL =2'b00)



Channel output prepare signal

As is shown in ***Figure 23-100. Output compare logic (when MCHxMSEL = 2'b00, x=0)***, ***Figure 23-101. Output compare logic (when MCHxMSEL = 2'b11, x=0)*** and ***Figure 23-102. 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.

Clear the channel output prepare signal

The OxCPRE and MOxCPRE signal can be cleared by the OCPRE_CLR_INT signal when the CHxCOMCEN or MCHxCOMCEN bit (in TIMERx_CHCTLy / TIMERx_MCHCTLy register) is set. This function can be used in the compare output modes which configured in CHxCOMCTL[3:0] or MCHxCOMCTL[3:0] bit-field (without the value of 4'b0100 and 4'b0101).

The OCPRE_CLR_INT is connected to the OCPRE_CLR inputs. The OxCPRE / MOxCPRE signal is cleared by the high level of the OCPRE_CLR_INT signal, and restored until the next update event occurs. The OCPRE_CLR inputs can be selected in OCRINSEL[2:0] bit-field in the TIMERx_AFCTL1 register.

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_CCHP0 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 23-19. Complementary outputs controlled by parameters (MCHxMSEL =2'b11)***.

Table 23-19. 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):	
			1	0	the CHx_O / MCHx_O output inactive level firstly: CHx_O	

Complementary Parameters					Output Status	
POEN	ROS	IOS	CHxEN	MCHxEN	CHx_O	MCHx_O
				1	= CHxP, MCHx_O = CHxNP; If the clock for deadtime generator is present, after a deadtime: CHx_O = ISOx, MCHx_O = ISOxN. (3)	
		1	x	x	CHx_O/ MCHx_O output “off-state”: the CHx_O/ MCHx_O output inactive level firstly: CHx_O = CHxP, MCHx_O = CHxNP; 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

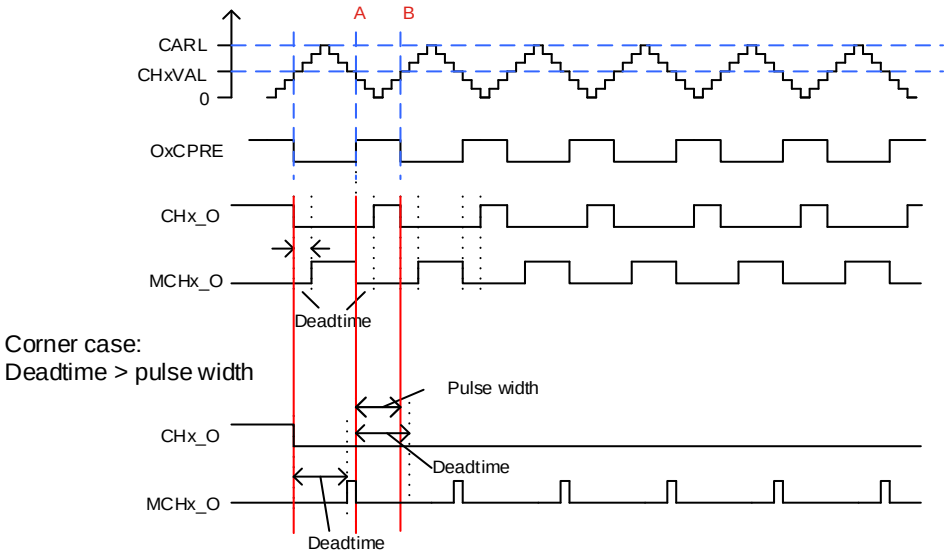
The dead time insertion is enabled when MCHxMSEL=2’b11 and both CHxEN and MCHxEN are configured to 1’b1, it is also necessary to configure POEN to 1. The field named DTCFG defines the dead time delay that can be used for all channels. Refer to the ***Complementary channel protection register 0 (TIMERx_CCHP0)*** 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 23-113. 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 23-113. Complementary output with dead-time insertion***).

Figure 23-113. Complementary output with dead-time insertion



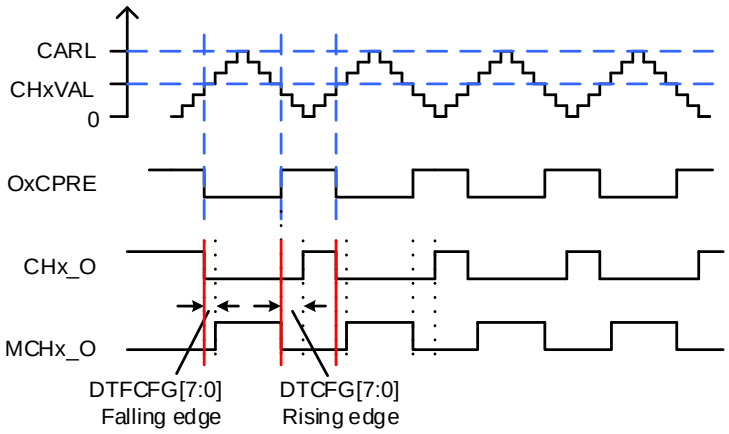
Different dead time insertion

The CHx_O and MCHx_O signal can output with different dead time when the DTDIFEN bit (in the TIMERx_CCHP1 register) is set to 1. As shown in ***Figure 23-114. 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 dead time can be modified on-the-fly when the CHx_O and MCHx_O signals are output. When DTMODEN bit in TIMERx_CCHP1 register is set, this function can be enabled. The new value of DTCFG[7:0] bit-field and DTFCFG[7:0] bit-field will be active when the next update event occurs.

Figure 23-114. Complementary output with different dead time(DTDIFEN=1)



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_CCHP0 register. The break input polarity is configured by the BRK0P bit in TIMERx_CCHP0 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_CCHP0 register, ISOx and ISOxN bits in the TIMERx_CTL1 register.

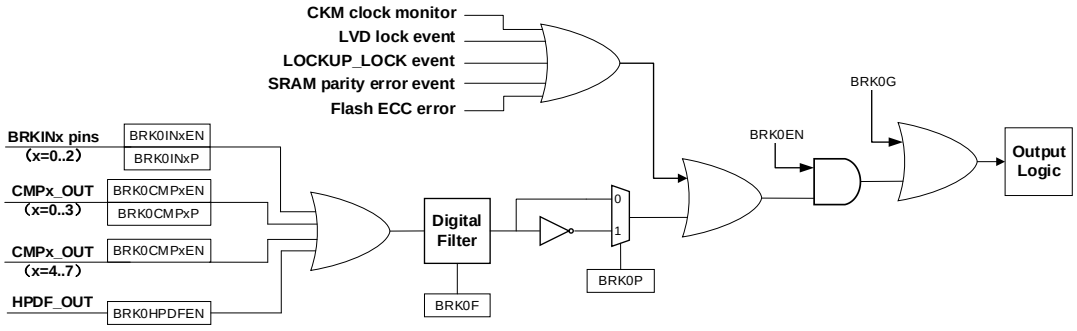
The break event is the result of logic ORed of all sources. The BREAK0 function can handle three types of event sources:

1) External sources: coming from BRKIN0 input;
2) System sources: HXTAL stuck event which is generated by Clock Monitor CKM in RCU, LVD lock event, Cortex®-M33 LOCKUP_LOCK event, SRAM parity error event or flash ECC error event;
3) On-chip peripheral events: input by comparator output or HPDF watchdog output.

BREAK0 events can also be generated by software using BRK0G bit in the TIMERx_SWEVG register.

Refer to ***Figure 23-115. BREAK0 function logic diagram***, BRKIN0 can select GPIO pins from the TRIGSEL module, which can select by ***Trigger selection for TIMER14_BRKIN register (TRIGSEL_TIMER14BRKIN)***.

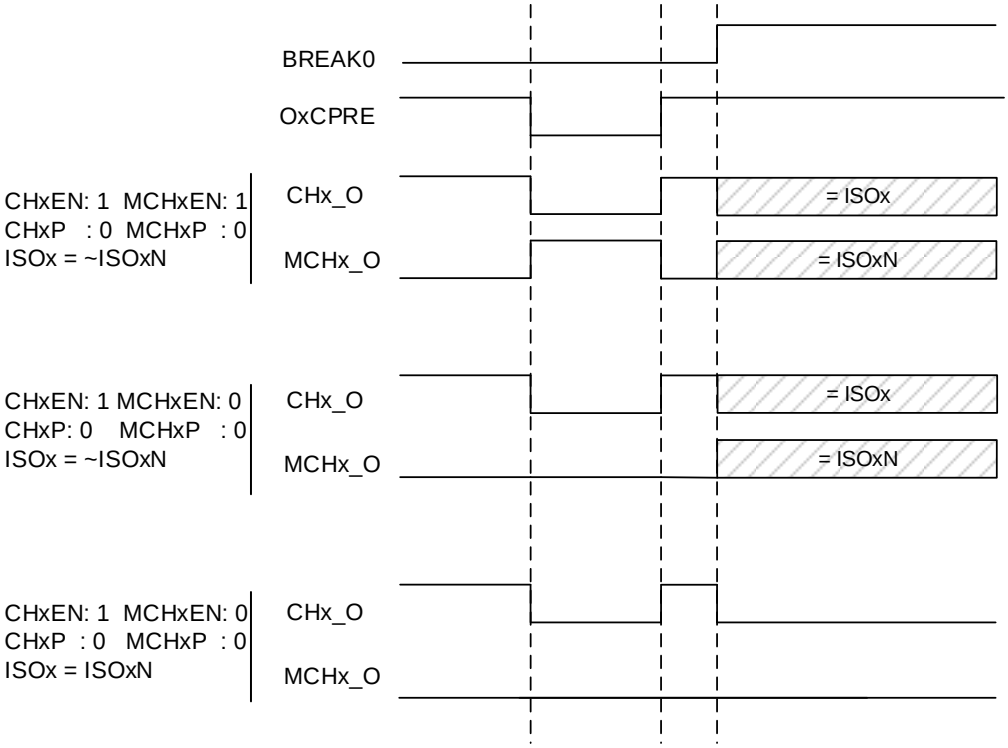
Figure 23-115. BREAK0 function logic diagram



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 23-116. 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 23-116. Output behavior of the channel in response to BREAK0 (the break input high active and IOS=1)



When a break occurs, the BRKIF bit in the TIMERx_INTF register will be set. If BRKIE is 1, an interrupt will be generated.

Locked break function

The BRKIN0 input pin of general timer have the locked break function, this function can be enabled by setting the BRK0LK bit in the TIMERx_CCHP0 register.

When the locked break function is enabled, the BRKIN0 pins need to be configured to open-drain output mode with low level active (BRK0P=0 and BRK0IN0P=0). When any break source requests occur, the BRKIN0 pin can be forced to low level. If the break input polarity is active high (BRK0P=1 and BRK0IN0P =1), the locked break function is invalid.

When the break function is enabled (the BRK0EN =1), the BRKIN0 pin can be forced to low level with the BRK0G bit setting to 1 by software.

When the break function is disabled (the BRK0EN =0), setting the BRK0G bit will have no effect on the BRKIN0 pin. The BRK0F bit will set and the channel outputs will be in a safe state.

The BRKIN0 pin can be released by setting the BRK0REL bit in the TIMERx_CCHP0 register. When the break input sources are inactive, the BRK0REL bit will cleared by hardware and the BRKIN0 pin will restore the locked break function.

In the following two cases, the BRKIN0 pin cannot be released:

1) Break input sources are active: the BRK0REL bit is set to 1 and the BRKIN0 pin locked break function is released. The break events are still active, because the break input sources are still active.
2) POEN=1: when the channel outputs are enabled, the BRKIN0 pin cannot be released even if the BRK0REL is set.

Table 23-20. Break function input pins locked / released conditions

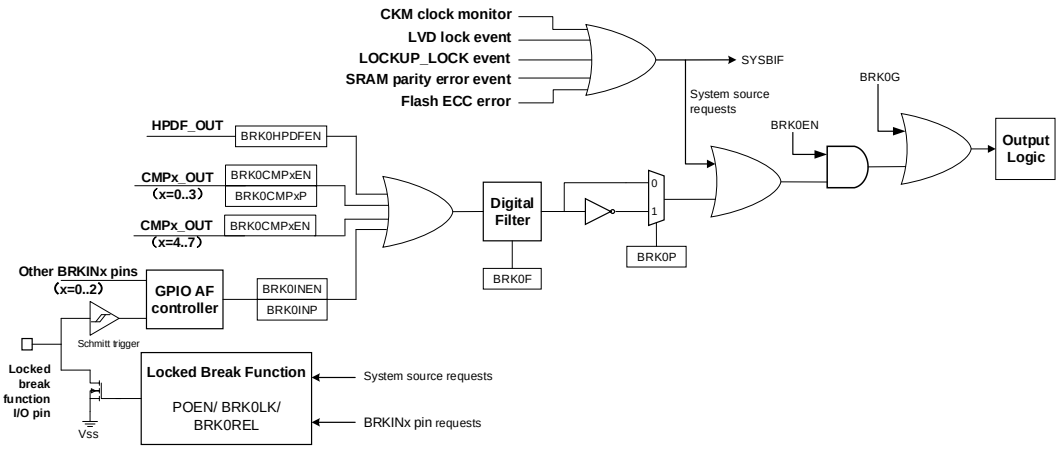
POEN	BRK0LK	BRK0REL	Break input pin state
0	1	0	Locked
	1	1	Released

The locked break function of the BREAK0 input pin BRKIN0 is enabled by default (BRK0REL=0). When the BREAK0 events occur, the following steps can be used to reconfigure the locked break function:

1) Set the BRK0REL bit to 1 and released the BRKIN0 pin;
2) The software waits for the system break sources inactive, and then clears the SYSBIF flag;
3) The software polls the state of BRK0REL bit, until the BRK0REL bit is cleared (cleared by hardware).

Then the locked break function of BREAK0 input pin is re-enabled, and the channel outputs can be restored by setting the POEN bit to 1 by software.

Figure 23-117. BRKIN0 pin logic with BREAK0 function

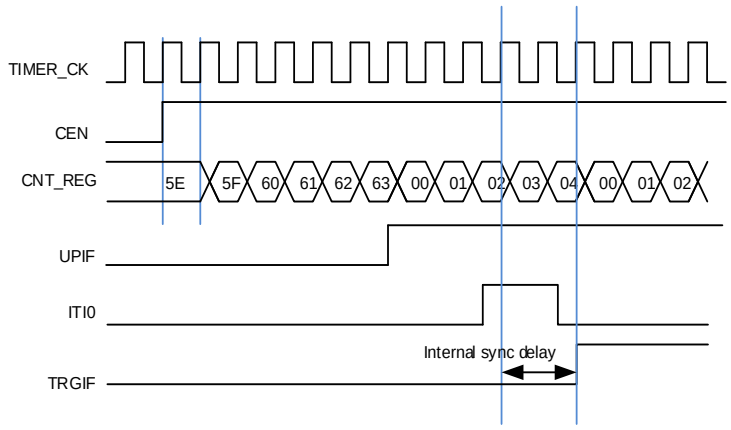
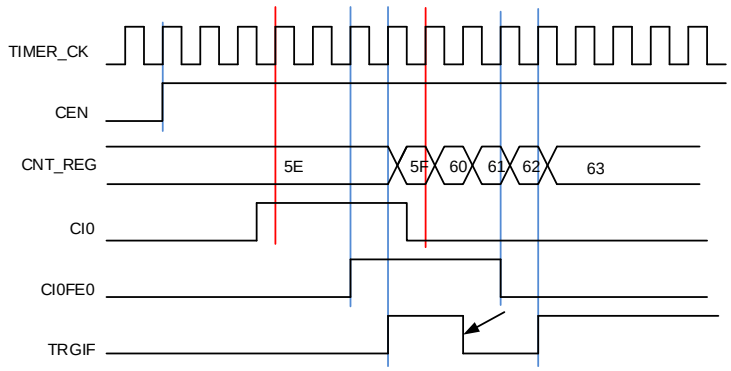


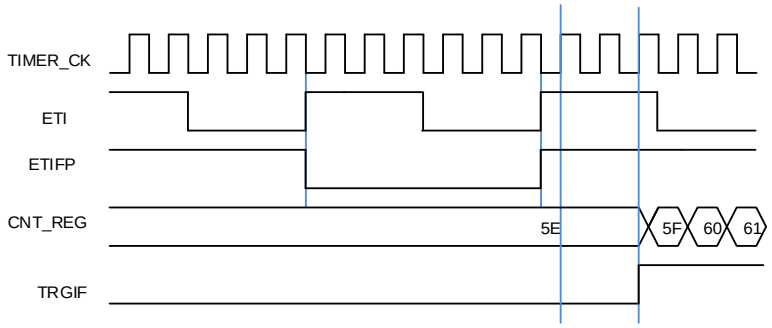
Master-slave management

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..7) in SYSCFG_TIMERxCFG(x=14).

Table 23-21. 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 y=7: restart + event mode	TSCFGy[4:0] 00000: Mode disable 00001: ITI0 00010: ITI1 00011: ITI2 00100: ITI3 00101: CI0F_ED 00110: CI0FE0 00111: CI1FE1 01000: Reserved 01001: ITI4 01010: ITI5 01011: ITI6 01100: ITI7 01101: ITI8 01110: ITI9 01111: ITI10 10000: Reserved 10001: Reserved 10010: Reserved 10011: ITI14 Others: Reserved	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.

<table>
<tr><th></th><th>Mode Selection</th><th>Source Selection</th><th>Polarity Selection</th><th>Filter and Prescaler</th></tr>
<tr><td rowspan="2">Exam1</td><td>Restart mode
The counter can be clear and restart when a rising trigger input.</td><td>TSCFG3[4:0]
5’b00001, ITI0 is the selection.</td><td>For ITI0, no polarity selector can be used.</td><td>For the ITI0, no filter and prescaler can be used.</td></tr>
<tr><td colspan="4">Figure 23-118. Restart mode
 
</td></tr>
<tr><td rowspan="2">Exam2</td><td>Pause mode
The counter can be paused when the trigger input is low.</td><td>TSCFG4[4:0]
=5’b00110,
CI0FE0 is the selection.</td><td>TI0S=0. (Non-xor) [MCHxP=0, CH0P=0] no inverted. Capture will be sensitive to the rising edge only.</td><td>Filter is bypass in this example.</td></tr>
<tr><td colspan="4">Figure 23-119. Pause mode
 
</td></tr>
<tr><td>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>
</table>

	Mode Selection	Source Selection	Polarity Selection	Filter and Prescaler
	Figure 23-120. Event mode  			
Exam4	**Restart + event mode** The counter is reinitialized and started when a rising edge of trigger input comes.			

Single pulse mode

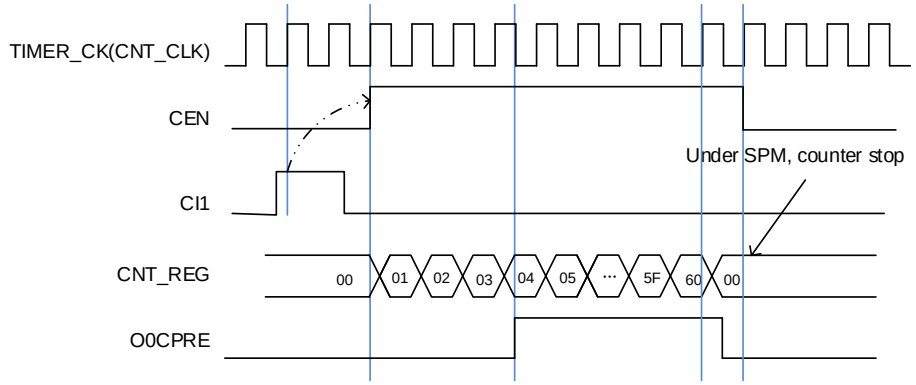
Single pulse mode is opposite to the repetitive mode, which can be enabled by setting SPM in TIMERx_CTL0. If SPM is set, the counter will be cleared and stopped automatically when the next update event occurs. In order to get a pulse waveform, the TIMERx is configured to PWM mode or compare mode by CHxCOMCTL 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 23-121. Single pulse mode TIMERx_CHxCV = 0x04 TIMERx_CAR=0x60



Delayable single pulse mode

Delayable single pulse mode is enabled by setting CHxCOMCTL[3:0]/ MCHxCOMCTL[3:0] in TIMERx_CHCTLx/ TIMERx_MCHCTLx registers. In this mode, the pulse width of OxCPRE/ MOxCPRE signal is determined by the TIMERx_CAR register.

Once the timer is set to the delayable single pulse mode, the following configuration is required:

- TIMERx need to work in slave mode and TSCFG7[4:0] != 5'b00000 in SYSCFG_TIMERxCFG(x=0, 7, 19);
- The CHxCOMCTL[3:0]/ MCHxCOMCTL[3:0] bit-field is setting to 4'b1000 (delayable single pulse mode 0) or 4'b1001 (delayable single pulse mode 1).

In delayable SPM mode 0. The behavior of OxCPRE/ MOxCPRE is performed as in PWM mode 0. When counting up, the OxCPRE/ MOxCPRE is active. When a trigger event occurs, the OxCPRE/ MOxCPRE is inactive. The OxCPRE/ MOxCPRE is active again at the next update event; When counting down, the OxCPRE/ MOxCPRE is inactive, when a trigger event occurs, the OxCPRE/ MOxCPRE is active. The OxCPRE/ MOxCPRE is inactive again at the next update event.

In delayable mode 1. The behavior of OxCPRE/ MOxCPRE is performed as in PWM mode 1. When counting up, the OxCPRE/ MOxCPRE is inactive, when a trigger event occurs, the OxCPRE/ MOxCPRE is active. The OxCPRE/ MOxCPRE is inactive again at the next update event; When counting down, the OxCPRE/ MOxCPRE is active. When a trigger event occurs, the OxCPRE/ MOxCPRE is inactive. The OxCPRE/ MOxCPRE is active again at the next update event.

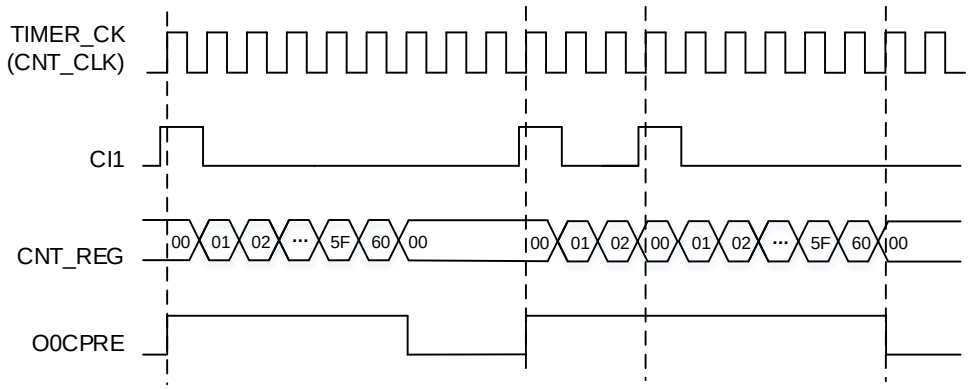
The PWM adjustment mode can also use in delayable SPM modes.

Note:

- The center-aligned counting mode cannot be used in this mode and the CAM[1:0] = 2'b00(in TIMERx_CTL0 register);
- When counter counting up (DIR = 0 in TIMERx_CTL0 register), the value of

TIMERx_CHxCV/ TIMERx_MCHxCV should be set to 0; When counting down (DIR =1 in TIMERx_CTL0 register), the value of TIMERx_CHxCV/ TIMERx_MCHxCV should be greater than or equal to the value of TIMERx_CAR register.

Figure 23-122. delayable single pulse mode TIMERx_CHxCV=0x00, TIMERx_CAR=0x60



Timers interconnection

Please refer to ***Advanced timer (TIMERx, x=0, 7,19) Timers interconnection***.

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.

UPIF bit backup

The UPIF bit backup function is enabled by setting UPIFBUEN in the TIMERx_CTL0 register. The UPIF and UPIFBU bits are fully synchronized and without latency.

By using this function, the UPIF bit in the TIMERx_INTF register will be backuped to the UPIFBU bit in the TIMERx_CNT register. This can avoid conflicts when reading the counter and interrupt processing.

Timer debug mode

When the Cortex®-M33 halted, and the TIMERx_HOLD configuration bit in DBG_CTL1 register set to 1, the TIMERx counter stops.

23.3.5. Register definition (TIMERx, x=14)

TIMER14 base address: 0x4001 4000

Control register 0 (TIMERx_CTL0)

Address offset: 0x00

Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			ADMEN	UPIFBUEN	Reserved	CKDIV[1:0]		ARSE	Reserved			SPM	UPS	UPDIS	CEN
			rw	rw		rw		rw				rw	rw	rw	rw

Bits	Fields	Descriptions
31:13	Reserved	Must be kept at reset value.
12	ADMEN	Adjustment mode enable 0: Adjustment mode disabled 1: Adjustment mode enabled Note: This bit can be modified only when the CEN bit is 0.
11	UPIFBUEN	UPIF bit backup enable 0: Backup disable. UPIF bit is not backuped to UPIFBU bit in TIMERx_CNT register. 1: Backup enabled. UPIF bit is backuped to UPIFBU bit in TIMERx_CNT register.
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
6:4	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: 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: 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
CCUC[2:1]		Reserved													
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					ISO1	ISO0N	ISO0	Reserved	MMC0[2:0]			DMAS	CCUC[0]	Reserved	CCSE
					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.
29:11	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, MCH0_O is set low. 1: When POEN bit is reset, MCH0_O is set high. This bit can be modified only when PROT[1:0] bits in TIMERx_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 MCH0_O is implemented. This bit can be modified only when PROT[1:0] bits in TIMERx_CCHP0 register is 00.
7	Reserved	Must be kept at reset value.
6:4	MMC0[2:0]	Master mode control 0 These bits control the selection of TRGO0 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 TRGO0 pulse occurs. And in the latter case, the signal on TRGO0 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 TRGO0. 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 TRGO0 output, except if the master-slave mode is selected. 010: Update. In this mode, the master mode controller selects the update event as TRGO0. 011: Capture/compare pulse. In this mode, the master mode controller generates a TRGO0 pulse when a capture or a compare match occurs in channel 0. 100: Compare. In this mode, the master mode controller selects the O0CPRE signal as TRGO0. 101: Compare. In this mode, the master mode controller selects the O1CPRE signal as TRGO0. 110: Reserved. 111: Reserved.
3	DMAS	DMA request source selection 0: DMA request of CHx/MCHx is sent when capture/compare event occurs.

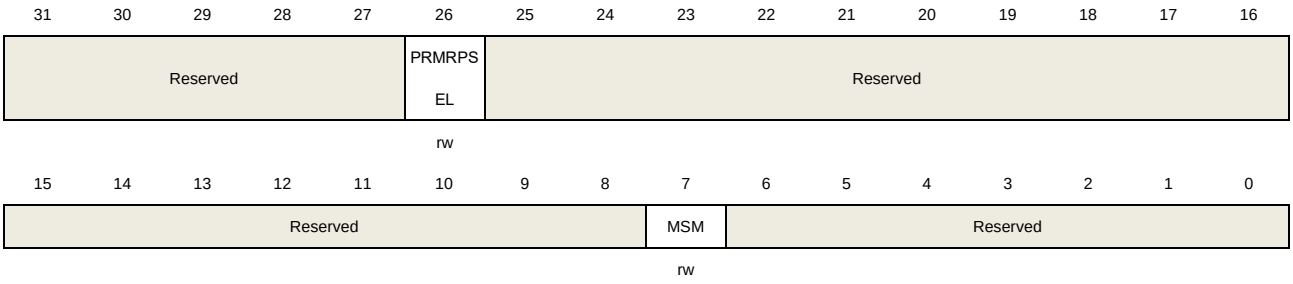
		1: DMA request of channel CHx/MCHx is sent when update event occurs.
2	CCUC[0]	Commutation control shadow register update control The CCUC[2:1] and CCUC[0] field are used to control the commutation control shadow register update. When the commutation control shadow registers (for CHxEN, MCHxEN and CHxCOMCTL 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. 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, 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).



Bits	Fields	Descriptions
31:27	Reserved	Must be kept at reset value.
26	PRMRPSEL	Pause + restart mode reset polarity selection 0: Counter is reset at falling edge

		1: Counter is reset at rising edge
25:8	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

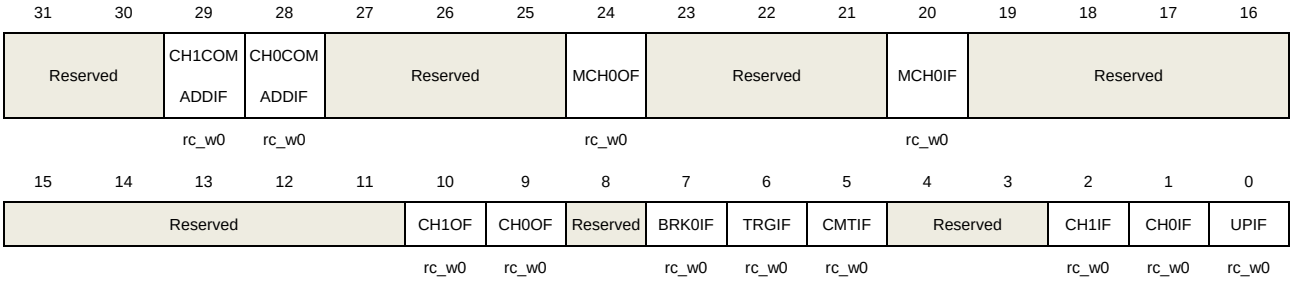
1	CH0IE	Channel 0 capture/compare interrupt enable 0: Disabled 1: Enabled
0	UPIE	Update interrupt enable 0: Disabled 1: Enabled

Interrupt flag register (TIMERx_INTF)

Address offset: 0x10

Reset value: 0x0000 0000

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



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]	CH1COMADDSEN	CH0COMADDSEN	CH1ADDUPS	CH0ADDUPS	Reserved	CH1COMCTL[3]	Reserved							CH0COMCTL[3]
		Reserved	Reserved	Reserved	Reserved		Reserved								Reserved
rw	rw	rw	rw	rw	rw		rw								rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1COMCEN	CH1COMCTL[2:0]			CH1COMSEN	CH1COMFEN	CH1MS[1:0]		CH0COMCEN	CH0COMCTL[2:0]			CH0COMSEN	CH0COMFEN	CH0MS[1:0]	
CH1CAPFLT[3:0]				CH1CAPPSC[1:0]				CH0CAPFLT[3:0]				CH0CAPPSC[1:0]			
rw				rw		rw		rw				rw		rw	

Output compare mode:

Bits	Fields	Descriptions
31	CH1MS[2]	Channel 1 I/O mode selection Refer to CH1MS[1:0]description
30	CH0MS[2]	Channel 0 I/O mode selection Refer to CH0MS[1:0] description
29	CH1COMADDSEN	Channel 1 additional compare output shadow enable Refer to CH0COMADDSEN description.
28	CH0COMADDSEN	Channel 0 additional compare output shadow enable When this bit is set, the shadow register of TIMERx_CH0COMV_ADD register which updates at each update event will be enabled. 0: Channel 0 additional compare output shadow disabled 1: Channel 0 additional compare output shadow enabled The PWM mode can be used without validating the shadow register only in single pulse mode (SPM bit in TIMERx_CTL0 register is set). This bit cannot be modified when PROT[1:0] bit-field in TIMERx_CCHP0 register is 11 and CH0MS bit-field is 000.
27	CH1ADDUPS	Channel 1 additional update source 0: TIMERx_CH1COMV_ADD register is updated when an update event occurs.

		1: TIMERx_CH1COMV_ADD register is updated when the counter matches the value of CH1VAL.
26	CH0ADDUPS	Channel 0 additional update source 0: TIMERx_CH0COMV_ADD register is updated when an update event occurs. 1: TIMERx_CH0COMV_ADD register is updated when the counter matches the value of CH0VAL.
25	Reserved	Must be kept at reset value.
24	CH1COMCTL[3]	Channel 1 compare output control Refer to CH0COMCTL[2:0] description
23:17	Reserved	Must be kept at reset value.
16	CH0COMCTL[3]	Channel 0 compare output control Refer to CH0COMCTL[2:0] description
15	CH1COMCEN	Channel 1 output compare clear enable Refer to CH0COMCEN description
14:12	CH1COMCTL[2:0]	Channel 1 compare output control Refer to CH0COMCTL[2:0] description
11	CH1COMSEN	Channel 1 output compare shadow enable Refer to CH0COMSEN description
10	CH1COMFEN	Channel 1 output compare fast enable Refer to CH0COMSEN 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=14) 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.

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: Delayable SPM mode 0. The behavior of O0CPRE is performed as in PWM mode 0. When counting up, the O0CPRE is active. When a trigger event occurs, the O0CPRE is inactive. The O0CPRE is active again at the next update event; When counting down, the O0CPRE is inactive, when a trigger event occurs, the O0CPRE is active. The O0CPRE is inactive again at the next update event.

1001: Delayable SPM mode 1. The behavior of O0CPRE is performed as in PWM mode 1. When counting up, the O0CPRE is inactive, when a trigger event occurs, the O0CPRE is active. The O0CPRE is inactive again at the next update event; When counting down, the O0CPRE is active. When a trigger event occurs, the O0CPRE is inactive. The O0CPRE is active again at the next update event.

1010~1111: 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. The minimum delay from an edge on the trigger input to activate CH0_O output is 5 clock cycles. 1: Channel 0 output quickly compare enable. The minimum delay from an edge on the trigger input to activate CH0_O output is 3 clock cycles.
1:0	CH0MS[1:0]	Channel 0 I/O mode selection This bit-field specifies the work mode of the channel and the input signal selection. The CH0MS[2:0] bit-field is writable only when the channel is not active (When MCH0MSEL[1:0] = 2’b00, the CH1EN bit in TIMERx_CHCTL2 register is reset; when MCH0MSEL[1:0] = 2’b11, the CH0EN and MCH0EN bits in TIMERx_CHCTL2 register are reset). 000: Channel 0 is configured as output. 001: Channel 0 is configured as input, IS0 is connected to CI0FE0. 010: Channel 0 is configured as input, IS0 is connected to CI1FE0. 011: Channel 0 is configured as input, IS0 is connected to ITS, this mode is working only if an internal trigger input is selected (through TSCFG15[4:0] bit-field in SYSCFG_TIMERxCFG2(x=14) 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															

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
31:8	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 MCH0_O output signal polarity. 0: Multi mode channel 0 output active high 1: Multi mode channel 0 output active low When CH0 is configured in input mode, in conjunction with CH0P, this bit is used to define the polarity of CH0. This bit cannot be modified when PROT[1:0] bit-field in TIMERx_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 MCH0_O signal in active state. When multi mode channel 0 is configured in input mode, setting this bit enables the capture event in multi mode channel 0. 0: Multi mode channel 0 disabled 1: Multi mode channel 0 enabled

1	CH0P	Channel 0 capture/compare polarity When channel 0 is configured in output mode, this bit specifies the output signal polarity. 0: Channel 0 active high 1: Channel 0 active low When channel 0 is configured in input mode, these bits specifie the channel 0 input signal's polarity. [MCH0P, CH0P] will select the active trigger or capture polarity for channel 0 input signals. 00: Channel 0 input signal's rising edge is the active signal for capture or trigger operation in slave mode. And channel 0 input signal will not be inverted. 01: Channel 0 input signal's falling edge is the active signal for capture or trigger operation in slave mode. And channel 0 input signal will be inverted. 10: Reserved. 11: Noninverted/both channel 0 input signal's edges. This bit cannot be modified when PROT[1:0] bit-field in TIMERx_CCHP0 register is 11 or 10.
0	CH0EN	Channel 0 capture/compare enable When channel 0 is configured in output mode, setting this bit enables CH0_O signal in active state. When channel 0 is configured in input mode, setting this bit enables the capture event in channel 0. 0: Channel 0 disabled 1: Channel 0 enabled

Counter register (TIMERx_CNT)

Address offset: 0x24
Reset value: 0x0000 0000

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

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

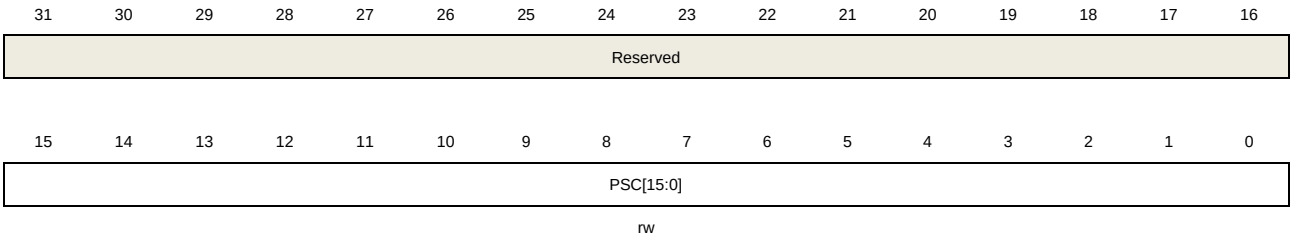
Bits	Fields	Descriptions
31	UPIFBU	UPIF bit backup This bit is a backup of UPIF bit in TIMERx_INTF register, and read-only. This bit is only valid when UPIFBUEN = 1. If the UPIFBUEN =0, this bit is reserved and read the result is 0.
30:16	Reserved	Must be kept at reset value.

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. When the PWMADMEN =0, this bit-field indicates the current counter value. Writing to this bit-field can change the value of the counter. When the PWMADMEN =1, this bit-field just indicates the integer part of the counter value, and without the fractional part.

Prescaler register (TIMERx_PSC)

Address offset: 0x28
Reset value: 0x0000 0000

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

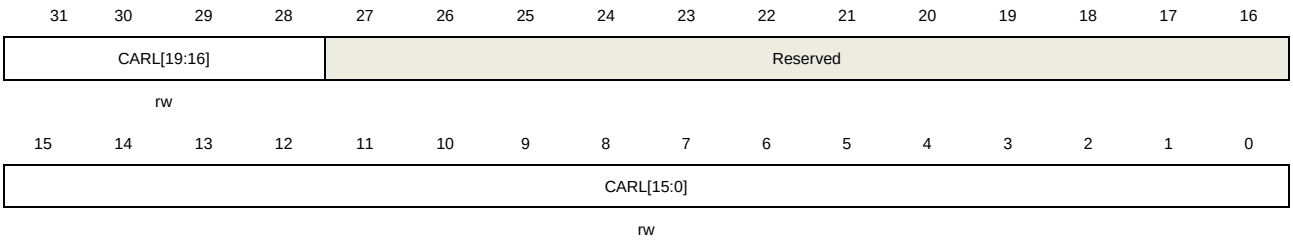


Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	PSC[15:0]	Prescaler value of the counter clock The PSC clock is divided by (PSC+1) to generate the counter clock. The value of this bit-field will be loaded to the corresponding shadow register at every update event.

Counter auto reload register (TIMERx_CAR)

Address offset: 0x2C
Reset value: 0x0000 FFFF

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



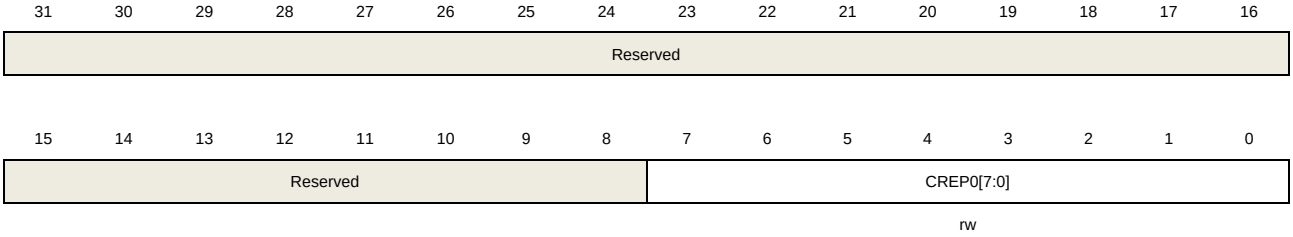
Bits	Fields	Descriptions
31:28	CARL[19:16]	Counter auto reload value (bit 16 to 19) When the PWMADMEN =0, CARL[19:16] bit-field is 0000.

		When the PWMADMEN =1, CARL[19:16] bit-field specifies fractional part of the auto reload value.
27:16	Reserved	Must be kept at reset value.
15:0	CARL[15:0]	Counter auto reload value When the PWMADMEN =0, CARL[15:0] bit-field specifies the auto reload value of the counter. When the PWMADMEN =1, CARL[15:0] bit-field specifies integer part of the auto reload value.

Counter repetition register 0 (TIMERx_CREP0)

Address offset: 0x30
Reset value: 0x0000 0000

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

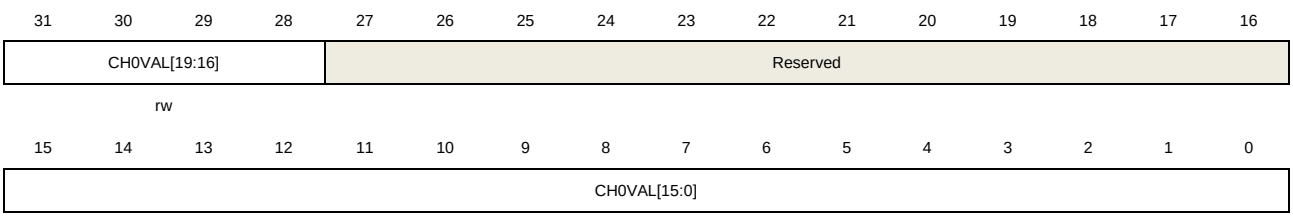


Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	CREP0[7:0]	Counter repetition value 0 This bit-field specifies the update event generation rate. Each time the repetition counter counts down to zero, an update event will be generated. The update rate of the shadow registers is also affected by this bit-field when these shadow registers are enabled. **Note**: This bit-field just used with CREPSEL =0 (in TIMERx_CFG register).

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).



rw

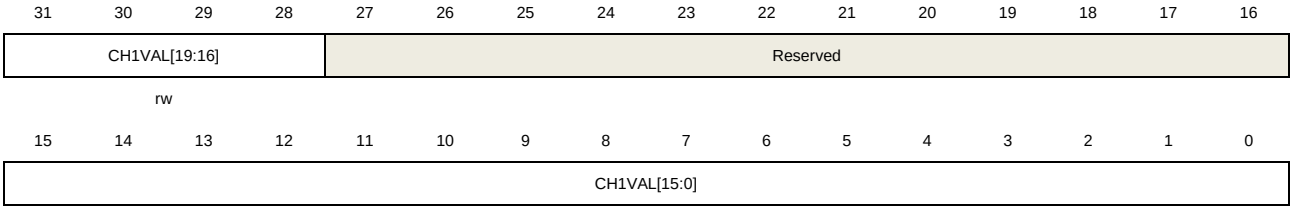
Bits	Fields	Descriptions
31:28	CH0VAL[19:16]	Capture/compare value of channel 0 (bit 16 to 19) When channel 0 is configured in input mode, CH0VAL[19:16] bit-field is 0000. When channel 0 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event. When the PWMADMEN = 0, the CH0VAL[19:16] bit-field is 0000. When the PWMADMEN =1, CH0VAL[19:16] bit-field specifies the fractional part.
27:16	Reserved	Must be kept at reset value.
15:0	CH0VAL[15:0]	Capture/compare value of channel 0 (bit 0 to 15) When channel 0 is configured in input mode, CH0VAL[15:0] 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. When the PWMADMEN = 0, the CH0VAL[15:0] bit-field specifies the compare value. When the PWMADMEN =1, CH0VAL[15:0] bit-field specifies integer part of the compare value.

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).



Bits	Fields	Descriptions
31:28	CH1VAL[19:16]	Capture/compare value of channel 1 (bit 16 to 19) When channel 1 is configured in input mode, CH1VAL[19:16] bit-field is 0000. 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. When the PWMADMEN = 0, the CH1VAL[19:16] bit-field is 0000.

		When the PWMADMEN =1, CH1VAL[19:16] bit-field specifies the fractional part.
27:16	Reserved	Must be kept at reset value.
15:0	CH1VAL[15:0]	Capture/compare value of channel 1 (bit 0 to 15) When channel 1 is configured in input mode, CH1VAL[15:0] 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. When the PWMADMEN = 0, the CH1VAL[15:0] bit-field specifies the compare value. When the PWMADMEN =1, CH1VAL[15:0] bit-field specifies integer part of the compare value.

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			BRK0LK	Reserved	BRK0REL	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
31:29	Reserved	Must be kept at reset value.
28	BRK0LK	BREAK0 input locked 0: BREAK0 input in input mode 1: BREAK0 input in locked mode When the BRK0LK is set to 1, the BREAK0 input is configured in open drain output mode. Any active BREAK0 event asserts a low logic level on the Break input to indicate an internal break event to external devices. This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00. **Note**: Every write operation to this bit needs a delay of 1 APB clock to active.
27	Reserved	Must be kept at reset value.
26	BRK0REL	BREAK0 input released This bit is cleared by hardware when the BREAK0 input is invalid.

0: BREAK0 input is unreleased
1: BREAK0 input is released
The locked output control (open drain mode in Hi-z state) is released by setting this bit with software. And when the fault is disappeared, this bit will reset by hardware.
Note: Every write operation to this bit needs a delay of 1 APB clock to active.

25:20 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 23-19. Complementary outputs controlled by parameters (MCHxMSEL =2’b11)***. 0: “off-state” disabled. If the CHxEN or CHxNEN bit is reset, the corresponding channel is output disabled. 1: “off-state” enabled. If the CHxEN or CHxNEN bit is reset, the corresponding channel is “off-state”. This bit cannot be modified when PROT [1:0] bit-filed in TIMERx_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 23-19. Complementary outputs controlled by parameters (MCHxMSEL =2’b11)***. 0: “off-state” disabled. If the CHxEN/CHxNEN bits are both reset, the channels are output disabled. 1: “off-state” enabled. No matter the CHxEN/CHxNEN bits, the channels are “off-state”. This bit cannot be modified when PROT [1:0] bit-filed in TIMERx_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.

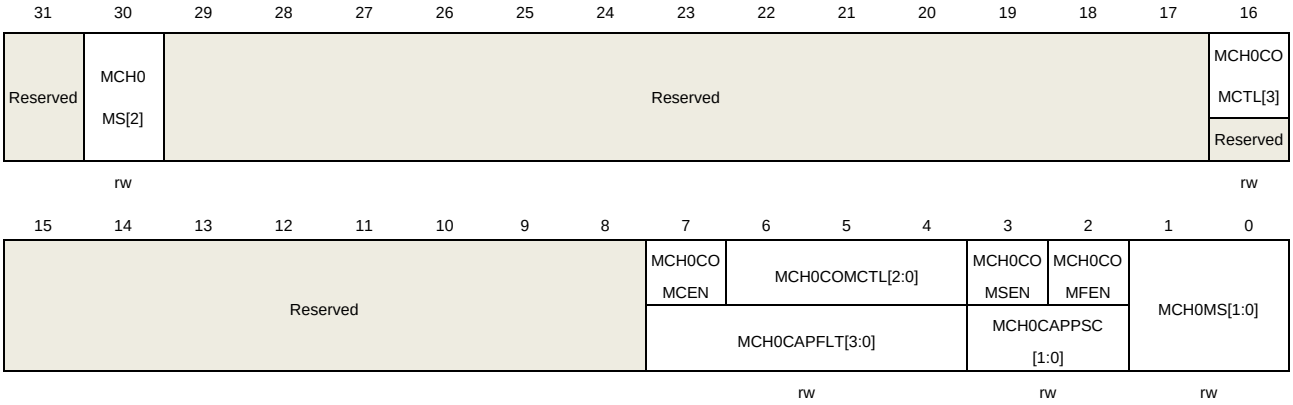
7:0	DTCFG[7:0]	Dead time configuration This bit-field controls the value of the dead-time, which is inserted before the output transitions. The relationship between the value of DTCFG and the duration of dead-time is as follow: DTCFG[7:5] = 3'b0xx: DT value = DTCFG[7:0] * t_{DT}, $t_{DT} = t_{DTS}$. 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.
15:8	Reserved	Must be kept at reset value.
7	MCH0COMCEN	Multi mode channel 0 output compare clear enable. When this bit is set, the MO0CPRE signal is cleared when high level is detected on ETIFP input. 0: Multi mode channel 0 output compare clear disabled. 1: Multi mode channel 0 output compare clear enabled.
6:4	MCH0COMCTL[2:0]	Multi mode channel 0 output compare control When multi mode channel 0 is configured in output mode, and the MCH0MSEL[1:0] = 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: Delayable SPM mode 0. The behavior of MO0CPRE is performed as in PWM mode 0. When counting up, the MO0CPRE is active. When a trigger event occurs, the MO0CPRE is inactive. The MO0CPRE is active again at the next update event; When counting down, the MO0CPRE is inactive, when a trigger event occurs, the

		O0CPR MO0CPRE E is active. The MO0CPRE is inactive again at the next update event. 1001: Delayable SPM mode 1. The behavior of MO0CPRE is performed as in PWM mode 1. When counting up, the MO0CPRE is inactive, when a trigger event occurs, the MO0CPRE is active. The MO0CPRE is inactive again at the next update event; When counting down, the MO0CPRE is active. When a trigger event occurs, the MO0CPRE is inactive. The MO0CPRE is active again at the next update event. 1010~1111: 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 CH0NMS bit-field is 00(compare mode).
3	MCH0COMSEN	Multi mode channel 0 output compare shadow enable When this bit is set, the shadow register of TIMERx_MCH0CV register which updates at each update event will be enabled. 0: Multi mode channel 0 output compare shadow disabled 1: Multi mode channel 0 output compare shadow enabled The PWM mode can be used without 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 PWM0 or PWM1 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. The minimum delay from an edge on the trigger input to activate MCH0_O output is 5 clock cycles. 1: Multi mode channel 0 output quickly compare enable. The minimum delay from an edge on the trigger input to activate MCH0_O output is 3 clock cycles.
1:0	MCH0MS[1:0]	Multi mode channel 0 I/O mode selection This bit-field specifies the work mode of the channel and the input signal selection. This bit-field is writable only when the channel is not active (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=14) 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 specific 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 specific 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).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		CH1C PWMEN	CH0C PWMEN	Reserved						MCH0MSEL[1:0]		Reserved			
		rw	rw							rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				CH1OMPSEL[1:0]		CH0OMPSEL[1:0]		Reserved							
				rw		rw									

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29	CH1CPWMEN	Channel 1 composite PWM mode enable 0: Disabled 1: Enabled
28	CH0CPWMEN	Channel 0 composite PWM mode enable 0: Disabled 1: Enabled
27:22	Reserved	Must be kept at reset value.
21:20	MCH0MSEL[1:0]	Multi mode channel 0 mode select 00: Independent mode, MCH0 is independent of CH0

		01: Reserved 10: Reserved 11: Complementary mode, only the CH0 is valid for input, and the outputs of MCH0 and CH0 are complementary
19:12	Reserved	Must be kept at reset value.
11:10	CH1OMPSEL[1:0]	Channel 1 output match pulse select When the match events occur, this bit is used to select the output of O1CPRE which drives CH1_O. 00: The O1CPRE signal is output normal with the configuration of CH1COMCTL[2:0] bits. 01: Only when the counter is counting up, the O1CPRE signal is output a pulse when the match events occurs, and the pulse width is one CK_TIMER clock cycle. 10: Reserved 11: Reserved
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: Reserved 11: Reserved
7:0	Reserved	Must be kept at reset value.

TIMERx alternate function control register 0 (TIMERx_AFCTL0)

Address offset: 0x8C

Reset value: 0x0000 0001

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			BRK0CMP3P	BRK0CMP2P	BRK0CMP1P	BRK0CMP0P	Reserved								BRK0IN0P
			rw	rw	rw	rw									rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BRK0CMP6EN	BRK0CMP5EN	BRK0CMP4EN	BRK0CMP3EN	BRK0CMP2EN	BRK0CMP1EN	BRK0CMP0EN	BRK0HPDFEN	BRK0CMP7EN	Reserved						BRK0IN0EN
rw	rw	rw	rw	rw	rw	rw	rw	rw							rw

Bits	Fields	Descriptions

31:29	Reserved	Must be kept at reset value.
28	BRK0CMP3P	BREAK0 CMP3 input polarity This bit is used to configure the CMP3 input polarity, and the specific polarity is determined by this bit and the BRK0P bit. 0: CMP3 input signal will not be inverted (BRK0P =0, the input signal is active low; BRK0P =1, the input signal is active high) 1: CMP3 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.
27	BRK0CMP2P	BREAK0 CMP2 input polarity This bit is used to configure the CMP2 input polarity, and the specific polarity is determined by this bit and the BRK0P bit. 0: CMP2 input signal will not be inverted (BRK0P =0, the input signal is active low; BRK0P =1, the input signal is active high) 1: CMP2 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.
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	BRK0CMP6EN	BREAK0 CMP6 enable 0: CMP6 input disabled 1: CMP6 input enabled This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.
14	BRK0CMP5EN	BREAK0 CMP5 enable 0: CMP5 input disabled 1: CMP5 input enabled This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.
13	BRK0CMP4EN	BREAK0 CMP4 enable 0: CMP4 input disabled 1: CMP4 input enabled This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.
12	BRK0CMP3EN	BREAK0 CMP3 enable 0: CMP3 input disabled 1: CMP3 input enabled This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.
11	BRK0CMP2EN	BREAK0 CMP2 enable 0: CMP2 input disabled 1: CMP2 input enabled This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.
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	BRK0HPDFEN	BREAK0 HPDF input (HPDF_BREAK[0]) enable 0: HPDF input disabled 1: HPDF input enabled This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.
7	BRK0CMP7EN	BREAK0 CMP7 enable 0: CMP7 input disabled 1: CMP7 input enabled This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.
6:1	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.

TIMER alternate function control register 1 (TIMERx_AFCTL1)

Address offset: 0x90

Reset value: 0x0000 0000

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

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

Bits	Fields	Descriptions
31:22	Reserved	must be kept at reset value
21:19	OCRINSEL[2:0]	OCPRE_CLR inputs selection 000: OCPRE_CLR0 001: OCPRE_CLR1 … 111: OCPRE_CLR7

OCPRE_CLR inputs selection	TIMER14
OCPRE_CLR0	CMP0_OUT

OCPRE_CLR1	CMP1_OUT
OCPRE_CLR2	CMP2_OUT
OCPRE_CLR3	CMP3_OUT
OCPRE_CLR4	CMP4_OUT
OCPRE_CLR5	CMP5_OUT
OCPRE_CLR6	CMP6_OUT
OCPRE_CLR7	CMP7_OUT

This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.

18:0	Reserved	must be kept at reset value

Counter repetition register 1 (TIMERx_CREP1)

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
CREP1[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CREP1[15:0]															
rw															

Bits	Fields	Descriptions
31:0	CREP1[31:0]	Counter repetition value 1 This bit-field is 32 bits and can be read on the fly. 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. **Note**: This bit-field just used with CREPSEL =1(in TIMERx_CFG register).

Complementary channel protection register 1 (TIMERx_CCHP1)

Address offset: 0x09C
Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:18	Reserved	Must be kept at reset value.
17	DTMODEN	Dead time modified on-the-fly enable 0: Dead time value modified on-the-fly disable 1: Dead time value modified on-the-fly enable This bit can be modified only when PROT [1:0] bit-filed in TIMERx_CCHP0 register is 00.
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_CCHP1 register. 1: The dead time on rising edge is defined in DTCFG[7:0] bit-field in TIMERx_CCHP1 register, and the dead time on falling edge is defined in DTFCFG [7:0] bit-field in TIMERx_CCHP2 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.

DMA configuration register (TIMERx_DMACFG)

Address offset: 0xE0

Reset value: 0x0000 0000

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

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

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

rw rw

Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value.
13:8	DMATC[5:0]	DMA transfer count This field defines the times of accessing(R/W) the TIMERx_DMATB register by DMA. 6'b000000: transfer 1 time 6'b000001: transfer 2 times … 6'b111000: transfer 57 times
7:6	Reserved	Must be kept at reset value.
5:0	DMATA[5:0]	DMA transfer access start address This field defines the start address of accessing the TIMERx_DMATB register by DMA. When the first access to the TIMERx_DMATB register is done, this bit-field specifies the address just accessed. And then the address of the second access to the TIMERx_DMATB register will be (start address + 0x4). 6'b000000: TIMERx_CTL0 6'b000001: TIMERx_CTL1 … In a word: start address = TIMERx_CTL0 + DMATA*4

DMA transfer buffer register (TIMERx_DMATB)

Address offset: 0xE4

Reset value: 0x0000 0000

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

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

Bits	Fields	Descriptions
31:0	DMATB[31:0]	DMA transfer buffer When a read or write operation is assigned to this register, the register located at the address ranges from (start address) to (start address + transfer count * 4) will be accessed. The transfer count is calculated by hardware, and ranges from 0 to DMATC.

Configuration register (TIMERx_CFG)

Address offset: 0xFC

Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												CCUSEL	CREPSEL	CHVSEL	OUTSEL
												rw	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	CREPSEL	The counter repetition register select This bit is used to select the counter repetition register. 0: The update event rate is depended to TIMERx_CREP0 register 1: The update event rate is depended to TIMERx_CREP1 register
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.

23.4. General level4 timer (TIMERx, x=15, 16)

23.4.1. Overview

The general level4 timer module (TIMER15 / 16) is a two-channel timer that supports both input capture and output compare. They can generate PWM signals to control motor or be used for power management applications. The general level4 timer has a 16-bit counter that can be used as an unsigned counter.

In addition, the general level4 timers can be programmed and be used for counting, their external events can be used to drive other timers.

Timer also includes a dead-time Insertion module which is suitable for motor control applications.

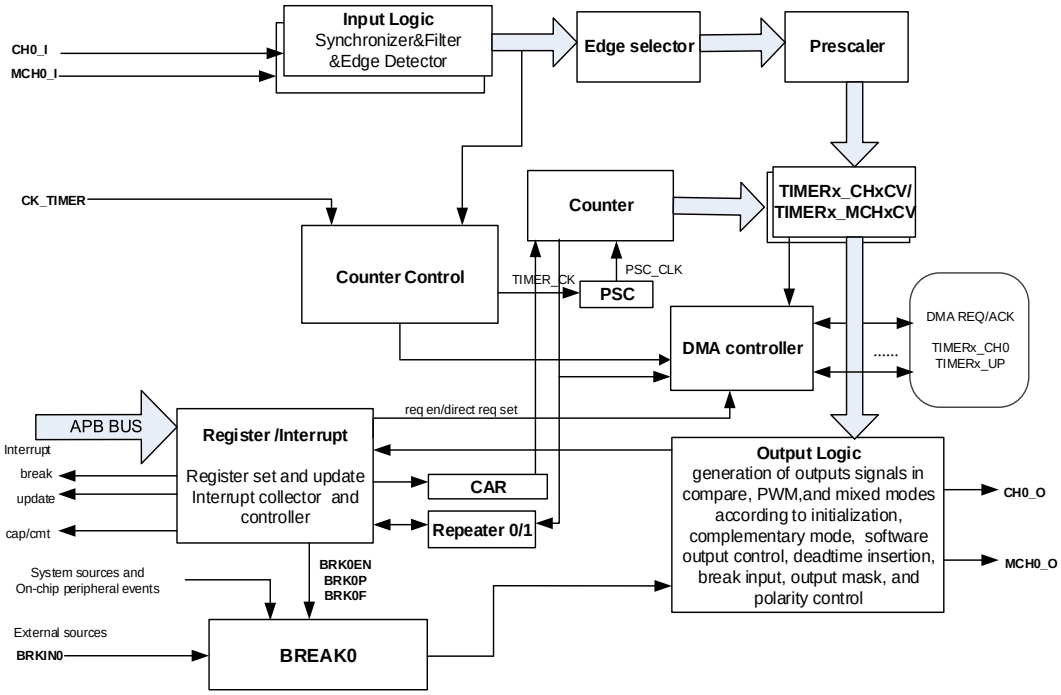
23.4.2. Characteristics

- Total channel num: 2.
- Counter width: 16-bit.
- Source of counter clock: internal clock.
- Counter modes: count up only.
- Programmable prescaler: 16-bit. The factor can be changed on the go.
- Each channel is user-configurable: input capture mode, output compare mode, programmable PWM mode, single pulse mode.
- Programmable dead time insertion.
- Auto reload function.
- Programmable counter repetition function.
- Break input function: BREAK0.
- Interrupt output or DMA request: update, compare/capture event, and break input.

23.4.3. Block diagram

Figure 23-123. General level4 timer block diagram provides details of the internal configuration of the general level4 timer.

Figure 23-123. General level4 timer block diagram



23.4.4. Function overview

Clock selection

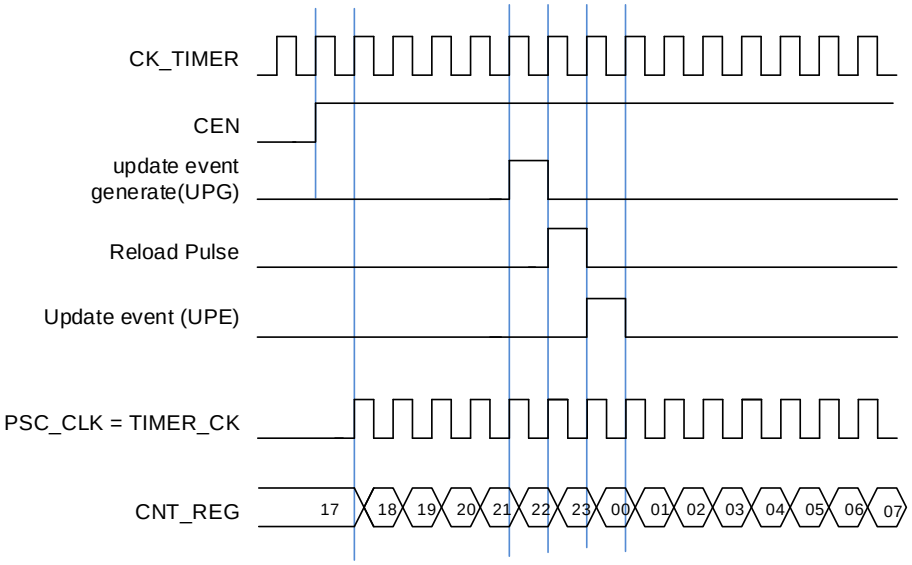
The general level4 TIMER can only being clocked by the CK_TIMER.

4) Internal timer clock CK_TIMER which is from module RCU

The general level4 TIMER has only one clock source which is the internal CK_TIMER, used to drive the counter prescaler. When the CEN is set, the CK_TIMER will be divided by PSC value to generate PSC_CLK.

The TIMER_CK, driven counter's prescaler to count, is equal to CK_TIMER which is from RCU

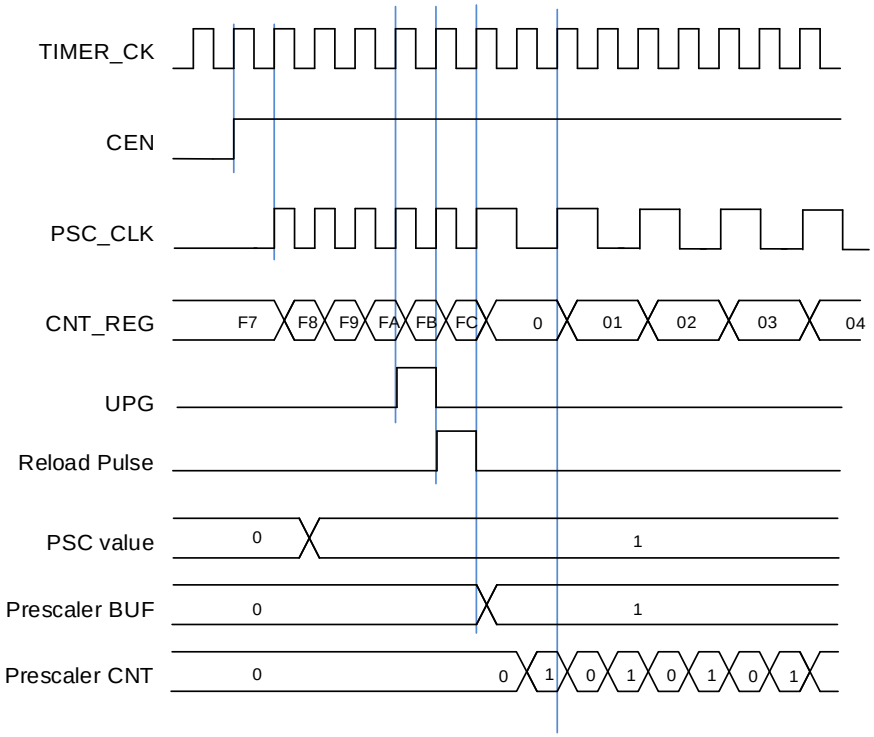
Figure 23-124. Normal mode, internal clock divided by 1



Prescaler

The prescaler can divide the timer clock (TIMER_CK) to a counter clock (PSC_CLK) by any factor 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 23-125. 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+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 23-126. Timing diagram of up counting mode, PSC=0/2 and *Figure 23-127. 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 23-126. Timing diagram of up counting mode, PSC=0/2

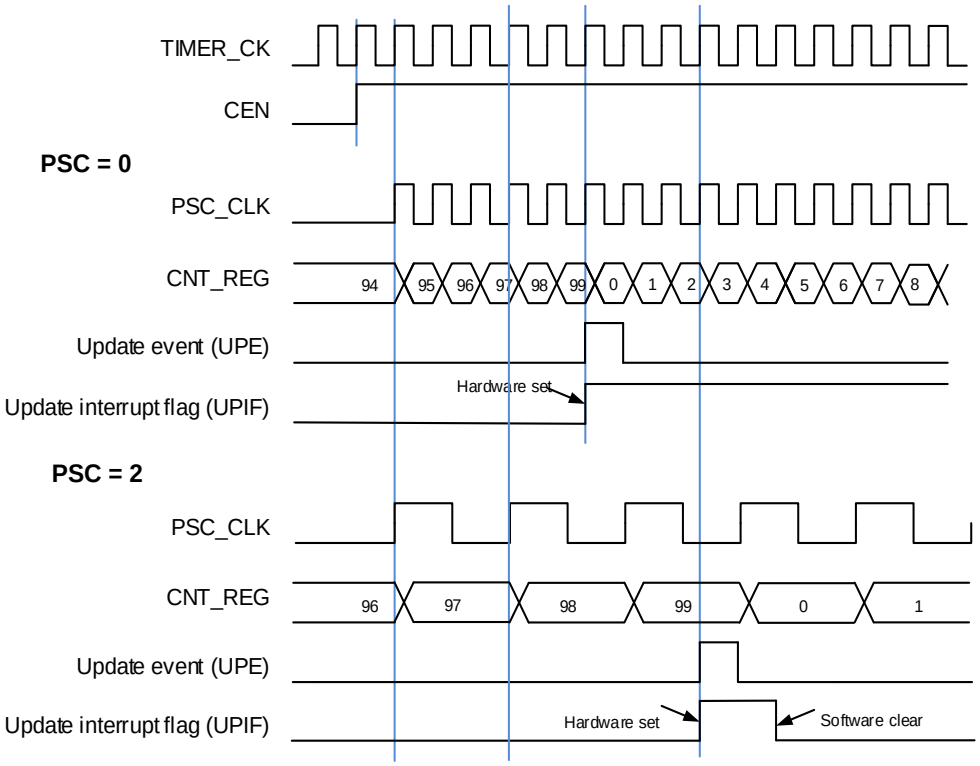
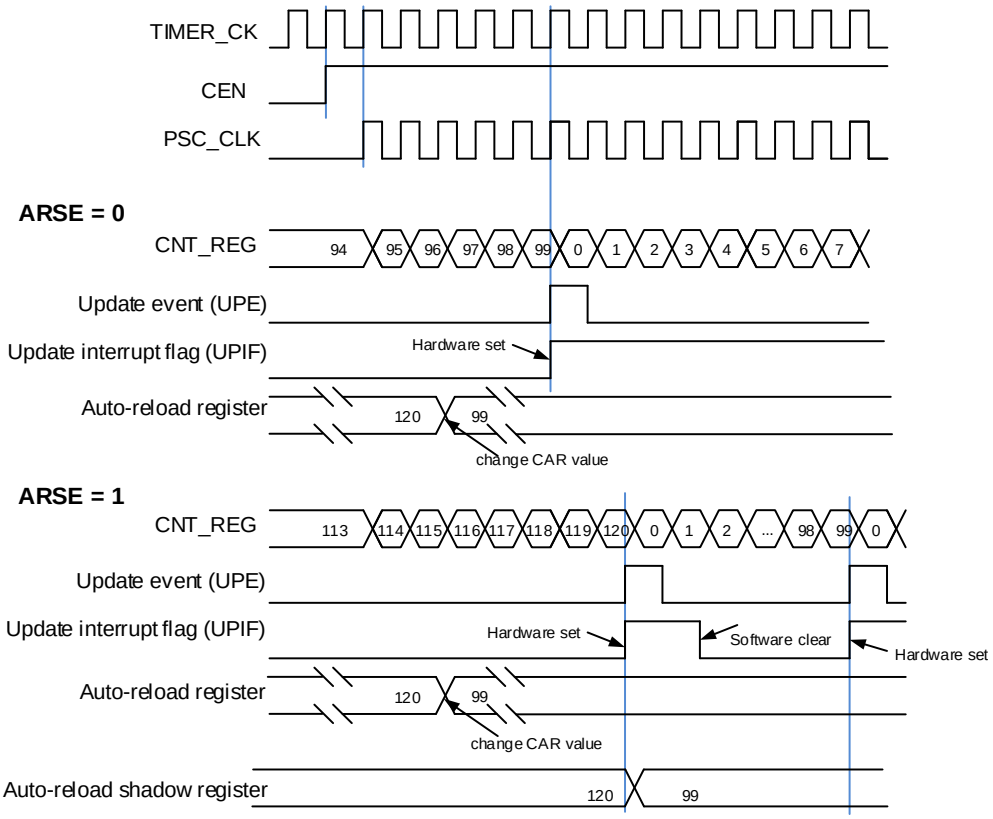


Figure 23-127. Timing diagram of up counting mode, change TIMERx_CAR on the go



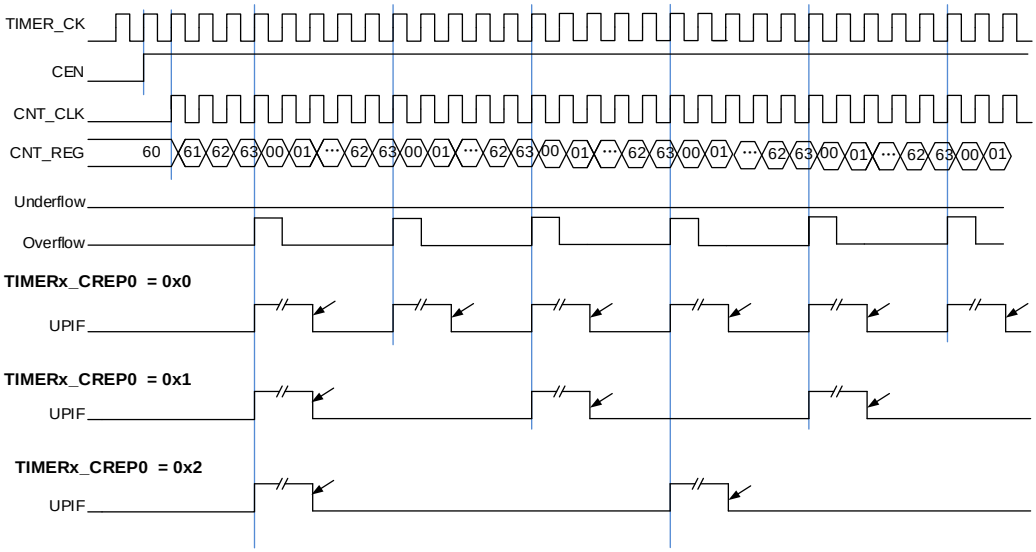
Counter repetition

The general timer has two repetitions counter TIMERx_CREP0/1, which can be selected by configuring the CPERSEL bit in the TIMERx_CFG register. The CPEP[7:0] bit-field is 8bits, the CPEP[31:0] bit-field is 32bits and can be read on the fly.

Counter repetition is used to generator update event or updates the timer registers only after a given number (N+1) of cycles of the counter, where N is CREP0/1 in TIMERx_CREP0/1 register. The repetition counter is decremented at each counter overflow in up-counting mode.

Setting the UPG bit in the TIMERx_SWEVG register will reload the content of CREP0/1 in TIMERx_CREP0/1 register and generator an update event.

Figure 23-128. Repetition timechart for up-counter



Capture/compare channels

The general level4 timer has two independent channels which can be used as capture inputs or compare match outputs. Each channel is built around a channel capture compare register including an input stage, channel controller and an output stage.

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_MCHxCV(x=0) registers, at the same time the CHxIF/ MCHxIF(x=0) bits are set and the channel interrupt is generated if it is enabled when CHxIE/ MCHxIE =1(x=0).

Figure 23-129. Input capture logic for channel 0

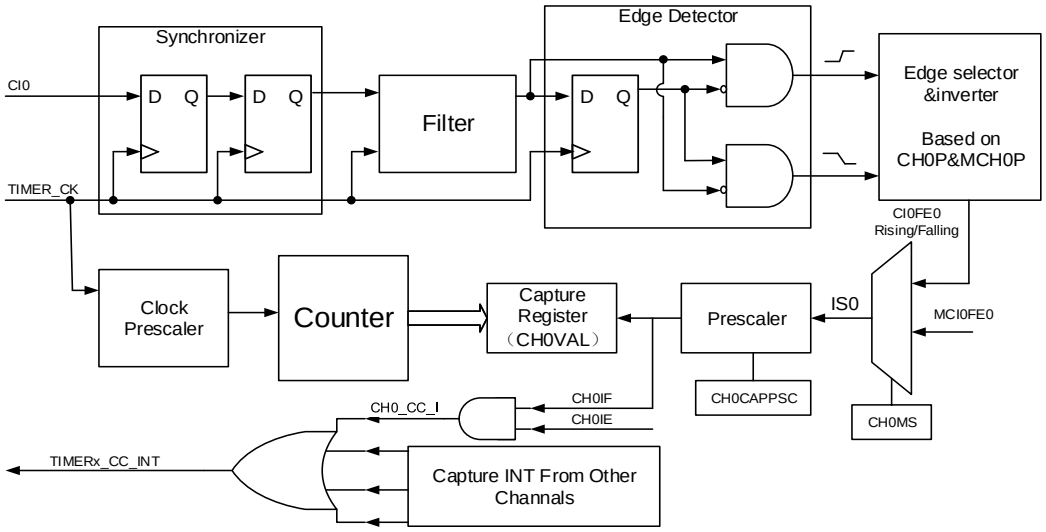
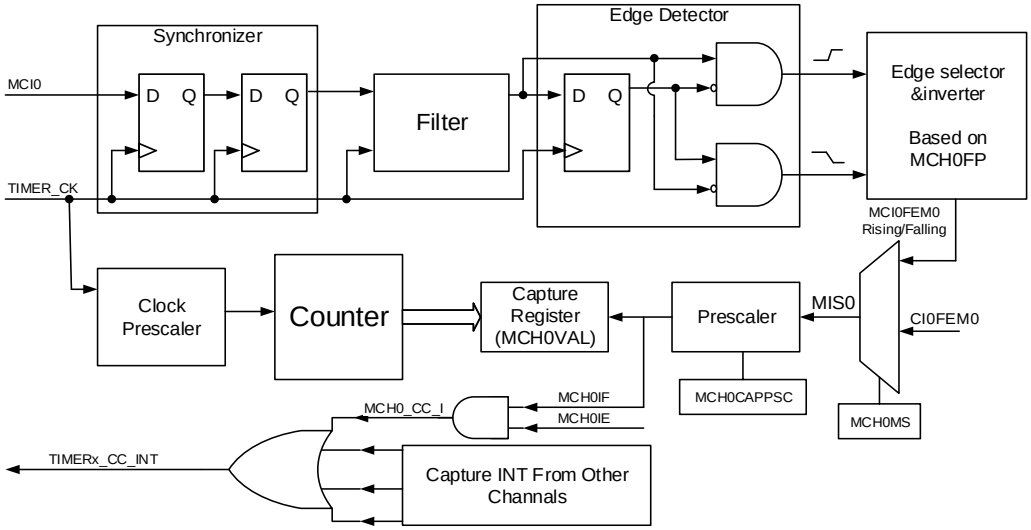


Figure 23-130. Input capture logic for multi mode channel 0



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.

■ Output compare mode

Figure 23-131. Output compare logic (when MCHxMSEL = 2'b00, x=0) and *Figure 23-132. Output compare logic (when MCHxMSEL = 2'b11, x=0)* show the logic circuit of output compare mode.

Figure 23-131. Output compare logic (when MCHxMSEL = 2'b00, x=0)

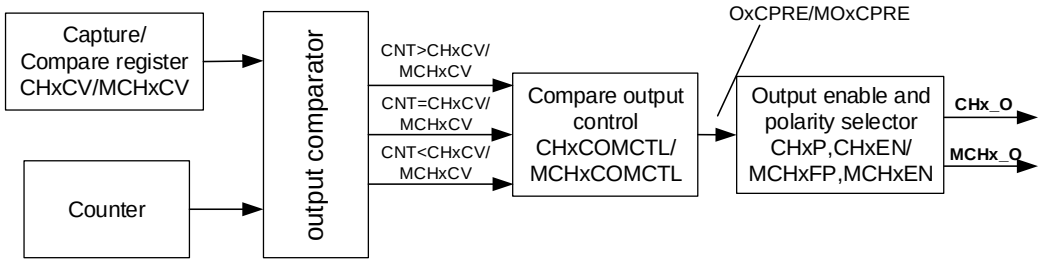
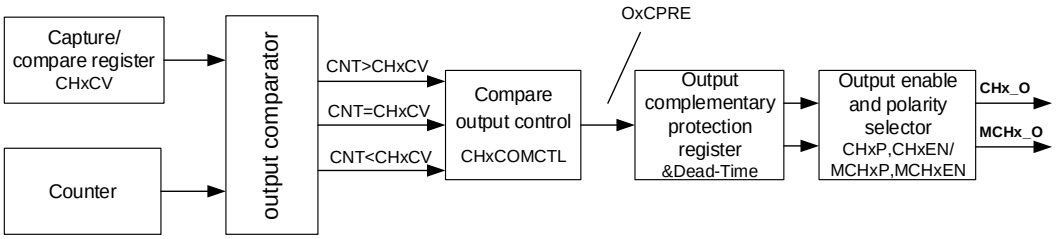


Figure 23-132. Output compare logic (when MCHxMSEL = 2'b11, x=0)



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 23-131. Output compare logic (when MCHxMSEL = 2'b00, 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 23-132. Output compare logic (when MCHxMSEL = 2'b11, x=0)***.

For examples (the MCHx_O output is independent from the CHx_O output):

5) 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.

6) 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_CCHP0 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.

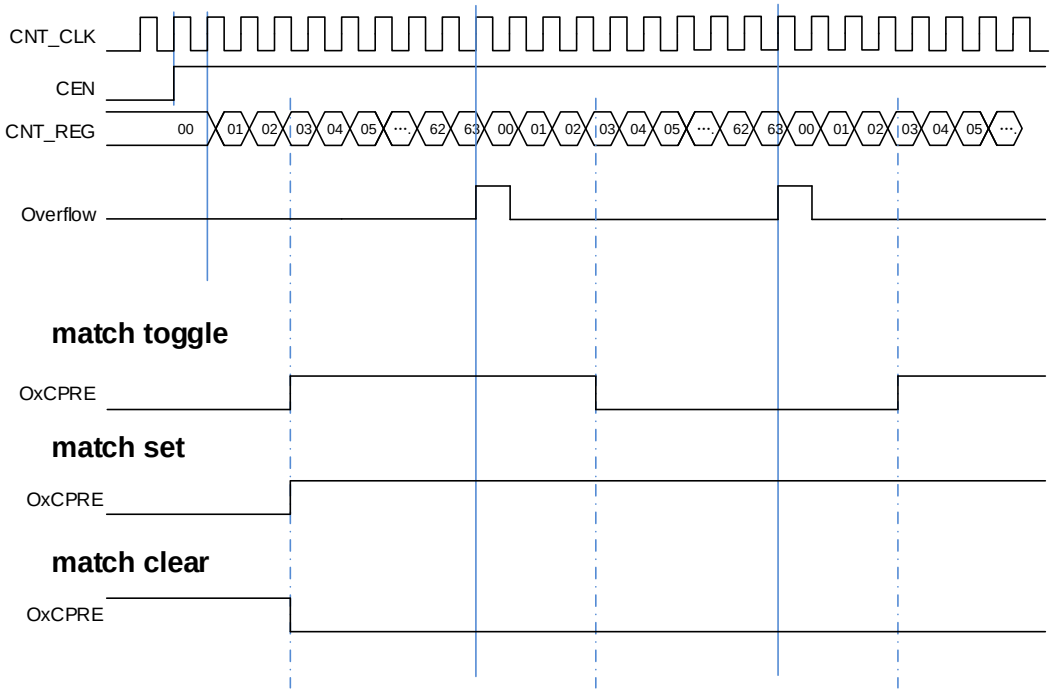
Step4: Compare output timing configuration by TIMERx_CAR and TIMERx_CHxCV/ TIMERx_MCHxCV.

The TIMERx_CHxCV/ TIMERx_MCHxCV can be changed ongoing to meet the expected waveform.

Step5: Start the counter by configuring CEN to 1.

Figure 23-133. Output-compare in three modes shows the three compare modes: toggle/set/clear. CARL=0x63, CHxVAL=0x3.

Figure 23-133. Output-compare in three modes



PWM mode

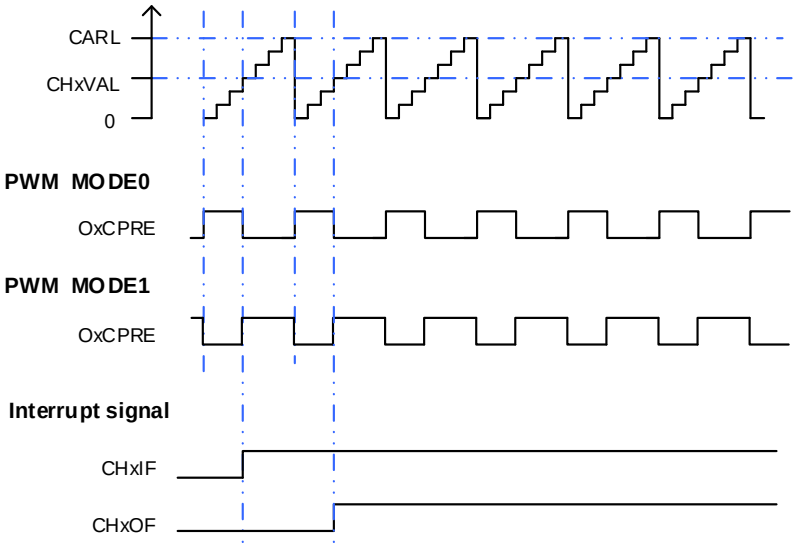
In the PWM output mode (by setting the CHxCOMCTL/ MCHxCOMCTL bit to 4'b0110 (PWM mode 0) or to 4'b0111(PWM mode 1)), the channel can generate PWM waveform according

to the TIMERx_CAR registers and TIMERx_CHxCV/ TIMERx_MCHxCV registers.

The EAPWM's period is determined by TIMERx_CAR and the duty cycle is determined by TIMERx_CHxCV/ TIMERx_MCHxCV. ***Figure 23-134. PWM mode timechart*** shows the EAPWM output and interrupts waveform.

In up counting mode, if the value of TIMERx_CHxCV/ TIMERx_MCHxCV is greater than the value of TIMERx_CAR, the output will be always active in PWM mode 0 (CHxCOMCTL/ MCHxCOMCTL =4'b0110). And if the value of TIMERx_CHxCV/ TIMERx_MCHxCV is greater than the value of TIMERx_CAR, the output will be always inactive in PWM mode 1 (CHxCOMCTL/ MCHxCOMCTL =4'b0111).

Figure 23-134. PWM mode timechart



Adjustment mode

Please refer to ***General level3 timer (TIMERx, x=14) Adjustment mode***.

Channel output prepare signal

As is shown in ***Figure 23-131. Output compare logic (when MCHxMSEL = 2'b00, x=0)*** and ***Figure 23-132. Output compare logic (when MCHxMSEL = 2'b11, x=0)***, 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.

Clear the channel output prepare signal

Please refer to ***General level3 timer (TIMERx, x=14) Clear the channel output prepare signal***.

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_CCHP0 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 23-22. Complementary outputs controlled by parameters (MCHxMSEL =2'b11)***.

Table 23-22. 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):	
			1	0	the CHx_O/ MCHx_O output inactive level firstly: CHx_O =	

Complementary Parameters					Output Status	
POEN	ROS	IOS	CHxEN	MCHxEN	CHx_O	MCHx_O
				1	CHxP, MCHx_O = CHxNP; If the clock for deadtime generator is present, after a deadtime: CHx_O = ISOx, MCHx_O = ISOxN. (3)	
		1	x	x	CHx_O/ MCHx_O output "off-state": the CHx_O/ MCHx_O output inactive level firstly: CHx_O = CHxP, MCHx_O = CHxNP; 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

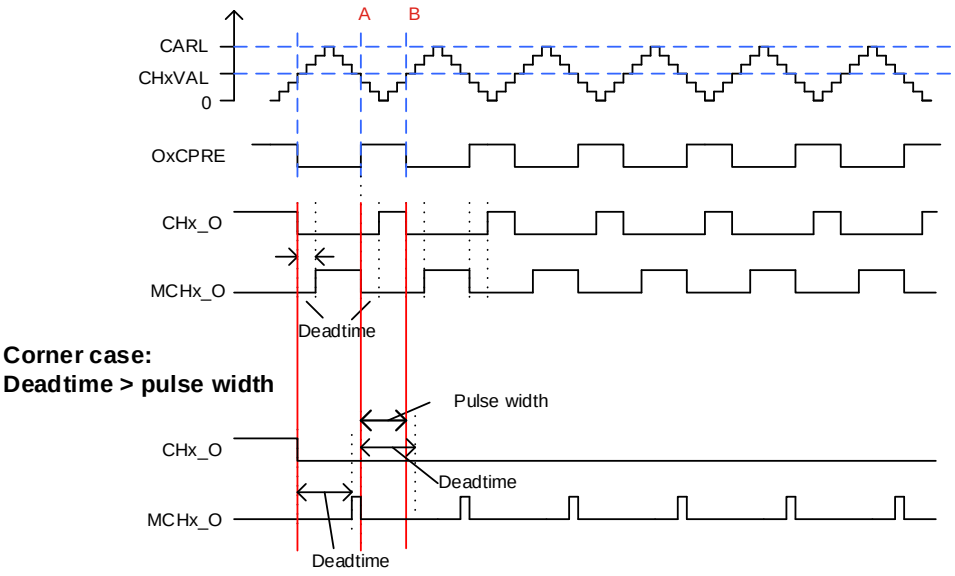
The dead time insertion is enabled when MCHxMSEL=2'b11 and both CHxEN and MCHxEN are configured to 1'b1, it is also necessary to configure POEN to 1. The field named DTCFG defines the dead time delay that can be used for all channels. Refer to the ***Complementary channel protection register 0 (TIMERx_CCHP0)*** 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 23-135. 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 23-135. Complementary output with dead-time insertion***).

Figure 23-135. Complementary output with dead-time insertion



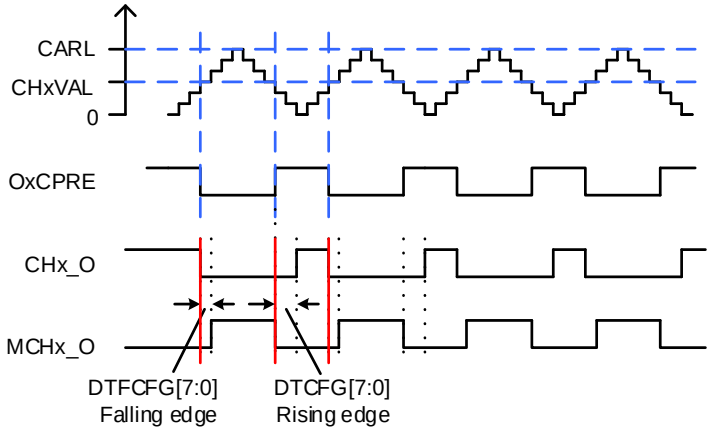
Different dead time insertion

The CHx_O and MCHx_O signal can output with different dead time when the DTDIFEN bit (in the TIMERx_CCHP1 register) is set to 1. As shown in ***Figure 23-136. 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 dead time can be modified on-the-fly when the CHx_O and MCHx_O signals are output. When DTMODEN bit in TIMERx_CCHP1 register is set, this function can be enabled. The new value of DTCFG[7:0] bit-field and DTFCFG[7:0] bit-field will be active when the next update event occurs.

Figure 23-136. Complementary output with different dead time(DTDIFEN=1)



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_CCHP0 register. The break input polarity is configured by the BRK0P bit in TIMERx_CCHP0 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_CCHP0 register, ISOx and ISOxN bits in the TIMERx_CTL1 register.

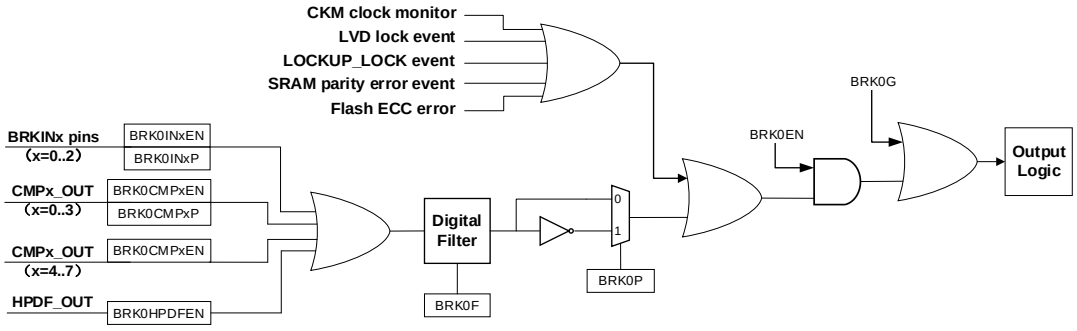
The break event is the result of logic ORed of all sources. The BREAK0 function can handle three types of event sources:

5) External sources: coming from BRKIN0 input;
6) System sources: HXTAL stuck event which is generated by Clock Monitor CKM in RCU, LVD lock event, Cortex®-M33 LOCKUP_LOCK event, SRAM parity error event or flash ECC error event;
7) On-chip peripheral events: input by comparator output or HPDF watchdog output.

BREAK0 events can also be generated by software using BRK0G bit in the TIMERx_SWEVG register.

Refer to ***Figure 23-137. BREAK0 function logic diagram***, BRKIN0 can select GPIO pins from the TRIGSEL module, which can select by ***Trigger selection for TIMER15 BRKIN register (TRIGSEL_TIMER15BRKIN)*** and ***Trigger selection for TIMER16 BRKIN register (TRIGSEL_TIMER16BRKIN)*** registers.

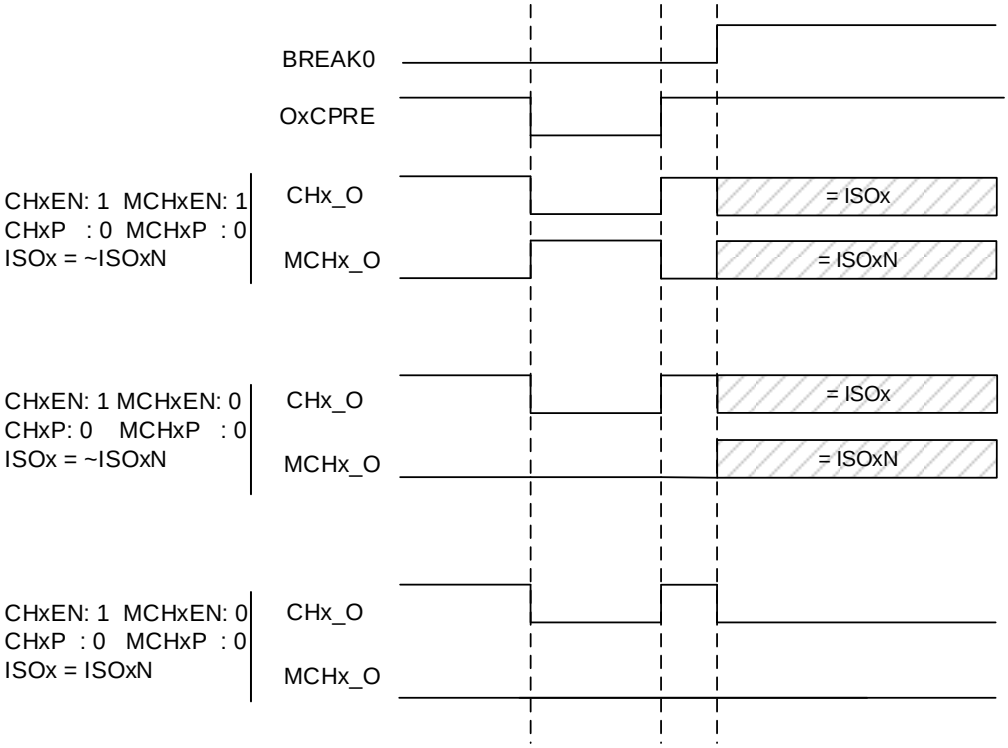
Figure 23-137. BREAK0 function logic diagram



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 23-138. 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 23-138. Output behavior of the channel in response to BREAK0 (the break input high active and IOS=1)



When a break occurs, the BRKIF bit in the TIMERx_INTF register will be set. If BRKIE is 1, an interrupt will be generated.

Locked break function

The BRKIN0 input pin of general timer have the locked break function, this function can be enabled by setting the BRK0LK bit in the TIMERx_CCHP0 register.

When the locked break function is enabled, the BRKIN0 pins need to be configured to open-drain output mode with low level active (BRK0P=0 and BRK0IN0P=0). When any break source requests occur, the BRKIN0 pin can be forced to low level. If the break input polarity is active high (BRK0P=1 and BRK0IN0P=1), the locked break function is invalid.

When the break function is enabled (the BRK0EN =1), the BRKIN0 pin can be forced to low level with the BRK0G bit setting to 1 by software.

When the break function is disabled (the BRK0EN =0), setting the BRK0G bit will have no effect on the BRKIN0 pin. The BRK0F bit will set and the channel outputs will be in a safe state.

The BRKIN0 pin can be released by setting the BRK0REL bit in the TIMERx_CCHP0 register. When the break input sources are inactive, the BRK0REL bit will cleared by hardware and the BRKIN0 pin will restore the locked break function.

In the following two cases, the BRKIN0 pin cannot be released:

8) Break input sources are active: the BRK0REL bit is set to 1 and the BRKIN0 pin locked break function is released. The break events are still active, becanse the break input sources are still active.
9) POEN=1: when the channel outputs are enabled, the BRKIN0 pin cannot be released even if the BRK0REL is set.

Table 23-23. Break function input pins locked/ released conditions

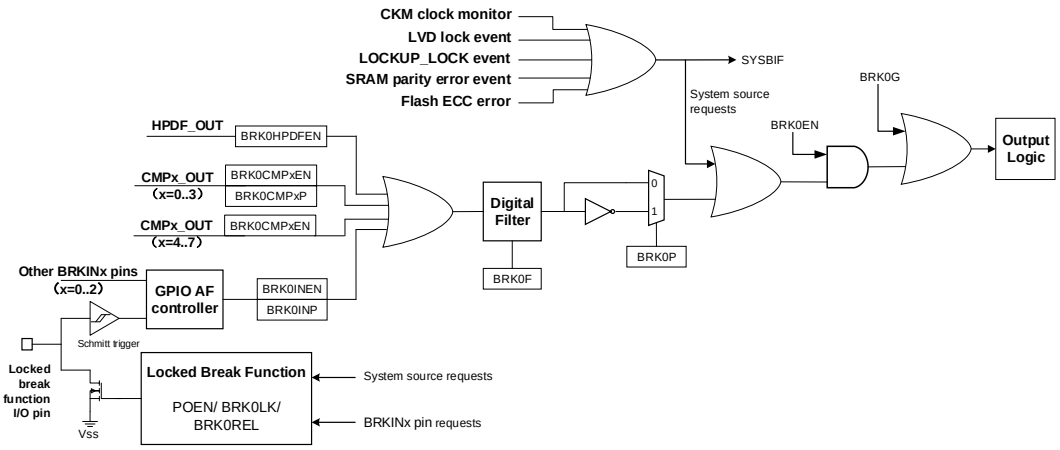
POEN	BRK0LK	BRK0REL	Break input pin state
0	1	0	Locked
	1	1	Released

The locked break function of the BREAK0 input pin BRKIN0 is enabled by default (BRK0REL=0). When the BREAK0 events occur, the following steps can be used to reconfigure the locked break function:

10) Set the BRK0REL bit to 1 and released the BRKIN0 pin;
11) The software waits for the system break sources inactive, and then clears the SYSBIF flag;
12) The software polls the state of BRK0REL bit, until the BRK0REL bit is cleared (cleared by hardware).

Then the locked break function of BREAK0 input pin is re-enabled, and the channel outputs can be restored by setting the POEN bit to 1 by software.

Figure 23-139. BRKIN0 pin logic with BREAK0 function



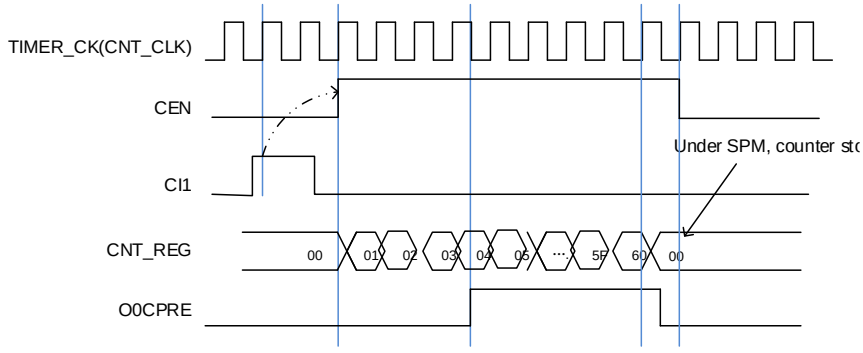
Single pulse mode

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.

Figure 23-140. Single pulse mode TIMERx_CHxCV = 0x04 TIMERx_CAR=0x60



Timer DMA mode

Timer's DMA mode is the function that configures timer's register by DMA module. The relative registers are TIMERx_DMACFG and TIMERx_DMATB. Of course, you have to enable a DMA request which will be asserted by some internal event. When the interrupt event was asserted, TIMERx will send a request to DMA, which is configured to M2P mode and PADDR is TIMERx_DMATB, then DMA will access the TIMERx_DMATB. In fact, register TIMERx_DMATB is only a buffer; timer will map the TIMERx_DMATB to an internal register, appointed by the field of DMATA in TIMERx_DMACFG. If the field of DMATC in TIMERx_DMACFG is 0(1 transfer), then the timer's DMA request is finished. While if TIMERx_DMATC is not 0, such as 3(4 transfers), then timer will send 3 more requests to DMA, and DMA will access timer's registers DMATA+0x4, DMATA+0x8, DMATA+0xc at the next 3 accesses to TIMERx_DMATB. In 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.

UPIF bit backup

The UPIF bit backup function is enabled by setting UPIFBUEN in the TIMERx_CTL0 register. The UPIF and UPIFBU bits are fully synchronized and without latency.

By using this function, the UPIF bit in the TIMERx_INTF register will be backuped to the UPIFBU bit in the TIMERx_CNT register. This can avoid conflicts when reading the counter and interrupt processing.

Timer debug mode

When the Cortex®-M33 halted, and the TIMERx_HOLD configuration bit in DBG_CTL1 register set to 1, the TIMERx counter stops.

23.4.5. Register definition (TIMERx, x=15, 16)

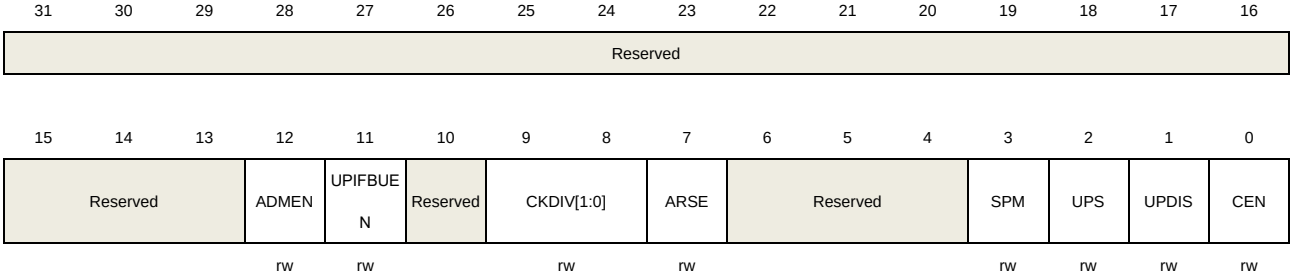
TIMER15 base address: 0x4001 4400

TIMER16 base address: 0x4001 4800

Control register 0 (TIMERx_CTL0)

Address offset: 0x00
Reset value: 0x0000 0000

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



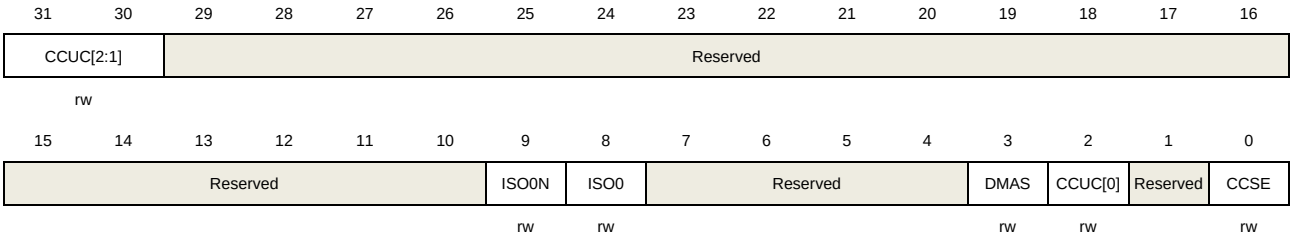
Bits	Fields	Descriptions
31:13	Reserved	Must be kept at reset value.
12	ADMEN	Adjustment mode enable 0: Adjustment mode disabled 1: Adjustment mode enabled Note: This bit can be modified only when the CEN bit is 0.
11	UPIFBUEN	UPIF bit backup enable 0: Backup disable. UPIF bit is not backuped to UPIFBU bit in TIMERx_CNT register. 1: Backup enabled. UPIF bit is backuped to UPIFBU bit in TIMERx_CNT register.
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

6:4	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: 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: Only counter overflow generates an update interrupt or a DMA request.
1	UPDIS	Update disable This bit is used to enable or disable the update event generation. 0: Update event enable. The update event is generated and the buffered registers are loaded with their preloaded values when one of the following events occurs: – The UPG bit is set. – The counter generates an overflow event. – The slave mode controller generates an update event. 1: Update event disable. The buffered registers keep their value, while the counter and the prescaler are reinitialized if the UG bit is set or the slave mode controller generates a hardware reset event.
0	CEN	Counter enable 0: Counter disable 1: Counter enable The CEN bit must be set by software when timer works in external clock mode, pause mode or decoder mode. While in event mode, the hardware can set the CEN bit automatically.

Control register 1 (TIMERx_CTL1)

Address offset: 0x04
Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:30	CCUC[2:1]	Commutation control shadow register update control Refer to CCUC [0] description.
29:10	Reserved	Must be kept at reset value.
9	ISO0N	Idle state of multi mode channel 0 complementary output 0: When POEN bit is reset, MCH0_O is set low. 1: When POEN bit is reset, MCH0_O is set high. This bit can be modified only when PROT[1:0] bits in TIMERx_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 MCH0_O is implemented. This bit can be modified only when PROT[1:0] bits in TIMERx_CCHP0 register is 00.
7:4	Reserved	Must be kept at reset value.
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: Reserved 100: The shadow registers update when the counter generates an overflow 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, 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.

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							MCH0DEN	Reserved			MCH0IE	Reserved			
							rw				rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						CH0DEN	UPDEN	BRKIE	Reserved	CMTIE	Reserved			CH0IE	UPIE
						rw	rw	rw		rw				rw	rw

Bits	Fields	Descriptions
31:25	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).
19:10	Reserved	Must be kept at reset value.
9	CH0DEN	Channel 0 capture/compare DMA request enable 0: Disabled 1: Enabled
8	UPDEN	Update DMA request enable 0: Disabled 1: Enabled
7	BRKIE	Break interrupt enable 0: Disabled 1: Enabled
6	Reserved	Must be kept at reset value.
5	CMTIE	Commutation interrupt enable

		0: Disabled 1: Enabled
4:2	Reserved	Must be kept at reset value.
1	CH0IE	Channel 0 capture/compare interrupt enable 0: Disabled 1: Enabled
0	UPIE	Update interrupt enable 0: Disabled 1: Enabled

Interrupt flag register (TIMERx_INTF)

Address offset: 0x10

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved							MCH0OF	Reserved			MCH0IF	Reserved			
							rc_w0				rc_w0				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						CH0OF	Reserved	BRK0IF	Reserved	CMTIF	Reserved			CH0IF	UPIF
						rc_w0		rc_w0		rc_w0				rc_w0	rc_w0

Bits	Fields	Descriptions
31:25	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
19:10	Reserved	Must be kept at reset value.
9	CH0OF	Channel 0 over capture flag When channel 0 is configured in input mode, this flag is set by hardware when a capture event occurs while CH0IF flag has already been set. This flag is cleared by software. 0: No over capture interrupt occurred 1: Over capture interrupt occurred
8	Reserved	Must be kept at reset value.
7	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 BREAK0 input has been detected. 1: An active level on BREAK0 input has been detected.
6	Reserved	Must be kept at reset value.
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
4:2	Reserved	Must be kept at reset value.
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											MCH0G	Reserved			
											w				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								BRK0G	Reserved	CMTG	Reserved			CH0G	UPG
								w		w				w	w

Bits	Fields	Descriptions
31:21	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 break event 1: Generate a break event
6	Reserved	Must be kept at reset value.
5	CMTG	Channel commutation event generation This bit is set by software and cleared by hardware automatically. When this bit is set, channel's capture/compare control registers (CHxEN, 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
4:2	Reserved	Must be kept at reset value.
1	CH0G	Channel 0 capture or compare event generation This bit is set by software to generate a capture or compare event in channel 0, it is automatically cleared by hardware. When this bit is set, the CH0IF flag will be set, and the corresponding interrupt or DMA request will be sent if enabled. In addition, if channel 0 is configured in input mode, the current value of the counter is captured to TIMERx_CH0CV register, and the CH0OF flag is set if the CH0IF flag has been

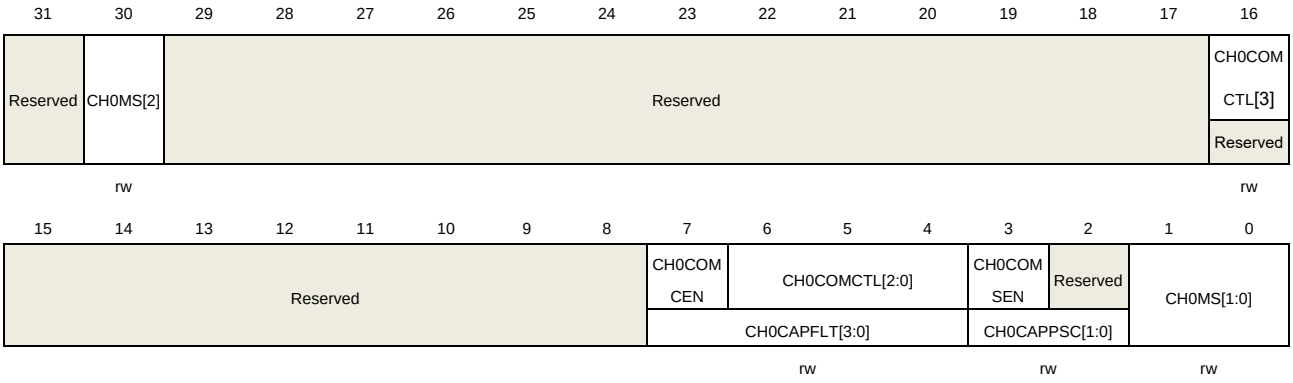
		set. 0: No generate a channel 0 capture or compare event 1: Generate a channel 0 capture or compare event
0	UPG	Update event generation This bit can be set by software, and automatically cleared by hardware. When this bit is set, the counter is cleared if the up counting mode is selected. The prescaler counter is cleared at the same time. 0: No generate an update event 1: Generate an update event

Channel control register 0 (TIMERx_CHCTL0)

Address offset: 0x18

Reset value: 0x0000 0000

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



Output compare mode:

Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30	CH0MS[2]	Channel 0 I/O mode selection Refer to CH0MS[1:0] description.
29:17	Reserved	Must be kept at reset value.
16	CH0COMCTL[3]	Channel 0 compare output control Refer to CH0COMCTL[2:0] description
15:8	Reserved	Must be kept at reset value.
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. **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**: 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	Reserved	Must be kept at reset value.
1:0	CH0MS[1:0]	Channel 0 I/O mode selection This bit-field specifies the work mode of the channel and the input signal selection. The CH0MS[2:0] bit-field is writable only when the channel is not active (When MCH0MSEL[1:0] = 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: Reserved 011: Reserved 100: Channel 0 is configured as input, IS0 is connected to MCI0FE0. 101~111: Reserved.

Input capture mode:

Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30	CH0MS[2]	Channel 0 I/O mode selection Refer to CH0MS[1:0] description.
29:8	Reserved	Must be kept at reset value.
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															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												MCH0P	MCH0EN	CH0P	CH0EN
												rw	rw	rw	rw

Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value.
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 MCH0_O output signal polarity. 0: Multi mode channel 0 output active high 1: Multi mode channel 0 output active low When CH0 is configured in input mode, in conjunction with CH0P, this bit is used to define the polarity of CH0. This bit cannot be modified when PROT[1:0] bit-field in TIMERx_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 MCH0_O signal in active state. When multi mode channel 0 is configured in input mode, setting this bit enables the capture event in multi mode channel 0. 0: Multi mode channel 0 disabled 1: Multi mode channel 0 enabled

1	CH0P	Channel 0 capture/compare polarity When channel 0 is configured in output mode, this bit specifies the output signal polarity. 0: Channel 0 active high 1: Channel 0 active low When channel 0 is configured in input mode, these bits specifie the channel 0 input signal's polarity. [MCH0P, CH0P] will select the active trigger or capture polarity for channel 0 input signals. 00: channel 0 input signal's rising edge is the active signal for capture or trigger operation in slave mode. And channel 0 input signal will not be inverted. 01: channel 0 input signal's falling edge is the active signal for capture or trigger operation in slave mode. And channel 0 input signal will be inverted. 10: Reserved. 11: Noninverted/both channel 0 input signal's edges. This bit cannot be modified when PROT[1:0] bit-field in TIMERx_CCHP0 register is 11 or 10.
0	CH0EN	Channel 0 capture/compare enable When channel 0 is configured in output mode, setting this bit enables CH0_O signal in active state. When channel 0 is configured in input mode, setting this bit enables the capture event in channel 0. 0: Channel 0 disabled 1: Channel 0 enabled

Counter register (TIMERx_CNT)

Address offset: 0x24
Reset value: 0x0000 0000

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

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

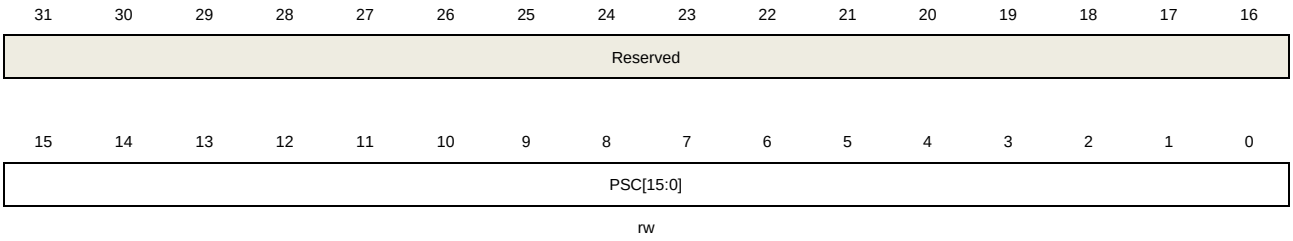
Bits	Fields	Descriptions
31	UPIFBU	UPIF bit backup This bit is a backup of UPIF bit in TIMERx_INTF register, and read-only. This bit is only valid when UPIFBUEN = 1. If the UPIFBUEN =0, this bit is reserved and read the result is 0.
30:16	Reserved	Must be kept at reset value.

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. When the PWMADMEN =0, this bit-field indicates the current counter value. Writing to this bit-field can change the value of the counter. When the PWMADMEN =1, this bit-field just indicates the integer part of the counter value, and without the fractional part.

Prescaler register (TIMERx_PSC)

Address offset: 0x28
Reset value: 0x0000 0000

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

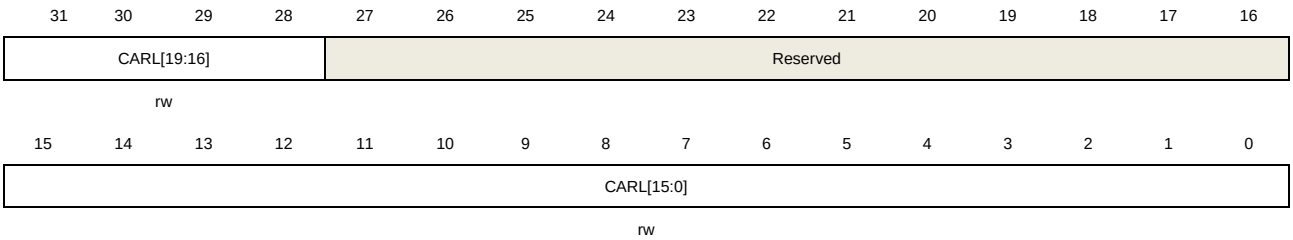


Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	PSC[15:0]	Prescaler value of the counter clock The PSC clock is divided by (PSC+1) to generate the counter clock. The value of this bit-field will be loaded to the corresponding shadow register at every update event.

Counter auto reload register (TIMERx_CAR)

Address offset: 0x2C
Reset value: 0x0000 FFFF

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



Bits	Fields	Descriptions
31:28	CARL[19:16]	Counter auto reload value (bit 16 to 19) When the PWMADMEN =0, CARL[19:16] bit-field is 0000.

		When the PWMADMEN =1, CARL[19:16] bit-field specifies fractional part of the auto reload value.
27:16	Reserved	Must be kept at reset value.
15:0	CARL[15:0]	Counter auto reload value (bit 0 to 15) When the PWMADMEN =0, CARL[15:0] bit-field specifies the auto reload value of the counter. When the PWMADMEN =1, CARL[15:0] bit-field specifies integer part of the auto reload value.

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. **Note**: This bit-field just used with CREPSEL =0 (in TIMERx_CFG register).

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[19:16]				Reserved											
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH0VAL[15:0]															

rw

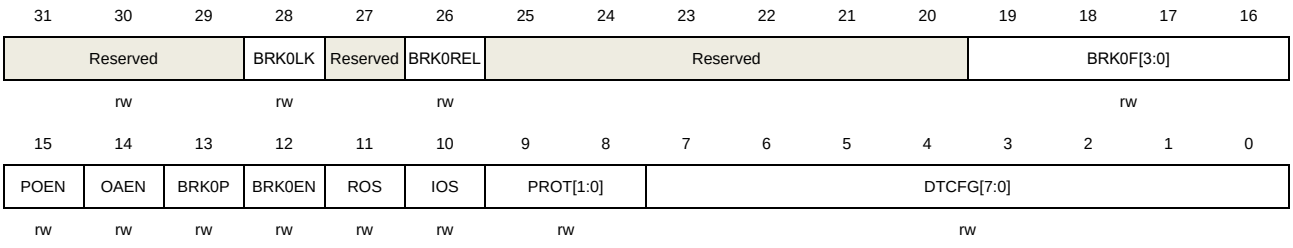
Bits	Fields	Descriptions
31:28	CH0VAL[19:16]	Capture/compare value of channel 0 (bit 16 to 19) When channel 0 is configured in input mode, CH0VAL[19:16] bit-field is 0000. When channel 0 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event. When the PWMADMEN = 0, the CH0VAL[19:16] bit-field is 0000. When the PWMADMEN =1, CH0VAL[19:16] bit-field specifies the fractional part.
27:16	Reserved	Must be kept at reset value.
15:0	CH0VAL[15:0]	Capture/compare value of channel 0 (bit 0 to 15) When channel 0 is configured in input mode, CH0VAL[15:0] 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. When the PWMADMEN = 0, the CH0VAL[15:0] bit-field specifies the compare value. When the PWMADMEN =1, CH0VAL[15:0] bit-field specifies integer part of the compare value.

Complementary channel protection register 0 (TIMERx_CCHP0)

Address offset: 0x44

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	BRK0LK	BREAK0 input locked 0: BREAK0 input in input mode 1: BREAK0 input in locked mode When the BRK0LK is set to 1, the BREAK0 input is configured in open drain output mode.

		Any active break event asserts a low logic level on the BREAK0 input to indicate an internal break event to external devices. This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00. **Note**: Every write operation to this bit needs a delay of 1 APB clock to active.
27	Reserved	Must be kept at reset value.
26	BRK0REL	BREAK0 input released This bit is cleared by hardware when the break input is invalid. 0: BREAK0 input is unreleased 1: BREAK0 input is released The locked output control (open drain mode in Hi-z state) is released by setting this bit with software. And when the fault is disappeared, this bit will reset by hardware. **Note**: Every write operation to this bit needs a delay of 1 APB clock to active.
25:20	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 23-22. Complementary outputs controlled by parameters (MCHxMSEL =2’b11)***
0: “off-state” disabled. If the CHxEN or CHxNEN bit is reset, the corresponding channel is output disabled.
1: “off-state” enabled. If the CHxEN or CHxNEN bit is reset, the corresponding channel is “off-state”.
This bit cannot be modified when PROT [1:0] bit-filed in TIMERx_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 23-22. Complementary outputs controlled by parameters (MCHxMSEL =2’b11)***.
0: “off-state” disabled. If the CHxEN/CHxNEN bits are both reset, the channels are output disabled.
1: “off-state” enabled. No matter the CHxEN/CHxNEN bits, the channels are “off-state”.

This bit cannot be modified when PROT [1:0] bit-filed in TIMERx_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.

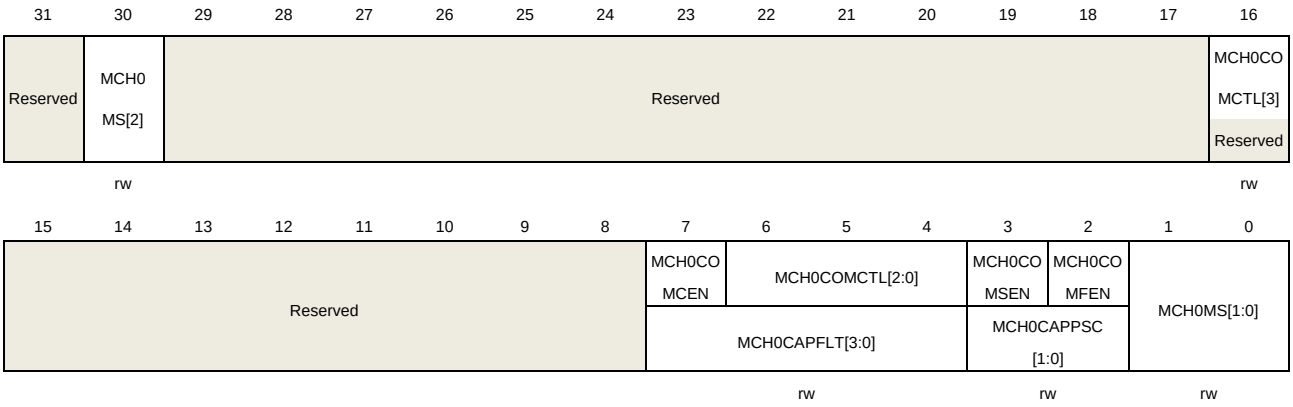
This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.

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.
15:8	Reserved	Must be kept at reset value.
7	MCH0COMCEN	Multi mode channel 0 output compare clear enable. When this bit is set, the MO0CPRE signal is cleared when high level is detected on ETIFP input. 0: Multi mode channel 0 output compare clear disabled. 1: Multi mode channel 0 output compare clear enabled.
6:4	MCH0COMCTL[2:0]	Multi mode channel 0 output compare control When multi mode channel 0 is configured in output mode, and the MCH0MSEL[1:0] = 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 CH0NMS bit-field is 00 (compare mode).
3	MCH0COMSEN	Multi mode channel 0 output compare shadow enable When this bit is set, the shadow register of TIMERx_MCH0CV register which updates at each update event will be enabled. 0: Multi mode channel 0 output compare shadow disabled 1: Multi mode channel 0 output compare shadow enabled The PWM mode can be used without 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 PWM0 or PWM1 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. The minimum delay from an edge on the trigger input to activate MCH0_O output is 5 clock cycles. 1: Multi mode channel 0 output quickly compare enable. The minimum delay from an edge on the trigger input to activate MCH0_O output is 3 clock cycles.
1:0	MCH0MS[1:0]	Multi mode channel 0 I/O mode selection This bit-field specifies the work mode of the channel and the input signal selection. This bit-field is writable only when the channel is not active (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: Reserved. 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.

Control register 2 (TIMERx_CTL2)

Address offset: 0x74

Reset value: 0x0030 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										MCH0MSEL[1:0]		Reserved			
										rw					

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

Bits	Fields	Descriptions
31:22	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
19:0	Reserved	Must be kept at reset value.

TIMER 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			BRK0CMP3P	BRK0CMP2P	BRK0CMP1P	BRK0CMP0P	Reserved								BRK0IN0P
			rw	rw	rw	rw									rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BRK0CMP6EN	BRK0CMP5EN	BRK0CMP4EN	BRK0CMP3EN	BRK0CMP2EN	BRK0CMP1EN	BRK0CMP0EN	BRK0HPDFEN	BRK0CMP7EN	Reserved						BRK0INON
rw	rw	rw	rw	rw	rw	rw	rw	rw							rw

Bits	Fields	Descriptions
31:29	Reserved	Must be kept at reset value.
28	BRK0CMP3P	BREAK0 CMP3 input polarity This bit is used to configure the CMP3 input polarity, and the specific polarity is determined by this bit and the BRK0P bit. 0: CMP3 input signal will not be inverted (BRK0P =0, the input signal is active low; BRK0P =1, the input signal is active high) 1: CMP3 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.
27	BRK0CMP2P	BREAK0 CMP2 input polarity This bit is used to configure the CMP2 input polarity, and the specific polarity is determined by this bit and the BRK0P bit. 0: CMP2 input signal will not be inverted (BRK0P =0, the input signal is active low; BRK0P =1, the input signal is active high) 1: CMP2 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.
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	BRK0CMP6EN	BREAK0 CMP6 enable 0: CMP6 input disabled 1: CMP6 input enabled This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.
14	BRK0CMP5EN	BREAK0 CMP5 enable 0: CMP5 input disabled 1: CMP5 input enabled This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.
13	BRK0CMP4EN	BREAK0 CMP4 enable 0: CMP4 input disabled 1: CMP4 input enabled This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.
12	BRK0CMP3EN	BREAK0 CMP3 enable 0: CMP3 input disabled 1: CMP3 input enabled This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.
11	BRK0CMP2EN	BREAK0 CMP2 enable 0: CMP2 input disabled 1: CMP2 input enabled This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.
10	BRK0CMP1EN	BREAK0 CMP1 enable 0: CMP1 input disabled

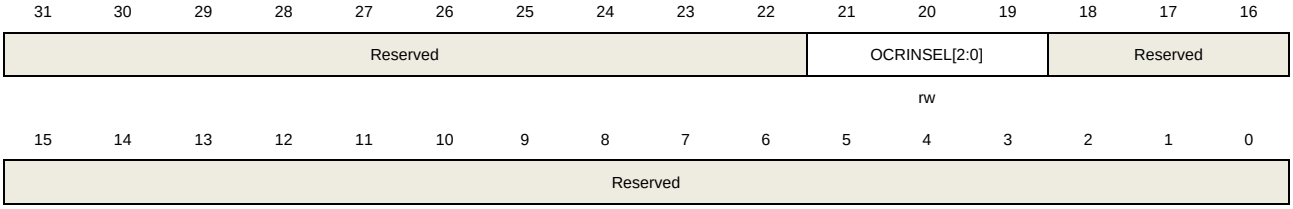
		1: CMP1 input enabled This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.
9	BRK0CMP0EN	BREAK0 CMP0 enable 0: CMP0 input disabled 1: CMP0 input enabled This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.
8	BRK0HPDFEN	BREAK0 HPDF input 0: HPDF input disabled 1: HPDF input enabled **Note**: HPDF inputs are different for TIMER15 (HPDF_BREAK[1]) and TIMER16 (HPDF_BREAK[2]). This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.
7	BRK0CMP7EN	BREAK0 CMP7 enable 0: CMP7 input disabled 1: CMP7 input enabled This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.
6:1	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.

TIMER alternate function control register 1 (TIMERx_AFCTL1)

Address offset: 0x90

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions

31:22	Reserved	must be kept at reset value
21:19	OCRINSEL[2:0]	OCPRE_CLR inputs selection 000: OCPRE_CLR0 001: OCPRE_CLR1 … 111: OCPRE_CLR7

OCPRE_CLR inputs selection	TIMER15 / 16
OCPRE_CLR0	CMP0_OUT
OCPRE_CLR1	CMP1_OUT
OCPRE_CLR2	CMP2_OUT
OCPRE_CLR3	CMP3_OUT
OCPRE_CLR4	CMP4_OUT
OCPRE_CLR5	CMP5_OUT
OCPRE_CLR6	CMP6_OUT
OCPRE_CLR7	CMP7_OUT

This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.

18:0	Reserved	must be kept at reset value

Counter repetition register 1 (TIMERx_CREP1)

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
CREP1[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CREP1[15:0]															
rw															

Bits	Fields	Descriptions
31:0	CREP1[31:0]	Counter repetition value 1 This bit-field is 32 bits and can be read on the fly. 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. **Note**: This bit-field just used with CREPSEL =1(in TIMERx_CFG register).

Complementary channel protection register 1 (TIMERx_CCHP1)

Address offset: 0x09C

Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:18	Reserved	Must be kept at reset value.
17	DTMODEN	Dead time modified on-the-fly enable 0: Dead time value modified on-the-fly disable 1: Dead time value modified on-the-fly enable This bit can be modified only when PROT [1:0] bit-filed in TIMERx_CCHP0 register is 00.
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_CCHP1 register. 1: The dead time on rising edge is defined in DTCFG[7:0] bit-field in TIMERx_CCHP1 register, and the dead time on falling edge is defined in DTFCFG [7:0] bit-field in TIMERx_CCHP2 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.

DMA configuration register (TIMERx_DMACFG)

Address offset: 0xE0

Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value.
13:8	DMATC[5:0]	DMA transfer count This field defines the times of accessing(R/W) the TIMERx_DMATB register by DMA. 6'b000000: transfer 1 time 6'b000001: transfer 2 times … 6'b111000: transfer 57 times
7:6	Reserved	Must be kept at reset value.
5:0	DMATA[5:0]	DMA transfer access start address This field defines the start address of accessing the TIMERx_DMATB register by DMA. When the first access to the TIMERx_DMATB register is done, this bit-field specifies the address just accessed. And then the address of the second access to the TIMERx_DMATB register will be (start address + 0x4). 6'b000000: TIMERx_CTL0 6'b000001: TIMERx_CTL1 … In a word: start address = TIMERx_CTL0 + DMATA*4

DMA transfer buffer register (TIMERx_DMATB)

Address offset: 0xE4

Reset value: 0x0000 0000

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

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

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

DMATB[15:0]
rw

Bits	Fields	Descriptions
31:0	DMATB[31:0]	DMA transfer buffer When a read or write operation is assigned to this register, the register located at the address ranges from (start address) to (start address + transfer count * 4) will be accessed. The transfer count is calculated by hardware, and ranges from 0 to DMATC.

Configuration register (TIMERx_CFG)

Address offset: 0xFC
Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												CCUSEL	CREPSEL	CHVSEL	OUTSEL
												rw	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	CREPSEL	The counter repetition register select This bit is used to select the counter repetition register. 0: The update event rate is depended to TIMERx_CREP0 register 1: The update event rate is depended to TIMERx_CREP1 register
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)

23.5.1. Overview

The basic timer module(TIMER5 / 6) has a 16-bit counter that can be used as an unsigned counter. The basic timer can be configured to generate a DMA request and a TRGO0 to connect to DAC.

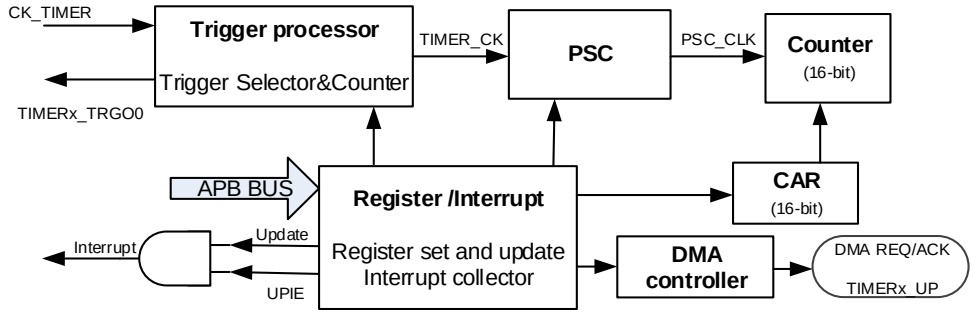
23.5.2. Characteristics

- Counter width: 16 bits.
- Source of count clock is internal clock only.
- Multiple counter modes: count up.
- Programmable prescaler: 16 bits. The factor can be changed ongoing.
- Auto reload function.
- Interrupt output or DMA request: update event.

23.5.3. Block diagram

Figure 23-141. Basic timer block diagram provides details on the internal configuration of the basic timer.

Figure 23-141. Basic timer block diagram



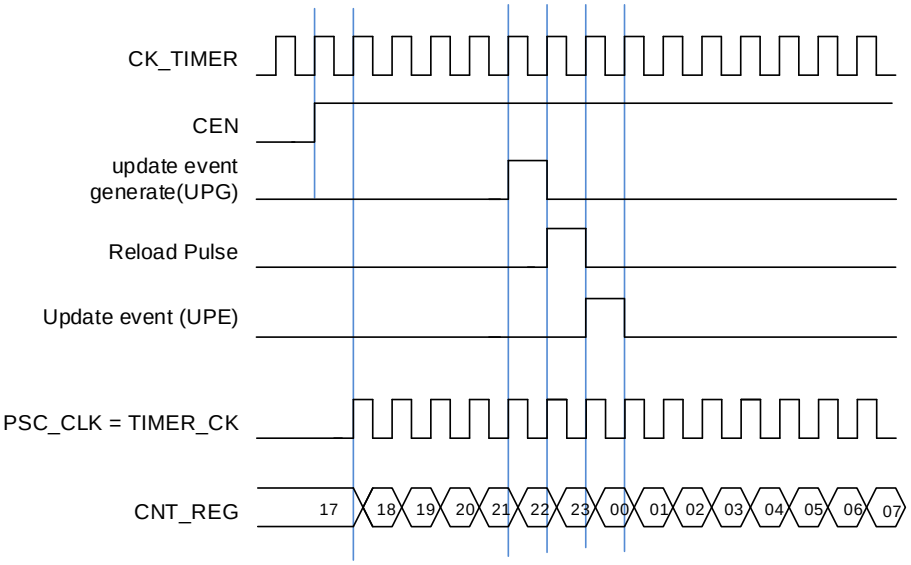
23.5.4. Function overview

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.

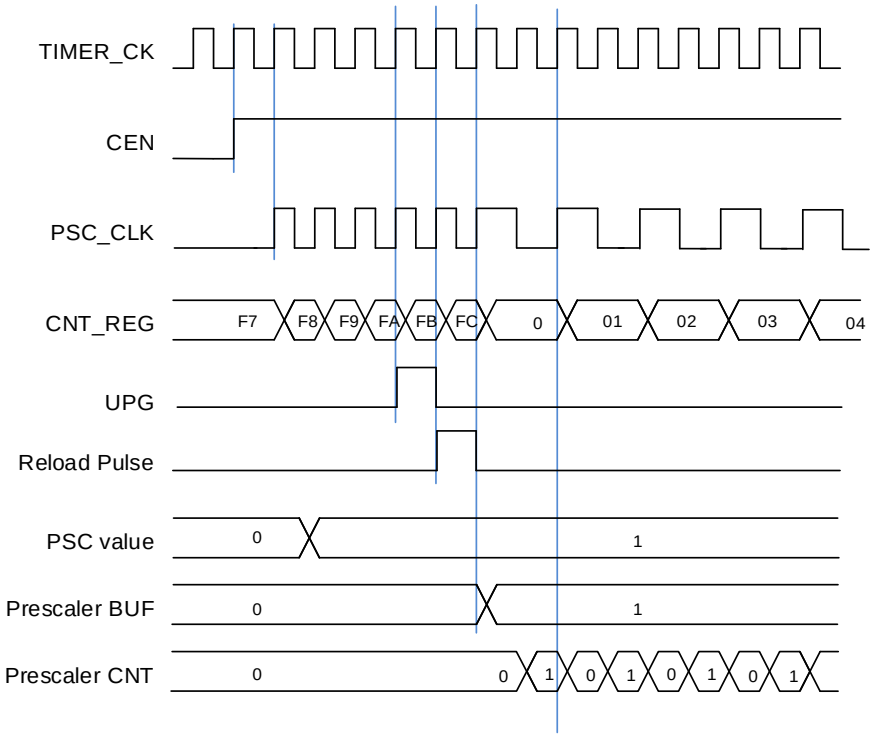
Figure 23-142. Normal mode, internal clock divided by 1



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 23-143. 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 23-144. Timing chart of up counting mode, PSC=0/1

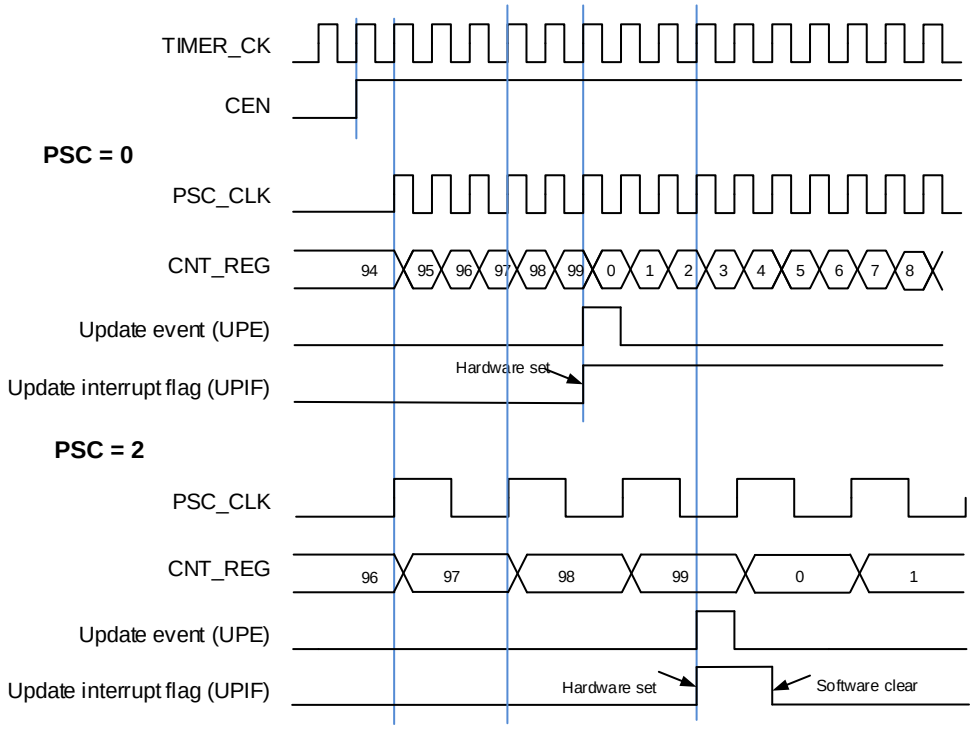
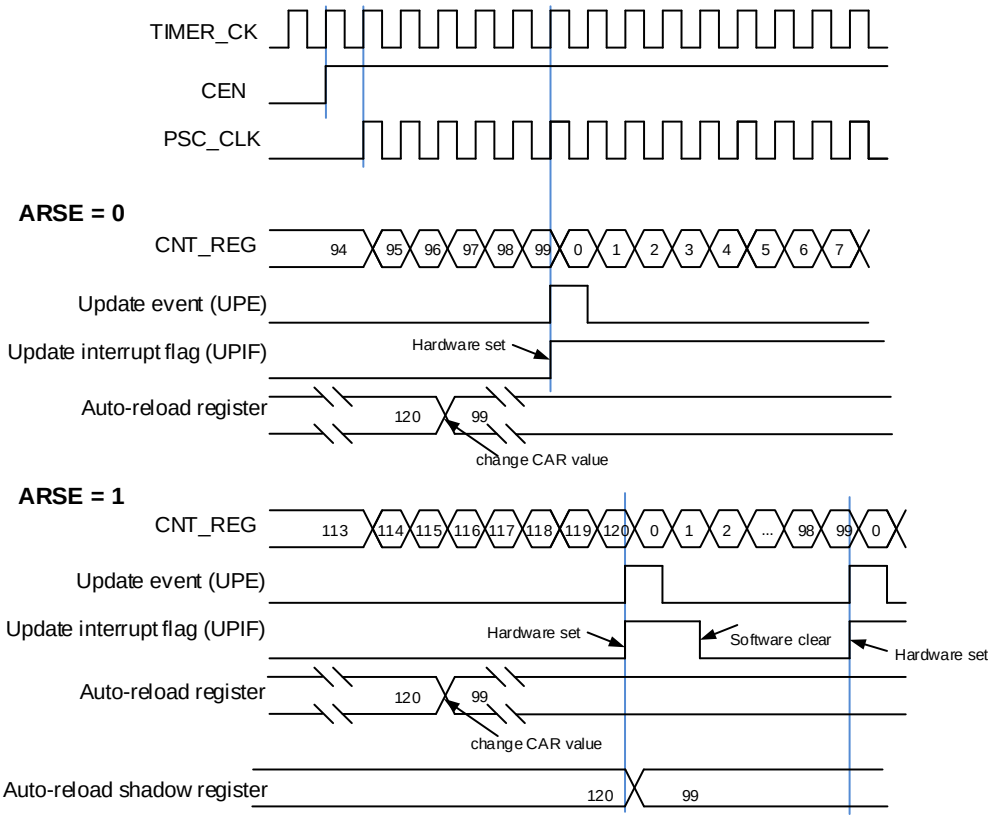


Figure 23-145. Timing chart of up counting mode, change TIMERx_CAR ongoing

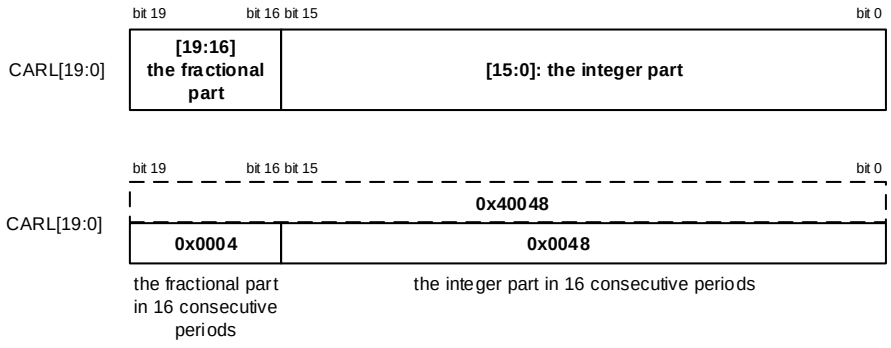


Adjustment mode

Adjustment mode is enabled by setting ADMEN bit in TIMERx_CTL0 register to 1, and this mode can improve the effective resolution of the time base. And the time base resolution can be improved by the CARL[19:0] bit-field in the TIMERx CAR register.

When the adjustment mode is enabled, the low 16 bits CARL[15:0] bit-field are used for the integer part and the high 4 bits CARL[19:16] are used for the fractional part. By adjust the CARL value over 16 consecutive periods (no more than one TIMER clock cycle at a time) in a predefined way, can increase 16-fold in resolution.

Figure 23-146. Adjustment mode: Data format and the register bit-field



Depending on the configuration of the ADMEN bit (set or clear), the CHxVAL and CARL bit-

field are automatically updated. To clear the ADMEN bit, must follow the following steps:

1. CEN bit and ARSE bits must be cleared;
2. CARL[19:16] bit-field must be cleared;
3. ADMEN bit must be cleared;
4. Set the CEN bit to 1.

The following formula to calculate the time base Resolution:

$$\text{Resolution} = f_{PSC_CLK}/f_{timebase} \quad (23\text{-}12)$$

According to Equation ***(23-12)***, when the adjustment mode is disabled (ADMEN=0), the time base minimum frequency $f_{timebase}$:

$$(f_{timebase})_{min} = f_{PSC_CLK}/65536 \quad (23\text{-}13)$$

When the adjustment mode is enabled (ADMEN=1),

$$(f_{timebase})_{min} = f_{PSC_CLK}/(65535+15/16) \quad (23\text{-}14)$$

When the adjustment mode is enabled, the max values of CARL[19:0] bit-field are 0xFFFFE (the integer part is 0xFFFE, the fractional part is 0xF)。

The changes of period within 16 consecutive periods are shown in ***Figure 23-147. Adjustment mode schematic diagram*** and ***Table 23-24. CARL bit-field change in edge-aligned***.

Figure 23-147. Adjustment mode schematic diagram

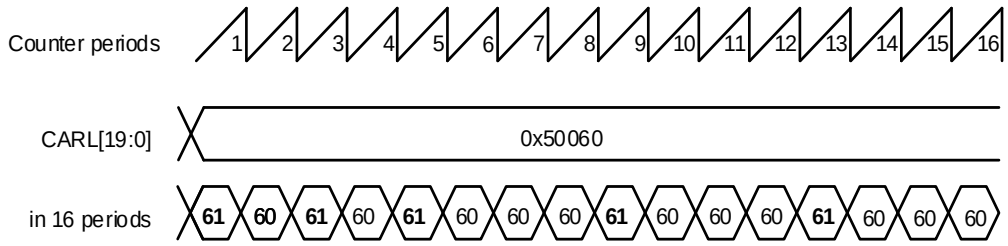


Table 23-24. CARL bit-field change in edge-aligned

CARL[19:16]	Period															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0000	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0001	+1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0010	+1	-	-	-	-	-	-	-	+1	-	-	-	-	-	-	-
0011	+1	-	-	-	+1	-	-	-	+1	-	-	-	-	-	-	-
0100	+1	-	-	-	+1	-	-	-	+1	-	-	-	+1	-	-	-
0101	+1	-	+1	-	+1	-	-	-	+1	-	-	-	+1	-	-	-
0110	+1	-	+1	-	+1	-	-	-	+1	-	+1	-	+1	-	-	-
0111	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	-	-
1000	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-
1001	+1	+1	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-
1010	+1	+1	+1	-	+1	-	+1	-	+1	+1	+1	-	+1	-	+1	-

CARL[19:16]	Period															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1011	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-	+1	-	+1	-
1100	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-
1101	+1	+1	+1	+1	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-
1110	+1	+1	+1	+1	+1	+1	+1	-	+1	+1	+1	+1	+1	+1	+1	-
1111	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	-

UPIF bit backup

The UPIF bit backup function is enabled by setting UPIFBUEN in the TIMERx_CTL0 register. The UPIF and UPIFBU bits are fully synchronized and without latency.

By using this function, the UPIF bit in the TIMERx_INTF register will be backuped to the UPIFBU bit in the TIMERx_CNT register. This can avoid conflicts when reading the counter and interrupt processing.

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)

TIMER5 base address: 0x4000 1000

TIMER6 base address: 0x4000 1400

Control register 0 (TIMERx_CTL0)

Address offset: 0x00
Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			ADMEN	UPIFBUEN	Reserved			ARSE	Reserved			SPM	UPS	UPDIS	CEN
			rw	rw				rw				rw	rw	rw	rw

Bits	Fields	Descriptions
31:13	Reserved	Must be kept at reset value.
12	ADMEN	Adjustment mode enable 0: Adjustment mode disabled 1: Adjustment mode enabled Note: This bit can be modified only when the CEN bit is 0.
11	UPIFBUEN	UPIF bit backup enable 0: Backup disable. UPIF bit is not backuped to UPIFBU bit in TIMERx_CNT register. 1: Backup enabled. UPIF bit is backuped to UPIFBU bit in TIMERx_CNT register.
10:8	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 TRGO0 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 TRGO0 pulse occurs. And in the latter case, the signal on TRGO0 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 TRGO0. 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 TRGO0 output, except if the master-slave mode is selected. 010: Update. In this mode, the master mode controller selects the update event as TRGO0. 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).

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

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

Bits	Fields	Descriptions
31:9	Reserved	Must be kept at reset value.
8	UPDEN	Update DMA request enable 0: Disabled 1: Enabled
7:1	Reserved	Must be kept at reset value.
0	UPIE	Update interrupt enable 0: Disabled 1: Enabled

Interrupt flag register (TIMERx_INTF)

Address offset: 0x10
Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															UPIF
															rc_w0

Bits	Fields	Descriptions
31:1	Reserved	Must be kept at reset value.
0	UPIF	Update interrupt flag This bit is set by hardware 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															UPG
															w

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).

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

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

Bits	Fields	Descriptions
31	UPIFBU	UPIF bit backup This bit is a backup of UPIF bit in TIMERx_INTF register and read-only. This bit is only valid when UPIFBUEN = 1. If the UPIFBUEN =0, this bit is reserved and read the result is 0.
30:16	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. When the PWMADMEN =0, this bit-field indicates the current counter value. Writing to this bit-field can change the value of the counter. When the PWMADMEN =1, this bit-field just indicates the integer part of the counter value, and without the fractional part.

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
CARL[19:16]				Reserved											
rw															

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

Bits	Fields	Descriptions
31:28	CARL[19:16]	Counter auto reload value (bit 16 to 19) When the PWMADMEN =0, CARL[19:16] bit-field is 0000. When the PWMADMEN =1, CARL[19:16] bit-field specifies fractional part of the auto reload value.
27:16	Reserved	Must be kept at reset value.
15:0	CARL[15:0]	Counter auto reload value When the PWMADMEN =0, CARL[15:0] bit-field specifies the auto reload value of the counter. When the PWMADMEN =1, CARL[15:0] bit-field specifies integer part of the auto reload value.

24. Low power timer (LPTIMER)

24.1. Overview

The LPTIMER is a 16-bit timer and it is able to keep running in all power modes except for standby mode with its diversity of clock sources. The LPTIMER provides a flexible mechanism of the clock, which reduces the power consumption to a minimum while also achieving the required functions and performance.

The LPTIMER can be used as a pulse counter with no internal clock source. The LPTIMER has the ability to wake up the system from the low-power modes, and it is suitable for realizing timeout mode with very low power consumption.

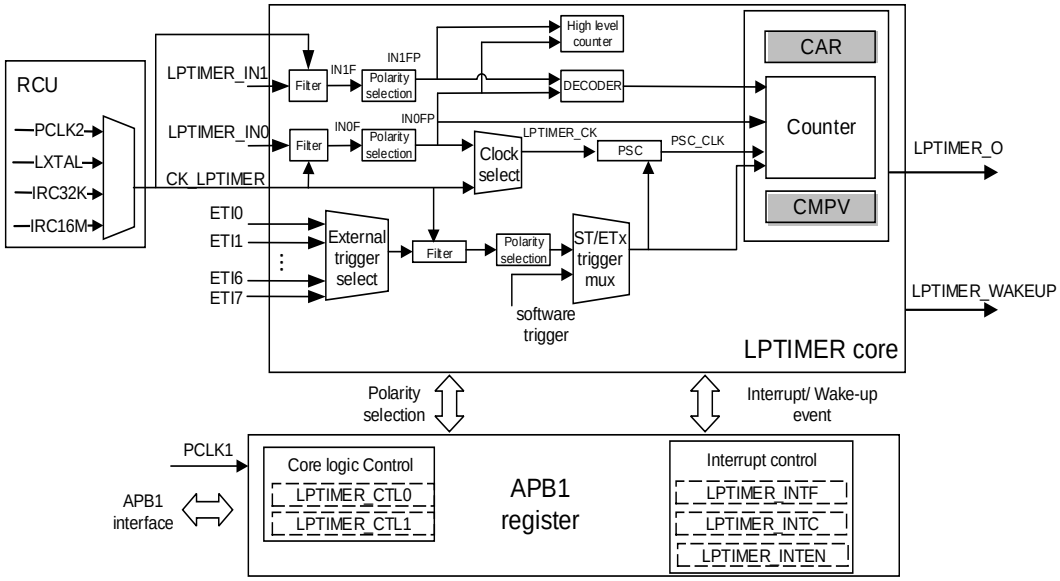
24.2. Characteristics

- Counter width: 16-bit.
- Source of counter clock is selectable:
  - Internal clock: an internal 8 MHz RC oscillator (IRC8M), an Internal 32 KHz RC oscillator (IRC32K), a 32.768 KHz Low Speed crystal oscillator (LXTAL), or an APB2 clock (PCLK2).
  - External clock: the sources through LPTIMER external input 0 (used as a pulse counter).
- Counter modes: count up.
- Operating mode: continuous counting mode or single counting mode.
- Programmable prescaler: 3 bit.
- Channel output is user-configurable:
  Programmable PWM mode, single pulse mode, set mode.
- Auto reload function.
- Interrupt output.
- Selectable trigger: software trigger or hardware input trigger.
- Decoder mode: decoder mode 0 and decoder mode 1.

24.3. Block diagram

Figure 24-1. LPTIMER block diagram provides details of the internal configuration of the low power timer.

Figure 24-1. LPTIMER block diagram



24.4. Function overview

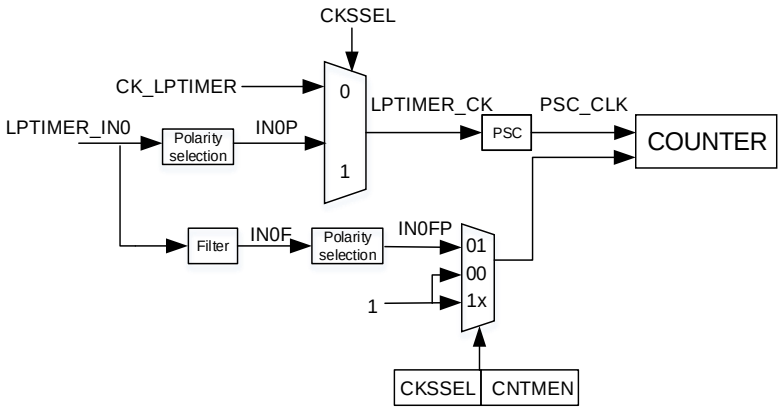
24.4.1. Clock selection

The LPTIMER can be clocked by several clock sources. It can be clocked using an internal clock signal: internal 8 MHz RC oscillator (IRC8M), internal 32 KHz RC oscillator (IRC32K), 32.768 KHz Low Speed crystal oscillator (LXTAL), APB2 clock (PCLK2) sources through the Reset and clock unit (RCU).

LPTIMER can also use an external clock signal on its external input 0 (LPTIMER_IN0) for clock control. When using an external clock source as the clock source, LPTIMER has the following two possible configurations:

- **Case 0:** When LPTIMER is clocked by an external signal, meanwhile APB1 or any other oscillator (including IRC8M, IRC32K and LXTAL) provides an internal clock signal to LPTIMER.
- **Case 1:** LPTIMER is only clocked by an external clock source on LPTIMER_IN0. When all oscillators are turned off after entering low power mode, this configuration is a configuration used to implement the timeout mode or pulse counter function.

Figure 24-2. LPTIMER clock source selection



LPTIMER has the capability of being clocked by either the internal clock signal or external clock signal controlled by bits CNTMEN and CKSSEL in LPTIMER_CTL0 register. The CKSSEL bit is used to select which clock drivies the counter prescaler and the default clock source is the PCLK2. The CNTMEN bit is used to select which clock drivies LPTIMER counter.

When LPTIMER use an external clock signal, the CKPSEL bits are used to configure the active edge used by the counter. The counter can be updated with a rising/ falling edge or both edges of an external clock signal, which depends on the value of the CKPSEL [1:0] bits.

Note that when external clock source signal is derived from the external input 0 (LPTIMER_IN0) pin, if both edges are configured to be active ones(CKPSEL=2'b10) or the pin sampled by a digital filter(ECKFLT≠2'b00), an internal clock signal should also be provided (Case 0). In this case, the internal clock signal frequency should be at least four times the frequency of the external clock signal.

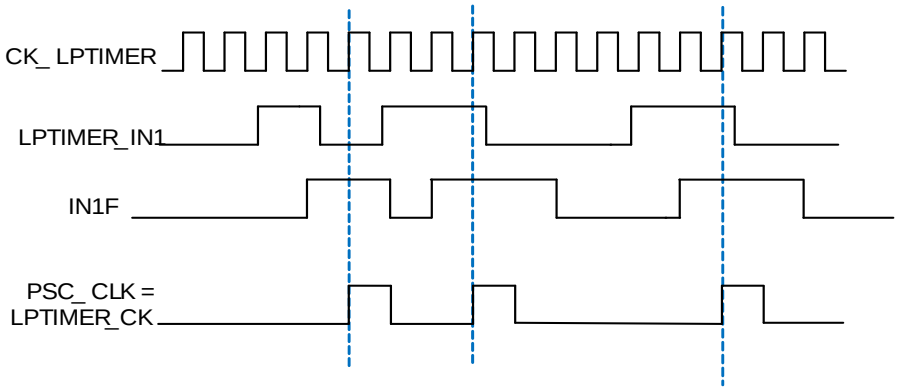
The following clock modes can be selected according to configuration of the CKSSEL bit and CNTMEN bit:

- CKSSEL = 0: the LPTIMER clock is provided by an internal clock signal
  - Internal clock mode 0 (CNTMEN = 0)
    The LPTIMER is clocked by an internal clock signal and the counter is count with every internal clock pulse.
  - Internal clock mode 1 (CNTMEN = 1)
    The external input 0 (LPTIMER_IN0) is sampled with the internal clock. Therefore, without losing any events, the change frequency of the external input signal should never exceed the frequency of the internal clock. And, the LPTIMER's internal clock of cannot be prescaled (PSC [2:0] = 000).
- CKSSEL = 1: the LPTIMER clock is provided by an external clock signal.
  In this case, the CNTMEN bit can be set or reset. The LPTIMER does not require an internal clock source (unless the input filter is enabled or both edges are configured to be active ones). The external signal on the LPTIMER_IN0 is used as LPTIMER's system clock, this is suitable for operation modes without an embedded oscillator.

For this case, the LPTIMER counter can be clocked either on rising or falling edges of the external input clock signal, but not on both edges.

Since the external signal added to the LPTIMER_IN0 pin is also used to clock the LPTIMER core logic, there is some initial delay (after the LPTIMER is enabled) before the counter is counting. Therefore, the first five active edges on theLPTIMER_IN0 are lost after the LPTIMER has enabled.

Figure 24-3. Internal clock mode1 (CKSSEL = 0 and CNTMEN = 1 and PSC[2:0] = 000)



24.4.2. LPTIMER enable

The LPTEN bit in the LPTIMER_CTL1 register is used to enable the LPTIMER core logic. After the LPTEN bit is set to 1, it is necessary to delay two LPTIMER_CK clocks before the LPTIMER is actually enabled.

The LPTIMER_CTL0 and LPTIMER_INTEN registers (except for INHLCOIE and HLCMVUPIF bit) must be modified only when the LPTIMER is disabled.

24.4.3. Prescaler

The prescaler can divide the timer clock (LPTIMER_CK) to the counter clock (PSC_CLK) by a configurable power of 2 prescaler. The prescaler is select by the PSC[2:0] bit-field in LPTIMER_CTL0 must only be modified when the LPTIMER is disabled (LPTEN bit is reset to ‘0’). The table below lists all the possible division ratios:

Table 24-1. Prescaler division factor

Prescaler divider	PSC[2:0] bit-filed
1/1	000
1/2	001
1/4	010
1/8	011
1/16	100
1/32	101
1/64	110

Prescaler divider	PSC[2:0] bit-filed
1/128	111

24.4.4. Input filter

The external (mapped to GPIOs) or internal (mapped on-chip peripherals, such as comparators) signals on the LPTIMER_INx needs to be filtered by a digital filter to prevent the glitches and noise interference from spreading in LPTIMER. This can be used to prevent false counts and triggers.

Before using the digital filters, it is necessary to provide an internal clock source to the LPTIMER to ensure the proper operation of the filters.

There are two types of digital filters:

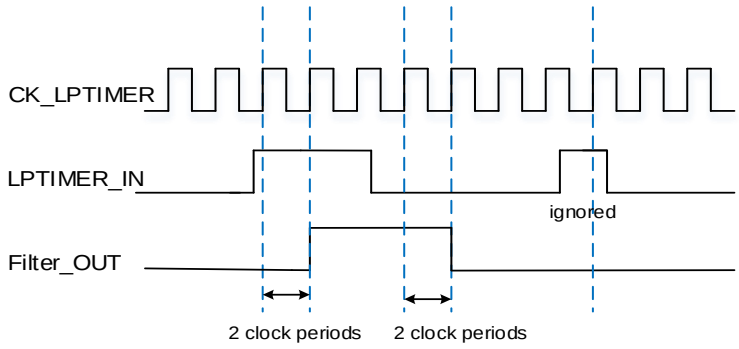
- The first type: protects the LPTIMER external inputs (LPTIMER_IN0/ LPTIMER_IN1). The digital filter is controlled by the ECKFLT [1:0] bits.
- The second type: protects the LPTIMER trigger inputs (ETIx). The digital filter is controlled by the TFLT [1:0] bits.

Note: The same type of digital filters should be keep the same configuration.

The sensitivity of the digital filters depends on the number of consecutive identical samples that should be detected on the LPTIMER inputs and treats the signal level changes as valid.

Figure 24-4. Input filter timing diagram (ECKFLT=2'b01) shows an example of 2 consecutive samples of the input filter.

Figure 24-4. Input filter timing diagram (ECKFLT=2'b01)



Note: If there is no internal clock signal, the ECKFLT and TFLT bits must be set to 0 to disable the digital filter. In this case, an external analog filter can be used to protect the LPTIMER external inputs against the disturbance.

24.4.5. External inputs high level counter

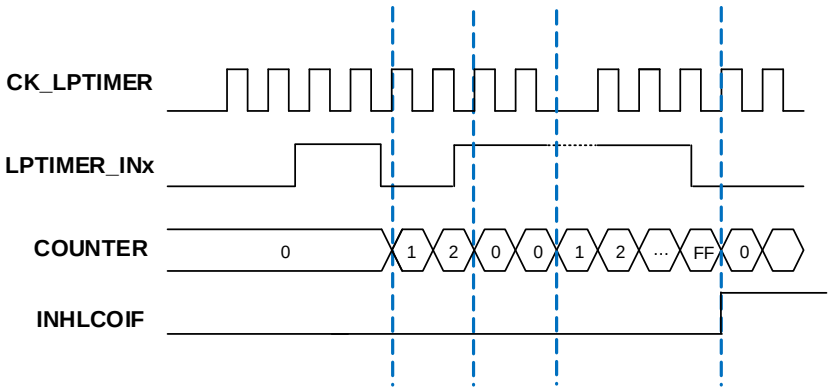
The INHLCEN bit is set to 1 to enable the high level count function of external inputs (LPTIMER_INx). In this mode, the high level counter is clocked by CK_LPTIMER (internal clock). The counter starts counting up when the hign level occurs, and once a low level occurs,

the counter is cleared to 0.

The INHLCOIF flag (in LPTIMER_INTF register) is set by hardware when the value of LPTIMER_INx high level counter equal to the value of INHLCMVAL bits (in LPTIMER_INHLCMV register). An interrupr will generated if the INHLCOIE bit is enabled (in LPTIMER_INTEN register). The INHLCOIF flag can be cleared by writing 1 to the INHLCOIC bit in the INTC register.

Figure 24-5. External inputs high level counter shows an example of the external inputs high level counter

Figure 24-5. External inputs high level counter



The APB bus and the CK_LPTIMER use different clocks, so there is a delay between the APB write and the time when these values are actually used in the LPTIMER_INHLCMV register. Within this delay time period, any additional write into this register must be avoided.

The HLCMVUPIF flag in the LPTIMER_INTF register is used to indicate when the write operation to the LPTIMER_INHLCMV register is completed.

24.4.6. Start counting mode

The LPTIMER counter may be triggered by software or by detecting a valid edge on one of the 8 trigger inputs. ETMEN [1:0] is used to configure the trigger mode of LPTIMER:

- ETMEN[1:0] = 2'b00: The LPTIMER counter is started as soon as the CTNMST bit or SMST bit is set by software.
- ETMEN[1:0] ≠ 2'b00: ETSEL [2:0] is used to select one of the 8 trigger inputs is used to start the counter. The remaining three non-zero values of the ETMEN [1:0] bits are used to configure the valid edge used by the trigger inputs. The LPTIMER counter starts as soon as a valid edge is detected.

The external triggers can be regarded as asynchronous signals of the LPTIMER. Therefore, once a trigger is detected, for synchronization, a delay of two counter clock period is required before the timer starts running. If a new trigger event occurs after the LPTIMER starts, the trigger event will be ignored (unless timeout mode is enabled).

Note: The LPTEN bits must be enabled before the SMST/CTNMST bits are set. When the

LPTEN bit equals “0”, any write on these bits will be discarded by hardware.

24.4.7. External trigger mapping

The LPTIMER external trigger mapping is shown in ***Table 24-2. External trigger mapping***.

Table 24-2. External trigger mapping

ETSEL[3:0]	External trigger mapping
ETI0	GPIO
ETI1	RTC Alarm 0
ETI2	RTC Alarm 1
ETI3	RTC_TAMP0
ETI4	RTC_TAMP1
ETI5	RTC_TAMP2
ETI6	CMP0_OUT
ETI7	CMP1_OUT
ETI8	CMP2_OUT
ETI9	CMP3_OUT
ETI10	CMP4_OUT
ETI11	CMP5_OUT
ETI12	CMP6_OUT
ETI13	CMP7_OUT

24.4.8. Counter operating mode

The LPTIMER counter works in two operating modes:

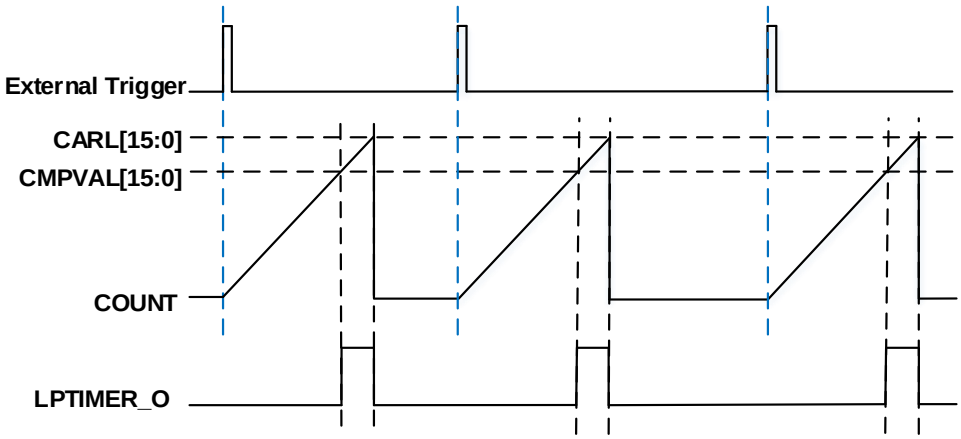
- Continuous counting mode: the LPTIMER counter is running continuously, the counter is started from a trigger event (software trigger or external trigger) and will not stop until the timer is disabled.
- Single counting mode: the LPTIMER counter is started from a trigger event (software trigger or external trigger) and stops after counting the value of CARL bits in LPTIMER_CAR register.

Single counting mode

The SMST bit is set to 1 to enable the the single counting mode. In this mode, a new trigger event will restart the LPTIMER counter. Any trigger event that occurs after the counter is started and before the counter reaches the value of CARL bits will be ignored.

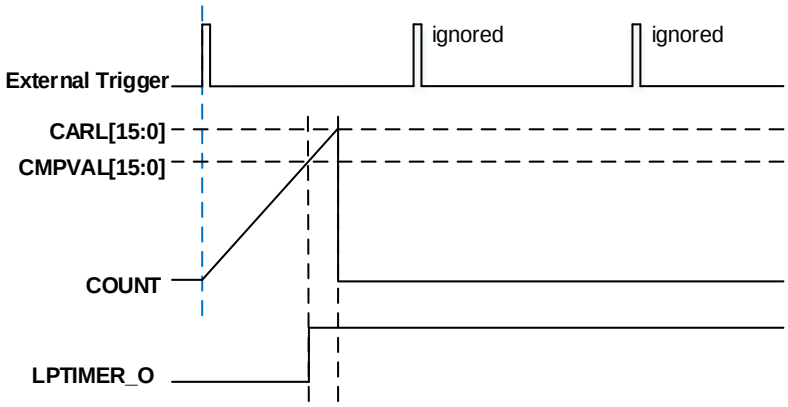
If an external trigger is selected to start LPTIMER counter, when the SMST bit is set, each external trigger event that arrives after the counter stops counting (the value of the CNT bits is zero), will start the counter in a new single counting cycle as shown in ***Figure 24-6. LPTIMER output with SMST = 1(16-bit)***.

Figure 24-6. LPTIMER output with SMST = 1(16-bit)



When the OMSEL bit in the LPTIMER_CTL0 register is set, the set mode is enable. In this case, the counter is only started once after the first trigger, and all subsequent trigger events is ignored, as shown in ***Figure 24-7. LPTIMER output with OMSEL = 1(16-bit)***.

Figure 24-7. LPTIMER output with OMSEL = 1(16-bit)



If ETMEN [1:0] = 2'b 00, the software trigger is enabled, setting the SMST bit will start the counter in single counting mode.

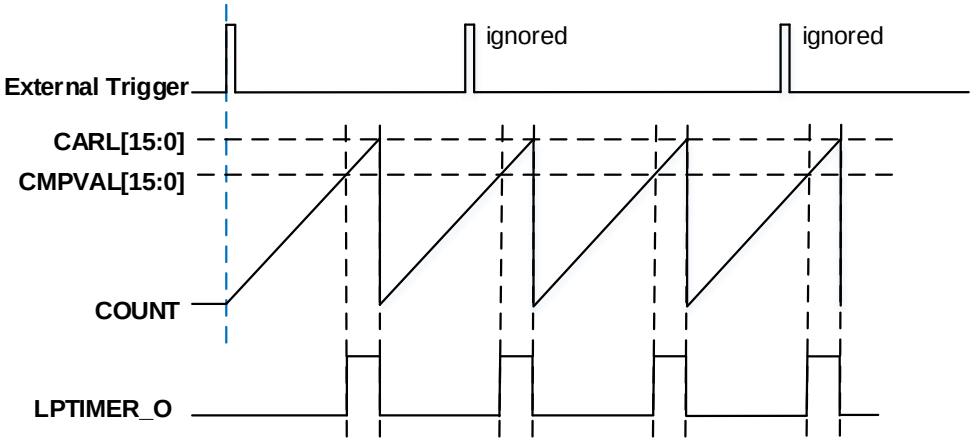
Continuous counting mode

The CTNMST bit is set to 1 to enable the the continuous counting mode.

If an external trigger is selected to start LPTIMER counter, an external trigger event arriving after the CTNMST bit is set will start the counter in continuous counting mode. Any trigger event that occurs after the counter is started will be ignored as shown in ***Figure 24-8. LPTIMER output with CTNMST = 1(16-bit)***.

If ETMEN [1:0] = 2'b 00, the software trigger is enabled, setting the CTNMST bit will start the counter in continuous counting mode.

Figure 24-8. LPTIMER output with CTNMST = 1(16-bit)



The SMST and CTNMST bits can be modified only when the timer is enabled (the LPTEN bit is set). And the single counting mode and continuous counting mode can be modified on the fly.

If the LPTIMER previously working in the continuous counting mode, setting SMST will switch the LPTIMER to the single counting mode. The counter will stop counting after counting the value of CARL bit-filed.

If the LPTIMER previously working in the single counting mode, setting CTNMST will switch the LPTIMER to the continuous counting mode. The counter will restart after counting to the value of CARL bit-filed.

24.4.9. Counter Reset

The LPTIMER counter can be reset to zero by software asynchronously or synchronously. The detail of two reset methods is as below:

- Counter reset asynchronously: the LPTIMER counter will be reset to zero after any read access to the LPTIMER_CNT register when set the RDRSTEN bit in the LPTIMER_CTL1 register to ‘1’.
- Counter reset synchronously: the LPTIMER can be reset to zero synchronously by setting the CNTRST bit in LPTIMER_CTL1 register to ‘1’. The synchronous reset means the reset will take 3 LPTIMER core clock cycles to synchronize when the LPTIMER uses an asynchronous clock (different from APB). The counter will still count during the synchronization. The CNTRST bit is automatically cleared by hardware after counter reset.

Note: These two reset methods can’t be used simultaneously. And the CNTRST can only be set to '1' before it is already cleared to '0' by hardware. Software should consequently check that CNTRST bit is already cleared to '0' before attempting to set it to '1' again.

24.4.10. Output Mode

By configuring the LPTIMER_CARL register and LPTIMER_CMPV register, the LPTIMER can output several different waveforms.

The LPTIMER can generate the following waveforms:

- PWM mode: the LPTIMER output is set when a match occurs between the value of LPTIMER_CMPV and the LPTIMER_CNT registers. The LPTIMER output is reset when a match occurs between the value of LPTIMER_CAR and the LPTIMER_CNT registers.
- Single pulse mode: the output waveform is same as the first pulse in PWM mode, and then the output will always be reset.
- Set mode: the output waveform is similar to the single pulse mode, except that the output remains at the last signal level (depends on the value of the OPSEL bit in the LPTIMER_CTL0 register).

There modes require that the value of LPTIMER_CAR register is greater than the value of the LPTIMER_CMPV register.

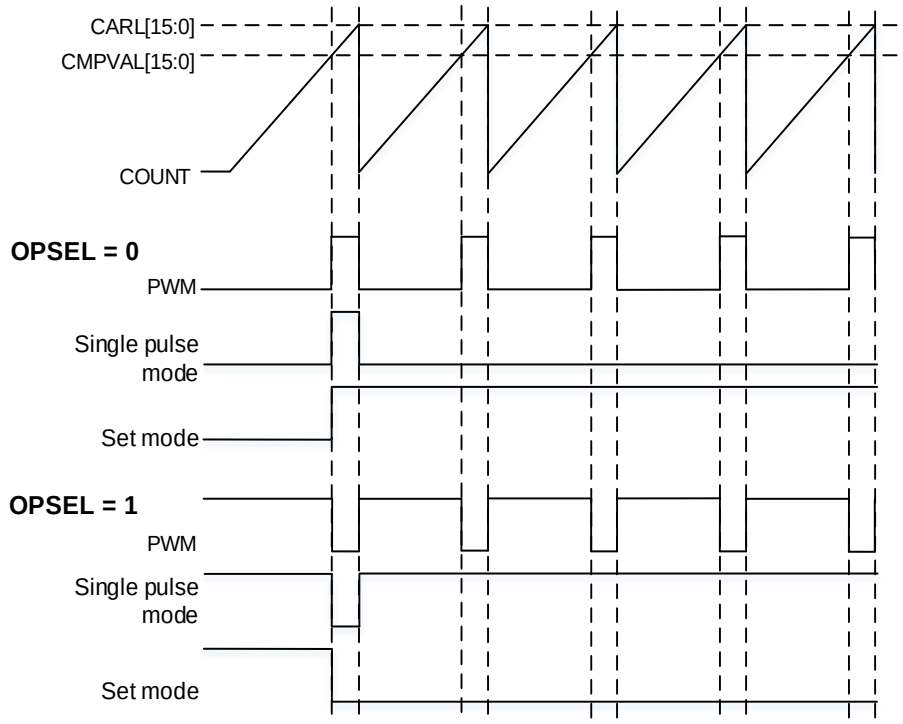
The OMSEL bit in the LPTIMER_CTL0 register is used to controls the output mode.

- OMSEL = 0: the LPTIMER to generate either a PWM mode waveform or a single pulse mode waveform (depending on the CTNMST bit or SMST bit is set).
- OMSEL = 1: the LPTIMER to generate a set mode waveform.

The OPSEL bit is used to configure the LPTIMER output polarity, the modification of this bit will take effect immediately. Therefore, any modification to the polarity configuration bit before enabling the LPTIMER will immediately change the output default value.

The maximum frequency of the generated signal is the LPTIMER clock frequency divided by 2. ***Figure 24-9. LPTIMER_O output mode with OPSEL bit(16-bit)*** below shows three waveforms output by the LPTIMER. Also, it shows the effect of the polarity change with the OPSEL bit.

Figure 24-9. LPTIMER_O output mode with OPSEL bit(16-bit)



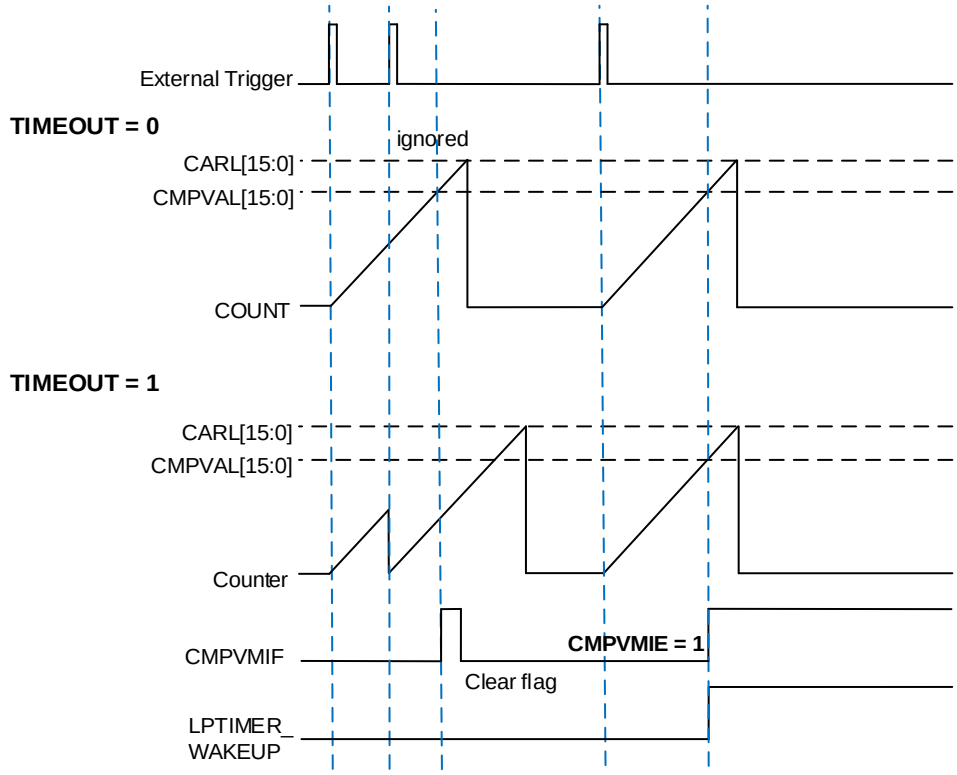
24.4.11. Timeout mode

By setting the TIMEOUT bit, a valid edge detected on a selected trigger input can be used to reset the LPTIMER counter. The first trigger event will start the timer, and subsequent trigger events will reset and restart the counter.

The LPTIMER can realize the low power consumption timeout mode, and the timeout value can be determined by the value of the LPTIMER_CMPV register.

If no trigger occurs within the comparison value range, when the counter counts to the comparison value, a comparison match interrupt will be generated to wake up the MCU.

Figure 24-10. LPTIMER timeout mode(16-bit)



24.4.12. Decoder mode

The LPTIMER has two decoder modes:

- Decoder mode 0: the inputs of LPTIMER_IN0 and LPTIMER_IN1 are quadrature signals, this mode is enabled when DECMEN = 1 and DECMSEL = 0.
- Decoder mode 1: the inputs of LPTIMER_IN0 and LPTIMER_IN1 are non-quadrature signals, this mode is enabled when DECMEN = 1 and DECMSEL = 1.

Decoder mode 0

The decoder mode 0 function uses two quadrature inputs derived from the LPTIMER_IN0 and LPTIMER_IN1 pins respectively to interact with each other to generate the counter value.

At first, the CTNMST bit is set to 1 to enable the continuous counting mode and the DECMEN bit is set to 1 to enable the decoder mode. Then setting DECMSEL = 0 to select that the decoder mode 0, and setting the CKPSEL [1:0] = 2b'00, 2b'01, or 2b'10 to select that the timer counting is determined only by the rising edge, only by falling edge, or both edges.

The counting direction is modified by hardware automatically during the voltage level change of IN0F (LPTIMER_IN0 after input filter) and IN1F (LPTIMER_IN1 after input filter) signals. The mechanism of changing the counter direction is shown in ***Table 24-3. Counting direction versus decoder signals***. The 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 LPTIMER_CAR register before the counter starts to count.

When the counter direction changes, the corresponding flag is set. When the counter direction moves from up to down, the DOWNIF bit is set. When the counter direction moves from down to up, the UPIF bit is set. The corresponding interrupt is generated if enabled by DOWNIE = 1 or UPIE = 1 in LPTIMER_INTEN register.

Table 24-3. Counting direction versus decoder signals

Counting Mode (CKPSEL [1:0])	Level	IN0F		IN1F	
		Rising	Falling	Rising	Falling
Decoder rising-edge-mode	IN0F = High	x	x	Up	-
	IN0F = Low	x	x	Down	-
	IN1F = High	Down	-	x	x
	IN1F = Low	Up	-	x	x
Decoder falling-edge-mode	IN0F = High	x	x	-	Down
	IN0F= Low	x	x	-	Up
	IN1F = High	-	Up	x	x
	IN1F = Low	-	Down	x	x
Decoder both-edge-mode	IN0F = High	x	x	Up	Down
	IN0F = Low	x	x	Down	Up
	IN1F = High	Down	Up	x	x
	IN1F = Low	Up	Down	x	x

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

Figure 24-11. Counter operation in decoder mode 0 with rising-edge-mode and *Figure 24-12. Counter operation in decoder mode 0 with falling-edge-mode* show the counting situations with rising-edge and falling-edge modes.

Figure 24-11. Counter operation in decoder mode 0 with rising-edge-mode

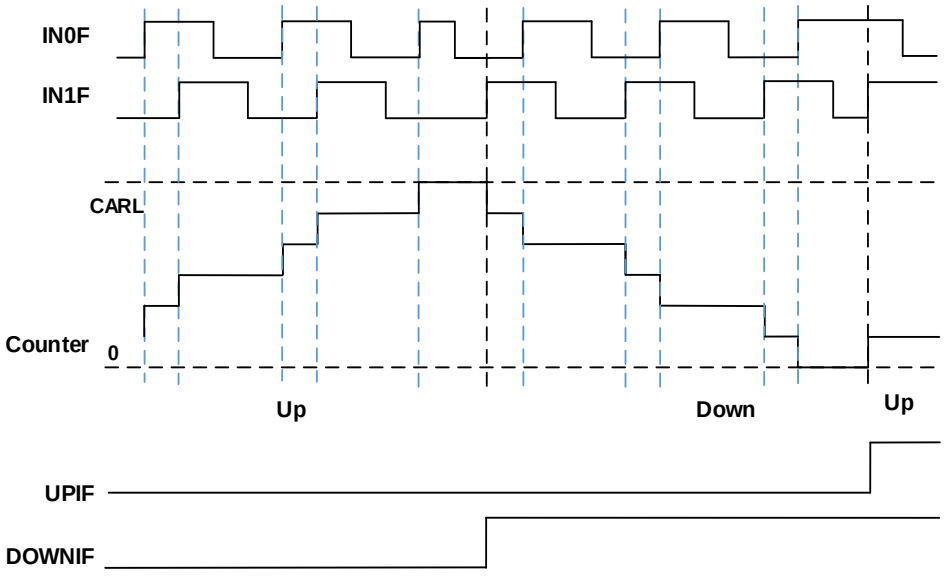
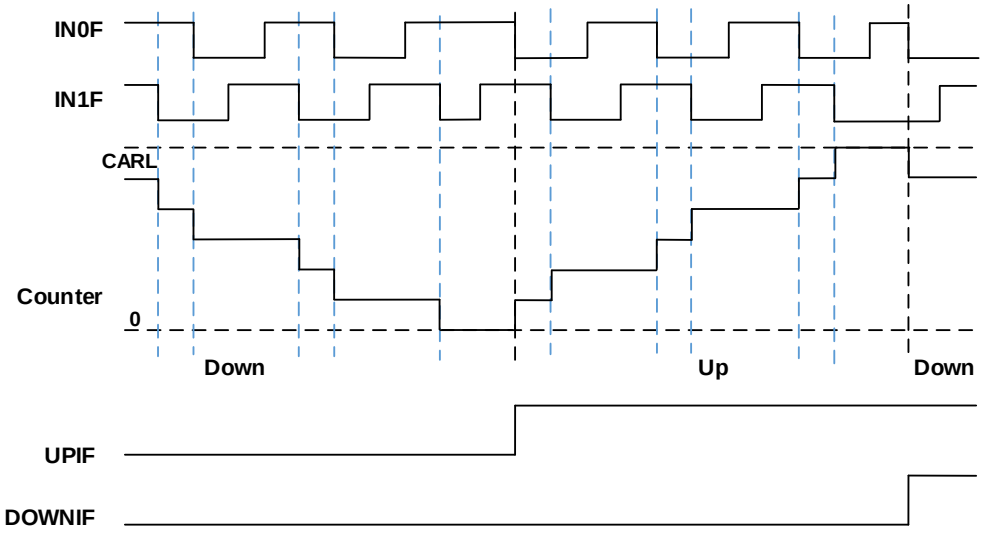


Figure 24-12. Counter operation in decoder mode 0 with falling-edge-mode



Decoder mode 1

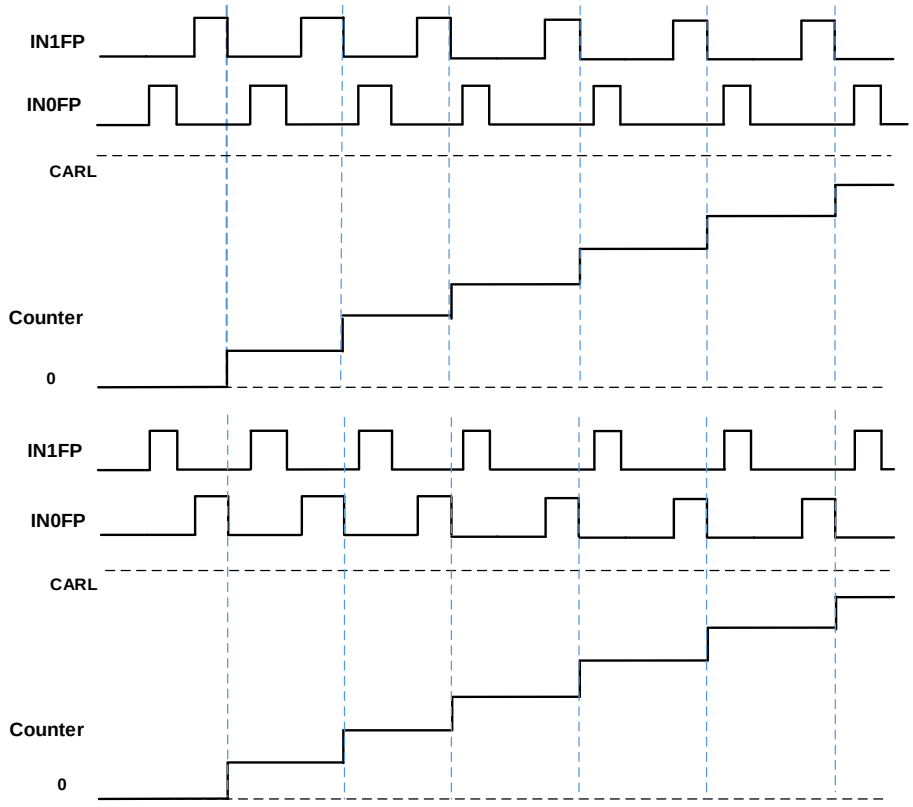
The decoder mode 1 function uses two non-quadrature inputs derived from the LPTIMER_IN0 and LPTIMER_IN1 pins respectively to generate the counter value.

At first, the CTNMST bit is set to 1 to enable the the continuous counting mode and the DECMEN bit is set to 1 to enable the decoder mode. Then setting DECMSEL = 1 to select that the decoder mode 1, and setting the CKPSEL [1:0] = 2b'00, 2b'01 to select that the inputs of LPTIMER_IN0 and LPTIMER_IN1 are non-inverted or inverted.

When two non-overlap pulses appear in IN0FP and IN1FP in sequence, the counter will increment once. ***Figure 24-13. Counter operation in decoder mode 1 with non-inverted*** shows two waveform timing diagrams for the counter can count correctly in decoder mode 1. The high level of IN0FP and IN1FP are not overlap.

The decoder can be regarded as an external clock. This means that the counter counts continuously from 0 to the counter-reload value. Therefore, users must configure the LPTIMER_CAR register before the counter starts to count.

Figure 24-13. Counter operation in decoder mode 1 with non-inverted



When the inputs of LPTIMER_IN0 and LPTIMER_IN1 do not meet the timing relationship in ***Figure 24-13. Counter operation in decoder mode 1 with non-inverted***, the counter cannot count. Depending on the input wavefroms the corresponding interrupt flags will be set (IN1EIF，IN0EIF，INRFOEIF，INHLOEIF), and the interrupts will generated if enabled by IN1EIE, IN0EIE ,INRFOEIE or INHLOEIE in LPTIMER_INTEN register.

Figure 24-14. Counter operation in decoder mode 1 with non-inverted(IN1EIF)

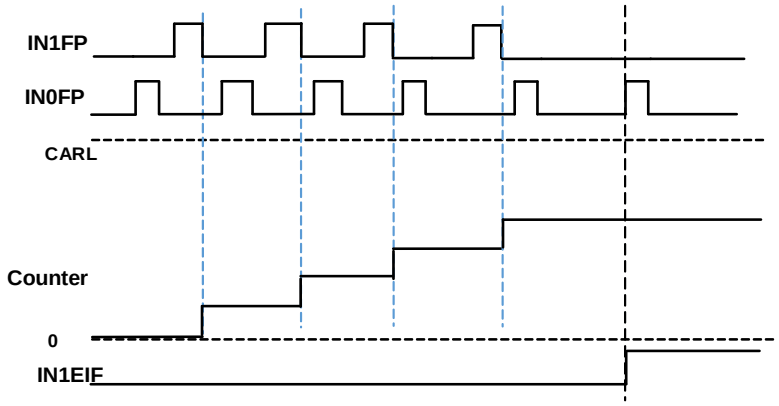


Figure 24-15. Counter operation in decoder mode 1 with non-inverted(IN0EIF)

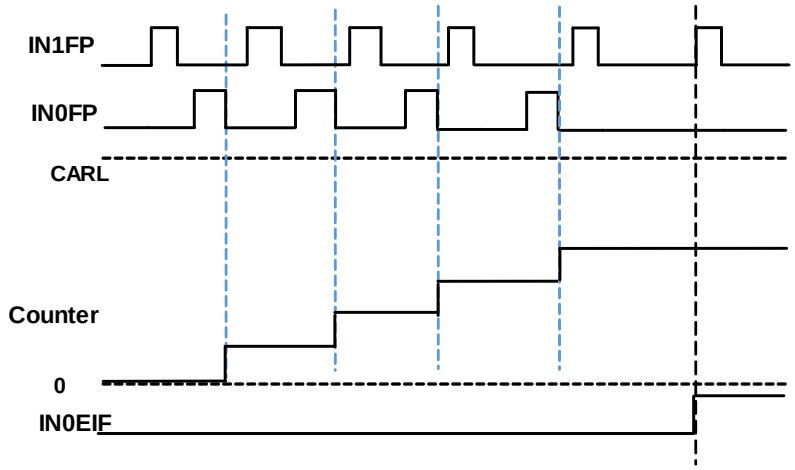


Figure 24-16. Counter operation in decoder mode 1 with non-inverted(INRFOEIF)

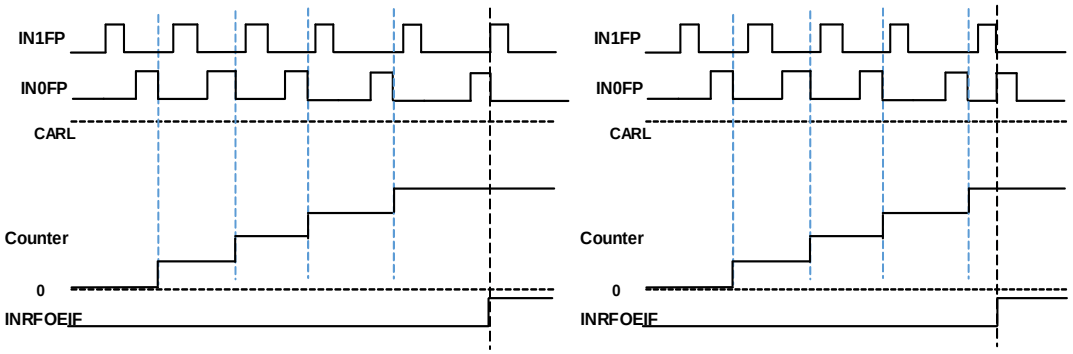
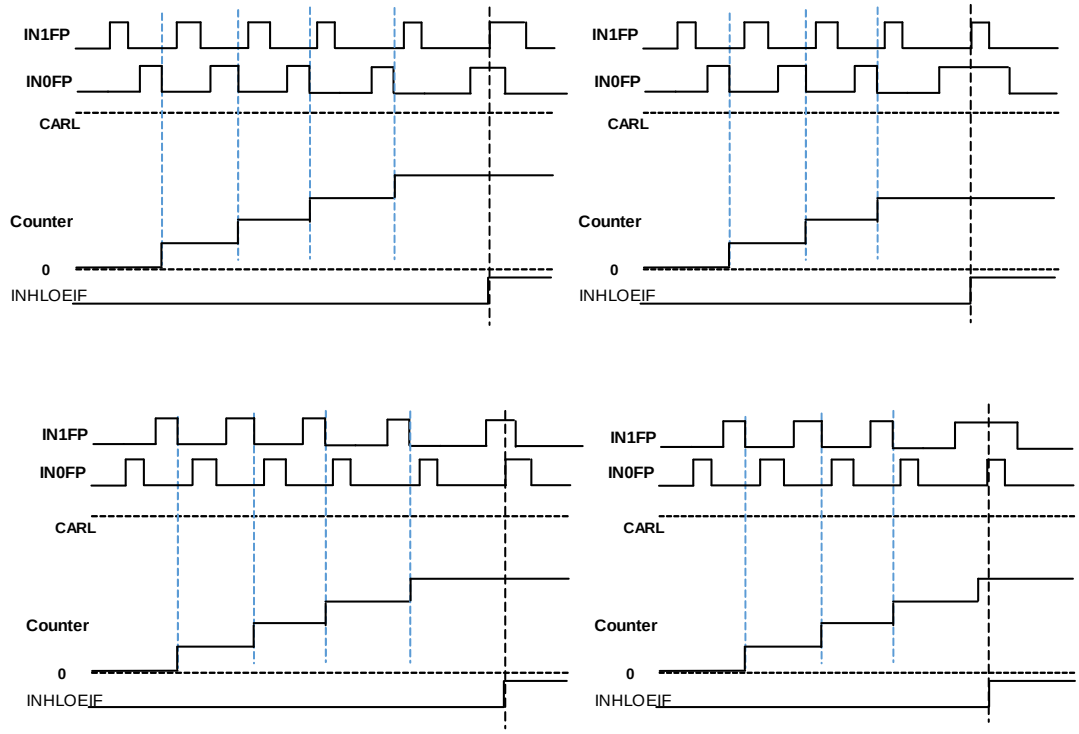


Figure 24-17. Counter operation in decoder mode 1 with non-inverted(INHLOEIF)



Note that when the LPTIMER used in decoder modes, an internal clock signal should also be

provided (CKSSEL = 0) and the internal clock of LPTIMER cannot be prescaled (PSC [2:0] = 000). In this case, the internal clock signal frequency should be at least four times the frequency of the external clock signal.

24.4.13. Register update operation

The LPTIMER_CAR register and LPTIMER_CMPV register are updated immediately after the APB bus completes the write operation, or updated at the end of the current period, when the LPTIMER has already started. The SHWEN bit is used to configure the update of the LPTIMER_CAR and the LPTIMER_CMPV registers:

- SHWEN = 0: after any write access, the LPTIMER_CAR and the LPTIMER_CMPV registers are updated immediately.
- SHWEN = 1: after the LPTIMER is started, the LPTIMER_CAR and the LPTIMER_CMPV registers are updated at the end of the current period.

The APB bus and the LPTIMER core use different clocks, so there is some delay between the APB write and when the LPTIMER_CAR register and LPTIMER_CMPV register actually use these values. During this delay time, any additional writes to these registers must be avoided.

The CARUPIF flag and the CMPVUPIF flag in the LPTIMER_INTF register are respectively used to indicate when the write operation to the LPTIMER_CAR register and the LPTIMER_CMPV register is completed.

After the LPTIMER_CAR register or the LPTIMER_CMPV register is written, only the previous write operation is completed, a new write operation to the same register can be performed.

Any continuous write operations performed before the CMPVUPIF flag or the CARUPIF flag are set respectively will cause unpredictable results.

24.4.14. Low-power modes

The LPTIMER is able to keep running in all power saving modes except for Standby mode with its diversity of clock sources. The LPTIMER has the ability to wake up the system from the low-power modes, and it is suitable for realizing timeout with very low power consumption.

Table 24-4. LPTIMER works in low-power modes

Mode	Description
Sleep mode	Operating normally. LPTIMER interrupts cause the device to exit Sleep mode.
Deep-sleep mode	When LPTIMER is clocked by LXTAL or Internal low speed RC oscillator, LPTIMER interrupts cause the device to exit Deep-sleep mode.

24.4.15. Interrupts

The following events can generate interrupts or wake-up events, if they are enabled through the LPTIMER_INTEN register:

- LPTIMER_IN1 error
- LPTIMER_IN0 error
- The falling and rising edges of LPTIMER_IN0 and LPTIMER_IN1 overlap error
- The high level of LPTIMER_IN0 and LPTIMER_IN1 overlap error
- LPTIMER_INx(x=0,1) high level counter overflow
- Input high level counter max value register update interrupt
- LPTIMER counter direction change up to down
- LPTIMER counter direction change down to up
- Counter auto reload register update
- Compare value register update
- External trigger edge event
- Counter auto reload register match
- Compare value register match

If the interrupt flag in the LPTIMER_INTF register is set before its corresponding interrupt enable bit in the LPTIMER_INTEN register is set, the interrupt is invalid.

Table 24-5. LPTIMER interrupt events

Interrupt event	Description
LPTIMER_IN1 error	Interrupt flag is set when the signal of LPTIMER_IN1 does not jump between the two consecutive rising edges of LPTIMER_IN0 (just used in decoder mode 1).
LPTIMER_IN0 error	Interrupt flag is set when the signal of LPTIMER_IN0 does not jump between the two consecutive rising edges of LPTIMER_IN1 (just used in decoder mode 1).
The falling and rising edges of LPTIMER_IN0 and LPTIMER_IN1 overlap	Interrupt flag is set when the falling edge of LPTIMER_IN0 and the rising edge of LPTIMER_IN1 occur simultaneously or the falling edge of LPTIMER_IN1 and the rising edge of LPTIMER_IN0 occur simultaneously (just used in decoder mode 1).
The high level of LPTIMER_IN0 and LPTIMER_IN1 overlap	Interrupt flag is set when the high level of LPTIMER_IN0 and LPTIMER_IN1 overlap (just used in decoder mode 1).
LPTIMER_INx(x=0,1) high level counter overflow	Interrupt flag is set when LPTIMER_INx high level counter equal to external input high level counter max value register (LPTIMER_INHLCMV).
Input high level counter max value register update	Interrupt flag is set when the APB bus write operation to the LPTIMER_INHLCMV register has been successfully completed.
LPTIMER counter direction change up to down	Interrupt flag is set when the counter direction moves from up to down.

Interrupt event	Description
LPTIMER counter direction change down to up	Interrupt flag is set when the counter direction moves from down to up.
Counter auto reload register update	Interrupt flag is set when the APB bus write operation to the LPTIMER_CAR register has been successfully completed.
Compare value register update	Interrupt flag is set when the APB bus write operation to the LPTIMER_CMPV register has been successfully completed.
External trigger edge event	Interrupt flag is set when an external trigger active edge event is detected.
Counter auto reload register match	Interrupt flag is set when the value of the Counter register (LPTIMER_CNT) matches the value of the Counter auto-reload register (LPTIMER_CAR).
Compare value register match	Interrupt flag is set when the value of the Counter register (LPTIMER_CNT) matches the value of the compare value register (LPTIMER_CMPV).

24.4.16. LPTIMER debug mode

When the Cortex®-M33 halted, and the corresponding LPTIMER_HOLD configuration bit in DBG_CTL1 register is set to 1, the LPTIMER counter stops.

24.5. Register definition

LPTIMER base address: 0x4000 9400

24.5.1. Interrupt flag register (LPTIMER_INTF)

Address offset: 0x00
Reset value: 0x0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IN1EIF	IN0EIF	INRFOEIF	INHLOEIF	INHLCOIF	HLCMV UPIF	Reserved									
r	r	r	r	r	r										

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									DOWNIF	UPIF	CARUPIF	CMPV UPIF	ETED EVIF	CARMIF	CMPV MIF
									r	r	r	r	r	r	r

Bits	Fields	Descriptions
31	IN1EIF	LPTIMER_IN1 error interrupt flag This flag is set by hardware when the signal of LPTIMER_IN1 does not jump between the two consecutive rising edges of LPTIMER_IN0. IN1EIF flag can be cleared by writing 1 to the IN1EIC bit in the INTC register. **Note:** This flag just used in decoder mode 1.
30	IN0EIF	LPTIMER_IN0 error interrupt flag This flag is set by hardware when the signal of LPTIMER_IN0 does not jump between the two consecutive rising edges of LPTIMER_IN1. IN0EIF flag can be cleared by writing 1 to the IN0EIC bit in the INTC register. **Note:** This flag just used in decoder mode 1.
29	INRFOEIF	The falling and rising edges of LPTIMER_IN0 and LPTIMER_IN1 overlap error interrupt flag. This flag is set by hardware when the falling edge of LPTIMER_IN0 and the rising edge of LPTIMER_IN1 occur simultaneously or the falling edge of LPTIMER_IN1 and the rising edge of LPTIMER_IN0 occur simultaneously. INRFOEIF flag can be cleared by writing 1 to the INRFOEIC bit in the INTC register. **Note:** This flag just used in decoder mode 1.
28	INHLOEIF	The high level of LPTIMER_IN0 and LPTIMER_IN1 overlap error interrupt flag. This flag is set by hardware when the high level of LPTIMER_IN0 and LPTIMER_IN1 overlap. INHLOEIF flag can be cleared by writing 1 to the INHLOEIC bit in the INTC register.

		Note: This flag just used in decoder mode 1.
27	INHLCOIF	LPTIMER_INx(x=0,1) high level counter overflow interrupt flag This flag is set by hardware when LPTIMER_INx high level counter equal to external input high level counter max value register (LPTIMER_INHLCMV). INHLCOIF flag can be cleared by writing 1 to the INHLCOIC bit in the INTC register.
26	HLCMVUPIF	Input high level counter max value register update interrupt flag This flag is set by hardware when the APB bus write operation to the LPTIMER_INHLCMV register has been successfully completed. HLCMVUPIF flag can be cleared by writing 1 to the HLCMVUPIC bit in the INTC register.
25:7	Reserved	Must be kept at reset value.
6	DOWNIF	LPTIMER counter direction change up to down interrupt flag In decoder mode 0, the DOWNIF bit is set by hardware when the counter direction moves from up to down. The DOWNIF flag can be cleared by writing 1 to the DOWNIC bit of the INTC register.
5	UPIF	LPTIMER counter direction change down to up interrupt flag In decoder mode 0, the UPIF bit is set by hardware when the counter direction moves from down to up. The UPIF flag can be cleared by writing 1 to the UPIC bit in the INTC register.
4	CARUPIF	Counter auto reload register update interrupt flag This flag is set by hardware when the APB bus write operation to the LPTIMER_CAR register has been successfully completed. The CARUPIF flag can be cleared by writing 1 to the CARUPIC bit in the INTC register.
3	CMPVUPIF	Compare value register update interrupt flag This flag is set by hardware when the APB bus write operation to the LPTIMER_CMPV register has been successfully completed. The CMPVUPIF flag can be cleared by writing 1 to the CMPVUPIC bit in the INTC register.
2	ETEDEVIF	External trigger edge event interrupt flag This flag is set by hardware when the active edge of the external trigger occurs. The ETEDEVIF flag can be cleared by writing 1 to the ETEDEVIC bit in the INTC register. **Note:** This flag will not be set when the active edge of the external trigger happened after LPTIMER started.
1	CARMIF	Counter auto reload register match interrupt flag This flag is set by hardware when the LPTIMER_CNT value matches the value of the LPTIMER_CAR register. The CARMIF flag can be cleared by writing 1 to the CARMIC bit in the INTC register.
0	CMPVMIF	Compare value register match interrupt flag This flag is set by hardware when the LPTIMER_CNT value matches the value of the LPTIMER_CMPV register. The CMPVMIF flag can be cleared by writing 1 to the

CMPVMIC bit in the INTC register.

24.5.2. Interrupt flag clear register (LPTIMER_INTC)

Address offset: 0x04

Reset value: 0x0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IN1EIC	IN0EIC	INRFOEIC	INHLOEIC	INHLCOIC	HLCMV UPIC	Reserved									
w	w	w	w	w	w										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									DOWNIC	UPIC	CARUPIC	CMPV UPIC	ETED EVIC	CARMIC	CMPV MIC
									w	w	w	w	w	w	w

Bits	Fields	Descriptions
31	IN1EIC	LPTIMER_IN1 error interrupt flag clear bit. Write 1 to this bit to clear the IN1EIF flag, and write 0 has no effect.
30	IN0EIC	LPTIMER_IN0 error interrupt flag clear bit. Write 1 to this bit to clear the IN0EIF flag, and write 0 has no effect.
29	INRFOEIC	The falling and rising edges of LPTIMER_IN0 and LPTIMER_IN1 overlap error interrupt flag clear bit. Write 1 to this bit to clear the INRFOEIF flag, and write 0 has no effect.
28	INHLOEIC	The high level of LPTIMER_IN0 and LPTIMER_IN1 overlap error interrupt flag clear bit. Write 1 to this bit to clear the INHLOEIF flag, and write 0 has no effect.
27	INHLCOIC	LPTIMER_INx(x=0, 1) high level counter overflow interrupt flag clear bit. Write 1 to this bit to clear the INHLCOIF flag, and write 0 has no effect.
26	HLCMVUPIC	Input high level counter max value register update interrupt flag clear bit. Write 1 to this bit to clear the HLCMVUPIF flag, and write 0 has no effect.
25:7	Reserved	Must be kept at reset value.
6	DOWNIC	LPTIMER counter direction change up to down interrupt flag clear bit. Write 1 to this bit to clear the DOWNIF flag, and write 0 has no effect.
5	UPIC	LPTIMER counter direction change down to up interrupt flag clear bit. Write 1 to this bit to clear the UPIF flag, and write 0 has no effect.
4	CARUPIC	Counter auto reload register update interrupt flag clear bit. Write 1 to this bit to clear the CARUPIF flag, and write 0 has no effect.

3	CMPVUPIC	Compare value register update interrupt flag clear bit. Write 1 to this bit to clear the CMPVUPIF flag, and write 0 has no effect.
2	ETEDEVIC	External trigger edge event interrupt flag clear bit. Write 1 to this bit to clear the ETEDEVIF flag, and write 0 has no effect.
1	CARMIC	Counter auto reload register match interrupt flag clear bit. Write 1 to this bit to clear the CARMIF flag, and write 0 has no effect.
0	CMPVMIC	Compare value register match interrupt flag clear bit. Write 1 to this bit to clear the CMPVMIF flag, and write 0 has no effect.

24.5.3. Interrupt enable register (LPTIMER_INTEN)

Address offset: 0x08

Reset value: 0x0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IN1EIE	IN0EIE	INRFOEIE	INHLOEIE	INHLCOIE	HLCMV UPIE	Reserved									
rw	rw	rw	rw	rw	rw										

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									DOWNIE	UPIE	CARUPIE	CMPV UPIE	ETED EVIE	CARMIE	CMPV MIE
									rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31	IN1EIE	LPTIMER_IN1 error interrupt enable bit 0: Disabled 1: Enabled This bit can be modified only when the LPTIMER is disabled (The LPTEN bit in LPTIMER_CTL1 register is 0).
30	IN0EIE	LPTIMER_IN0 error interrupt enable bit 0: Disabled 1: Enabled This bit can be modified only when the LPTIMER is disabled (The LPTEN bit in LPTIMER_CTL1 register is 0).
29	INRFOEIE	The falling and rising edges of LPTIMER_IN0 and LPTIMER_IN1 overlap error interrupt enable bit 0: Disabled 1: Enabled This bit can be modified only when the LPTIMER is disabled (The LPTEN bit in LPTIMER_CTL1 register is 0).

28	INHLOEIE	The high level of LPTIMER_IN0 and LPTIMER_IN1 overlap error interrupt enable bit. 0: Disabled 1: Enabled This bit can be modified only when the LPTIMER is disabled (The LPTEN bit in LPTIMER_CTL1 register is 0).
27	INHLCOIE	LPTIMER_INx(x=0,1) high level counter overflow interrupt enable bit 0: Disabled 1: Enabled This bit can be modified only when the LPTIMER is external input high level counter disabled (The INHLCEN bit in LPTIMER_CTL1 register is 0).
26	HLCMVUPIE	Input high level counter max value register update interrupt enable bit 0: Disabled 1: Enabled This bit can be modified only when the LPTIMER is external input high level counter disabled (The INHLCEN bit in LPTIMER_CTL1 register is 0).
25:7	Reserved	Must be kept at reset value.
6	DOWNIE	LPTIMER counter direction change up to down interrupt enable bit 0: Disabled 1: Enabled This bit can be modified only when the LPTIMER is disabled (The LPTEN bit in LPTIMER_CTL1 register is 0).
5	UPIE	LPTIMER counter direction change down to up interrupt enable bit 0: Disabled 1: Enabled This bit can be modified only when the LPTIMER is disabled (The LPTEN bit in LPTIMER_CTL1 register is 0).
4	CARUPIE	Counter auto reload register update interrupt enable bit 0: Disabled 1: Enabled This bit can be modified only when the LPTIMER is disabled (The LPTEN bit in LPTIMER_CTL1 register is 0).
3	CMPVUPIE	Compare value register update interrupt enable bit 0: Disabled 1: Enabled This bit can be modified only when the LPTIMER is disabled (The LPTEN bit in LPTIMER_CTL1 register is 0).
2	ETEDEVIE	External trigger edge event interrupt enable bit 0: Disabled

		1: Enabled This bit can be modified only when the LPTIMER is disabled (The LPTEN bit in LPTIMER_CTL1 register is 0).
1	CARMIE	Counter auto reload register match interrupt enable bit 0: Disabled 1: Enabled This bit can be modified only when the LPTIMER is disabled (The LPTEN bit in LPTIMER_CTL1 register is 0).
0	CMPVMIE	Compare value register match interrupt enable bit 0: Disabled 1: Enabled This bit can be modified only when the LPTIMER is disabled (The LPTEN bit in LPTIMER_CTL1 register is 0).

24.5.4. Control register 0 (LPTIMER_CTL0)

Address offset: 0x0C

Reset value: 0x0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		ETSEL[3]	Reserved			DECMSEL	DECMEN	CNTMEN	SHWEN	OPSEL	OMSEL	TIMEOUT	ETMEN[1:0]		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
ETSEL[2:0]			Reserved	PSC[2:0]			Reserved	TFLT [1:0]		Reserved	ECKFLT[1:0]		CKPSEL[1:0]		CKSSEL
rw				rw				rw			rw		rw		rw

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29	ETSEL[3]	External trigger select Refer to ETSEL[2:0].
28:26	Reserved	Must be kept at reset value.
25	DECMSEL	Decoder mode select 0: Decoder mode 0 1: Decoder mode 1 This bit can be modified only when the LPTIMER is disabled (The LPTEN bit in LPTIMER_CTL1 register is 0).
24	DECMEN	Decoder mode enabled 0: Decoder mode disabled 1: Decoder mode enabled This bit can be modified only when the LPTIMER is disabled (The LPTEN bit in

		LPTIMER_CTL1 register is 0).
23	CNTMEN	Counter mode select This bit is used to select the clock source of the LPTIMER counter. 0: The counter is count with each internal clock pulse 1: The counter is count with each active clock pulse on the LPTIMER_IN0. This bit can be modified only when the LPTIMER is disabled (The LPTEN bit in LPTIMER_CTL1 register is 0).
22	SHWEN	LPTIMER_CAR and LPTIMER_CMPV shadow registers enable 0: The shadow registers are disable. The registers are updated immediately after every APB bus write access. 1: The shadow registers are enable. The registers are updated at the end of the LPTIMER period. This bit can be modified only when the LPTIMER is disabled (The LPTEN bit in LPTIMER_CTL1 register is 0).
21	OPSEL	Output polarity select This bit is used to controls the output polarity. 0: The output is non-inverted. When counting up, the output is set as long as the counter is match the value of LPTIMER_CMPV; The output is reset as long as the counter is match the value of LPTIMER_CAR. 1: The output is inverted. When counting up, the output is reset as long as the counter is match the value of LPTIMER_CMPV; The output is set as long as the counter is match the value of LPTIMER_CAR. This bit can be modified only when the LPTIMER is disabled (The LPTEN bit in LPTIMER_CTL1 register is 0).
20	OMSEL	Output Mode select This bit is used to controls the output mode. 0: PWM mode or single pulse mode (CTNMST bit for PWM mode and SMST for single pulse mode). 1: Set mode. This bit can be modified only when the LPTIMER is disabled (The LPTEN bit in LPTIMER_CTL1 register is 0).
19	TIMEOUT	Timeout mode enable This bit is used to controls the timeout mode. 0: A new trigger event will be ignored after LPTIMER started 1: A new trigger event will reset and restart the count after LPTIMER started This bit can be modified only when the LPTIMER is disabled (The LPTEN bit in LPTIMER_CTL1 register is 0).
18:17	ETMEN[1:0]	External Trigger mode enable The ETMEN bits are used to configure the trigger mode for LPTIMER. 00: External trigger disable (Software trigger)

		01: Rising edge of external trigger enable 01: Falling edge of external trigger enable 11: Rising and falling edges of external trigger enable These bits can be modified only when the LPTIMER is disabled (The LPTEN bit in LPTIMER_CTL1 register is 0).
16	Reserved	Must be kept at reset value.
15:13	ETSEL[2:0]	External trigger select These bits and ETSEL[3] bit are used to select the external trigger source for LPTIMER. 0000: ETI0 (GPIO) 0001: ETI1 (RTC Alarm 0) 0010: ETI2 (RTC Alarm 1) 0011: ETI3 (RTC_TAMP0) 0100: ETI4 (RTC_TAMP1) 0101: ETI5 (RTC_TAMP2) 0110: ETI6 (CMP0_OUT) 0111: ETI7 (CMP1_OUT) 1000: ETI8 (CMP2_OUT) 1001: ETI9 (CMP3_OUT) 1010: ETI10 (CMP4_OUT) 1011: ETI11 (CMP5_OUT) 1100: ETI12 (CMP6_OUT) 1101: ETI13 (CMP7_OUT) These bits can be modified only when the LPTIMER is disabled (The LPTEN bit in LPTIMER_CTL1 register is 0).
12	Reserved	Must be kept at reset value.
11:9	PSC[2:0]	Clock prescaler selection The PSC bits are used to configure the prescaler to divide the timer clock (LPTIMER_CK) to a counter clock (PSC_CLK). 000: $f_{PSC_CLK}=f_{LPTIMER_CK}$ 001: $f_{PSC_CLK}=f_{LPTIMER_CK}$ /2 010: $f_{PSC_CLK}=f_{LPTIMER_CK}$ /4 011: $f_{PSC_CLK}=f_{LPTIMER_CK}$ / 8 100: $f_{PSC_CLK}=f_{LPTIMER_CK}$ / 16 101: $f_{PSC_CLK}=f_{LPTIMER_CK}$ / 32 110: $f_{PSC_CLK}=f_{LPTIMER_CK}$ / 64 111: $f_{PSC_CLK}=f_{LPTIMER_CK}$ / 128 These bits can be modified only when the LPTIMER is disabled (The LPTEN bit in LPTIMER_CTL1 register is 0).
8	Reserved	Must be kept at reset value.

7:6	TFLT[1:0]	Trigger filter The TFLT bits are used to configure the digital filter for triggers. An internal clock source must be used in this function. 00: Filter disabled, any active level of the trigger is valid. 01: The active level change of the trigger need to be maintained at least 2 clock periods. 10: The active level change of the trigger need to be maintained at least 4 clock periods. 11: The active level change of the trigger need to be maintained at least 8 clock periods. These bits can be modified only when the LPTIMER is disabled (The LPTEN bit in LPTIMER_CTL1 register is 0).
5	Reserved	Must be kept at reset value.
4:3	ECKFLT[1:0]	External clock filter The ECKFLT bits are used to configure the digital filter for external clock. An internal clock source must be used in this function. 00: Filter disabled, any active level change of external clock is valid. 01: The active level change of the external clock need to be maintained at least 2 clock periods. 10: The active level change of the external clock need to be maintained at least 4 clock periods. 11: The active level change of the external clock need to be maintained at least 8 clock periods. These bits can be modified only when the LPTIMER is disabled (The LPTEN bit in LPTIMER_CTL1 register is 0).
2:1	CKPSEL[1:0]	Clock polarity select If LPTIMER is clocked by an external clock source, CKPSEL bits is used to configure the active edge or edges used for the counter counting: 00: the rising edge is the active edge used for counting. If the LPTIMER is configured in decoder mode 0 (DECMEN = 1, DECMSEL = 0), the decoder rising-edge-mode is active. If the LPTIMER is configured in decoder mode 1 (DECMEN = 1, DECMSEL = 1), the inputs of LPTIMER_IN0 and LPTIMER_IN1 are non-inverted. If the LPTIMER external input high level counter enable（INHLCEN=1），the inputs of LPTIMER_IN0 and LPTIMER_IN1 are non-inverted. 01: the falling edge is the active edge used for counting If the LPTIMER is configured in decoder mode 0, the decoder falling-edge-mode is active. If the LPTIMER is configured in decoder mode 1, the inputs of LPTIMER_IN0 and LPTIMER_IN1 are inverted. If the LPTIMER external input high level counter enable（INHLCEN=1），the inputs of LPTIMER_IN0 and LPTIMER_IN1 are inverted.

		10: both edges are the active edge used for counting. When the both edges of two external clock signals are considered as valid edges, the LPTIMER must be clocked by an internal clock source, and the internal clock frequency is at least equal to four times the external clock frequency. If the LPTIMER is configured in decoder mode 0, the decoder both-edge-mode is active. This is not allowed if the LPTIMER is configured in decoder mode 1. If the LPTIMER external input high level counter enable（INHLCEN=1），the inputs of LPTIMER_IN0 and LPTIMER_IN1 are non-inverted. 11: not allowed These bits can be modified only when the LPTIMER is disabled (The LPTEN bit in LPTIMER_CTL1 register is 0).
0	CKSSEL	Clock source select This bit is used to select the clock source for LPTIMER. 0: LPTIMER is clocked by internal clock source. 1: LPTIMER is clocked by external clock source on the LPTIMER_IN0. This bit can be modified only when the LPTIMER is disabled (The LPTEN bit in LPTIMER_CTL1 register is 0). Note：when the decoder mode 0 enabled (DECMEN=1), the CKSSEL bit will cleared by hardware.

24.5.5. Control register 1 (LPTIMER_CTL1)

Address offset: 0x10

Reset value: 0x0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INHLCEN	LPTENF	Reserved													
rw	r														
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											RDRSTEN	CNTRST	CTNMST	SMST	LPTEN
											rw	rs	rw	rw	rw

Bits	Fields	Descriptions
31	INHLCEN	LPTIMER external input high level counter enable. 0: disabled 1: enabled
30	LPTENF	LPTIMER enabled from LPTIMER core flag. This bit is set and reset by hardware. 0: LPTIMER is disabled

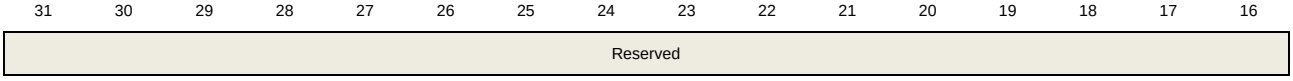
		1: LPTIMER is enabled
29:5	Reserved	Must be kept at reset value.
4	RDRSTEN	Read cause LPTIMER_CNT asynchronously reset enable This bit is set and reset by software. 0: Read cause LPTIMER_CNT register asynchronously reset disabled 1: Read cause LPTIMER_CNT register asynchronously reset enabled This bit can be modified only when the LPTIMER is enabled (The LPTEN bit in LPTIMER_CTL1 register is 1).
3	CNTRST	Counter (LPTIMER_CNT) synchronously reset This bit is set by software and cleared by hardware. When set to '1', the LPTIMER_CNT counter register will be reset synchronously. When the LPTIMER uses an asynchronous clock (different from APB), the reset will take 3 LPTIMER core clock cycles to synchronize. **Note:** The CNTRST can only be set to '1' by software before it is already cleared to '0' by hardware. Software should consequently check that CNTRST bit is already cleared to '0' before attempting to set it to '1'. This bit can be set only when the LPTIMER is enabled and automatically reset by hardware after reset completed.
2	CTNMST	LPTIMER start for continuous counting mode. This bit is set by software and reset by hardware. This bit can be modified only when the LPTIMER is enabled (The LPTEN bit in LPTIMER_CTL1 register is 1).
1	SMST	LPTIMER start for single counting mode. This bit is set by software and reset by hardware. This bit can be modified only when the LPTIMER is enabled (The LPTEN bit in LPTIMER_CTL1 register is 1).
0	LPTEN	LPTIMER enable This bit is set and reset by software. 0: LPTIMER is disabled 1: LPTIMER is enabled

24.5.6. Compare value register (LPTIMER_CMPV)

Address offset: 0x14

Reset value: 0x0000

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
CMPVAL[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CMPVAL[15:0]	Compare value This bit-filed specifies the compare value of the counter. This bit-filed can be modified only when the LPTIMER is enabled (The LPTEN bit in LPTIMER_CTL1 register is 1).

24.5.7. Counter auto reload register (LPTIMER_CAR)

Address offset: 0x18
Reset value: 0x0001

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

31	30	29	28	27	26	25	24	23	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. This bit-filed can be modified only when the LPTIMER is enabled (The LPTEN bit in LPTIMER_CTL1 register is 1).

24.5.8. Counter register (LPTIMER_CNT)

Address offset: 0x1C
Reset value: 0x0000

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

31	30	29	28	27	26	25	24	23	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]															

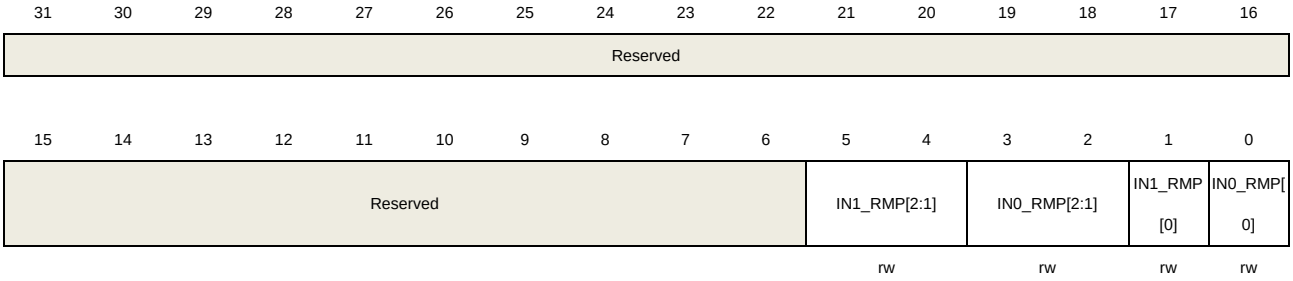
r

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CNT[15:0]	Counter value **Note:** When the LPTIMER uses an asynchronous clock, reads the LPTIMER_CNT register may return unreliable values. So it is necessary to perform two consecutive read operations and confirm whether the two read values are the same.

24.5.9. External input remap register (LPTIMER_EIRMP)

Address offset: 0x20
Reset value: 0x0000

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



Bits	Fields	Descriptions
31:6	Reserved	Must be kept at reset value.
5:4	IN1_RMP[2:1]	LPTIMER input1 remap extension These bits are set and cleared by software. 00: External input is remaped to CMP1_OUT 01: External input is remaped to CMP3_OUT 10: External input is remaped to CMP5_OUT 11: External input is remaped to CMP7_OUT
3:2	IN0_RMP[2:1]	LPTIMER input0 remap extension These bits are set and cleared by software. 00: External input is remaped to CMP0_OUT 01: External input is remaped to CMP2_OUT 10: External input is remaped to CMP4_OUT 11: External input is remaped to CMP6_OUT
1	IN1_RMP[0]	External input1 remap This bit is set and cleared by software. 0: External input is remaped to GPIO. 1: External input is remaped to CMPx_OUT according to IN1_RMP[2:1].

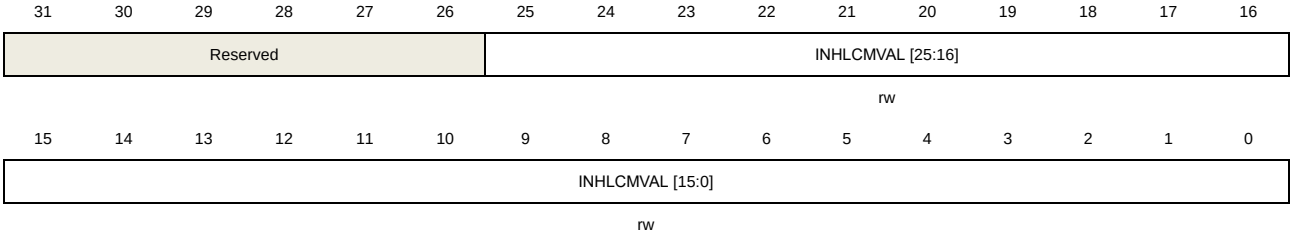
0	IN0_RMP[0]	External input0 remap This bit is set and cleared by software. 0: External input is remaped to GPIO. 1: External input is remaped to CMPx_OUT according to IN0_RMP[2:1].

24.5.10. Input high level counter max value register (LPTIMER_INHLCMV)

Address offset: 0X24

Reset value: 0x0000

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



Bits	Fields	Descriptions
31:26	Reserved	Must be kept at reset value.
25:0	INHLCMVAL	Input high level counter max value This bit can be modified only when the LPTIMER is external input high level counter enabled (The INHLCEN bit in LPTIMER_CTL1 register is 1).

25. High-Resolution Timer (HRTIMER)

25.1. Overview

HRTIMER has a high-resolution counting clock and can be used for high-precision timing. It can generate 16 high resolution and flexible digital signals to control motor or be used for power management applications. The 16 digital signals can be output independently or coupled into 8 pairs of complementary signals.

It has a flexible capture function and can be used to capture timing the input signal. It has multiple internal signals connected to the ADC and DAC. It can be used for control and monitoring purposes.

It can handle various fault input for safe purposes.

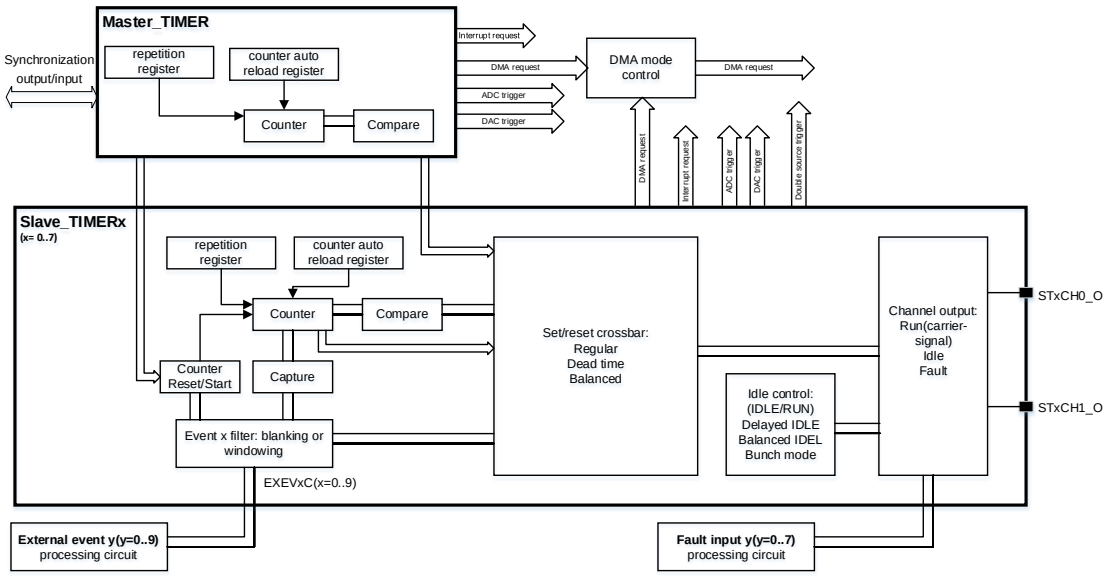
25.2. Characteristics

- High- resolution timing units: Master_TIMER, Slave_TIMERx (x=0.. 7).
- 16 digital signals outputs: they can be controlled by any timing unit and output independently or coupled into 8 pairs.
- Synchronization outputs: synchronize external resources as master.
- Synchronization inputs: be synchronized as a slaver.
- Multiple internal signals connected to the ADC and DAC.
- Various fault input protection scheme: fault input channel and system fault.
- Bunch mode controller to handle light-load operation.
- 10 interrupt vectors: Master_TIMER interrupt, Slave_TIMERx (x=0..7) interrupt, fault interrupts.
- 9 DMA request: Master_TIMER requests, Slave_TIMERx (x=0..7) requests.
- DMA mode for multiple registers update.

25.3. Block diagram

Figure 25-1. HRTIMER block diagram provides details of the internal configuration of the HRTIMER timer.

Figure 25-1. HRTIMER block diagram



25.4. Function overview

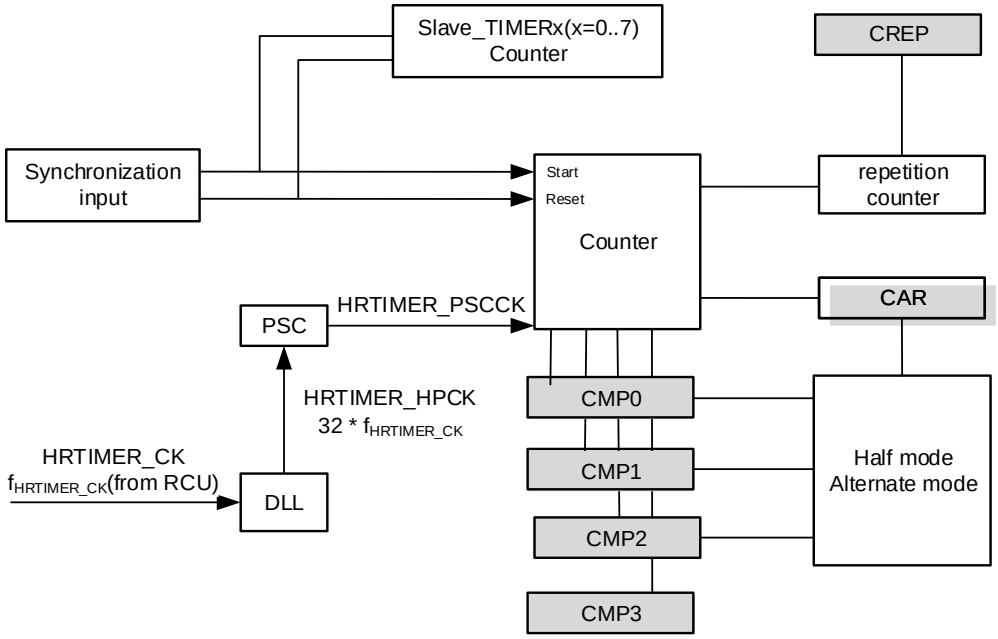
25.4.1. Master_TIMER unit

The Master_TIMER unit is built around the following components:

- 16-bit counter.
- Auto reload register: counting period.
- Repetition counter.
- Compare y (y=0..3) register.

Figure 25-2. Master_TIMER diagram provides details of the internal configuration of the Master_TIMER timer

Figure 25-2. Master_TIMER diagram



The auto-reload register and compare y (y=0..3) register have the following limitations:

When CNTCKDIV[2:0] < 3'b101

- The minimum value must be greater than or equal to the count value corresponding to 3 $t_{HRTIMER_CK}$ period.
- The maximum value must be less than or equal to 0xFFFF – (count value corresponding to 1 $t_{HRTIMER_CK}$ period).

Note: The count value corresponding to each $t_{HRTIMER_CK}$ period = $f_{HRTIMER_PSSCK}$ / $f_{HRTIMER_CK}$.

When CNTCKDIV[2:0] >= 3'b101

- The minimum value must be greater than or equal to 0x0003.
- The maximum value must be less than or equal to 0XFFFE.

Refer to ***Table 25-1. The limitations of auto-reload and compare y (y=0..3) register***.

Table 25-1. The limitations of auto-reload and compare y (y=0..3) register

CNTCKDIV[2:0]	Min value	Max value
3'b000	0x0060	0xFFDF
3'b001	0x0030	0xFFEF
3'b010	0x0018	0xFFF7
3'b011	0x000C	0xFFFB
3'b100	0x0006	0xFFFD
3'b101	0x0003	0xFFFE
3'b110	0x0003	0xFFFE
3'b111	0x0003	0xFFFE

Counter clock

The clock source of Master_TIMER is the HRTIMER_CK from module RCU. The DLL is used to produce a high resolution clock HRTIMER_HPCK ($f_{HRTIMER_HPCK}$ = 32 *$f_{HRTIMER_CK}$). Refer to ***DLL calibrate*** for more information.

The prescaler (PSC) can divide the high resolution clock (HRTIMER_HPCK) to the counter clock (HRTIMER_PSCCK) by factor $2^{CNTCKDIV[2:0]}$ which is controlled by CNTCKDIV[2:0] bit-field in HRTIMER_MTCTL0 register. The frequency relationship between them can be expressed below:

$$f_{HRTIMER_PSCCK} = f_{HRTIMER_HPCK}/2^{CNTCKDIV[2:0]} \quad (25\text{-}1)$$

Note: The clock division CNTCKDIV[2:0] cannot be modified once the Master_TIMER is enabled. CNTCKDIV[2:0] is in HRTIMER_MTCTL0 register.

Figure 25-3. Counter clock when divided by 16 shows some behavior of the counter when the register HRTIMER_MTCAR is set to 0x104 as well as the field CNTCKDIV[2:0] is set to 3'b100.

Figure 25-3. Counter clock when divided by 16

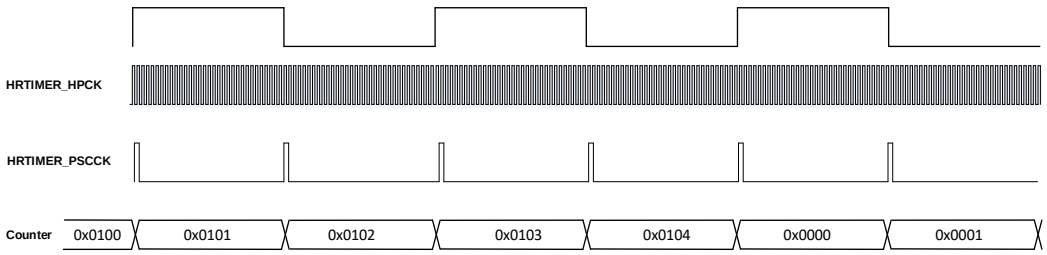


Table 25-2. Resolution with fHRTIMER_CK = 216MHz shows the various resolutions when the $f_{HRTIMER_CK}$ is 216MHz.

Table 25-2. Resolution with $f_{HRTIMER_CK}$ = 216MHz

CNTCKDIV[2:0]	$f_{HRTIMER_PSCCK}$	Resolution
3'b000	216*32MHz=6.912GHz	144.68ps
3'b001	216*16MHz=3.456GHz	289.35ps
3'b010	216*8MHz=1.728GHz	578.70ps
3'b011	216*4MHz=864MHz	1.16ns
3'b100	216*2MHz=432MHz	2.31ns
3'b101	216*1MHz=216MHz	4.63ns
3'b110	216/2MHz=108MHz	9.26ns
3'b111	216/4MHz=54MHz	18.52ns

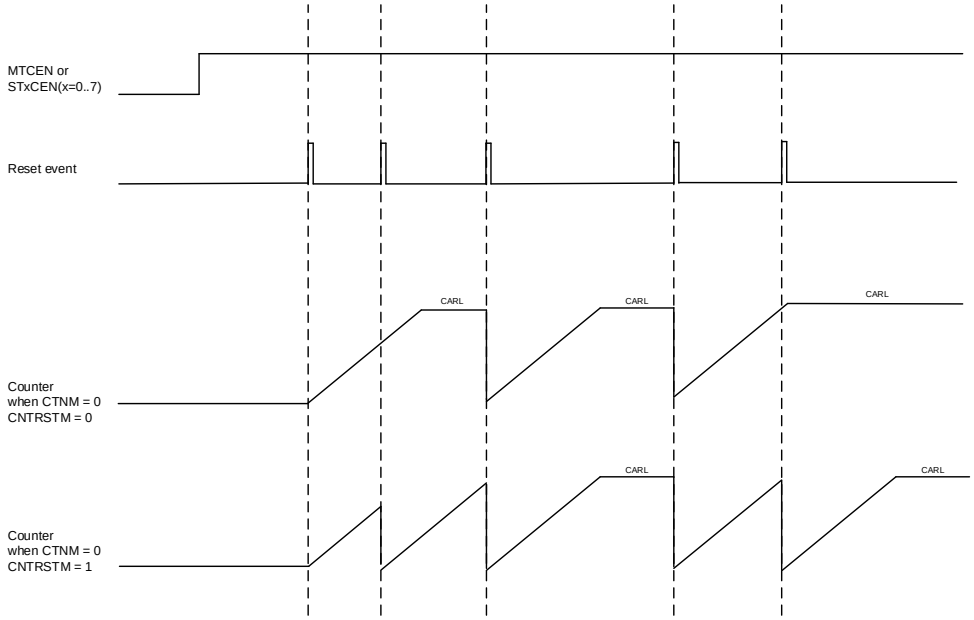
Up counting mode

The counter counts up continuously from 0 to the counter-reload value, which is defined in the HRTIMER_MTCAR register. There are two counter operating mode: either single pulse mode (CTNM = 0 in HRTIMER_MTCTL0 register) or continuous mode (CTNM = 1 in

HRTIMER_MTCTL0 register).

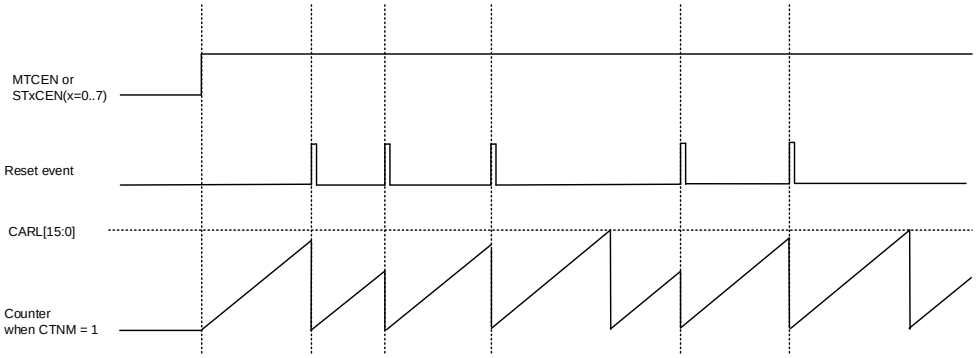
In single pulse mode, after setting the bit MTCEN in HRTIMER_MTCTL0 register, the first reset event will start the counter. When counting up to the counter-reload value, the counter stops and generates a period event. Then the other reset event will reset and restart the counter. During counting process, the reset event will reset and restart the counter if CNTRSTM = 1 in HRTIMER_MTCTL0 register, otherwise it will be ignored. ***Figure 25-4. Counter behavior in single pulse mode*** shows the counter operation diagram in single pulse mode.

Figure 25-4. Counter behavior in single pulse mode



In continuous mode, the counter starts immediately as soon as MTCEN bit in HRTIMER_MTCTL0 register is set to 1. When counting up to the counter-reload value, the counter restarts from 0 and a roll-over event is generated. Different from single pulse mode, the reset events generated at any time will reset and restart the counter.***Figure 25-5. Counter behavior in continuous mode*** shows counter operation diagram in continuous mode.

Figure 25-5. Counter behavior in continuous mode



Repetition counter

When MTCEN bit in HRTIMER_MTCTL0 register is set to 1, the repetition counter load the value of HRTIMER_MTCREP register. The repetition counter is decremented when the counter is cleared due to either a roll-over event in continuous mode or a reset event. When the repetition counter has reached zero, the coming roll-over event in continuous mode or reset event will generate a repetition event and reload the value of HRTIMER_MTCREP register.

The repetition event will set REPIF bit in HRTIMER_MTINTF register to 1, and a repetition interrupt or DMA request is issued if enabled (REPIE = 1 or REPDEN = 1 bits in HRTIMER_MTDMAINTEN register). The repetition interrupt flag can be cleared by writing 1 to REPIFC bit in HRTIMER_MTINTFC.

Figure 25-6. Repetition counter behavior in continuous mode shows repetition counter operation diagram in continuous mode.

Figure 25-6. Repetition counter behavior in continuous mode

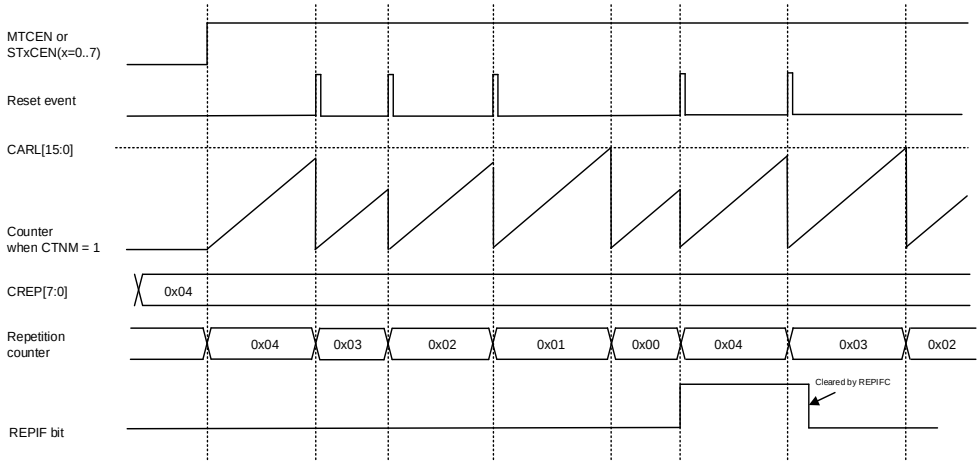


Figure 25-7. Repetition counter behavior in single pulse mode with CNTRSTM = 0 shows repetition counter operation diagram in single pulse mode with CNTRSTM = 0.

Figure 25-7. Repetition counter behavior in single pulse mode with CNTRSTM = 0

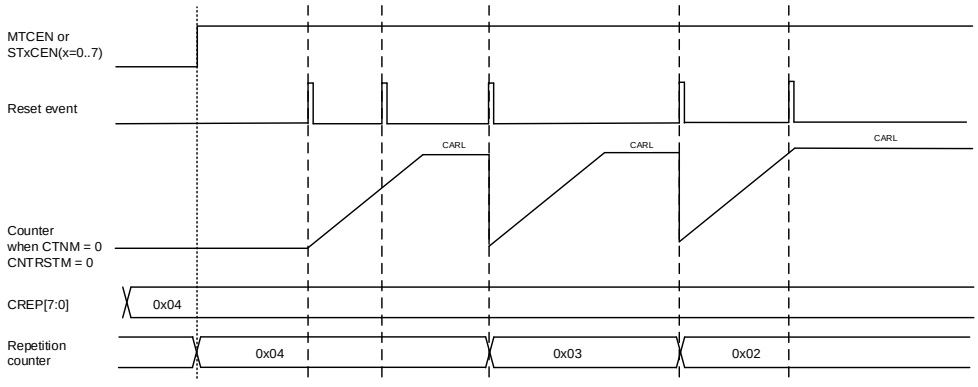
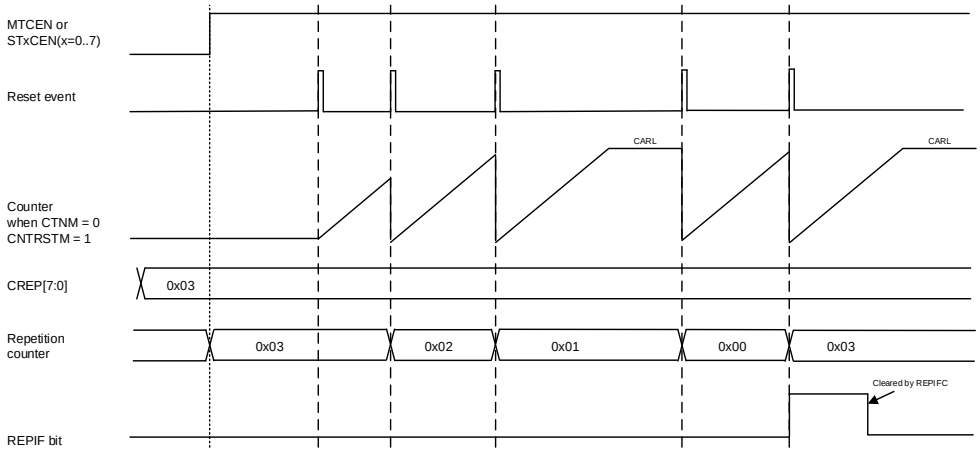


Figure 25-8. Repetition counter behavior in single pulse mode with CNTRSTM = 1 shows

repetition counter operation diagram in single pulse mode with CNTRSTM = 1.

Figure 25-8. Repetition counter behavior in single pulse mode with CNTRSTM = 1



Counter reset

The counter can be reset to 0 by software or synchronous input only once the counter is enabled (MTCEN = 1).

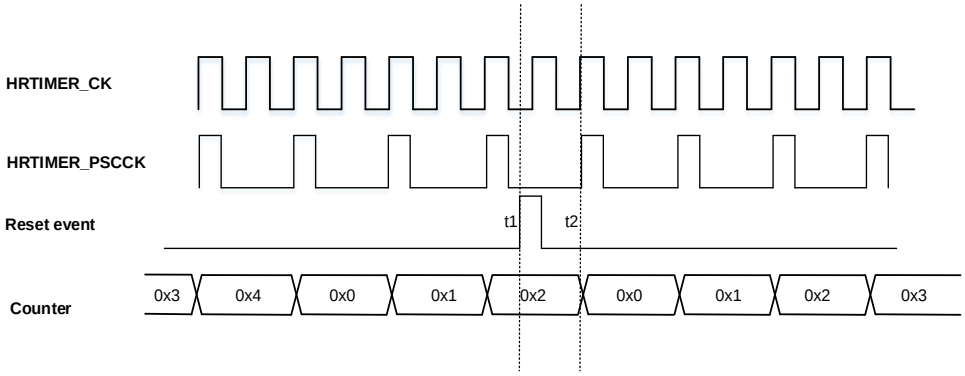
Writing 1 to the MTSRST bit, cleared by hardware automatically, makes the counter reset.

When SYNIRST is set to 1 in HRTIMER_MTCTL0 register, the synchronous input can reset the counter. The details refer to ***Synchronization input***.

When the counter clock HRTIMER_PSCCK prescaling ratio is above 32 (CNTCKDIV[2:0] > 3'b101), the counter reset event is delayed to the next rising edge of the HRTIMER_PSCCK.

Figure 25-9 Reset event resynchronization when prescaling ratio is 64 show CNTCKDIV[2:0] = 3'b110 in continuous mode with HRTIMER_MTCAR = 0x4.

Figure 25-9 Reset event resynchronization when prescaling ratio is 64



Compare

The Master_TIMER unit has four compare registers: HRTIMER_MTCMPxV(x=0..3). When the counter value matches the compare registers value, a coincident compare event is generated.

The compare event will set the corresponding compare interrupt flag to 1 (CMPxIF bit in HRTIMER_MTINTF where x=0..3), and a compare interrupt or DMA request is issued if enabled (CMPxIE = 1 or CMPxDEN = 1 bits in HRTIMER_MTDMAINTEN register where x=0..3). The compare interrupt flag can be cleared by writing 1 to CMPxIFC bit in HRTIMER_MTINTFC where x=0..3.

Half mode

When HALFM bit in HRTIMER_MTCTL0 is set 1, the half mode is enabled. This mode forces the value of compare 0 active register to be half of the counter-reload value, but the value of HRTIMER_MTCMP0V register is not updated with the HRTIMER_MTCAR / 2 value. It is mainly used to generate a square wave with a fixed duty cycle of 50%.

When SHWEN bit in HRTIMER_MTCTL0 is set to 1, the shadow registers are enabled and the compare 0 active register is refreshed on the update event. Otherwise, the compare 0 active register is refreshed as soon as the new value is written into HRTIMER_MTCAR.

Alternate mode

This mode helps achieve an alternate topology that complements the half mode. The compare value registers are automatically recalculated when the HRTIMER_MTCAR value is updated. alternate mode is selected by ALTM[1:0] bits in HRTIMER_MTCTL0 and HRTIMER_STxCTL0.

Table 25-3. Alternate mode selection

ALTM[1:0]	Update event.
00	Disable
01	Triple alternate (120°)
10	Quad alternate (90°)
11	Reserved

Table 25-4. Alternate mode selection shows the compare values for the alternate modes.The contents of the compare register will be overwritten.The corresponding compare event can be used as a trigger, to set or reset the slave timer.

Table 25-4. Alternate mode selection

Mode	Triple alternate (120°)	Quad alternate (90°)
HRTIMER_MTCMP0V	HRTIMER_MTCAR /3	HRTIMER_MTCAR /4
HRTIMER_MTCMP1V	2 x HRTIMER_MTCAR /3	HRTIMER_MTCAR /2
HRTIMER_MTCMP2V	No affect	3 x HRTIMER_MTCAR /4

Note: In alternate modes, the compare registers are controlled by hardware and there is no effect when writing these registers. However the preload register stores the campare value, and will be active when exit alternate mode.

Synchronization input start/reset counter

Synchronous input can generate a counter reset event when SYNIRST is set to 1 in HRTIMER_MTCTL0 register and start counter when SYNISTRT is set to 1 in HRTIMER_MTCTL0 register. Refer to ***Synchronization input*** for more information.

A synchronization input request will set the SYNIIF bit in HRTIMER_MTINTF register to 1, and a interrupt or a DMA request is issued if enabled(SYNIIE = 1 or SYNIDEN = 1 bits in HRTIMER_MTDMAINTEN register).The synchronization input interrupt flag can be cleared by writing 1 to SYNIIFC bit in HRTIMER_MTINTFC.

Update event and shadow registers

Some registers in Master_TIMER contain shadow registers. The shadow registers are disabled after MCU reset.

If the SHWEN bit in HRTIMER_MTCTL0 register is cleared to 0, shadow registers are disabled. The values written into these registers are transferred into active register and take effect immediately.

If the SHWEN bit in HRTIMER_MTCTL0 register is set to 1, shadow registers are enabled and these registers are preloaded. The values written into these registers are transferred into the shadow register and do not take effect immediately. Their content of the shadow is transferred into active register and take effect immediately on update event.

Table 25-5. Master TIMER shadow registers and update event lists the registers containing shadow registers and the relevant event.

Table 25-5. Master_TIMER shadow registers and update event

Registers that contain shadow registers	Shadow registers enable bit	Update event.
HRTIMER_MTDMAINTEN	SHWEN bit in HRTIMER_MTCTL0 register	Software(MTSUP bit)
HRTIMER_MTCAR		Repetition event(UPREP = 1)
HRTIMER_MTCREP		DMA mode end event(UPSEL[1:0] = 2'b01)
HRTIMER_MTCMP0V		Roll-over event following a DMA mode end event (UPSEL[1:0] = 2'b10)
HRTIMER_MTCMP1V		
HRTIMER_MTCMP2V		
HRTIMER_MTCMP3V		

The Master_TIMER has 4 update options:

1. Software generates an update event. Writing 1 to MTSUP bit in HRTIMER_CTL1 register can generate an update event and cancel all the pending hardware update events, regardless of how UPSEL[1:0] in HRTIMER_MTCTL0 register is configured
2. A repetition event generates an update event. If UPREP bit is set to 1 in HRTIMER_MTCTL0 register, the repetition event due to either roll-over or reset events

can generate an update event. If UPSEL[1:0]=2'b10 in HRTIMER_MTCTL0 register, the repetition event can't generate update event.

3. An update event generated when the DMA transfer is completed in DMA mode. If UPSEL[1:0]=2'b01 in HRTIMER_MTCTL0 register, An update event is automatically generated by the hardware when the DMA transfer is completed in DMA mode. It is also possible to generate update event by software or repetition event.
4. Update event generated on counter roll-over following a DMA transfer completion in DMA mode. If UPSEL[1:0]=2'b10 in HRTIMER_MTCTL0 register, An update event is automatically generated by the hardware when a counter roll-over event following a DMA transfer completion is generated in DMA mode. It is also possible to generate update event by software.

A update event will set UPIF bit in HRTIMER_MTINTF register to 1, and an interrupt or a DMA request is issued if enabled(UPIE = 1 or UPDEN = 1 bits in HRTIMER_MTDMAINTEN register).The update event interrupt flag can be cleared by writing 1 to UPIFC bit in HRTIMER_MTINTFC.

DAC Trigger

When the Master_TIMER update event occurs, a DAC trigger request can be generated on HRTIMER_DACTRIGOy(y=0..2) if DACTRGS[1:0] != 2'b00 in HRTIMER_MTCTL0 register. If DACTRGS[1:0] = 2'b00 in HRTIMER_MTCTL0 register, it won't generate DAC trigger request. HRTIMER_DACTRIGOy(y=0..2) is the internal signal connected from Master_TIMER to the DAC module. Refer to ***DAC trigger*** for more information.

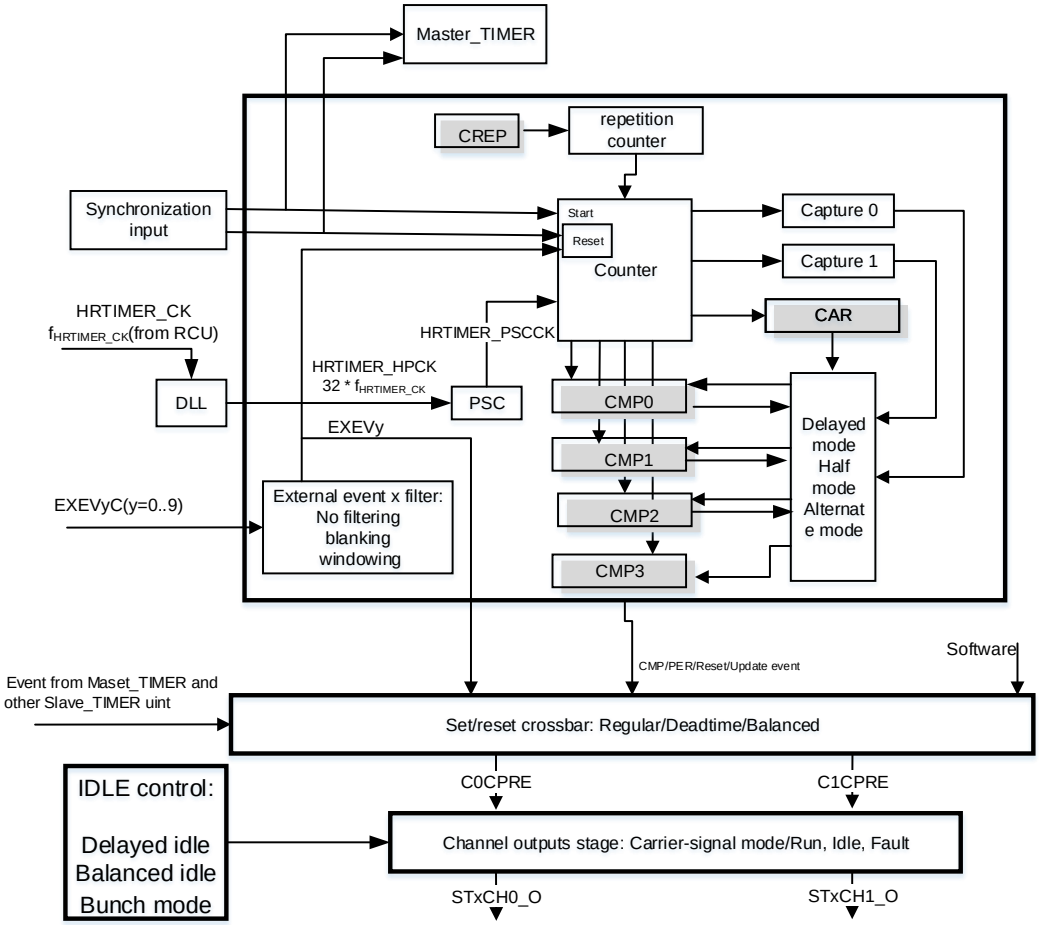
25.4.2. Slave_TIMERx(x=0..7) unit

The HRTIMER has 8 slave timers with similar structure: Slave_TIMERx(x=0..7). Each unit is built around the following components:

- 16-bit counter.
- Auto reload register: counting period.
- Repetition counter.
- Compare y (y=0..3) register.
- Capture y(y=0,1) register.
- Set/reset crossbar.
- IDEL control stage
- Channel output stage.

Figure 25-10. Slave_TIMERx diagram shows the structure of Slave_TIMERx.

Figure 25-10. Slave_TIMERx diagram



The auto-reload register and compare y (y=0..3) registers have the following limitations:

When CNTCKDIV[2:0] < 3'b101

- The minimum value must be greater than or equal to the count value corresponding to 3 $t_{HRTIMER_CK}$ period.
- The maximum value must be less than or equal to 0xFFFF – (count value corresponding to 1 $t_{HRTIMER_CK}$ period).

Note: The count value corresponding to each $t_{HRTIMER_CK}$ period = $f_{HRTIMER_PSSCK}$ / $f_{HRTIMER_CK}$.

When CNTCKDIV[2:0] >= 3'b101

- The minimum value must be greater than or equal to 0x0003.
- The maximum value must be less than or equal to 0XFFFE.

Refer to ***Table 25-1. The limitations of auto-reload and compare y (y=0..3) register***.

The counter and capture y(y=0,1) value registers also have the following limitations: for counter clock division below 32, the least significant bits are not significant. They cannot be written and read 0. Refer to ***Table 25-6. The limitations of counter and capture y(y=0,1) value registers***.

Table 25-6. The limitations of counter and capture y(y=0,1) value registers

CNTCKDIV[2:0]	Bits which are not significant
3'b000	Bit4 to bit0
3'b001	Bit3 to bit0
3'b010	Bit2 to bit0
3'b011	Bit1 to bit0
3'b100	Bit0
3'b101	x
3'b110	x
3'b111	x

Note: "x" means that all bits are significant.

Counter clock

The clock source of Slave_TIMERx is the HRTIMER_CK from module RCU. The DLL is used to produce a high resolution clock HRTIMER_HPCK ($f_{HRTIMER_HPCK} = 32 * f_{HRTIMER_CK}$). Refer to ***DLL calibrate*** for more information.

The prescaler (PSC) can divide the high resolution clock (HRTIMER_HPCK) to the counter clock (HRTIMER_PSCCK) by factor $2^{CNTCKDIV[2:0]}$ which is controlled by CNTCKDIV[2:0] bit-field in HRTIMER_STxCTL0 register. The frequency relationship between them can be expressed below:

$$f_{HRTIMER_PSCCK} = f_{HRTIMER_HPCK} / 2^{CNTCKDIV[2:0]} \quad (25\text{-}2)$$

Note: The clock division CNTCKDIV[2:0] cannot be modified once the Slave_TIMERx is enabled. CNTCKDIV[2:0] is in HRTIMER_STxCTL0 register.

Refer to ***Figure 25-3. Counter clock when divided by 16*** and ***Table 25-2. Resolution with fHRTIMER_CK = 216MHz*** for more information.

Up counting mode

The counter counts up continuously from 0 to the counter-reload value, which is defined in the HRTIMER_STxCAR register. There are two counter operating modes: single pulse mode (CTNM = 0 in HRTIMER_STxCTL0 register) and continuous mode (CTNM = 1 in HRTIMER_STxCTL0 register).

In single pulse mode, after setting the bit STxCEN in HRTIMER_MTCTL0 register, the first reset event will start the counter. When counting up to the counter-reload value, the counter stops and generates a period event. Then the other reset event will reset and restart the counter. During counting process, the reset event will reset and restart the counter if CNTRSTM = 1 in HRTIMER_STxCTL0 register, otherwise it will be ignored. Refer to ***Figure 25-4. Counter behavior in single pulse mode***.

In continuous mode, the counter starts immediately as soon as STxCEN bit in HRTIMER_MTCTL0 register is set to 1. When counting up to the counter-reload value, the

counter restarts from 0 and a roll-over event is generated. Different from single pulse mode, the reset events generated at any time will reset and restart the counter. Refer to ***Figure 25-5. Counter behavior in continuous mode***.

Repetition counter

When STxCEN bit in HRTIMER_MTCTL0 register is set to 1, the repetition counter load the value of HRTIMER_STxCREP register. The repetition counter is decremented when the counter is cleared due to either a roll-over event in continuous mode or a reset event. When the repetition counter has reached zero, the coming roll-over event in continuous mode or reset event will generate a repetition event and reload the value of HRTIMER_STxCREP register.

The repetition event will set REPIF bit in HRTIMER_STxINTF register to 1, and a repetition interrupt or DMA request is issued if enabled (REPIE = 1 or REPDEN = 1 bits in HRTIMER_STxDMAINTEN register).The repetition interrupt flag can be cleared by writing 1 to REPIFC bit in HRTIMER_STxINTFC.

The repetition counter behavior in continuous mode refer to ***Figure 25-6. Repetition counter behavior in continuous mode***.

The repetition counter behavior in single pulse mode with CNTRSTM = 0 refer to ***Figure 25-7. Repetition counter behavior in single pulse mode with CNTRSTM = 0***.

The repetition counter behavior in single pulse mode with CNTRSTM = 1 refer to ***Figure 25-8. Repetition counter behavior in single pulse mode with CNTRSTM = 1***.

Counter reset

The counter can be reset by three types of signal sources:

1. Software. Writing 1 to the STxSRST bit in HRTIMER_CTL1 register.
2. Synchronization input start/reset counter.
3. Events configured in HRTIMER_STxCNTRST register and HRTIMER_STxCNTRSTA register.

All these sources are logical ORed, they can be valid simultaneously. If multiple reset events occur in the same $t_{HRTIMER_CK}$ cycle, only the last one is valid. The counter reset requests are taken into account only once the related Slave_TIMERx are enabled

Note: If the external events is configured with level sensitivity, only one external events can be enabled in the HRTIMER_STxCNTRST register.

Writing 1 to the STxSRST bit, cleared by hardware automatically, makes the counter reset. These control bits of Master_TIMER and Slave_TIMERx(x=0..7) are grouped into HRTIMER_CTL1 register to allow the simultaneous reset of several counters.

When SYNIRST is set to 1 in HRTIMER_STxCTL0 register, the synchronous input can reset the counter. Refer to ***Synchronization input***.

There are 39 events that can be selected simultaneously to reset counter in the HRTIMER_STxCNTRST and HRTIMER_STxCNTRSTA register and these can be classified into four categories:

- Slave_TIMERx itself: update event, compare 1 event and compare 3 event.
- Other Slave_TIMERy (for instance x=1, then y=0, 2..7) : compare 0 event, compare 1 event and compare 3 event.
- Master_TIMER: compare 0 event, compare 1 event, compare 2 event, compare 3 event and reset event.
- External event y(y=0..9): EXEVy conditioned by external event filter in Slave_TIMERx.

When the counter clock HRTIMER_PSCCK prescaling ratio is above 16 (CNTCKDIV[2:0] > 3'b101), the counter reset event is delayed to the next rising edge of the HRTIMER_PSCCK. The details refer to ***Figure 25-9 Reset event resynchronization when prescaling ratio is 64***.

The counter reset event will set RSTIF bit in HRTIMER_STxINTF register to 1, and a counter reset interrupt or DMA request is issued if enabled(RSTIE = 1 or RSTDEN = 1 bits in HRTIMER_STxDMAINTEN register).The counter reset interrupt flag can be cleared by writing 1 to RSTIFC bit in HRTIMER_STxINTFC.

Capure

Capture feature not only allows the Slave_TIMERx to perform measurements such as pulse interval, frequency, period, duty cycle and so on, but also to update compare 1 and compare 3 values in delayed mode (refer to ***Delayed mode***).

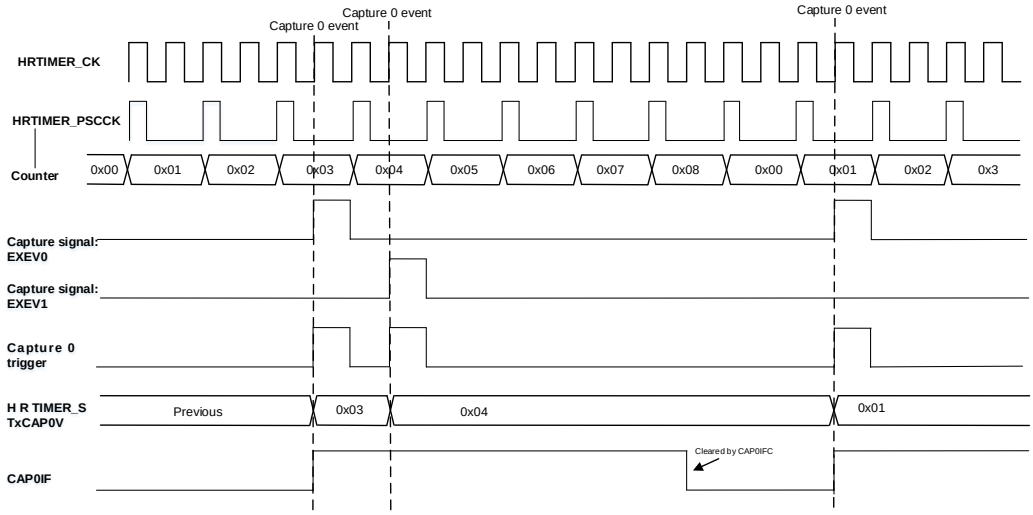
When a selected trigger signal occurs, the current value of the counter is captured into the HRTIMER_STxCAPyV(y=0,1) register. At the same time the CAPyIF(y=0,1) bit in HRTIMER_STxINTF register is set and the capture interrupt and DMA request is generated if enabled by CAPyIE(y=0,1) = 1 and CAPyDEN(y=0,1) = 1 in HRTIMER_STxDMAINTEN register. The capture interrupt flag CAPyIF can be cleared by writing 1 to CAPyIFC bit in HRTIMER_STxINTFC.

The capture 0 trigger events are defined in HRTIMER_STxCAP0TRG register and the capture 1 trigger events are defined in HRTIMER_STxCAP1TRG register. All the trigger events are logical ORed and they are all valid when multiple trigger events are selected.

Note: If the external events is configured with level sensitivity, only one external events can be enabled in the HRTIMER_STxCAPyTRG(y=0,1) register.

The over-capture is not prevented. Even if the previously captured value is not read or the capture flag is not cleared, the new capture will still be triggered and the new captured value will override the previous value. Refer to ***Figure 25-11. Capture 0 triggered by EXEV0 and EXEV1***.

Figure 25-11. Capture 0 triggered by EXEV0 and EXEV1

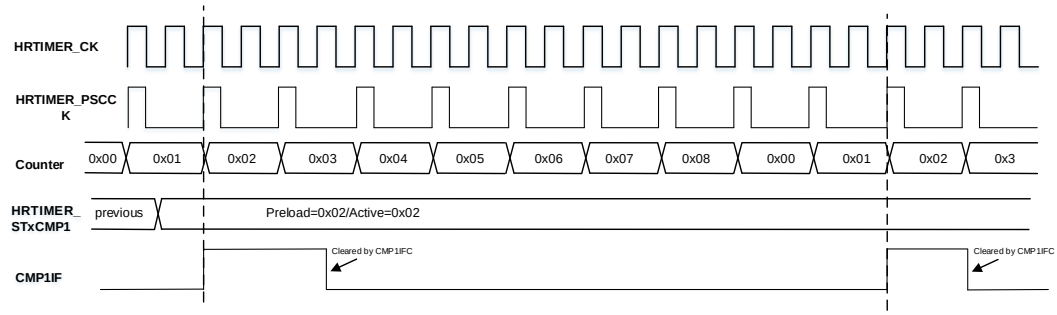


Compare

The Slave_TIMERx unit has four compare registers: HRTIMER_STxCMPyV(y=0..3). When the counter value matches the compare registers value, a coincident compare event is generated. Refer to ***Figure 25-12 . Compare 1 behavior with STxCAR=0x8, STxCMP1V=0x02***.

The compare event will set the corresponding compare interrupt flag to 1 (CMPyIF bit in HRTIMER_STxINTF where y=0..3), and a compare interrupt or DMA request is issued if enabled (CMPyIE = 1 or CMPyDEN = 1 bits in HRTIMER_STxDMAINTEN register where y=0..3). The compare interrupt flag can be cleared by writing 1 to CMPyIFC bit in HRTIMER_STxINTFC.

Figure 25-12. Compare 1 behavior with STxCAR=0x8, STxCMP1V=0x02



Half mode

When HALFM bit in HRTIMER_STxCTL0 is set to 1, the half mode is enabled. This mode forces the value of compare 0 active register to be half of the counter-reload value, but the value of HRTIMER_STxCMP0V register is not updated with the HRTIMER_STxCAR/2 value. It is mainly used to generate a square wave with a fixed duty cycle of 50%.

When SHWEN bit in HRTIMER_STxCTL0 is set to 1, the shadow registers are enabled and

the compare 0 active register is refreshed on the update event. Otherwise, the compare 0 active register is refreshed as soon as the new value is written.

Alternate mode

This mode helps achieve an alternate topology that complements the half mode. The compare value registers are automatically recalculated when the HRTIMER_STxCAR value is updated. alternate mode is selected by ALTM[1:0] bits in HRTIMER_STxCTL0 and HRTIMER_STxCTL0, alternate mode is only vaild when HALFM bit is 0.

Table 25-7. Alternate mode selection

ALTM[1:0]	Alternate mode
00	Disable
01	Triple alternate (120°)
10	Quad alternate (90°)
11	Reserved

Table 25-4. Alternate mode selection shows the active register compare values for the alternate modes.The contents of the compare register will be overwritten.The corresponding compare event can be used as a trigger, to set or reset the slave timer.

Table 25-8. Compare values for the two alternate modes

Mode	Triple alternate (120°)	Quad alternate (90°)
HRTIMER_STxCMP0V	HRTIMER_STxCAR /3	HRTIMER_STxCAR /4
HRTIMER_STxCMP1V	2 x HRTIMER_STxCAR /3	2 x HRTIMER_STxCAR /4
HRTIMER_STxCMP2V	No affect	3 x HRTIMER_STxCAR /4

Note: In alternate modes, the compare registers are controlled by hardware and there is no effect when writing these registers. However the preload register stores the compare value, and will be active when exit alternate mode.

Empty duty cycle abnormal case

The HRTIMER is not support output pulse less than 3 t_{HPTMER_CK} periods,refer to ***Output prepare signal: narrow pulses management***. For example 0x60 if CNTCKDIV[2:0] = 3'b000, 0x30 if CNTCKDIV[2:0] = 3'b001, 0x18 if CNTCKDIV[2:0] = 3'b010. Pulse which is smaller than the above value cannot be output normally.

By writing a null value in HRTIMER_STxCMP0V and HRTIMER_STxCMP2V and configure the following operations. a pulse will be skip in HRTIMER period.

- Update event generates channel 0 “set request”.
- Compare 0(Compare 2) event generates “reset request”.
- Compare 0(Compare 2) event is only useful in timer unit which the compare event belongs.

In the above application condition, re-write the comparison value HRTIMER_STxCMP0V and HRTIMER_STxCMP2V, and the difference value between them is greater than 3, then the normal output is restored.

Exchange mode

The two outputs CH0 and CH1 of slave timer can be exchange bysetting EXCx in HRTIMER_CTL1.and the output of two channels is effective when next update event occur.

The two outputs CH0 and CH1 are exchanged before SET and RESET. crossbar as follows:

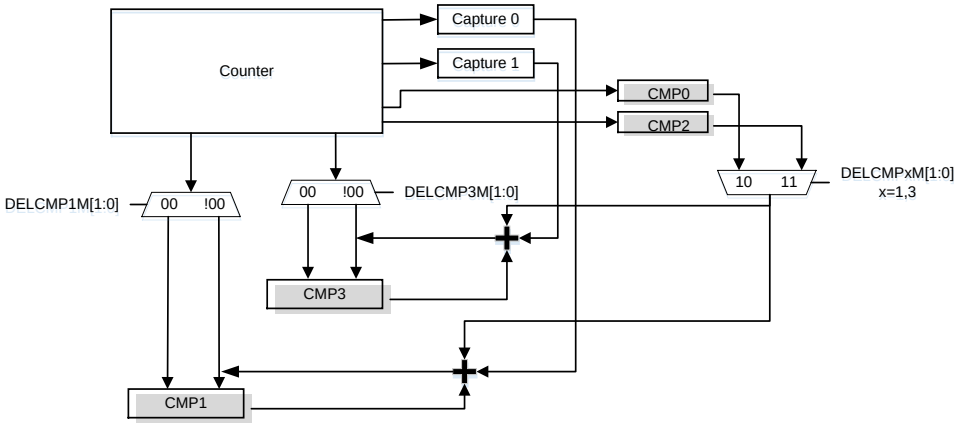
- If EXCx = 0, STxCH0SET and STx0RST control the output of CH0, and STxCH1SET and STxCH1RST control the output of CH1.
- If EXCx = 1, STxCH0SET and STx0RST control the output of CH1, and STxCH0SET and STxCH0RST control the output of CH1.

Note: The exchange mode is only affecting the preload register, the shadow register must be enable when using exchange mode.

Delayed mode

This mode is available only for compare y(y=1,3) and controlled by DELCMPyM[1:0] bit-field in HRTIMER_STxCTL0 register. The actual value compared to the counter is the recalculated value which is recomputed following a capture0/1 trigger or compare0/2 event, as shown in ***Figure 25-13. Compare delayed mode chart***. This mode allows the generation of waveforms to synchronize with the capture trigger by hardware.

Figure 25-13. Compare delayed mode chart



In delay mode, the compare y(y=1, 3) event is valid from the relative capture/compare event up to the period event. When the counter has reached the period value, the compare y(y=1, 3) event is disabled until a new capture/compare comes.

When no relative capture0/1 trigger or compare0/2 event occurs, no compare y event is generated. Once the relative capture is triggered, the value in compare y active register is summed with the relative HRTIMER_STxCAP0V/ HRTIMER_STxCAP1V, and it is compared to the counter. Compare 1 is associated with capture 0 and compare 0/2, while compare 3 is

associated with capture 1 and compare 0/2.

Note: The recalculated value is transferred to an internal register which cannot read.

The DELCMP3M[1:0] (compare 3) and DELCMP1M[1:0] (compare 1) in HRTIMER_STxCTL0 register can be used to configure the delay mode. Take DELCMP1M[1:0] for example：

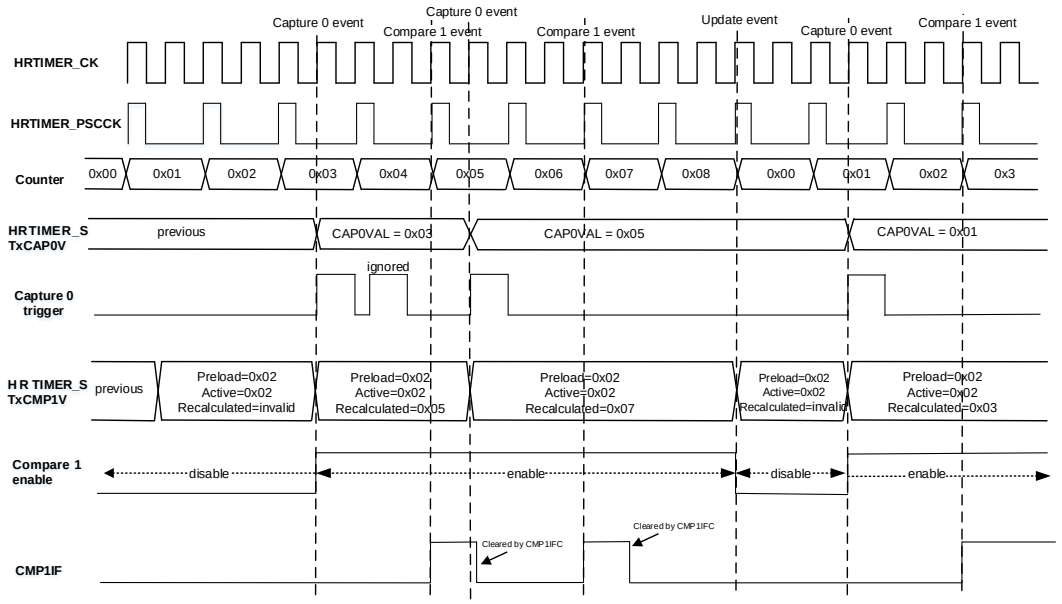
- 2'b00, compare 1 delayed mode disable.

Compare 1 delayed mode disable. Compare match occurs as soon as the counter value is equal to the value of compare 1 active register. Refer to ***Figure 25-12. Compare 1 behavior with STxCAR=0x8, STxCMP1V=0x02***.

- 2'b01, compare 1 delayed mode 0.

After a capture 0 event, the value of compare 1 register is recalculated (compare 1 active register value + capture 0 value). Compare 1 event occurs as soon as the counter value is equal to the recomputed compare 1 value. Refer to ***Figure 25-14. Compare 1 delayed mode 0***.

Figure 25-14. Compare 1 delayed mode 0



- 2'b10, compare 1 delayed mode 1.

After a capture 0 event or compare 0 event, the value of compare 1 register is recalculated.

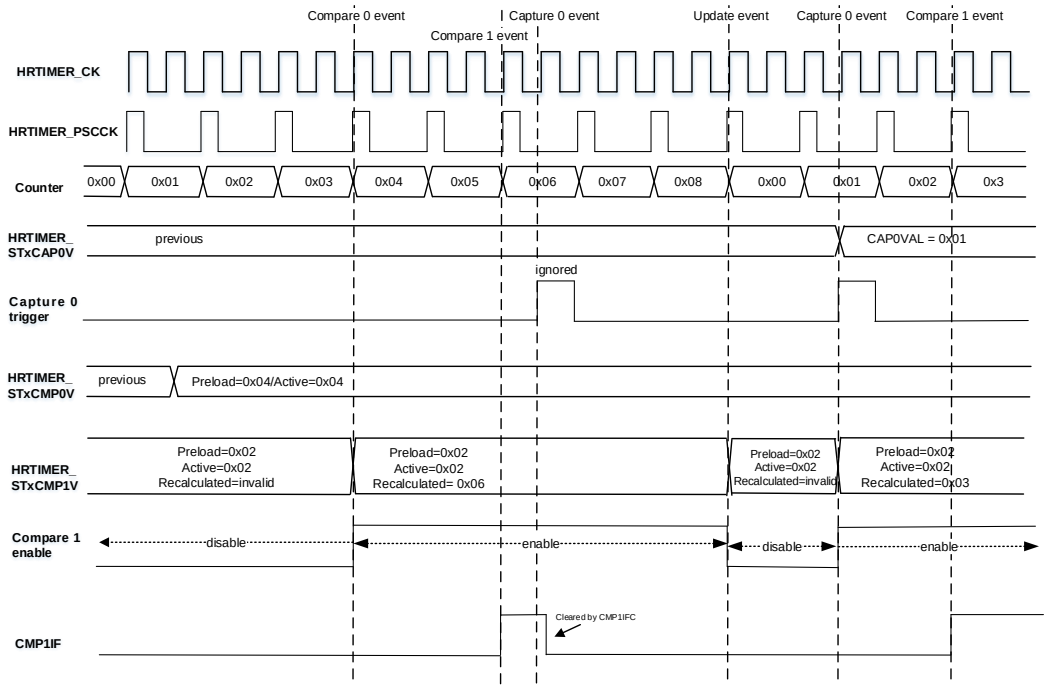
For capture 0 trigger, the recomputed value of compare 1 register = compare 1 active register value + capture 0 value for capture 0 event

For compare 0 event, the recomputed value of compare 1 register = compare 1 active register value previous + compare 0 active register value.

Compare 1 event occurs as soon as the counter value is equal to the recomputed compare 1 value. If capture 0 trigger occurs first, the later compare 0 event is ignored. If compare 0 event occurs first, the capture 0 trigger after the compare 1 event is ignored. Refer to ***Figure 25-15.***

Compare 1 delayed mode 1.

Figure 25-15. Compare 1 delayed mode 1



■ 2'b11, compare 1 delayed mode 2.

This mode is similar to compare 1 delayed mode 1.

After a capture 0 event or compare 2 event, the value of compare 1 register is recalculated.

For capture 0 event, the recalculated value of compare 1 register = compare 1 active register value + capture 0 value for capture 0 event

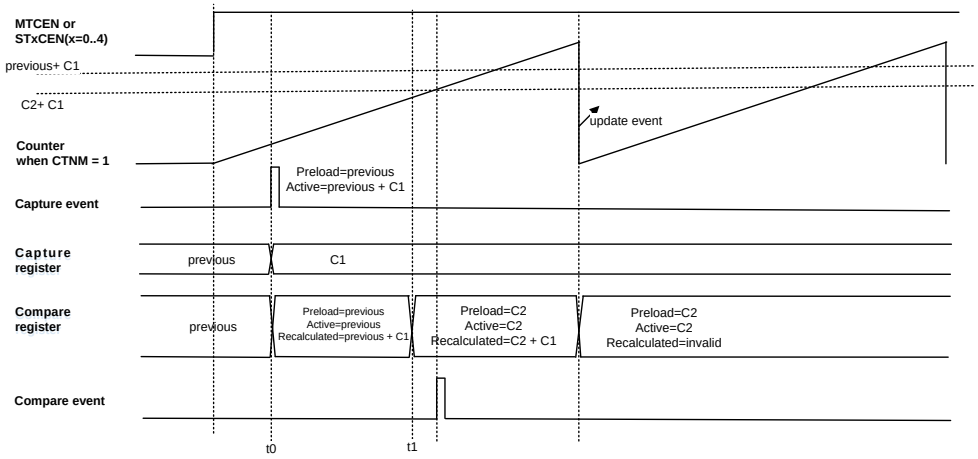
For compare 2 event, the recalculated value of compare 1 register = compare 1 active register value + compare 2 active register value.

Compare match occurs as soon as counter equal the recalculated value of compare 1 register. The captures are not taken into account if they are triggered after (compare 1 active register value + compare 2 active register value). Refer to ***Figure 25-15. Compare 1 delayed mode 1***

When the shadow registers are disabled (SHWEN = 0), the new compare value is taken into the active registers immediately, even if the HRTIMER_STxCMP0V or HRTIMER_STxCMP2V are modified after the capture event has occurred. ***Figure 25-16. Compare delayed mode with SHWEN = 0*** shows an example:

At t0, a capture event occurs and C1 is captured into the register. The recalculated value = the value of compare active register + C1. At t1, new compare value (C2) is written into compare register and the recalculated value = C2 + C1.

Figure 25-16. Compare delayed mode with SHWEN = 0



When the delayed mode is enabled (DELCMPyM[1:0]=01/10/11,y=1,3), the over-capture is prevented. In the same counting cycle (determined by HRTIMER_STxCAR), only the first capture event is taken into account. New capture event are effective in three cases as follows:

- When the recalculated value of compare register matches the counter value.
- When the period event occurs.
- When the counter is reset.

Variable frequency half mode

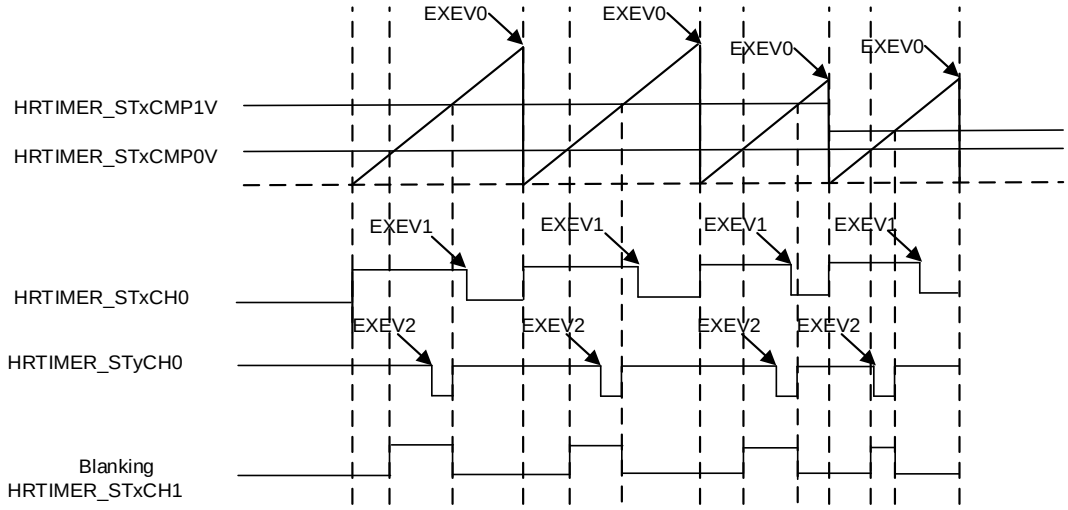
The variable frequency half mode is complementary to the half mode, The output signal frequency is adjustable while the 180° phase shift is guaranteed. The principle of this mode is mainly the master-slave mode, the slave converter(Slave_TIMERy) is continuously adjusted by the period of master vonverter(Slave_TIMERx).

The capture enent occurs on master vonverter(Slave_TIMERx), and the half of the captured value is stored in HRTIMER_STxCMP1V by hardware. the SET and RESET event of slave converter(Slave_TIMERy) can be triggered by master vonverter(Slave_TIMERx) compare 1 event.

By setting TRGHALFM in HRTIMER_STxCTL1, the variable frequency half mode is enable, and this bit can not be change when STxCEN is set in HRTIMER_MTCTL0.

User can write the initial value of HRTIMER_STyCMP1V and when the first capture is occur, the initial value is ignored. And HRTIMER_STyCMP1V will not be preload when TRGHALFM in HRTIMER_STxCTL1 is reset.

Figure 25-17. Variable frequency half mode



As shown in ***Figure 25-17. Variable frequency half mode***.

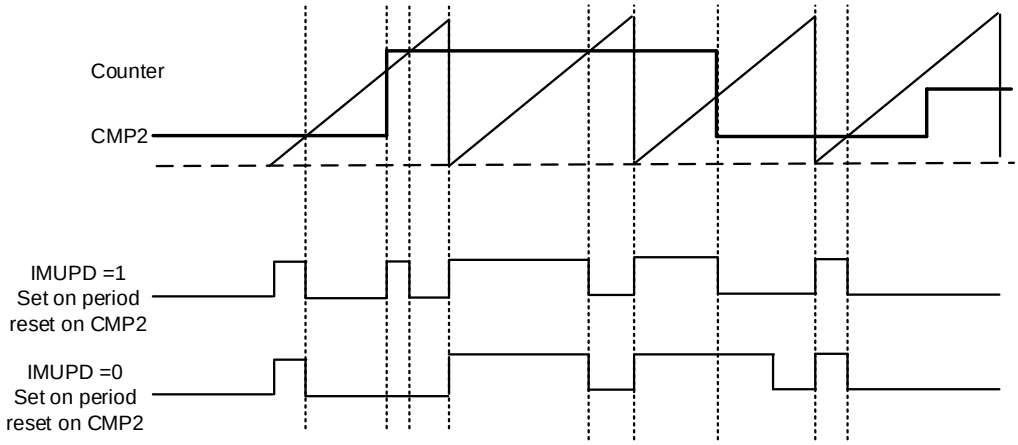
- HRTIMER_STxCH0 is SET by external event 0 (EXEV0), and RESET by external event 1 (EXEV1), EXEV0 trigger the capture event of master vonverter(Slave_TIMERx).
- HRTIMER_STyCH0 is SET by HRTIMER_STxCMP1V RESET by EXEV2.
- HRTIMER_STxCH1 is SET by HRTIMER_STxCMP0V and RESET by HRTIMER_STxCMP1V.

Immediately update mode

The immediately update mode of HRTIMER is available for compare 0 reset event and compare 2 reset event, and this mode is enabled by setting IMUPDxV bit in HRTIMER_STxCTL1 register. The output PWM waveform is updated immediately without waiting for the end of the current period when the immediate update mode is enabled. In the following situation, the PWM waveform is changed immediately.

- When the compare value is changed in running, if the new compare value is less than the counter value and the current compare value is greater than the counter value, the output PWM waveform is reset immediately.
- When the compare value is changed in running, if the new compare value is greater than the counter value and the current compare value is greater than the counter value, the output PWM waveform is set immediately.
- The output PWM waveform doesn't change when the new compare value and the current value are less than the counter value.

Figure 25-18. PWM waveform when IMPUD = 1 and IMUPD = 0



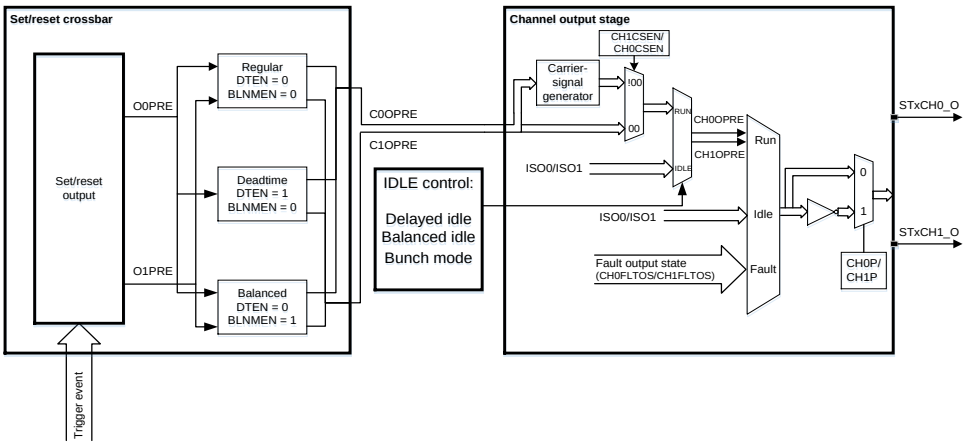
Set/reset crossbar

The channel output waveform can be divided into three stages:

- Set/reset crossbar.
- IDLE control.
- Channel output stage.

Figure 25-19. Channel output diagram shows the structure of the three stages.

Figure 25-19. Channel output diagram



The crossbar has three output modes: regular mode, dead-time mode and balanced mode. Only one of them can be chosen.

Output prepare signal for counter up mode

Slave_TIMERx has a set/reset output module. The module can generate two output prepare signals: O0PRE and O1PRE. O0PRE is controlled by HRTIMER_STxCH0SET and HRTIMER_STxCH0RST registers. O1PRE is controlled by HRTIMER_STxCH1SET and HRTIMER_STxCH1RST registers. The high level of OyPRE(y=0,1) is active level, while the low level is inactive level.

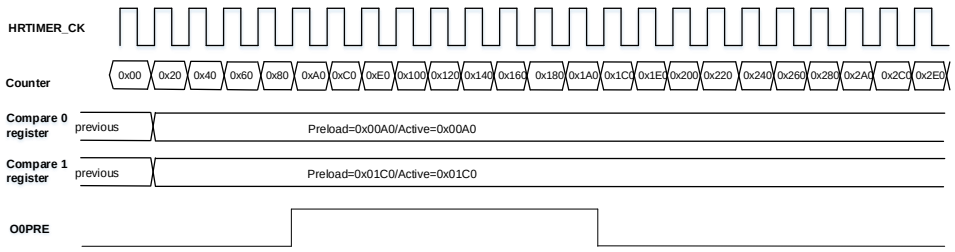
When the event configured in HRTIMER_STxCHySET register occurs, this module produces a “set request” and make the OyPRE high. When the event configured in HRTIMER_STxCHyRST occurs, this module produces a “reset request” and make the OyPRE low. If the same event is configured in HRTIMER_STxCHySET and HRTIMER_STxCHyRST registers, this module produces a “toggle request” and make the OyPRE toggle the output when the configured event occurs.

Note: If the CNTCKDIV[2:0] field in the HRTIMER_STxCTL0 and HRTIMER_STxACTL registers equals to ‘3’b110’ or ‘3’b111’, the same event in HRTIMER_STxCH0SET and HRTIMER_STxCH0RST registers must not be set simulaneously.

Figure 25-20. O0PRE wave: set on CMP0, reset on CMP1 shows O0PRE wave with the following configuration:

- CNTCKDIV[2:0] = 3’b000 in HRTIMER_STxCTL0 registers.
- HRTIMER_STxCH0SET = 0x0000 0008: compare 0 event produces “set request” and O0PRE will output high level.
- HRTIMER_STxCH0RST = 0x0000 0010: compare 1 event produces “reset request” and O0PRE will output low level.
- HRTIMER_STxCMP0V = 0x00A0
- HRTIMER_STxCMP1V = 0x01C0

Figure 25-20. O0PRE wave: set on CMP0, reset on CMP1



There are up to 34 events that can be selected for OyPRE(y=0,1):

- Slave_TIMERx itself: update event, reset event, period event and compare y(y=0..3) event.
- Master_TIMER: period event and compare y(y=0..3) event.
- Slave_TIMERx interconnection event: there are 11 interconnect events from other Slave_TIMERy (for instance x=1, then y=0, 2..7). Refer to ***Table 25-9. Slave_TIMER interconnection event***
- External event y(y=0..9): EXEVy conditioned by external event filter in Slave_TIMERx
- Software event.

Software event is always valid regardless of whether the STxCEN bit is 1. But the other events are only considered when the STxCEN is 1.

Table 25-9. Slave_TIMER interconnection event

Inter-connection		T0 ST0	T0 ST1	T0 ST2	T0 ST3	T0 ST4	T0 ST5	T0 ST6	T0 ST7
From ST0	CMP0	×	0	×	0	×	×	×	0
	CMP1	×	1	0	×	×	×	0	×
	CMP2	×	×	1	×	×	0	×	×
	CMP3	×	×	×	1	0	×	×	×
From ST1	CMP0	0	×	×	×	×	1	×	1
	CMP1	1	×	2	2	×	×	×	2
	CMP2	×	×	3	×	1	×	1	×
	CMP3	×	×	×	3	2	2	2	×
From ST2	CMP0	×	×	×	×	3	3	3	×
	CMP1	2	×	×	×	4	×	×	×
	CMP2	3	2	×	×	×	×	4	3
	CMP3	×	3	×	4	×	4	×	4
From ST3	CMP0	4	×	×	×	5	×	5	×
	CMP1	5	×	4	×	6	×	6	5
	CMP2	×	4	×	×	×	5	×	×
	CMP3	×	5	5	×	×	6	×	6
From ST4	CMP0	×	6	×	5	×	×	×	7
	CMP1	×	7	×	×	×	7	7	×
	CMP2	6	×	6	×	×	8	8	×
	CMP3	7	×	7	6	×	×	×	8
From ST5	CMP0	×	×	×	7	×	×	×	×
	CMP1	×	×	8	×	×	×	×	9
	CMP2	×	8	×	8	7	×	9	×
	CMP3	8	×	×	×	8	×	×	×
From ST6	CMP0	×	×	9	×	×	×	×	×
	CMP1	×	9	×	9	×	×	×	×
	CMP2	9	×	×	×	9	×	×	10
	CMP3	×	×	×	×	×	9	×	×
From ST7	CMP0	×	×	×	10	×	×	×	×
	CMP1	×	×	10	×	10	×	10	×
	CMP2	×	10	×	×	×	10	×	×
	CMP3	10	×	×	×	×	×	×	×

Note: (1) The numbers in the table represent the Slave_TIMERx interconnection event.

(2) "×" means not available.

Multiple event sources can be selected simultaneously (logic ORed) and when they occur in the same t_{HRTMER_CK} period, an arbitration is performed.

Output prepare signal for Center-aligned mode

In the center-aligned counting mode, the counter counts up from 0 to the counter-reload value

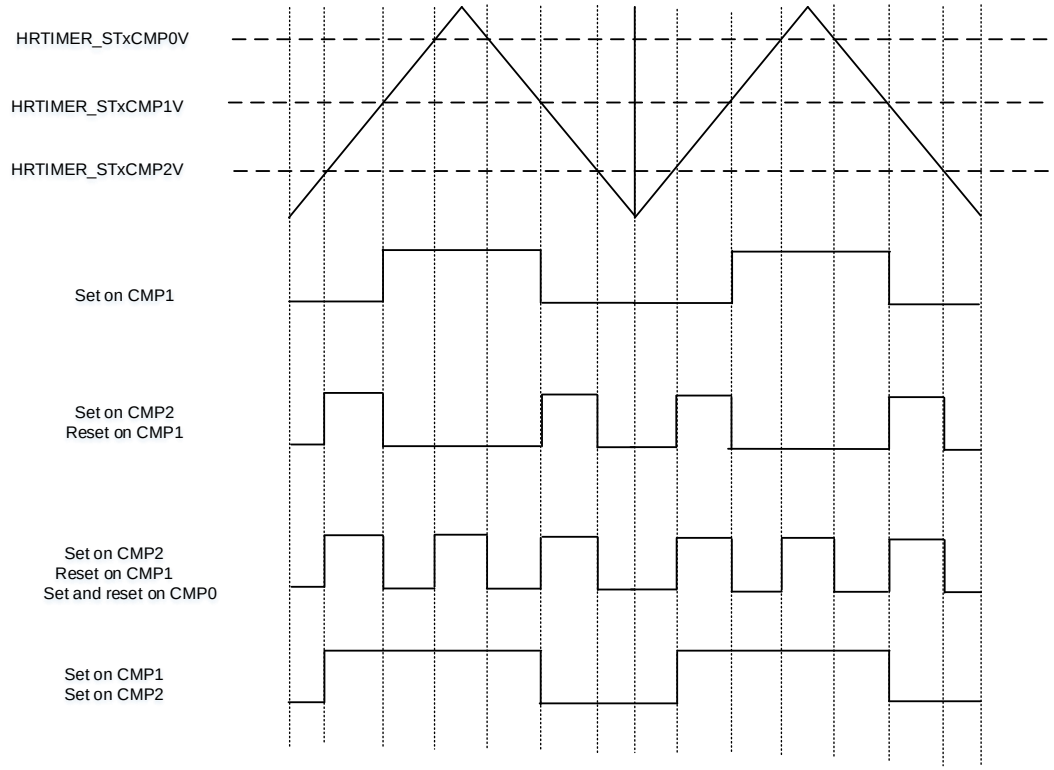
and then counts down to 0 alternatively. The mode is enabled by setting CAM bit in HRTIMER_STxCTL1, Center-aligned counting mode only used in Slave_TIMERx(x = 0..7) not for Master_TIMER.

The period of Slave_TIMERx in HRTIMER_STxCAR only can be preloaded when period event or reset event occurs.

When the event configured in HRTIMER_STxCHySET register occurs, this module produces a “set request” and make the OyPRE high during up counting and make the OyPRE low during down counting. When the event configured in HRTIMER_STxCHyRST occurs, this module produces a “reset request” and make the OyPRE low during up counting and make the OyPRE high during down counting. If the same event is configured in HRTIMER_STxCHySET and HRTIMER_STxCHyRST registers, this module produces a “toggle request” and make the OyPRE toggle the output when the configured event occurs.

There are up to 34 events that can be selected for OyPRE(y=0,1),Refer to ***Output prepare signal for counter up mode***.

Figure 25-21. OyPRE wave in center-aligned mode



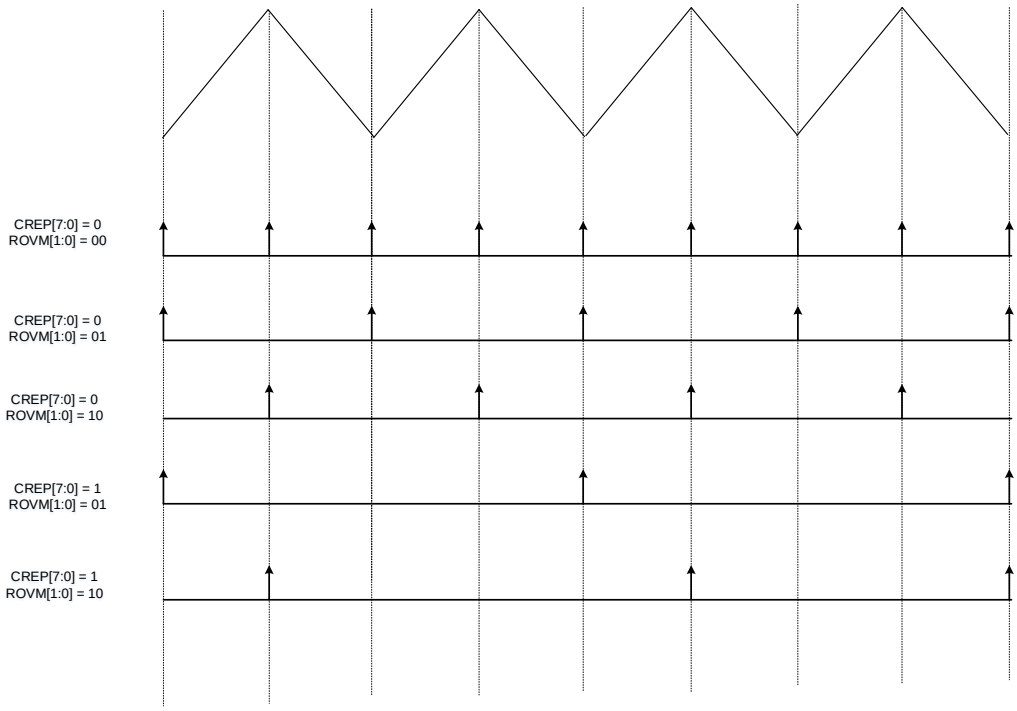
Center aligned mode can be used in half mode,dead-time mode, balanced mode,delayed idele mode, bunch mode, immediately update mode.

The capture value of HRTIMER_STxCAPyV(x =0..7,y=0..3) is referred to start value when up counting, HRTIMER_STxCAPyV = HRTIMER_STxCNT, the capture value of HRTIMER_STxCAPyV(x =0..7,y=0..3) is referred to auto reload value when down counting, HRTIMER_STxCAPyV = HRTIMER_STxCAR - HRTIMER_STxCNT. The DIR in

HRTIMER_STxCAPyV register shows the direction of the count.

The period event used to set or reset OxPRE is defined in ROVM[1:0] bits in HRTIMER_STxCTL1 register. When ROVM[1:0] = 2b00, the period event is generated when the counter is equal to 0 or to HRTIMER_STxCAR value, when ROVM[1:0] = 2b01 the period event is generated when the counter is equal to 0, when ROVM[1:0] = 2b10 the period event is generated when the counter is equal to HRTIMER_STxCAR.

Figure 25-22. Counter repetition value CREP[7:0] and ROVM[1:0] in center aligned mode



HRTIMER has blanking mode and windowing mode, the time of blanking and windowing is different in up counting mode and center aligned mode. Refer to ***Table 25-10. Blanking and windowing in up counting mode and center aligned mode***.

Table 25-10. Blanking and windowing in up counting mode and center aligned mode

EXEV0FM[4:0]	Up counting mode	Center aligned mode
00010	Blanking from period event to compare 1	Blanking from compare 0 to compare 1, only during the up-counting in center aligned mode
00100	Blanking from period event to compare 3	Blanking from compare 2 to compare 3, only during the up-counting in center aligned mode
01101	Windowing from period event to compare 1	Windowing from compare 1 to compare 2, only during the up-counting in center aligned mode
01110	Windowing from period event to	Windowing from compare 1 to

EXEV0FM[4:0]	Up counting mode	Center aligned mode
	compare 2	compare 2, only during the down-counting in center aligned mode
01111	Windowing from another timing unit	Windowing from compare 1 during the up-counting to compare 2 during the down-counting in center aligned mode

Arbitration mechanism

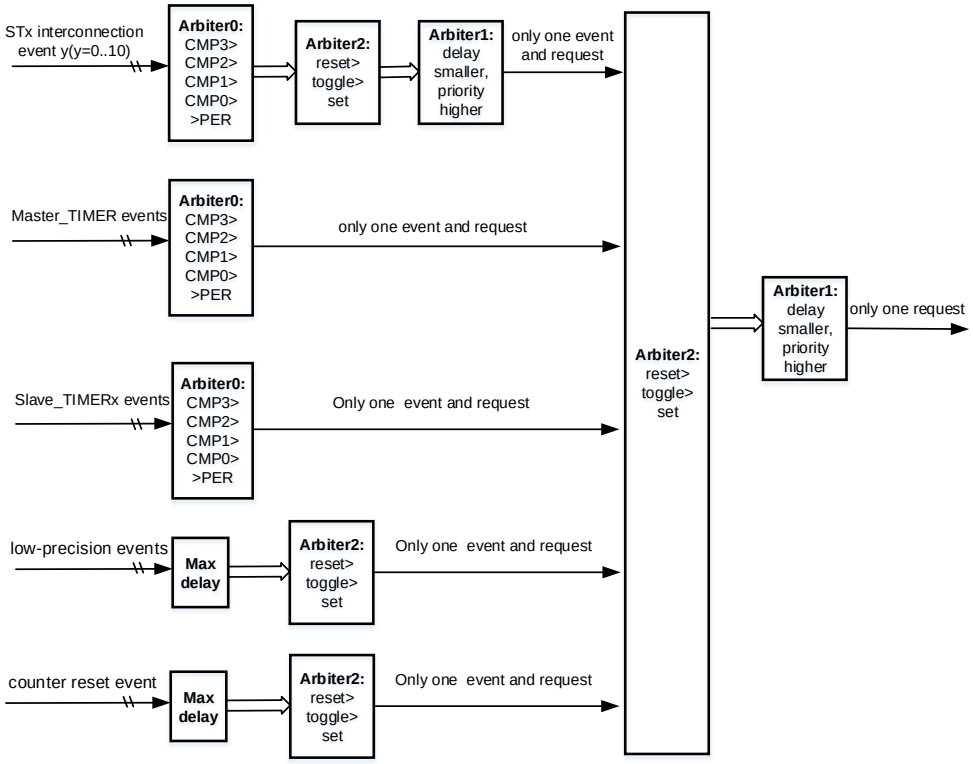
When multiple event configured in HRTIMER_STxCH1SET and HRTIMER_STxCH1RST registers occur during a t_{HPTMER_CK} period, an arbitration is performed and only one will be valid to change the OyPRE(y=0,1).

These 35 events can be divided into five types:

- From Slave_TIMERx itself: period event and compare y(y=0..3) event.
- From Master_TIMER: compare y(y=0..3) event, period event.
- From Slave_TIMERx interconnection event: interconnect event y(y=0..10)
- Low-precision events: Slave_TIMERx update event and reset event, external event y(y=0..9), software event.
- Counter reset event: use max delay.

The arbitration process for each type of event is shown in ***Figure 25-23. Arbitration mechanism during each tHRTMER_CK period***

Figure 25-23. Arbitration mechanism during each t_{HRTMER_CK} period



The functions of the three arbiters are as follows:

- The priority order of arbiter 0(from the highest to the lowest priority):

Compare 3 event > compare 2 event > compare 1 event > compare 0 event > period event.

- Arbitrator 1 arbitrates priority according to the delay of the event during t_{HPTMER_CK}:

The smaller the delay, the higher the priority.

- Arbitrator 2 arbitrates priority according to the effect of the event on the OyPRE(y=0,1):

Reset request > toggle request > set request.

Take O0PRE in Slave_TIMER0 for example and the configuration is:

- HRTIMER_STxCH0SET = 0x0060 5898, and selected events producing “set request” are:

From Master_TIMER: compare 2 event, period event.

From Slave_TIMER0 itself: compare 1 event, compare 0 event.

Interconnection event to Slave_TIMER0: interconnection event 0 (Slave_TIMER1 compare 0 event), interconnection event 2 (Slave_TIMER1 compare 3 event).

Low-precision events: external event 0(EXEV0), external event 1(EXEV1)

Counter reset event.

- HRTIMER_STxCH0RST = 0x0198 0344 and selected events producing “reset request” are:

From Master_TIMER: compare 0 event, compare 1 event.

From Slave_TIMER0 itself: compare 3 event, period event.

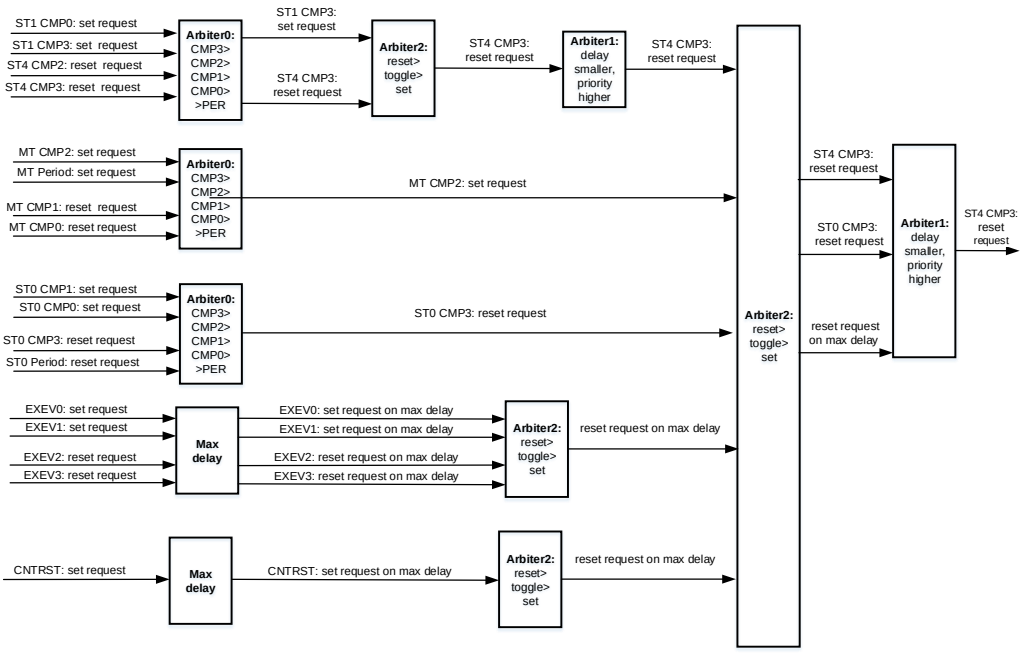
Interconnection event to Slave_TIMER0: interconnection event 7 (Slave_TIMER4 compare 2 event), interconnection event 8 (Slave_TIMER4 compare 3 event).

Low-precision events: external event 2(EXEV2), external event 3(EXEV3)

- The delay: Slave_TIMER4 compare 3 < Slave_TIMER0 compare 3

If the selected events above occur during one t_{HPTMER_CK} period, the arbitration process and results are shown in ***Figure 25-24. Arbitration mechanism example***. Finally, the “reset request” generated by Slave_TIMER4 compare 3 event is valid during the t_{HPTMER_CK} period and O0PRE will be set to low level.

Figure 25-24. Arbitration mechanism example



Output prepare signal: narrow pulses management

When several “set and/or reset requests” are occurring within 3 consecutive t_{HPTMER_CK} periods, OyPRE(y=0,1) is a narrow pulse. The management of the narrow pulse is different depending on CNTCKDIV[2:0]. It is described in two cases:

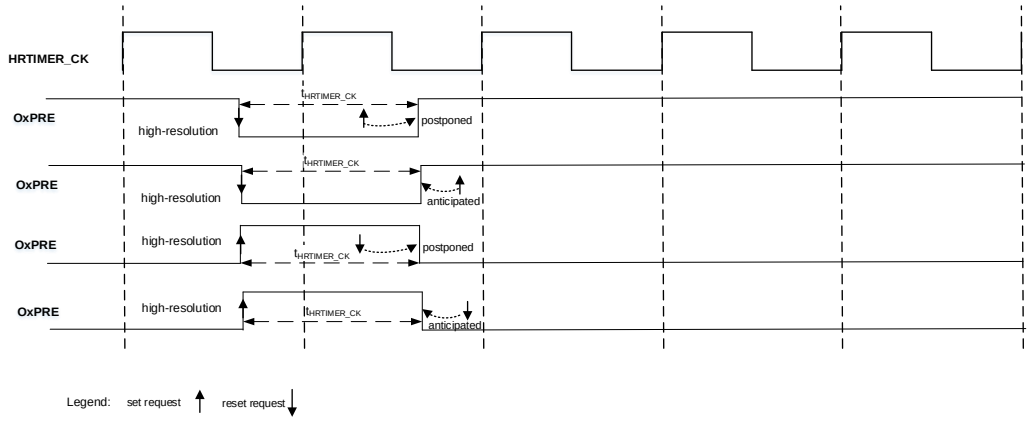
- Case 0: CNTCKDIV[2:0] < 3’b101
- Case 1: CNTCKDIV[2:0] >= 3’b101

Case 0: CNTCKDIV[2:0] < 3’b101

If the “set and reset requests” are generated within two successive t_{HPTMER_CK} period, a pulse

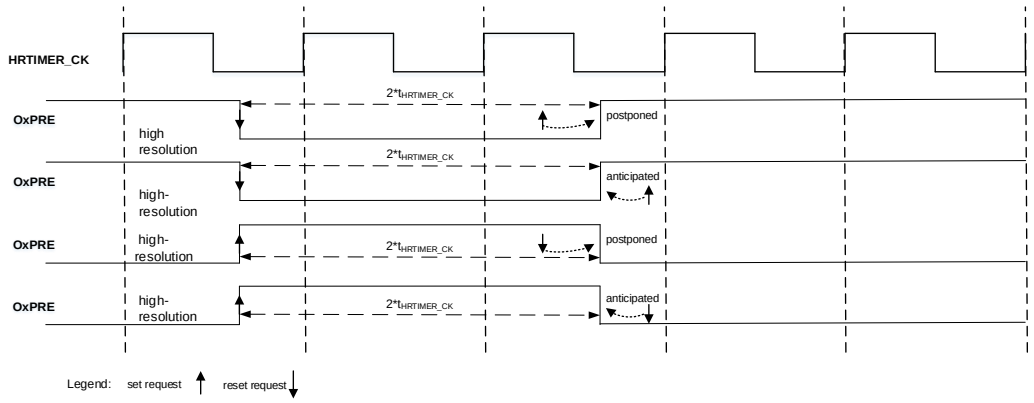
of 1 t_{HPTMER_CK} period is generated. Refer to ***Figure 25-25. A pulse of 1 tHRTMER_CK period***.

Figure 25-25. A pulse of 1 t_{HRTMER_CK} period



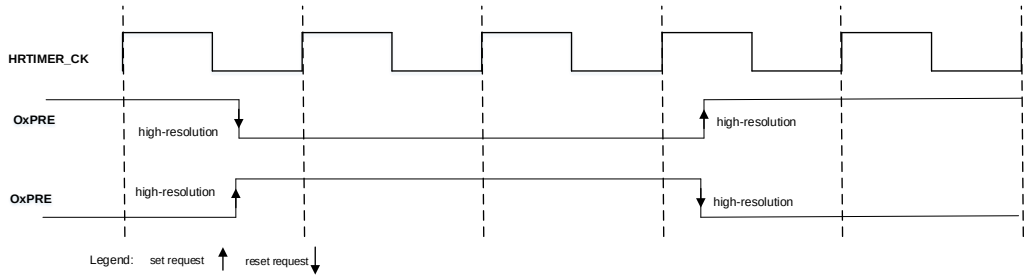
If the “set and reset requests” are generated with an interval including one complete t_{HPTMER_CK} period, a pulse of 2 t_{HPTMER_CK} periods is generated. Refer to ***Figure 25-26. A pulse of 2 tHRTMER_CK period***.

Figure 25-26. A pulse of 2 t_{HRTMER_CK} period



If the “set and reset requests” are generated with an interval including more than two complete t_{HPTMER_CK} periods, the high-resolution is available. Refer to ***Figure 25-27. High-resolution OxPRE wave***.

Figure 25-27. High-resolution OxPRE wave

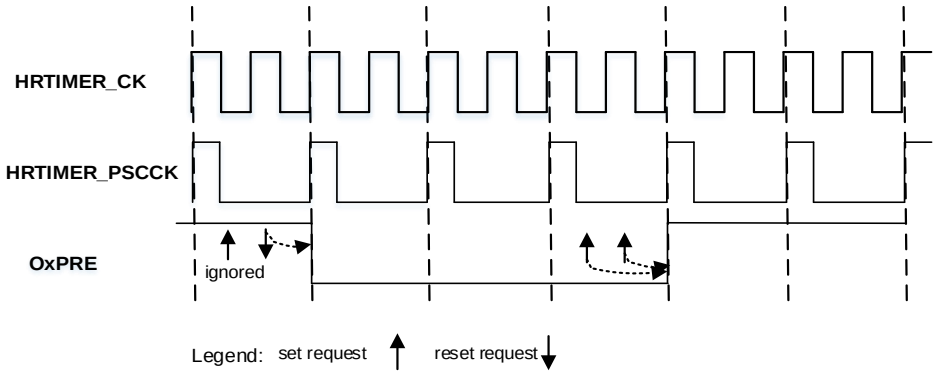


Case 1: CNTCKDIV[2:0] >= 3’b101

An “set or reset request” occurring within the HRTIMER_PSCCK cycle is delayed to the next active edge of the HRTIMER_PSCCK, even if the arbitration is still performed every $t_{HRTIMER_CK}$ cycle.

When “set and reset requests” from different event sources simultaneously occur in a tHRTIMER_CK cycle, the “reset request” has the highest priority. In HRTIMER_PSCCK cycle, subsequent requests override previous requests, and only the last request of that cycle is valid. Refer to ***Figure 25-28. OxPRE wave with CNTCKDIV[2:0] = 3’b110***.

Figure 25-28. OxPRE wave with CNTCKDIV[2:0] = 3’b110



Regular mode

When DTEN = 0 in HRTIMER_STxCHOCTL register and BLNMEN = 0 in HRTIMER_STxCTL0 register, the set/reset crossbar run in regular mode.

In this mode, C0OPRE and C1OPRE are independent. C0OPRE (C1OPRE) is directly connected to O0PRE(O1PRE).

When the Slave_TIMERx(x=0..7) runs in the RUN or IDEL state and the C0OPRE goes from inactive to active level, CH0OAIF bit in HRTIMER_STxINTF register will be set to 1, and a output active interrupt or a DMA request is issued if enabled (CH0OAIE = 1 or CH0OADEN = 1 bit in HRTIMER_STxDMAINTEN register). The CH0OAIF interrupt flag can be cleared by writing 1 to CH0OAIFC bit in HRTIMER_STxINTFC.

When the Slave_TIMERx(x=0..7) runs in the RUN or IDEL state and the C0OPRE goes from active to inactive level, CH0ONAIF bit in HRTIMER_STxINTF register will be set to 1, and a output inactive interrupt or a DMA request is issued if enabled (CH0ONAIE = 1 or CH0ONADEN = 1 bit in HRTIMER_STxDMAINTEN register). The CH0ONAIF interrupt flag can be cleared by writing 1 to CH0ONAIFC bit in HRTIMER_STxINTFC.

The channel 1 is similar to channel 0.

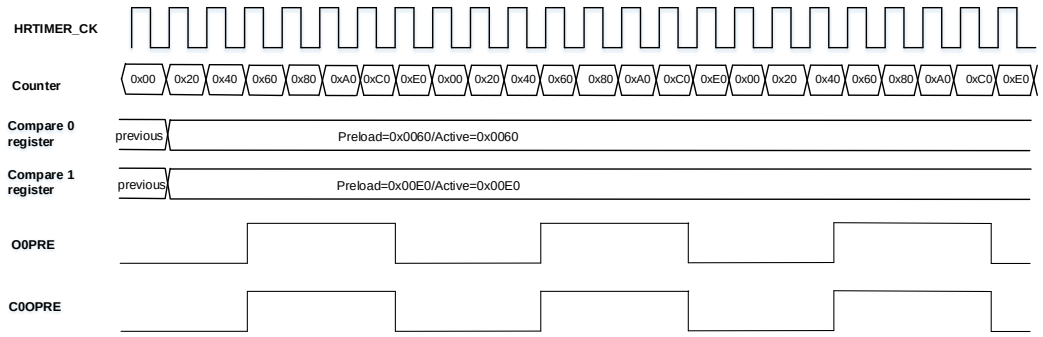
Figure 25-29. C0OPRE wave in regular mode shows C0OPRE wave with the following configuration:

- CNTCKDIV[2:0] = 3’b000 in HRTIMER_STxCTL0 register.
- HRTIMER_STxCH0SET = 0x0000 0008: compare 0 event produces “set request” and

O0PRE will be set to high level.

- HRTIMER_STxCH0RST = 0x0000 0010: compare 1 event produces “reset request” and O0PRE will be set to low level.
- HRTIMER_STxCMP0V = 0x0060
- HRTIMER_STxCMP1V = 0x00E0

Figure 25-29. C0OPRE wave in regular mode



Dead-time mode

When DTEN = 1 in HRTIMER_STxCHOCTL register and BLNMEN = 0 in HRTIMER_STxCTL0 register, the set/reset crossbar run in dead-time mode.

In dead-time mode, only O0PRE is programmed to drive C0OPRE and C1OPRE. C0OPRE and C1OPRE are a couple of complementary signals with programmable dead-time insertion between active state transitions.

The dead-time values are defined with DTFCFG[15:0] and DTRCFG[15:0] bit-fields. DTFCFG[15:0] bit-field defines the value of the dead-time following a falling edge of O0PRE, and DTRCFG[15:0] bit-field defines the value of the dead-time following a rising edge of O0PRE.

Note: DTFCFG[8:0] and DTRCFG[8:0] bit-fields are in HRTIMER_STxDTCTL register. DTFCFG[15:9] and DTRCFG[15:9] bit-fields are in HRTIMER_STxACTL register

The dead-time values can be positive or negative controlled by DTRS bit and DTFS bit in HRTIMER_STxDTCTL register. Negative dead-time values can be defined when some waves overlap is required.

The dead-time values are based on a specific clock division according to DTGCKDIV[3:0] bit-field in HRTIMER_STxDTCTL register.

When the Slave_TIMERx(x=0..7) runs in the RUN or IDEL state and the C0OPRE goes from inactive to active level, CH0OAIF bit in HRTIMER_STxINTF register will be set to 1, and a output active interrupt or a DMA request is issued if enabled (CH0OAIE = 1 or CH0OADEN = 1 bit in HRTIMER_STxDMAINTEN register). The CH0OAIF interrupt flag can be cleared by writing 1 to CH0OAIFC bit in HRTIMER_STxINTFC.

When the Slave_TIMERx(x=0..7) runs in the RUN or IDEL state and the C0OPRE goes from

active to inactive level, CH0ONAIF bit in HRTIMER_STxINTF register will be set to 1, and a output inactive interrupt or a DMA request is issued if enabled (CH0ONAIE = 1 or CH0ONADEN = 1 bit in HRTIMER_STxDMAINTEN register). The CH0ONAIF interrupt flag can be cleared by writing 1 to CH0ONAIFC bit in HRTIMER_STxINTFC.

The channel 1 is similar to channel 0.

Figure 25-30. C0OPRE and C1OPRE complementary wave with dead-time shows C0OPRE and C1OPRE wave with O0PRE pulse width greater than the dead-time.

Figure 25-30. C0OPRE and C1OPRE complementary wave with dead-time

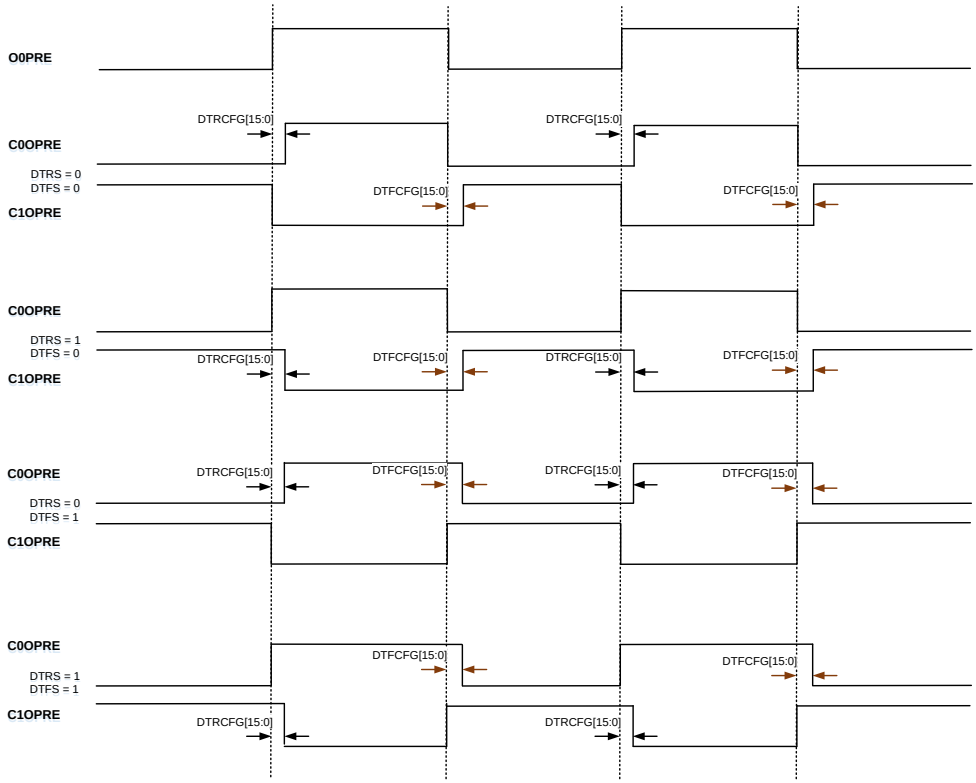
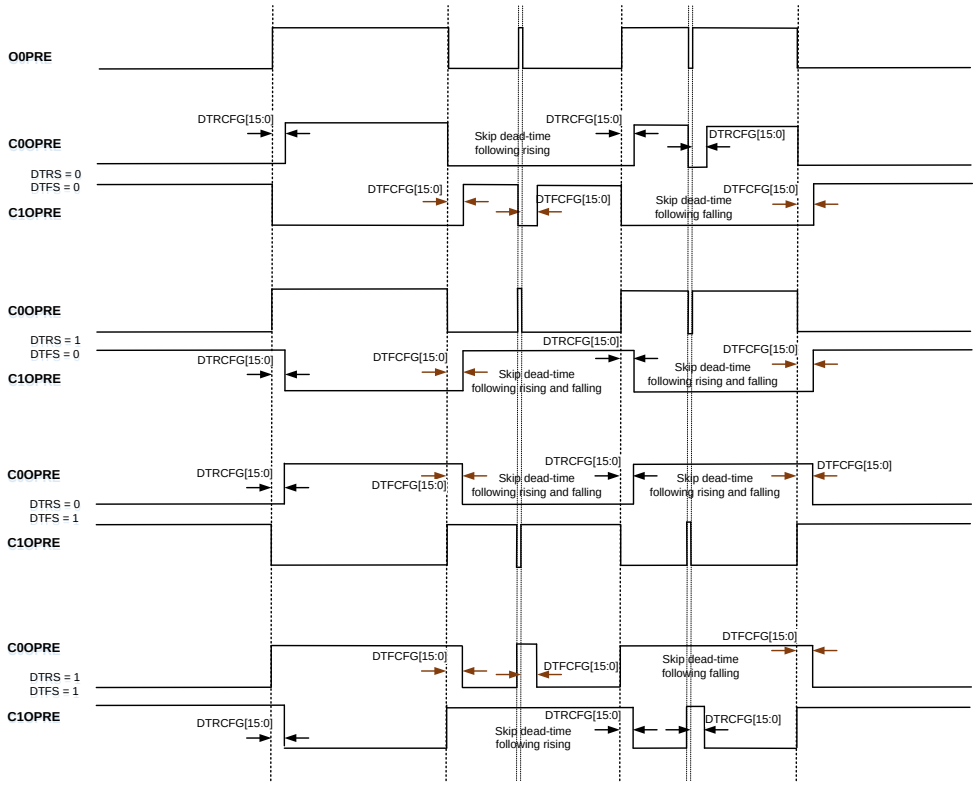


Figure 25-31. Complementary wave with pulse width less than dead-time shows C0OPRE and C1OPRE wave with O0PRE pulse width less than the dead-time.

Figure 25-31. Complementary wave with pulse width less than dead-time

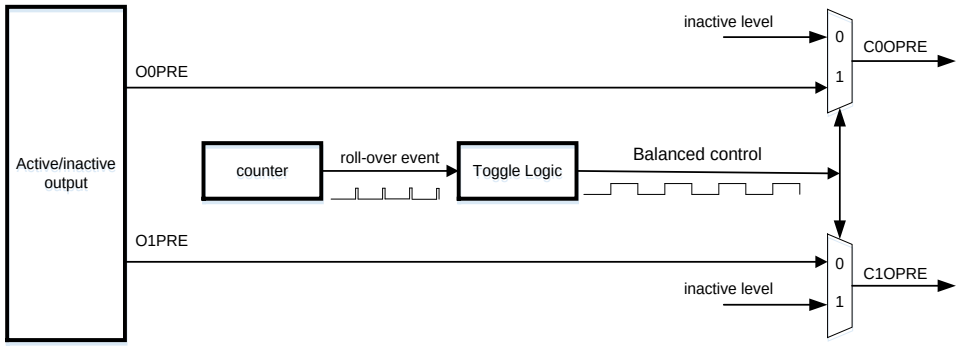


Balanced mode

When DTEN = 0 in HRTIMER_STxCHOCTL register and BLNMEN = 1 in HRTIMER_STxCTL0 register, the set/reset crossbar run in balanced mode. The balanced mode is only available when the counter operates in continuous mode and the counter must not be reset once it has been enabled.

Figure 25-32. Structure chart in balanced mode shows the signal control diagram in balanced mode.

Figure 25-32. Structure chart in balanced mode



Once receiving roll-over event, the output of toggle logic module toggles. When the output of toggle logic module is 1(high level), C0OPRE is connected to O0PRE and C1OPRE is set to inactive level (low level). When the output of toggle logic module is 0(low level), C1OPRE is connected to O1PRE and C0OPRE is set to inactive level (low level).

It is advised to make HRTIMER_STxCH0SET = HRTIMER_STxCH1SET and HRTIMER_STxCH0RST = HRTIMER_STxCH1RST, in order to achieve a balanced operation with identical waveforms. Still, it is possible to have different programming on both outputs for other uses.

The bit CBLNF in HRTIMER_STxINTF register which is reset when the balanced mode is disabled, indicates which channel is currently outputting the signal (O0PRE or O1PRE).

When the Slave_TIMERx(x=0..7) runs in the RUN or IDEL state and the C0OPRE goes from inactive to active level, CH0OAIF bit in HRTIMER_STxINTF register will be set to 1, and a output active interrupt or a DMA request is issued if enabled (CH0OAIE = 1 or CH0OADEN = 1 bit in HRTIMER_STxDMAINTEN register). The CH0OAIF interrupt flag can be cleared by writing 1 to CH0OAIFC bit in HRTIMER_STxINTFC.

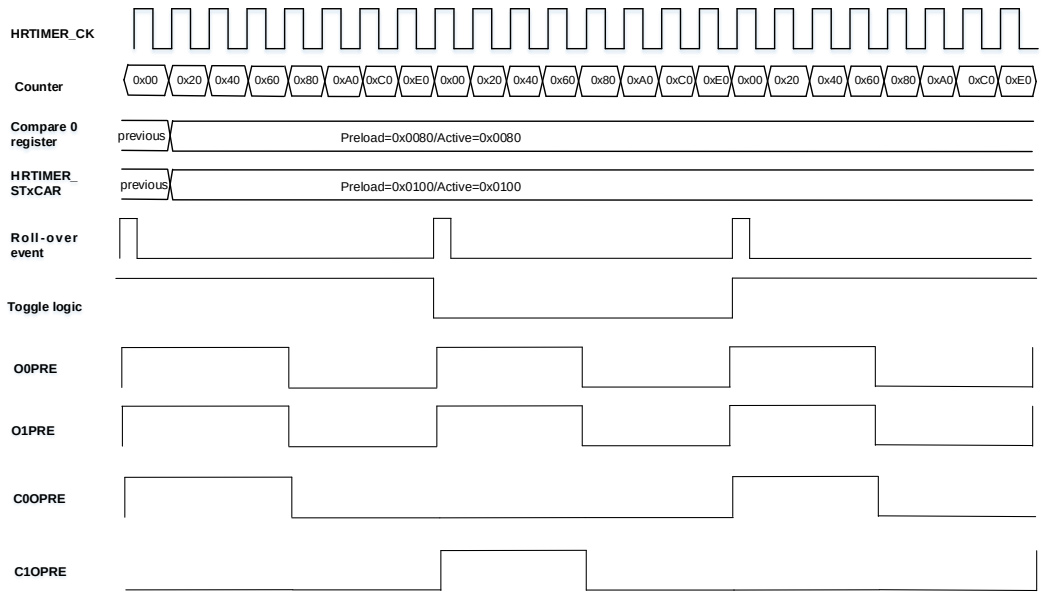
When the Slave_TIMERx(x=0..7) runs in the RUN or IDEL state and the C0OPRE goes from active to inactive level, CH0ONAIF bit in HRTIMER_STxINTF register will be set to 1, and a output inactive interrupt or a DMA request is issued if enabled (CH0ONAIE = 1 or CH0ONADEN = 1 bit in HRTIMER_STxDMAINTEN register). The CH0ONAIF interrupt flag can be cleared by writing 1 to CH0ONAIFC bit in HRTIMER_STxINTFC.

The channel 1 is similar to channel 0.

Figure 25-33. C0OPRE and C1OPRE wave in balanced mode shows C0OPRE and C1OPRE waves with the following configuration:

- HRTIMER_STxCH0SET = HRTIMER_STxCH1SET = 0x0000 0004: period event produces “set request”.O0PRE and O1PRE will output high level.
- HRTIMER_STxCH0RST = HRTIMER_STxCH1RST = 0x0000 0008: compare 0 event produces “reset request”. O0PRE and O1PRE will output low level.

Figure 25-33. C0OPRE and C1OPRE wave in balanced mode



IDLE control

The stage has three ways to control the IDLE state:

- Delayed IDLE
- Balanced IDLE
- IDLE controlled by bunch mode

Delayed IDLE and balanced IDLE cannot use at the same time. Balanced IDLE is only available in balanced mode. Delayed IDLE or balanced IDLE can use with bunch mode at the same time, but bunch mode has a lowest priority. When set/reset crossbar operates in different modes, different IDLE control operation modes can be used .Refer to ***Table 25-11. Crossbar and IDLE control stage work together.***

Table 25-11. Crossbar and IDLE control stage work together

Set/reset **crossbar operation mode**	**IDLE control stage operation mode**
Regular mode	Delayed IDLE, IDLE controlled by bunch mode
Dead-time mode	Delayed IDLE, Idle controlled by bunch mode
Balanced mode	Either delayed IDLE or balanced IDLE, IDLE controlled by bunch mode

The bits CHyF(y=0,1) in HRTIMER_STxINTF register indicates the level of CHyOPRE.

Delayed IDLE

When DLYISMEN bit in HRTIMER_STxCHOCTL register is set to 1, the delayed IDLE is enabled. In delayed IDLE, the “set request” or “reset request” following the selected external event (EXEV5/6 for Slave_TIMER0/1/2 or EXEV7/8 for Slave_TIMER3/4/5/6/7) causes the CHyOPRE (y=0,1) to enter IDLE state. It is associated with ISOy/CHyP(y=0,1) in HRTIMER_STxCHOCTL register. Refer to ***Table 25-12. Request to enter in IDLE and exit IDLE state*** ISOy define the CHyOPRE level in IDLE state. The IDLE mode is permanently maintained but the counter continues to run, until the output is re-enabled to exit delayed IDLE. It is re-enabled by “set request” or “reset request” following overwriting STxCH0EN and STxCH1EN bits to 1.

Table 25-12. Request to enter in IDLE and exit IDLE state

ISOy/CHyP y(=0,1) value	**Request to enter in IDLE state**	**Request exit IDLE state**
ISOy = 0, CHyP = 0	“reset request”	“set request” and “reset request”
ISOy = 1, CHyP = 0	“set request”	“set request” and “reset request”
ISOy = 0, CHyP = 1	“set request”	“set request” and “reset request”
ISOy = 1, CHyP = 1	“reset request”	“set request” and “reset request”

The delayed IDLE mode can be applied to a single output (CHyOPRE) or to both outputs (CH0OPRE and CH1OPRE) decided by the bit-field DLYISCH[2:0] in HRTIMER_STxCHOCTL register, as follows:

- DLYISCH[2:0] = 3'b000: The delayed IDLE mode is applied to CH0OPRE.
- DLYISCH[2:0] = 3'b001: The delayed IDLE mode is applied to CH1OPRE.
- DLYISCH[2:0] = 3'b010: The delayed IDLE mode is applied to CH0OPRE and CH1OPRE.

As soon as the selected external event (EXEV5/6 or EXEV7/8) arrives, the DLYIIF bit in HRTIMER_STxINTF register is set to 1, and an interrupt or a DMA request is issued if enabled (DLYIIE = 1 or DLYIDEN = 1 bits in HRTIMER_STxDMAINTEN register). The interrupt flag can be cleared by writing 1 to DLYIIFC bit in HRTIMER_STxINTFC.

This CHyDLYF(y=0,1) bit in HRTIMER_STxINTF register indicates the signal CHyOPRE state when the delayed IDLE was triggered by the selected external event (EXEV5/6 or EXEV7/8).

The following four figures show CH0OPRE wave in delayed IDLE with the following configuration:

- C0OPRE is in regular mode: DTEN = 0 in HRTIMER_STxCHOCTL register and BLNMEN = 0 in HRTIMER_STxCTL0 register
- Compare 0 event produces "set request".
- Compare 1 event produces "reset request".

Figure 25-34. ISO0 = 0 and CHOP = 0 in delayed IDLE

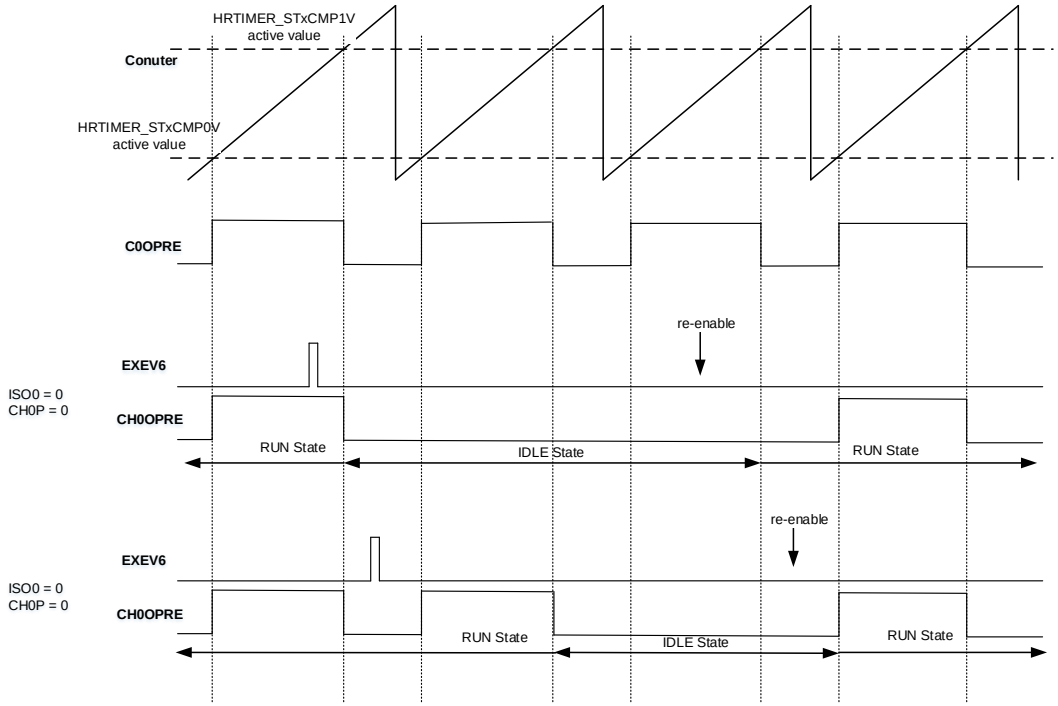


Figure 25-35. ISO0 = 1 and CHOP = 0 in delayed IDLE

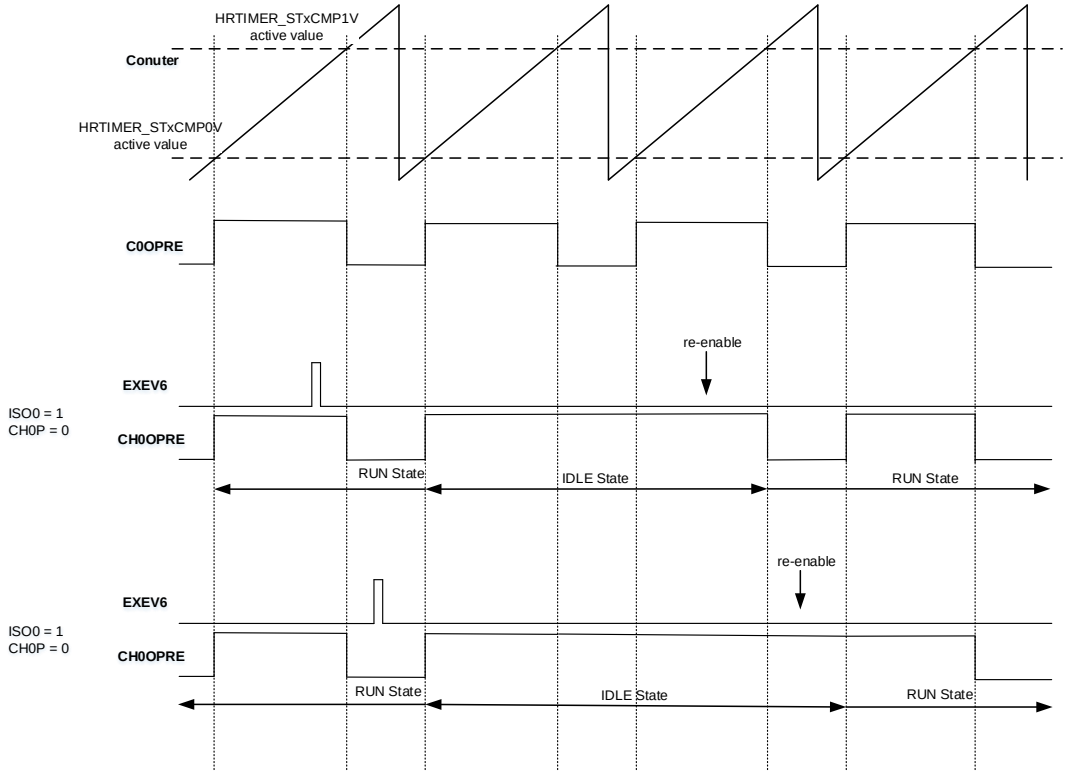


Figure 25-36. ISO0 = 0 and CHOP = 1 in delayed IDLE

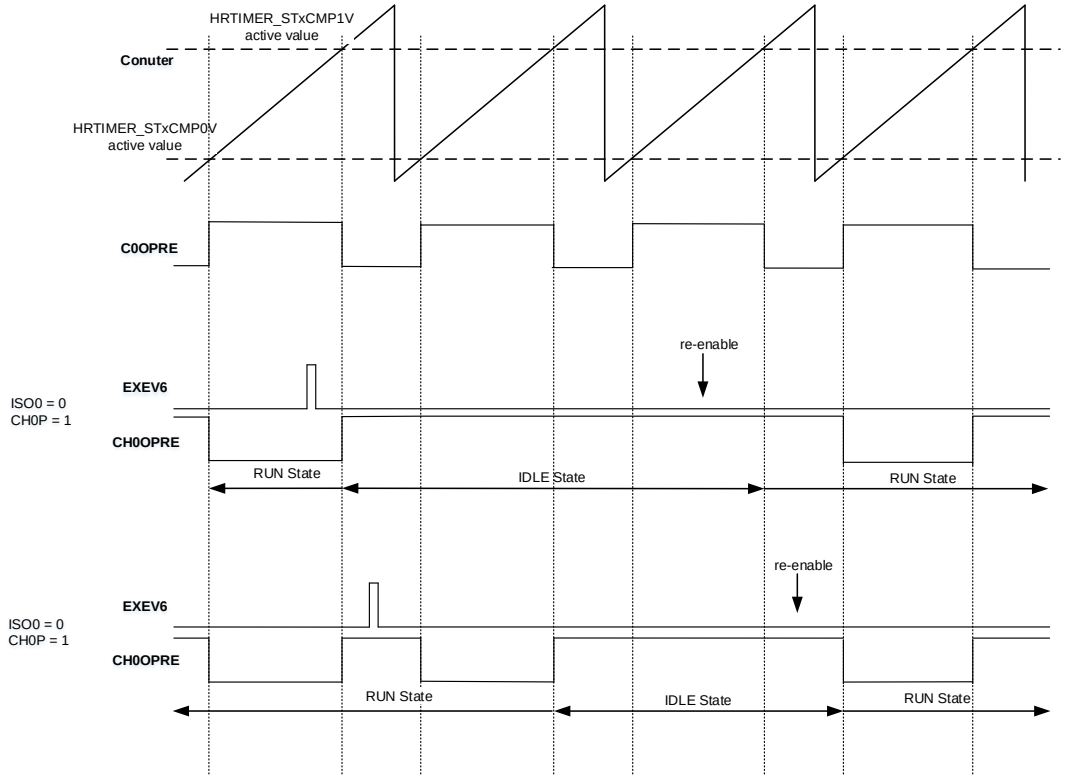
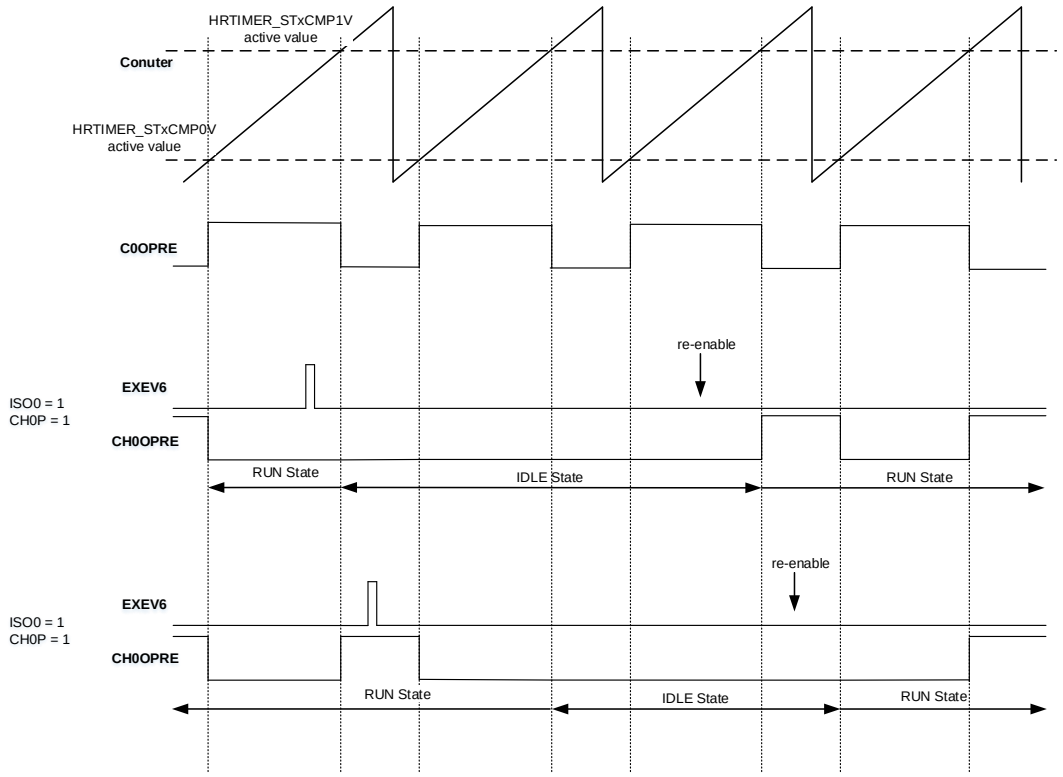


Figure 25-37. ISO0 = 1 and CHOP = 1 in delayed IDLE



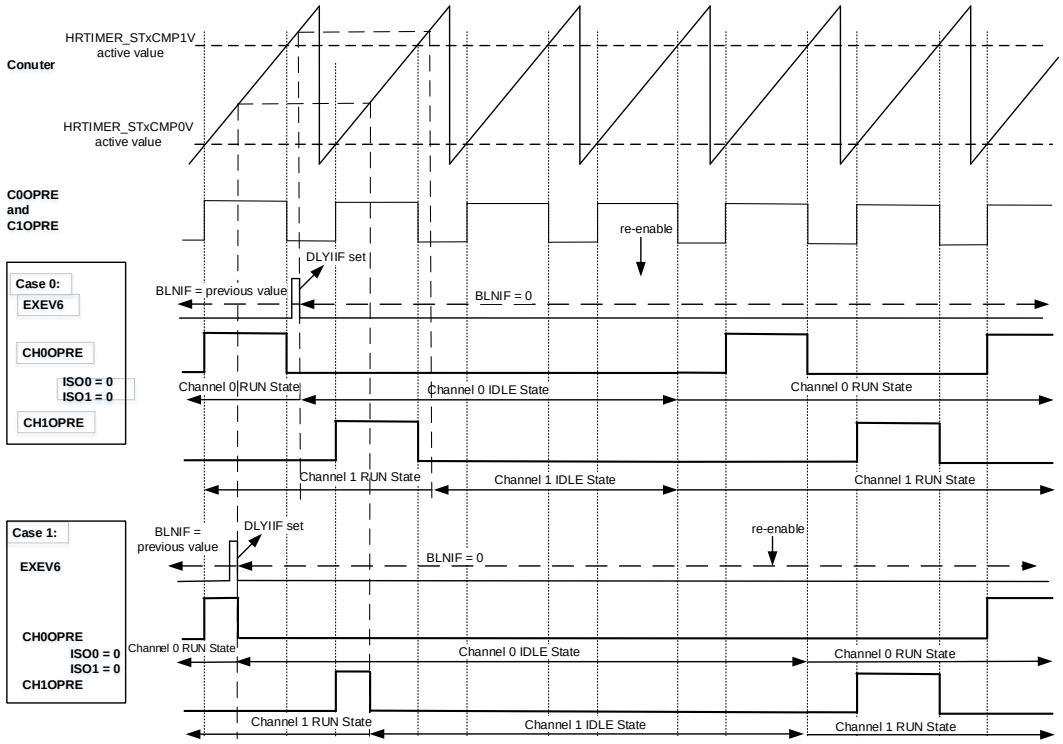
Balanced IDLE

Balanced IDLE is only available in balanced mode. Balanced IDLE is enabled by writing 3'bx011 in DLYISCH[2:0] bit-field in HRTIMER_STxCHOCTL register. Balanced IDLE is available with external event 5/6 (EXEV5/6) for Slave_TIMER0/1/2, and external event 7/8 (EXEV7/8) for Slave_TIMER3/4/5/6/7.

When the selected event arrives, the CHyOPRE (y=0,1) enters IDLE state and takes the level defined by ISOy bits in the HRTIMER_STxCHOCTL register. Meanwhile the DLYIIF in HRTIMER_STxINTF register is set to 1.The selected external event triggers a capture of the counter value into the compare 3 active register (this value is not user-accessible). The balanced mode is maintained for one additional period so that the opposite output CHzOPRE (z=0,1 and z ≠y) can repeat the shorten pulse on CHyOPRE: ***Figure 25-38. Balanced IDEL with ISO0 = 0 and ISO1 = 0*** shows CH0OPRE/ CH1OPRE wave in Slave_TIMER0 in balanced IDLE with the following configuration:

- C0OPRE is in balanced mode: DTEN = 0 in HRTIMER_STxCHOCTL register and BLNMEN = 1 in HRTIMER_STxCTL0 register
- Compare 0 event produces "set request".
- Compare 1 event produces "reset request".
- Channel 0 and channel 1 output balanced IDLE on external event 6: DLYISCH[2:0] = 3'b 111 for Slave_TIMER0

Figure 25-38. Balanced IDEL with ISO0 = 0 and ISO1 = 0



This BLNIF in HRTIMER_STxINTF register indicates which channel is outputting the signal when a balanced IDLE event entry occurred. For ***Figure 25-38. Balanced IDEL with ISO0 = 0 and ISO1 = 0***, external event 6 (EXEV6) arrives while channel 0 outputs the signal and channel 1 output inactive and BLNIF bit is reset to 0.

The IDLE mode is permanently maintained while the counter continues to run, until the output is re-enabled to exit delayed IDLE. It is re-enabled by “set request” or “reset request” following overwriting STxCH0EN and STxCH1EN bits to 1 simultaneously.

Balanced IDLE can be used together with the bunch mode under the following conditions:

- BMSTx bit must be reset (counter clock(HRTIMER_PSCCK) is maintained and the counter operates normally)
- No balanced IDLE are triggered while the outputs are in IDLE state controlled by bunch mode.

Balanced IDLE automatic recovery

The C0OPRE and C1OPRE can automatic recovery from Balanced idle when setting BALIAR,and the balanced idle automatic recovery mode can be used when DLYISCH[2:0] = 0x011 or 0x111 bit in HRTIMER_STxCHOCTL register. When an narrow pulse has been load to the output, The pulse of C0OPRE and C1OPRE are restored to normal output.

Balanced IDLE automatic recovery can be only used when the counter auto reload value CARL[15:0] in HRTIMER_STxCAR is more than 6 periods of $t_{HRTIMER_CK}$, for example 0xC0 when CNTCKDIV[2:0] = 0, 0x60 when CNTCKDIV[2:0] = 1.

IDLE controlled by bunch mode

In bunch mode, the IDLE state is controlled by bunch controller. Refer to ***Bunch mode*** for more information.

The balanced IDLE and delayed IDLE has a higher priority than the bunch mode: when the balanced IDLE or delayed IDLE has been triggered, the output is kept with IDLE state after exiting from bunch mode. On the contrary, if the balanced IDLE and delayed IDLE is exited while the bunch mode is active, the bunch mode will be resumed normally.

The bunch mode controller is able to take over the control of any two outputs CHyOPRE (y=0,1). The state of each output during IDLE state controlled by bunch mode is programmed using ISOy and BMCHyIEN(y=0,1) bits in the HRTIMER_STxCHOCTL register, as in ***Table 25-13. Output during IDEL state controlled by bunch mode***.

Table 25-13. Output during IDEL state controlled by bunch mode

ISOy	BMCHyIEN	CHyOPRE (y=0,1)
x	0	No action: the output is not affected by the bunch controller
0	1	Output inactive during IDLE state controlled by bunch mode
1	1	Output active during IDLE state controlled by bunch mode

Channel output stage

Each Slave_TIMERx unit controls a pair of outputs (STxCH0_O and STxCH1_O). The output stage has three operating states:

- Run state: STxCHy_O(y=0,1) can take the level of CHyOPRE(y=0,1).
- Idle state: STxCHy_O(y=0,1) can take the level defined by ISOy in HRTIMER_STxCHOCTL register.
- Fault state: STxCHy_O(y=0,1) can be permanently active, inactive or Hi-Z (defined in CHyFLTOS in HRTIMER_STxCHOCTL register). Refer to ***Fault input*** for more information.

The output stage status is indicated by STxCHyEN bit in HRTIMER_CHOUTEN register and STxCHyDISF bit in HRTIMER_CHOUTDISF register, as in ***Table 25-14. Output stage status programming (x=0..7, y=0,1).***

Table 25-14. Output stage status programming (x=0..7, y=0,1)

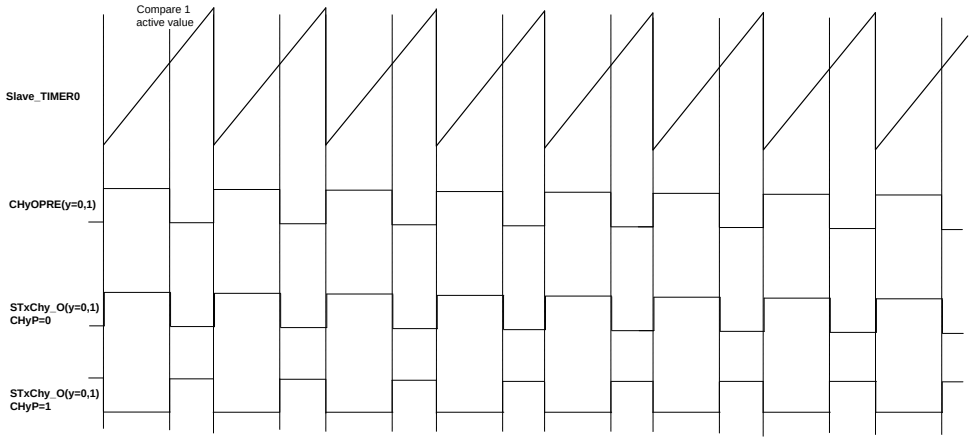
STxCHyEN	STxCHyDISF	output stage status
1	x	Run state
0	0	Idle state
0	1	Fault state

Note: “×” means “0/1”.

Writing 1 to STxCHyDIS in HRTIMER_CHOUTDIS register will disable output and make output stage enters the Idle state. The priority order of the three states is: Idle state > Fault state > Run state.

The output polarity is programmed using CHyP bits in HRTIMER_STxCHOCTL register. When CHyP = 0, the polarity is output active high. When CHyP = 1, the polarity is output active low. Refer to ***Figure 25-39. STxCHy_O wave with CHyP=0 or CHyP=1.***

Figure 25-39. STxCHy_O wave with CHyP=0 or CHyP=1



The output level in the Fault state is configured using CHyFLTOS[1:0] bits in HRTIMER_STxCHOCTL register, for each output, as follows:

- 2'b00: output never enters the Fault state and stays in Run or Idle state
- 2'b01: output is active level when in Fault state
- 2'b10: output is inactive level when in Fault state
- 2'b11: output is tri-stated when in Fault state.

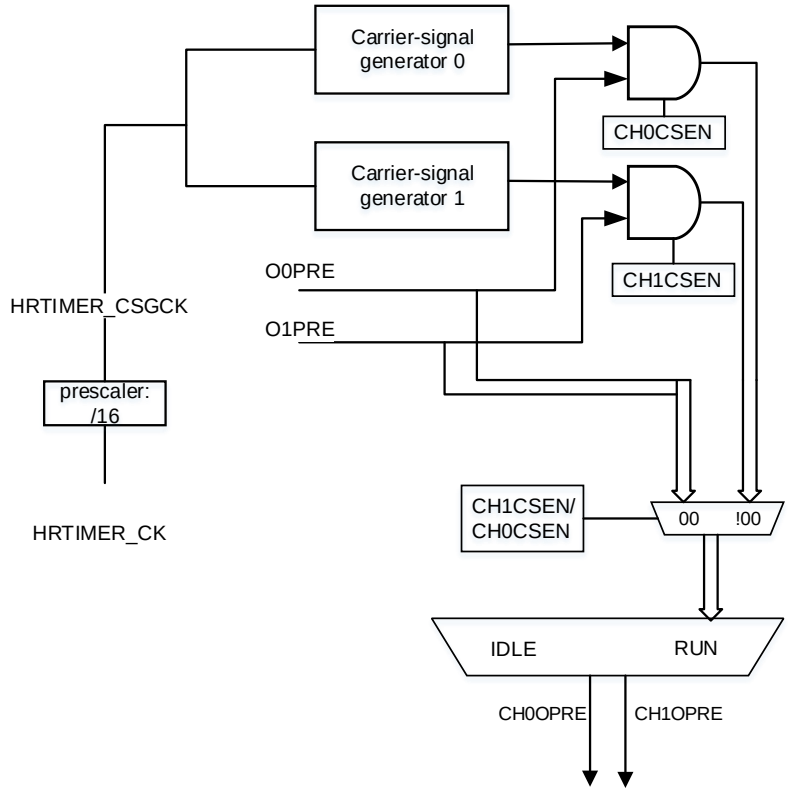
The level of the output in Idle state is configured using ISOy bit in HRTIMER_STxCHOCTL register, as follows:

- 2'b0: output is inactive level when in Idle state
- 2'b1: output is active level when in Idle state

Carrier-signal mode

A high-frequency carrier-signal can be added on top of the OyPRE(y=0,1), as follow ***Figure 25-40. Carrier-signal structure diagram***.

Figure 25-40. Carrier-signal structure diagram



In carrier-signal mode, it is possible to define a specific pulse width before the beginning of the carrier-signal. The frequency and duty cycle of the carrier-signal are configurable. Refer to ***Figure 25-41. HRTIMER output with carrier-signal mode enabled.***

Set CH0CSEN and CH1CSEN bits in the HRTIMER_STxCHOCTL register to 1 to enable carrier-signal mode on channel 0 and 1 respectively.

The pulse width of the first pulse is configured with CSFSTPW[3:0] bit-field in HRTIMER_STxCSCTL register, following the formula:

$$t_{CSFSTPW}=(CSFSTPW[3:0]+1)* t_{HRTIMER_CSGCK},\ \text{where}\ \ t_{HRTIMER_CSGCK}=16 *t_{HRTIMER_CK}$$

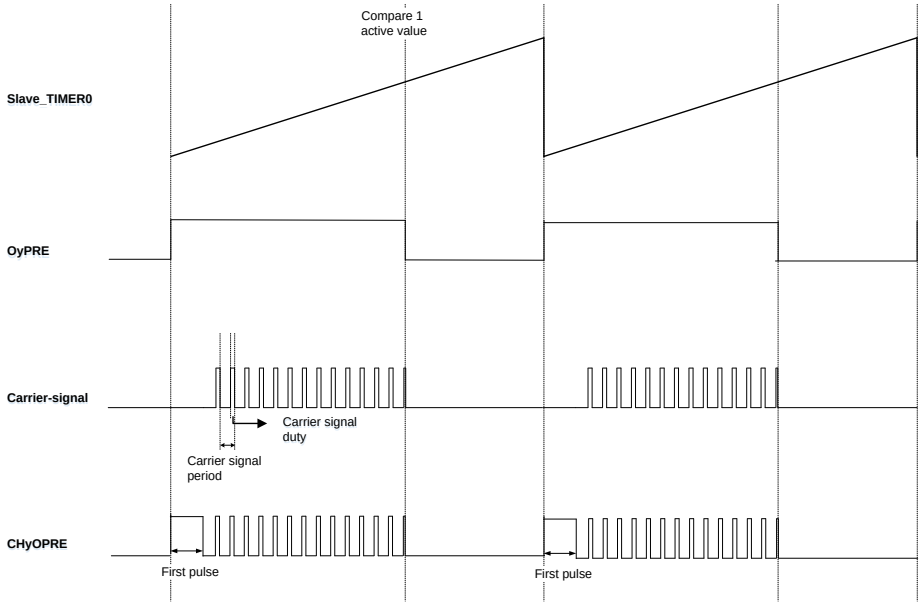
The frequency of carrier-signal is configured with CSPRD[3:0] bit-field in HRTIMER_STxCSCTL register, following the formula:

$$t_{CSPRD}=(CSPRD[3:0]+1)* t_{HRTIMER_CSGCK},\ \text{where}\ \ t_{HRTIMER_CSGCK}=16 *t_{HRTIMER_CK}$$

The duty cycle of carrier-signal is configured with CSDTY[2:0] bit-field in HRTIMER_STxCSCTL register by 12.5% per step.

In carrier-signal mode, the output of carrier-signal generator is combined with the OyPRE (logic ANDed). The carrier-signal signal is stopped as soon as the OyPRE(y=0,1) is inactive, even if the current carrier period is not completed. Refer to ***Figure 25-41. HRTIMER output with carrier-signal mode enabled***

Figure 25-41. HRTIMER output with carrier-signal mode enabled



Synchronization input start/reset counter

Synchronous input can generate a counter reset event when SYNIRST set 1 in HRTIMER_STxCTL0 register. Synchronous input can start counter when SYNISTRT set 1 in HRTIMER_STxCTL0 register. Refer to ***Synchronization input*** for more information.

A synchronization input request will set SYNIIF bit in HRTIMER_MTINTF register to 1, and a interrupt or a DMA request is issued if enabled(SYNIIE = 1 or SYNIDEN = 1 bits in HRTIMER_MTDMAINTEN register).The synchronization input interrupt flag can be cleared by writing 1 to SYNIIFC bit in HRTIMER_MTINTFC.

Update event and shadow registers

Some registers in Slave_TIMERx contain shadow registers. The shadow registers are disabled after MCU reset. If the SHWEN bit in HRTIMER_STxCTL0 register is cleared to 0, shadow registers are disabled. The values written into these registers are transferred into active register and take effect immediately.

If the SHWEN bit in HRTIMER_STxCTL0 register is set to 1, shadow registers are enabled and these registers listed in ***Table 25-15. Slave_TIMERx shadow registers and common registers*** **and update event** are preloaded. The values written into these registers are transferred into the shadow register and do not take effect immediately. Their content of the shadow is transferred into active register and take effect immediately on update event.

Note: Update event occurs only when SHWEN = 1.

Table 25-15. Slave_TIMERx shadow registers and common registers **and update event** lists the registers that contain shadow registers and corresponding update events.

Table 25-15. Slave_TIMERx shadow registers and common registers and update event

Registers that contain shadow registers	Shadow registers enable bit	Update event.
HRTIMER_STxDMAINTEN HRTIMER_STxCAR HRTIMER_STxCREP HRTIMER_STxCMP0V HRTIMER_STxCMP0CP HRTIMER_STxCMP1V HRTIMER_STxCMP2V HRTIMER_STxCMP3V HRTIMER_STxDTCTL HRTIMER_STxCH0SET HRTIMER_STxCH0RST HRTIMER_STxCH1SET HRTIMER_STxCH1RST HRTIMER_STxCNTRST HRTIMER_STxCNTRSTA DTFCFG[15:9] and DTRCFG[15:9] in HRTIMER_STxACTL	SHWEN bit in HRTIMER_STxCTL0 register	Software(STxSUP bit) Repetition event(UPREP = 1) Counter reset event or roll-over event(UPRST = 1) Update event from other timers(UPBSTy for Slave_TIMERy, UPBMT for Master_TIMER) DMA mode end event (UPSEL[1:0] = 4'b0001) Update event following a DMA mode end event (UPSEL[1:0] = 4'b0010) Update event generated on the rising edge of STxUPINy(y=0..2) Update event following the rising edge of STxUPINy(y=0..2)
HETIMER_ADCTRIGS0 HETIMER_ADCTRIGS0A HETIMER_ADCTRIGS1 HETIMER_ADCTRIGS1A HETIMER_ADCTRIGS2 HETIMER_ADCTRIGS2A HETIMER_ADCTRIGS3 HETIMER_ADCTRIGS3A	Master_TIMER or Slave_TIMERx(x = 0..7) update event,denpending on SHWEN bit in HRTIMER_MTCTL0 or HRTIMER_STxCTL0 register.	

The update enable inputs STxUPINy(y=0..2) are chip internal signal coming from the general-purpose timers and rising-edge sensitive. Refer to ***Table 25-16. STxUPINy(y=0..2) and chip internal signal***

Table 25-16. STxUPINy(y=0..2) and chip internal signal

Update enable input STxUPINy(y=0..2)	Chip internal signal
STxUPIN0	TIMER15_CH0_O
STxUPIN1	TIMER16_CH0_O
STxUPIN2	TIMER5_TRGO

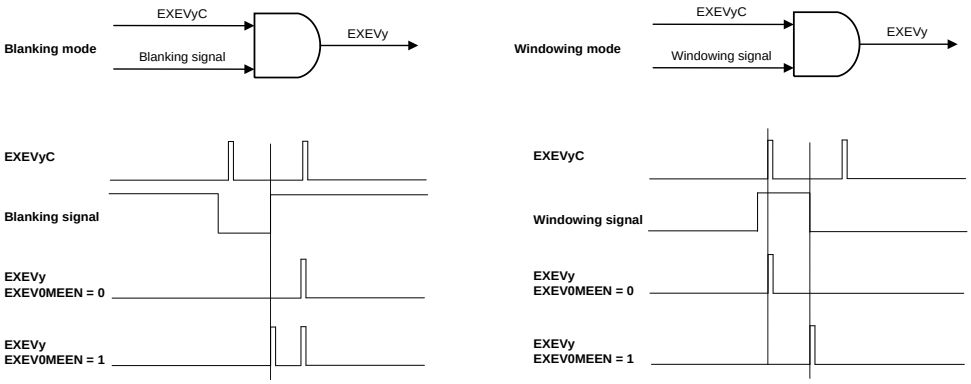
External event filter

The external event EXEVyC(y=0..9) can be filtered to act at a specified time. There are two filter mode:

- Blanking mode: external events that occur within a specified time are ignored.
- Windowing mode: external events that occur within a specified time are taken into account.

Refer to ***Figure 25-42. Blanking mode and windowing mode***

Figure 25-42. Blanking mode and windowing mode



Refer to ***External event*** about the external event EXEVyC(y=0..9) for more information.

Blanking mode

In blanking mode, the external event EXEVyC(y=0..9) that occurs within a specified time is ignored, otherwise it is taken into account. During specified time the blanking signal is low level. This mode is configured with EXEVyFM[4:0] bit-field values ranging from 5'b00001 to 5'b01100 and 5'b10000 to 5'b10001.

There are two types of blanking signal sources:

- The Slave_TIMERx itself: the specified time is the duration from the counter reset to the compare event. (EXEVyFM[4:0] = 5'b00001 to 5'b00100 for compare 0 to compare 3 event)
- STBLKSRCz(z=0..9) from other Slave_TIMER units (EXEVyFM[4:0] = 5'b00101 to 5'b01100) and 5'b10000 to 5'b1000.: the specified time can the duration from the selected Slave_TIMER counter reset to the compare event. It can also be CH1OPRE in the selected Slave_TIMER (in this case, events are ignored as long as the CH1OPRE is low level). Refer to ***Table 25-17. blanking signal remap in blanking mode***.

Table 25-17. blanking signal remap in blanking mode

		T0 ST0	T0 ST1	T0 ST2	T0 ST3	T0 ST4	T0 ST5	T0 ST6	T0 ST7
From ST0	CMP0	×	STBLK SRC0	×	STBLK SRC0	×	×	×	STBLK SRC0
	CMP1	×	×	STBLK SRC0	×	STBLK SRC0	×	×	×
	CMP3	×	STBLK SRC1	×	×	×	STBLK SRC0	×	×

		T0 ST0	T0 ST1	T0 ST2	T0 ST3	T0 ST4	T0 ST5	T0 ST6	T0 ST7
	CH1OPRE	×	STBLK SRC2	×	×	×	×	STBLK SRC0	×
From ST1	CMP0	STBLK SRC0	×	STBLK SRC1	×	STBLK SRC1	×	STBLK SRC1	×
	CMP1	×	×	×	STBLK SRC1	×	STBLK SRC1	×	STBLK SRC1
	CMP3	STBLK SRC1	×	STBLK SRC2	×	×	×	×	×
	CH1OPRE	STBLK SRC2	×	×	×	×	×	×	×
From ST2	CMP0	STBLK SRC3	STBLK SRC3	×	STBLK SRC2	STBLK SRC2	×	×	×
	CMP1	×	STBLK SRC4	×	STBLK SRC3	×	×	×	×
	CMP3	STBLK SRC4	×	×	×	×	STBLK SRC2	×	STBLK SRC2
	CH1OPRE	×	×	×	STBLK SRC4	×	×	STBLK SRC2	×
From ST3	CMP0	STBLK SRC6	×	STBLK SRC4	×	STBLK SRC5	×	×	×
	CMP1	×	STBLK SRC6	×	×	×	STBLK SRC3	STBLK SRC3	×
	CMP3	×	×	STBLK SRC5	×	STBLK SRC6	STBLK SRC4	×	STBLK SRC3
	CH1OPRE	×	×	STBLK SRC6	×	STBLK SRC7	×	×	×
From ST4	CMP0	×	STBLK SRC7	×	STBLK SRC5	×	STBLK SRC5	×	×
	CMP1	STBLK SRC7	×	×	×	×	×	×	STBLK SRC4
	CMP3	×	×	STBLK SRC7	STBLK SRC6	×	STBLK SRC6	STBLK SRC4	×
	CH1OPRE	×	×	×	×	×	STBLK SRC7	×	×
From ST5	CMP0	STBLK SRC5	×	STBLK SRC3	×	×	×	STBLK SRC5	STBLK SRC5
	CMP1	×	STBLK SRC5	×	×	×	×	×	×
	CMP3	×	×	×	STBLK SRC7	STBLK SRC3	×	STBLK SRC6	STBLK SRC6
	CH1OPRE	×	×	×	×	STBLK	×	STBLK	×

		T0 ST0	T0 ST1	T0 ST2	T0 ST3	T0 ST4	T0 ST5	T0 ST6	T0 ST7
						SRC4		SRC7	
From ST6	CMP0	×	STBLK SRC8	×	STBLK SRC8	×	STBLK SRC8	×	STBLK SRC7
	CMP1	STBLK SRC8	×	STBLK SRC8	×	STBLK SRC8	×	×	STBLK SRC8
	CMP3	×	×	×	×	×	×	×	×
	CH1OPRE	×	×	×	×	×	×	×	STBLK SRC9
From ST7	CMP0	STBLK SRC9	×	STBLK SRC9	×	STBLK SRC9	×	STBLK SRC8	×
	CMP1	×	STBLK SRC9	×	STBLK SRC9	×	STBLK SRC9	STBLK SRC9	×
	CMP3	×	×	×	×	×	×	×	×
	CH1OPRE	×	×	×	×	×	×	×	×

When EXEVyMEEN is set to 1, external event memorized is enabled. The external event is not taken into account immediately. It is memorized and generated as soon as the specified time is completed.

Windowing mode

In windowing mode, the external event EXEVyC(y=0..9) that occurs within a specified time are taken into account, otherwise it is ignored. During specified time the windowing signal is high level. This mode is configured with EXEVyFM[4:0] bit-field values ranging from 5'b01101 to 5'b01111.

If no EXEVyC(y=0..9) occurs within a specified time, the timeout event will be generated at the end of the specified time.

There are two types of windowing signal sources:

- The Slave_TIMERx itself: the specified time is the duration from the counter reset to the compare event. (EXEVyFM[4:0] = 5'b01101 and 5'b01110 ,respectively compare 1 and compare 2).
- STWDSRC from other Slave_TIMER units (EXEVyFM[5:0] = 5'b01111): the specified time is the duration from the selected Slave_TIMER counter reset to compare event. Refer to ***Table 25-18. Filtering signals mapping in windowing mode***.

Table 25-18. Filtering signals mapping in windowing mode

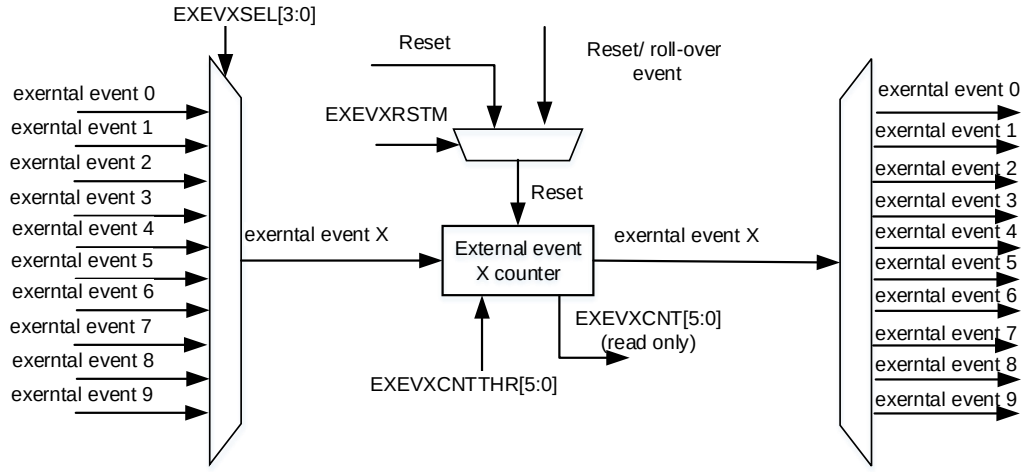
To / From	T0 ST0	T0 ST1	T0 ST2	T0 ST3	T0 ST4	T0 ST5	T0 ST6	T0 ST7
STWDSRC	Slave_TIMER1 compare 1	Slave_TIMER0 compare 1	Slave_TIMER3 compare 1	Slave_TIMER2 compare 1	Slave_TIMER5 compare 1	Slave_TIMER4 compare 1	Slave_TIMER7 compare 1	Slave_TIMER6 compare 1

When EXEVyMEEN is set to 1, external event memorized is enabled. The external event is not taken into account immediately. It is memorized and generated as soon as the specified time is completed.

External event counter

The 10 external events can be filtered by external event X counter module. As shown in *Figure 25-43. External event X counter*.

Figure 25-43. External event X counter



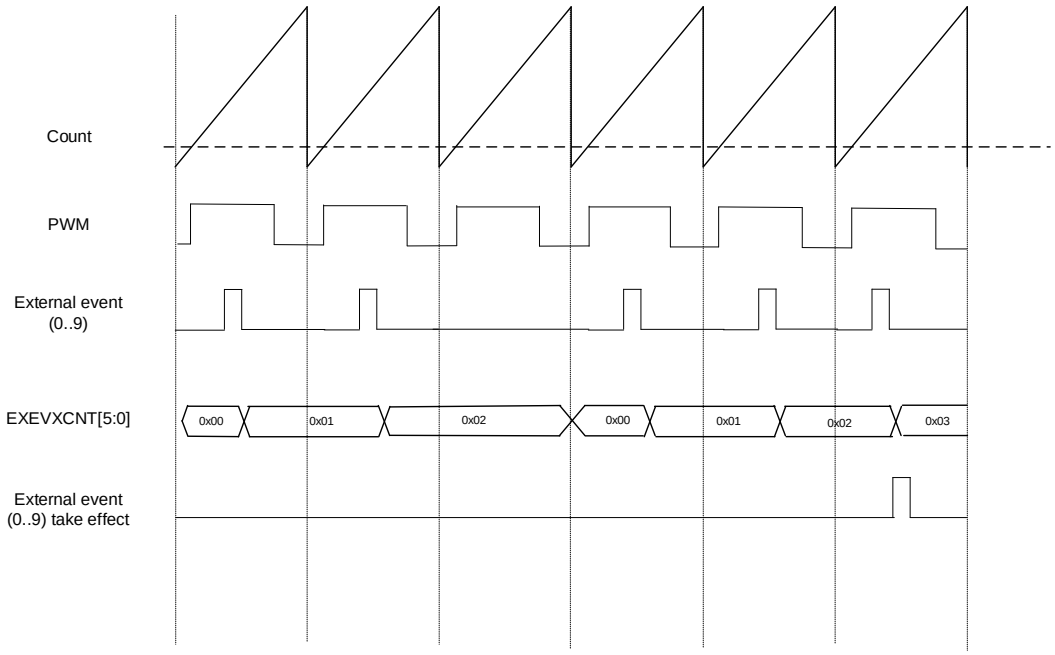
External event X counter is enabled by setting EXEVXCEN bit in HRTIMER_STxEXEVFCFG2 register, external event is only valid when the value of EXEVXCNT[5:0] is grater or equal to EXEVXCNTTHR[5:0] + 1.

When EXEVXRSTM bit is reset, the EXEVXCNT[5:0] bits is reset when reset and update event occurs, the external event is considered active only if it occurs multiple times within one PWM period. When EXEVXRSTM bit is set, the EXEVXCNT[5:0] bits is accumulation when an external event is generated within each PWM period, and the EXEVXCNT[5:0] bits is reset when no external event appears in the last PWM period.

The EXEVXCEN bit should be set after writing EXEVXCNTTHR[5:0] bits, and the EXEVXCNTTHR[5:0] bits can be changed when External event X counter is enabled. And the new value of EXEVXCNTTHR[5:0] will be take effect in next period. The EXEVSEL[3:0] can not be changed when external event X counter is enabled.

Figure 25-44. External event X counter when EXEVXCEN bit is 1 and

EXEVXCNTTHR[5:0] = 2



Fast external events mode

The processing time of external events can be adjusted dynamically according to the actual requirements. When EXEVxFAST bit is reset in HRTIMER_EXEVCFG0 register, the external events need to be resampled before takeing effect, thus some latency is added, and the highresolution pulse can be generate. When EXEVxFAST bit is set in HRTIMER_EXEVCFG0 register, the fast external events mode is enable. In this mode, the latency is minimized.

The fast external events mode is only valid for level events, not edge events, thus the EXEVxEG[1:0] bit in HRTIMER_EXEVCFG0 register must be 0x00. External events filtering can be used in this mode, and the external event memory mode must be disabled.

The same event can't be configured both in HRTIMER_STxCHySET and HRTIMER_STxCHyRST registers, and reset events have a higher priority when the reset event and the set event occur at the same time.

In fast external events mode, When an external event occurs and takes effect, other external events that occur within 11 $t_{HRTIMER_CK}$ are invalid.

25.4.3. DLL calibrate

The DLL can produce and calibrate a high resolution clock HRTIMER_HPCK ($f_{HRTIMER_HPCK}$ = 32 *$f_{HRTIMER_CK}$). DLL can calibrate the high resolution clock HRTIMER_HPCK either once or periodically.

When CLBPEREN bit in HRTIMER_DLLCCTL register is set to 1, it enables the periodic DLL calibration and the bit-field CLBPER[1:0] decides the calibration period. The DLL will

periodically calibrate the clock during the entire HRTIMER running.

When CLBPEREN bit in HRTIMER_DLLCCTL register is cleared to 0, DLL calibrates the high resolution clock HRTIMER_HPCK only once by writing CLBSTRT to 1.

25.4.4. Bunch mode

The bunch mode controller allows to have the CHyOPRE (y=0,1) alternatively in IDLE and RUN state by hardware. This mode is enabled with BMEN bit in the HRTIMER_BMCTL register and usually used in light load situations.

The bunch mode controller consists of:

- A counter called BM-counter.
- A compare register: HRTIMER_BMCMPV, to define IDLE duration.
- A period register: HRTIMER_BMCAR, to define the sum of the IDLE and RUN duration.

BM-counter counting mode

BM-counter can operate in continuous or single pulse mode.

When BMCTN = 1, BM-counter operate in continuous mode. The BM-counter counts up continuously from 0 to the counter-reload value(defined in HRTIMER_BMCAR register). When counts up to the counter-reload value (HRTIMER_BMCAR), the counter restarts from 0. The bunch mode operation is maintained until BMOPTF bit in HRTIMER_BMCTL is reset to terminate it.

When BMCTN = 0, BM-counter operate in single pulse mode. The BM-counter counts up continuously from 0 to the counter-reload value (HRTIMER_BMCAR). When counts up to the counter-reload value (HRTIMER_BMCAR), the BM-counter is stopped.

When counts up to the counter-reload value (HRTIMER_BMCAR), BMPERIF in HRTIMER_INTF register is set to 1 and bunch mode controller generates a bunch mode period interrupt request if BMPERIE=1 in HRTIMER_INTEN register. The BMPERIF bit can be cleared by writing 1 to BMPERIFC bit in HRTIMER_INTC register.

Bunch mode timing

The BM-counter can be clocked by several sources, selected with BMCLKS[3:0] bits in the HRTIMER_BMCTL register. When the rising edge of the selected clock source signal arrives, BM-counter increments by 1.

When BMCLKS[3:0]=4'b1010, the clock source of BM-counter is the $f_{HRTIMER_CK}$ prescaled by a factor defined with BMPSC[3:0] bit-field in HRTIMER_BMCTL register.

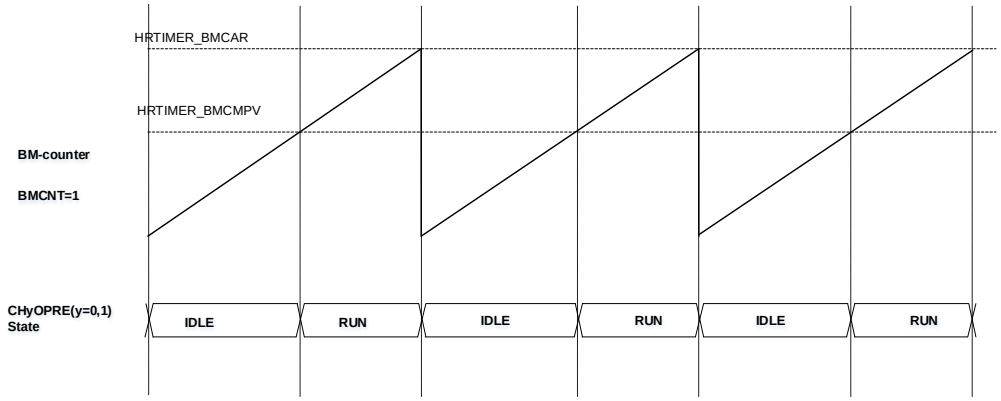
When BMCLKS[3:0]= 4'b0110 to 1001, the clock source of BM-counter is chip internal signal : BMCLKy(y=0..3). Refer to ***Table 25-19. Chip internal signal in bunch mode.***

Table 25-19. Chip internal signal in bunch mode

BMCLKy(y=0..3)	Chip internal signal
BMCLK0	TIMER15_CH0_O
BMCLK1	TIMER16_CH0_O
BMCLK2	TIMER6_TRGO
BMCLK3	Reserved

The duration of the IDLE is defined with HRTIMER_BMCMPV register, and HRTIMER_BMCAR register defines the bunch mode period which is the sum of the IDLE and RUN duration.Refer to ***Figure 25-45. Bunch mode timing chart***.

Figure 25-45. Bunch mode timing chart



When BMSE is set， the HRTIMER_BMCMPV and HRTIMER_BMCAR registers are preloaded and the transfer from preload to active register happens:

- when the bunch mode is enabled (BMEN = 1)
- at the end of the bunch mode period

Notice that a write into the HRTIMER_BMCAR register disables the update temporarily, until the HRTIMER_BMCMPV compare register is written.

Bunch mode entry

There are 44 events defined in HRTIMER_BMSTRG register and HRTIMER_BMSTRGA register can be used to trigger the bunch mode operation. These trigger events can be selected simultaneously and are logic ORed. During BM-counter counting process, these trigger events are ignored. These trigger events are divided into seven categories:

1. Events from Master_TIMER: repetition event, reset/roll-over event, compare 0 to 3 event
2. Events from Slave_TIMERx: repetition event, reset/roll-over event, compare 0 and 1 event
3. External event: EXEV6 and EXEV7
4. Slave_TIMER0 period event following EXEV6
5. Slave_TIMER3 period event following EXEV7
6. Chip internal signal: TIMER6_TRGO
7. Software: Writing 1 to the SWTRG bit in HRTIMER_BMSTRG register.

When the trigger event occurs, there are two ways to enter bunch mode: regular entry and delayed entry.

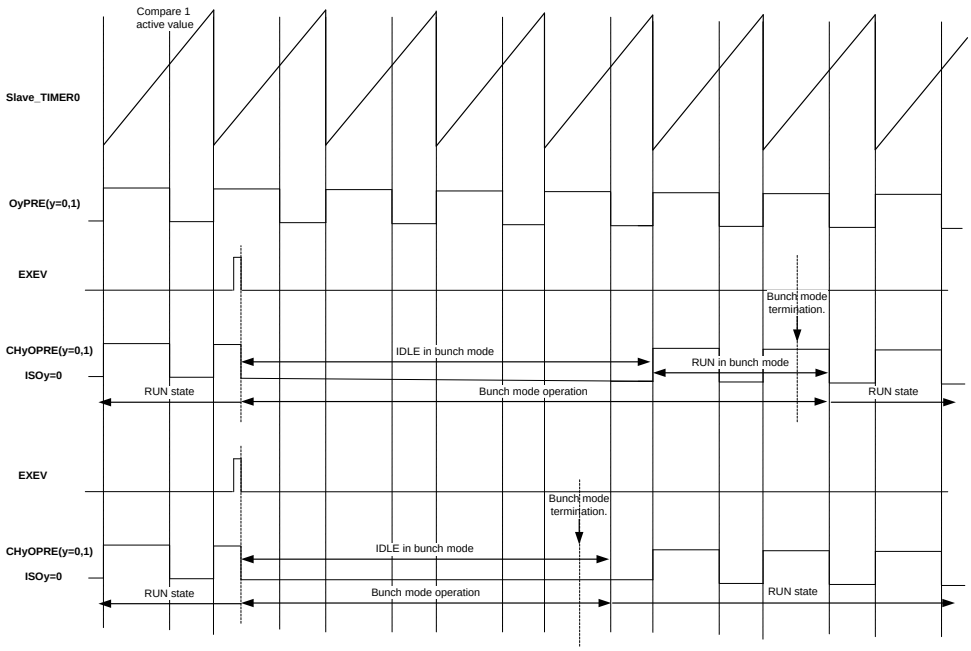
Regular entry

When BMCHyDTI(y=0,1) bit in HRTIMER_STxCHOCTL register is 0, the bunch mode entry is regular. The output will enter the bunch mode and take their idle level (as per ISO0 and ISO1 setting) on the first BM-counter counting clock after the selected event occurs.

Figure 25-46. Regular entry for bunch mode shows CHyOPRE wave in Slave_TIMER0 with the following configuration:

- CyOPRE is in regular mode: DTEN = 0 in HRTIMER_ST0CHOCTL register and BLNMEN = 0 in HRTIMER_ST0CTL0 register
- Period event produces “set request”.
- Compare 1 event produces “reset request”.
- Clock source of BM-counter is roll-over event in Slave_TIMER0: BMCLKS[3:0]=4’b0001

Figure 25-46. Regular entry for bunch mode



Delayed entry

When BMCHyDTI(y=0,1) bit in HRTIMER_STxCHOCTL register is 1, the bunch mode entry is delayed. CHyOPRE forces to a dead-time insertion before entering bunch mode.

Each CHyOPRE has its own dead-time value:

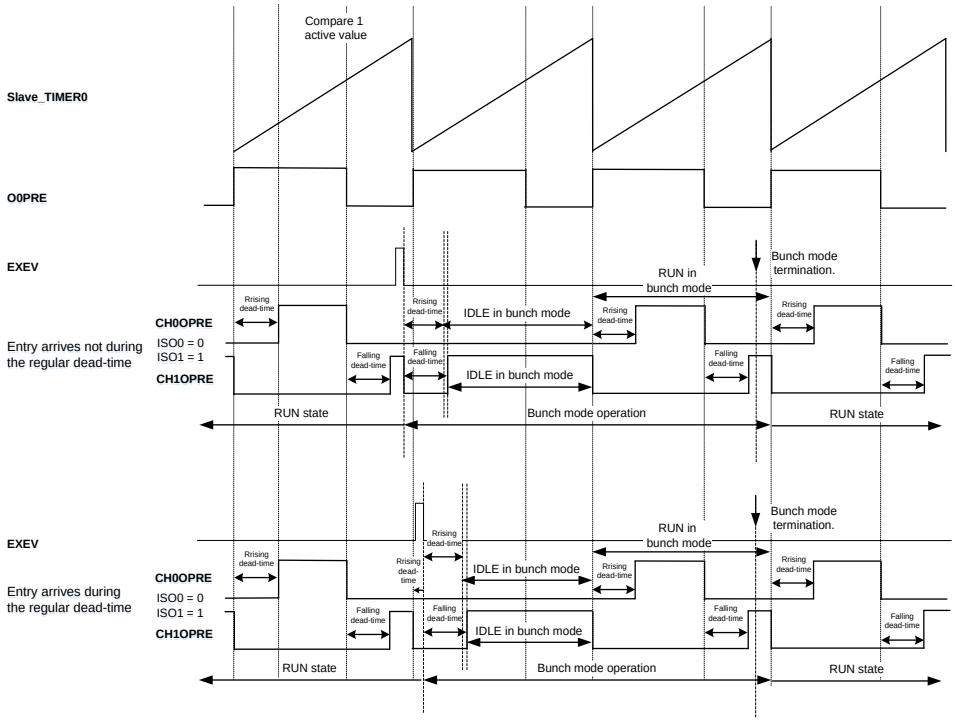
- DTRCFG[15:0] on CH0OPRE when BMCH0DTI=1
- DTFCFG[15:0] on CH1OPRE when BMCH1DTI=1

Delayed entry applies to the situation that one of the CHyOPRE has an active IDLE level (ISOy = 1) and dead-times are positive (DTRS/DTFS set to 0).

When the bunch mode entry arrives during the regular dead-time, it is aborted and a new dead-time is re-started corresponding to the inactive period. Refer to ***Figure 25-47. Delayed entry for bunch mode*** with the following configuration:

- CyOPRE is in dead-time mode: DTEN = 1 in HRTIMER_ST0CHOCTL register and BLNMEN = 0 in HRTIMER_ST0CTL0 register
- Period event produces “set request”.
- Compare 1 event produces “reset request”.
- Clock source of BM-counter is roll-over event in Slave_TIMER0: BMCLKS[3:0]=4’b0001

Figure 25-47. Delayed entry for bunch mode



Bunch mode exit

In continuous mode, the bunch mode exit is forced by software. The bunch mode exit by “set request” or “reset request” following overwriting BMOPTF or BMEN bit to 0. Refer to ***Figure 25-46. Regular entry for bunch mode*** and ***Figure 25-47. Delayed entry for bunch mode***.

In single pulse mode, the bunch mode exit once the IDLE period is elapsed.

Counter clock in bunch mode

The counters of the Master_TIMER and Slave_TIMERx(x=0..7) units can be stopped and reset during the bunch operation (RUN+IDLE).The bits BMMT and BMSTx(x=0..7) in HRTIMER_BMCTL register is used for this purpose:

- BMMT or BMSTx(x=0..7) equal to 0: Master_TIMER or Slave_TIMERx(x=0..7) counter clock(HRTIMER_PSCCK) is maintained and the counter operates normally

- BMMT or BMSTx(x=0..7) equal to 1: Master_TIMER or Slave_TIMERx(x=0..7) counter clock(HRTIMER_PSCCK) is stopped and the counter is reset.

Use HRTIMER_STxCMP0CP register to emulate bunch mode

The HRTIMER_STxCMP0CP register can be used to produce a waveform similar to that controlled by bunch mode. To do this, the following configuration is required:

- Compare 0 event is used to produce “reset request”.
- Period event is used to produce “set request”.
- Write two 32-bit data to the HRTIMER_STxCMP0CP register consecutively using DMA (upon repetition event), as below:

HRTIMER_STxCMP0CP = {CREP[7:0] = (RUN number of periods - 1); CMP0VAL[15:0] = duty cycle}

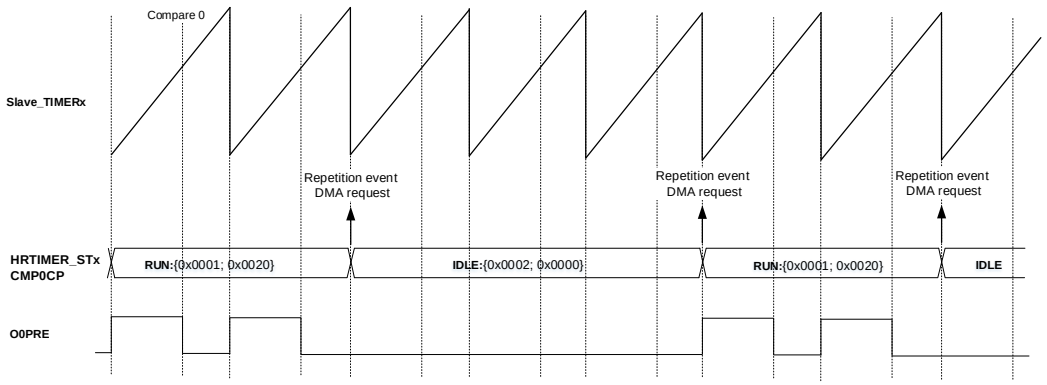
HRTIMER_STxCMP0CP = {CREP[7:0] = (IDLE number of periods - 1); CMP0VAL[15:0] = 0}

For example, to generate a PWM wave with 2 periods active every 5 PWM periods, the following values can be used:

- RUN: HRTIMER_STxCMP0CP = {0x0001; 0x0020}
- IDLE: HRTIMER_STxCMP0CP = {0x0002; 0x0000}

Its behavior is shown ***Figure 25-48. Emulate bunch mode example***.

Figure 25-48. Emulate bunch mode example



25.4.5. Synchronization input/output

The synchronization circuitry is controlled inside the Master_TIMER:

- Synchronization output: the HRTIMER can generate a synchronization signal as a master.
- Synchronization input: HRTIMER can also wait for a trigger to be synchronized as a slaver.

Synchronization output

HRTIMER can be configured to synchronize external resources and act as a master unit.

The bit-field SYNOSRC[1:0] in HRTIMER_MTCTL0 register can be configured to selecte the source to be sent to the synchronization output. There are four sources available:

- 2'b00: Master_TIMER start event. There are three situations in which the generated start event can be used as synchronous output: when MTCEN bit is set to 1, when the timer is re-started after having reached the period value in single pulse mode, a reset which occurs during the counting with CTNM or CNTRSTM bits set to 1.
- 2'b01: Master_TIMER compare 0 event
- 2'b10: Slave_TIMER0 reset and start event. It is similar to Master_TIMER start event, except for the following: counter roll-over in continuous mode, discarded reset request in single pulse mode with CNTRSTM=0.
- 2'b11: Slave_TIMER0 compare 0 event

The bit-field SYNOPLS[1:0] in HRTIMER_MTCTL0 register specifies the polarity of the synchronization output signal:

- 2'b00: Pulse generated disable. No pulse on the synchronization output pad HRTIMER_SCOUT.
- 2'b01: Reserved.
- 2'b10: Positive pulse generated on the synchronization output pad HRTIMER_SCOUT. The length of the positive pulse is 16 $t_{HRTIMER_CK}$ cycles.
- 2'b11: Negative pulse generated on the synchronization output pad HRTIMER_SCOUT. The length of the negative pulse is 16 $t_{HRTIMER_CK}$ cycles.

Synchronization input

HRTIMER can wait for a trigger to be synchronized as a slaver. The bit-field SYNISRC[1:0] in HRTIMER_MTCTL0 register can be configured to select the synchronization input source. There are four trigger sources available:

- 2'b00: Synchronization input disable.
- 2'b01: Reserved.
- 2'b10: Chip internal signal. TIMER0_TRGO in the advanced timer TIMER0
- 2'b11: Chip external pin. A positive pulse on the chip external pin (HRTIMER_SCIN) is effective (rising-edge sensitive).

The Master_TIMER behavior is defined with the bits SYNISTRT and SYNIRST in HRTIMER_MTCTL0 register. The Slave_TIMERx behavior is defined with the bits SYNISTRT and SYNIRST in HRTIMER_STxCTL0 register.

When SYNISTRT is set to 1, the synchronization input signal starts the timer's counter provided that the counter must first enable by setting STxCEN or MTCEN bit to 1. In continuous mode, the counter will not start even if STxCEN or MTCEN bit is set to 1, but only after the synchronous input signal arrives.

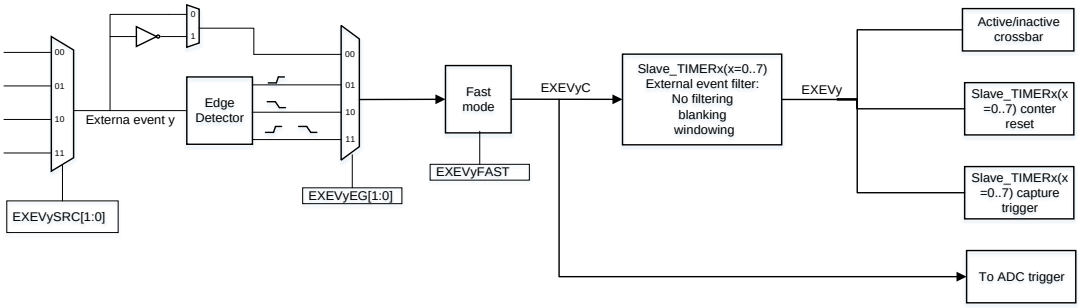
When SYNIRST is set to 1, the synchronization input signal resets the counter and decrements the repetition counter as any other reset event.

25.4.6. External event

There are 10 external events that can be simultaneously used on 8 Slave_TIMER. The external event y(y=0..4) are configured using the HRTIMER_EXEVCFG0 register , and the external event y(y=5..9) are configured using the HRTIMER_EXEVCFG1 and HRTIMER_EXEVDFCTL registers.

The process by which external event y(y=0..4) are processed is shown in ***Figure 25-49. Extern event y(y=0..4) processed diagram***.

Figure 25-49. Extern event y(y=0..4) processed diagram

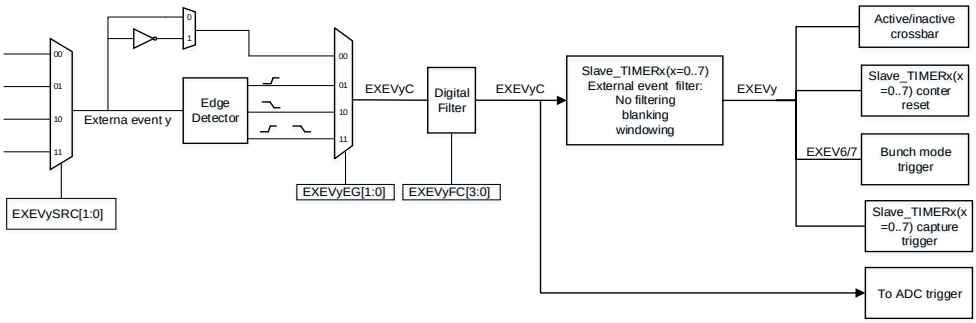


The configuration of external event y(y=0..4) is as follows:

- Up to 4 sources: configured with the EXEVySRC[1:0] bit-field.
- The sensitivity selection: configured with EXEVyEG[1:0] bit-field. It can be either level sensitive or edge sensitive (rising, falling or both).
- The polarity selection: configured with EXEVyP bit in case of a level sensitivity (EXEVyEG[1:0]=2'b00).
- External event is acting asynchronously on outputs when EXEVyFAST bit is 1.

The process by which external event y(y=5..9) are processed is shown in ***Figure 25-50. Extern event y(y=5..9) processed diagram***.

Figure 25-50. Extern event y(y=5..9) processed diagram



The configuration of external event y(y=5..9) is as follows:

- Up to 4 sources: configured with the EXEVySRC[1:0] bit-field.

- The sensitivity selection: configured with EXEVyEG[1:0] bit-field. It can be either level sensitive or edge sensitive (rising, falling or both).
- The polarity selection: configured with EXEV0P bit in case of a level sensitivity (EXEVyEG[1:0]=2'b00).
- Digital filters configuration: configured with EXEVyFC[3:0] bit-field in the HRTIMER_EXEVDFCTL register.

The digital filters sampling clock $f_{HRTIMER_EXEVFCK}$ is defined with EXEVFDIV[2:0] bit-field in HRTIMER_EXEVDFCTL register.

These external events sources EXEVySRCz(y=0..9,z=0..4) can come from comparators, digital input pins, ADC's analog watchdogs and TIMER_TRGO. Refer to ***Table 25-20. External events mapping.***

Table 25-20. External events mapping

External event	EXEVySRC0	EXEVySRC1	EXEVySRC2	EXEVySRC3
External event 0	PC12	Comparator 1	TIMER0_TRGO	ADC0_WD0_OUT
External event 1	PC11	Comparator 3	TIMER1_TRGO	ADC0_WD1_OUT
External event 2	PB7	Comparator 5	TIMER2_TRGO	ADC0_WD2_OUT
External event 3	PB6	Comparator 0	Comparator 4	ADC1_WD0_OUT
External event 4	PB9	Comparator 2	Comparator 6	ADC1_WD1_OUT
External event 5	PB5	Comparator 1	Comparator 0	ADC1_WD2_OUT
External event 6	PB4	Comparator 3	TIMER6_TRGO	ADC2_WD0_OUT
External event 7	PB8	Comparator 5	Comparator 2	ADC3_WD0_OUT
External event 8	PB3	Comparator 4	TIMER14_TRGO	Comparator 3
External event 9	PC5/PC6	Comparator 6	TIMER5_TRGO	ADC3_WD0_OUT

Note: "×" means not available.

The extern event y(y=0..9) can be used and also be filtered to have an action limited in a specified time. ***Refer to External event filter.***

25.4.7. Fault input

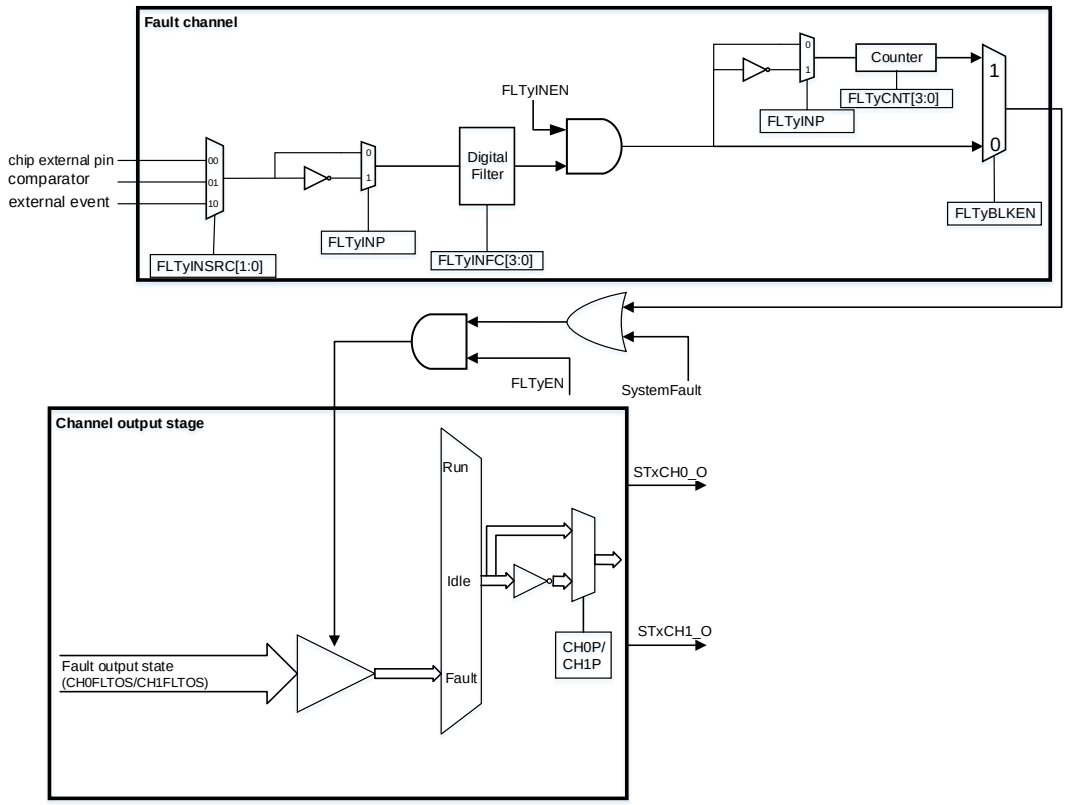
HRTIMER has a fault protection mechanism which is available for each Slave_TIMERx. Refer to ***Figure 25-51. Fault input diagram.***

When a fault event occurs, the output (STxCHy_O, x=0..7,y=0,1)is at a predefined level.

The predefined level is configured by bit-field CHyFLTOS[1:0] in HRTIMER_STxCHOCTL register. The protection mechanism can handle three types of fault sources:

- Fault channel: fault event from digital input pin or comparator.
- System fault: signals coming from inside the MCU, for example the Cortex®-M33-lockup signal.
- fault event: fault event from extern event y(y=0..9).

Figure 25-51. Fault input diagram



Fault input is enabled using FLTyEN bit in HRTIMER_STxFLTCTL register. A write-once FLTENPROT bit in the HRTIMER_STxFLTCTL register allows to protect FLTyEN bits for safety purpose. When FLTENPROT bit is set to 1, FLTyEN bits are writing protected (read-only).

Fault channel

The fault channel is fully configurable using HRTIMER_FLTINCFG0 and HRTIMER_FLTINCFG1 registers.

FLTyINSRC (y=0..7) bit selects the source of the fault channel, that can be from either a digital input pin or comparator output or external event. Each Slave_TIMERx has 8 fault channels (Fault channel 0 - 7) available. Refer to ***Table 25-21. Fault channel mapping***

Table 25-21. Fault channel mapping

Fault channel	FLTyINSRC = 00 (chip external pin)	FLTyINSRC = 01 (internal signal)	FLTyINSRC = 10 (external event)
Fault channel 0	PA12	Comparator 1	external event 0
Fault channel 1	PA15	Comparator 3	external event 1
Fault channel 2	PB10	Comparator 5	external event 2
Fault channel 3	PB11	Comparator 0	external event 3
Fault channel 4	PB0/PC7	Comparator 2	external event 4
Fault channel 5	PC10	Comparator 4	external event 5
Fault channel 6	PC3	Comparator 6	external event 6
Fault channel 7	PC4	Comparator 7	external event 7

Note: "×" means not available.

The polarity of the signal can be configured by the FLTyINP polarity bit in HRTIMER_FLTINCFG0 and HRTIMER_FLTINCFG1 registers. If FLTyINP = 0, the signal is active at low level; if FLTyINP = 1, it is active when high.

The digital filters of the signal after the polarity setting can be configured by the FLTyINFC[3:0] bit-field in HRTIMER_FLTINCFG0 and HRTIMER_FLTINCFG1 registers. The digital filters sampling clock $f_{HRTIMER_FLTFCK}$ is defined with FLTFDIV[2:0] bit-field in HRTIMER_FLTINCFG1 register.

The fault channel y(y=0..7) can be enabled using bits FLTyINEN in HRTIMER_FLTINCFG0 and HRTIMER_FLTINCFG1 registers, and they can be selected simultaneously.

A write-once FLTyINPROT bit in HRTIMER_FLTINCFG0 and HRTIMER_FLTINCFG1 registers allows to protect FLT0INEN, FLT0INP, FLT0INSRC and FLT0INFC[3:0] bits for safety purpose. When FLTyINPROT bit is set to 1, these bits are writing protected (read-only).

Fault blank

In fault blank mode, the fault channel that occurs within a specified time is ignored, Refer to ***Table 25-22. Fault channel blank.***

Table 25-22. Fault channel blank

Fault channel	FLTxBLKS = 0,reset and compare windows		FLTxBLKS = 1, compare and compare windows	
	Blank start	Blank stop	Blank start	Blank stop
Fault channel 0	Slave_TIMER0 reset / update	Slave_TIMER0 Comparator 2	Slave_TIMER0 Comparator 3	Slave_TIMER0 Comparator 2
Fault channel 1	Slave_TIMER1 reset / update	Slave_TIMER1 Comparator 2	Slave_TIMER1 Comparator 3	Slave_TIMER1 Comparator 2
Fault channel 2	Slave_TIMER2 reset / update	Slave_TIMER2 Comparator 2	Slave_TIMER2 Comparator 3	Slave_TIMER2 Comparator 2
Fault channel 3	Slave_TIMER3	Slave_TIMER3	Slave_TIMER3	Slave_TIMER3

Fault channel	FLTxBLKS = 0,reset and compare windows		FLTxBLKS = 1, compare and compare windows	
	Blank start	Blank stop	Blank start	Blank stop
	reset / update	Comparator 2	Comparator 3	Comparator 2
Fault channel 4	Slave_TIMER4 reset / update	Slave_TIMER4 Comparator 2	Slave_TIMER4 Comparator 3	Slave_TIMER4 Comparator 2
Fault channel 5	Slave_TIMER5 reset / update	Slave_TIMER5 Comparator 2	Slave_TIMER5 Comparator 3	Slave_TIMER5 Comparator 2
Fault channel 6	Slave_TIMER6 reset / update	Slave_TIMER6 Comparator 2	Slave_TIMER6 Comparator 3	Slave_TIMER6 Comparator 2
Fault channel 7	Slave_TIMER7 reset / update	Slave_TIMER7 Comparator 2	Slave_TIMER7 Comparator 3	Slave_TIMER7 Comparator 2

The fault counter threshold is configurated by FLTyCNT[3:0] (y = 0..7) in HRTIMER_FLTINCFG2 register and HRTIMER_FLTINCFG3 register. When the number of fault channel event is equal to (FLTyCNT[3:0]+1), the fault is takeing effect.

When FLTxRST bit is reset, the fault counter is reset when reset and roll-over event occurs, refer to ***Table 25-23. Source of counter reset***.

When FLTxRST bit is set, fault counter reset mode is determined by HRTIMER_FLTRECCTL register. When the FLTRECCTL bit in HRTIMER_FLTRECCTL register is set to 0, fault counter reset on each reset / roll-over event only if no fault event is active. When the FLTRECCTL bit in HRTIMER_FLTRECCTL register is set to 1, fault counter reset on each reset / roll-over event only if no fault event happens in one period.

Note: When FLTxRST bit is reset, the fault counter reset source depends on ***Table 25-23. Source of counter reset.*** When FLTxRST bit is set, the fault counter reset source comes from the Slave_TIMERx connected to the fault channel. The reset / roll-over event of the Slave_TIMERx connected to the fault channel is used to reset the fault counter.

Figure 25-52. Fault counter when FLTxRST bit is 1, FLTRECCTL bit is 1 and FLTxCNT[3:0] = 0x03

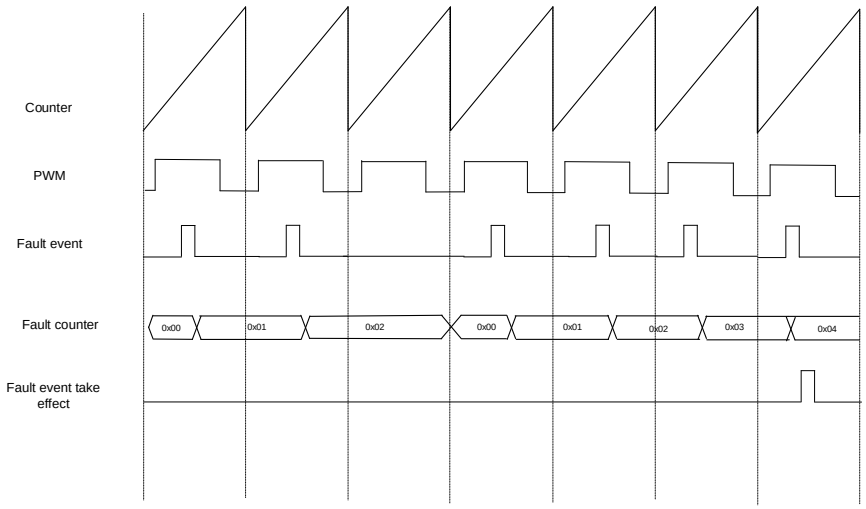


Table 25-23. Source of counter reset

Fault channel	Fault counter reset source
Fault channel 0	Slave_TIMER0 reset / roll-over
Fault channel 1	Slave_TIMER1 reset / roll-over
Fault channel 2	Slave_TIMER2 reset / roll-over
Fault channel 3	Slave_TIMER3 reset / roll-over
Fault channel 4	Slave_TIMER4 reset / roll-over
Fault channel 5	Slave_TIMER5 reset / roll-over
Fault channel 6	Slave_TIMER6 reset / roll-over
Fault channel 7	Slave_TIMER7 reset / roll-over

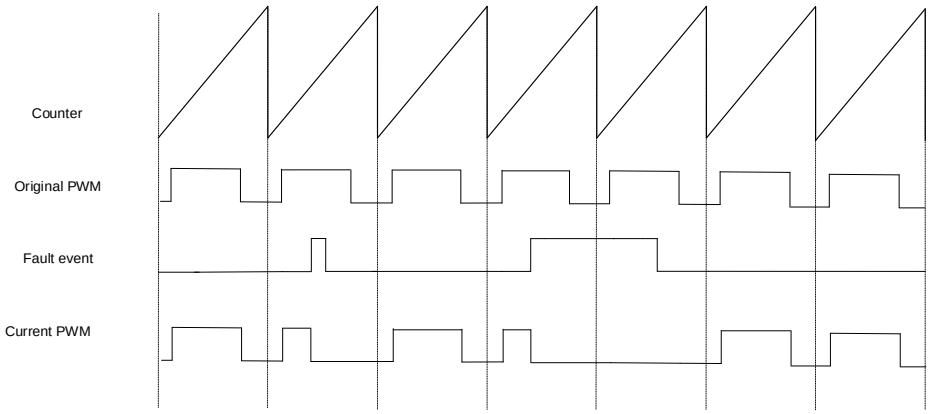
Fault resume

Fault hardware automatic resume

When the fault source inactive, the HRTIMER channel resumes PWM wave output.

When the FLTAR bit in HRTIMER_STxACTL register and the FLTxRST bit in HRTIMER_FLTINCFG2 and HRTIMER_FLTINCFG3 registers are used simultaneously, i.e., both FLTxRST (x = 0..7) and FLTAR bits are set, the channel can automatically resume PWM wave output in the next period after the fault inactive.

Figure 25-53. Fault automatic resume when FLTxRST bit is 1 and FLTAR = 1



Fault resume protection

When the FLTxRST (x = 0..7) bits are set to 1 and the FLTRECCTL bit is set to 0, the software cannot resume PWM output until the fault event is disappeared. PWM output can only be resumed after the fault source inactive and the channel output is re-enabled (Set the STxCHyEN bit in HRTIMER_CHOUTEN register, x = 0..7, y = 0,1).

Note: The STxCHyEN bit in the HRTIMER_CHOUTEN register does not correctly indicate the actual output state of the corresponding channel. The specific issues are as follows:

1) After a fault hardware automatic resume, the PWM channel resumes output and the PWM

waveform is restored, but the STxCHyEN bit remains incorrectly set to 0 instead of recovering to 1.

2) When the fault is active, even if the STxCHyEN bit is manually set via software, the PWM output remains disabled, while the STxCHyEN bit incorrectly displays a value of 1.

Due to these issues, when using the fault input function, the STxCHyEN bit cannot reliably indicate the channel state. Therefore, it is recommended to avoid relying on reading this bit for status verification.

System fault

The system fault comes from the signal inside the chip:

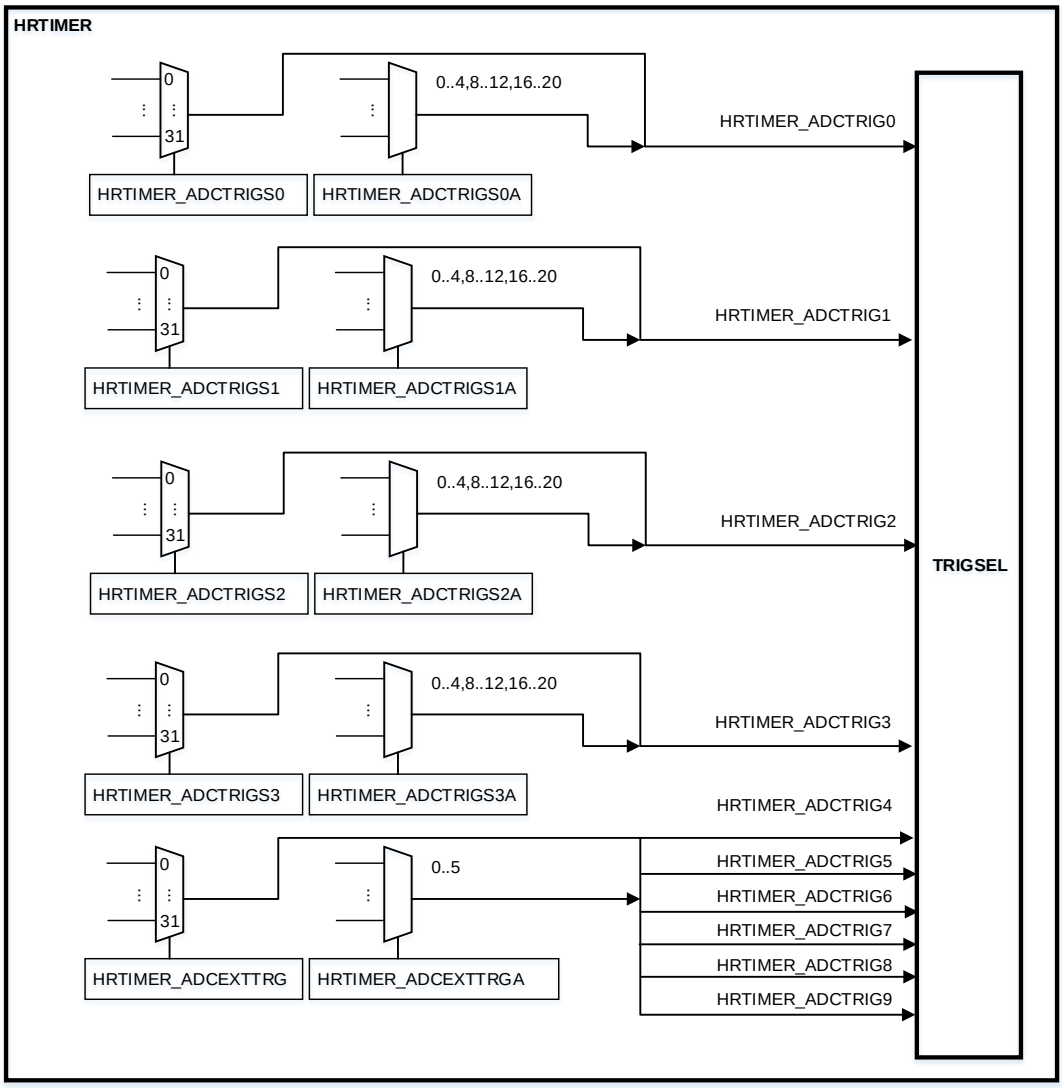
- HXTAL failure event generated by Clock Monitor.
- The Cortex®-M33-lockup signal.
- The output of Low Voltage Detector.

The system fault is valid when FLTyEN bit in HRTIMER_STxFLTCTL register is set to 1. The system fault can overrides the fault channel input (logic ORed).

25.4.8. Trigger to ADC

The ADCs can be triggered by the TRIGSEL. Ten independent ADC triggers (HRTIMER_ADCTRIG0 - HRTIMER_ADCTRIG9) are available to start the routine sequence of the ADCs. Refer to ***Figure 25-54. Trigger to ADC selection overview.***

Figure 25-54. Trigger to ADC selection overview



For ADC triggers 0 to 3 (HRTIMER_ADCTRIG0 - HRTIMER_ADCTRIG3), there are up to 47 events which can be combined (ORed) for trigger output(HRTIMER_ADCTRIGy(y = 0..3)) , 32 events are defined in HRTIMER_ADCTRIGSy(y=0..3) and 15 events are defined in HRTIMER_ADCTRIGSyA(y=0..3)

For ADC triggers 4 to 9 (HRTIMER_ADCTRIG4 - HRTIMER_ADCTRIG9), each trigger output can only be connected to one event at a time (selected from 47 events). The 47 events are defined in the HRTIMER_ADCEXTTRG and HRTIMER_ADCEXTTRGA registers.

HRTIMER_ADCTRIGSy(y=0..3) registers, HRTIMER_ADCTRIGSyA(y=0..3) registers, HRTIMER_ADCEXTTRG register and HRTIMER_ADCEXTTRGA register are preloaded and can be updated synchronously with the timer they are related to. The update source is defined with ADTGyUSRC[3:0] bit-field in the HRTIMER_CTL0 register and HRTIMER_ADCTRGUPD register. For example, ADTGyUSRC[3:0] = 4'b0001 and Slaver_TIMER0 is update source:

- If SHWEN = 1 in HRTIMER_STxCTL0 register, HRTIMER_ADCTRIGSy(y=0..3) and

HRTIMER_ADCTRIGSyA(y=0..3) registers is preloaded and can be updated synchronously with Slaver_TIMER0.

- If SHWEN = 0 in HRTIMER_STxCTL0 register, HRTIMER_ADCTRIGSy(y=0..3) and HRTIMER_ADCTRIGSyA(y=0..3) registers is not preloaded and a write access will result in an immediate update of the trigger source.

Trigger frequency division and other functions are set in HRTIMER_ADCTRIGS0-HRTIMER_ADCTRIGS9 in order to be more compatible with ADC triggering. It should be noted that these trigger sources are connected to TRIGSEL, and TRIGSEL decides what module to trigger.

ADC trigger division

The trigger frequency of the ADC can be divided by configuring the HRTIMER_ADCPSCRy (y=0,1) register.

In counter up mode, the trigger frequency of the ADC is only related to ADCxPSC[4:0] bits in HRTIMER_ADCPSCRy (y=0,1) register, and in center aligned counting mode, The ADC trigger frequency is also related to ADCROVM[1:0] bits in HRTIMER_STxCTL1 register.

- ADCROVM[1:0] = 00: ADC trigger event generated when counter in up direction and in down direction
- ADCROVM[1:0] = 01: ADC trigger event generated when counter in down direction
- ADCROVM[1:0] = 10: ADC trigger event generated when counter in up direction

Figure 25-55. ADC trigger division in up counting mode

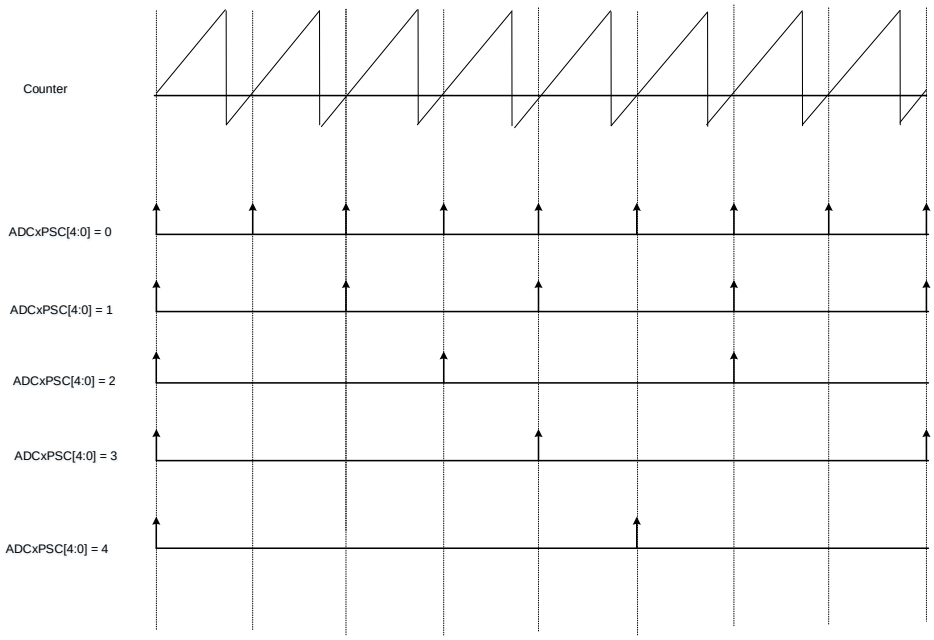
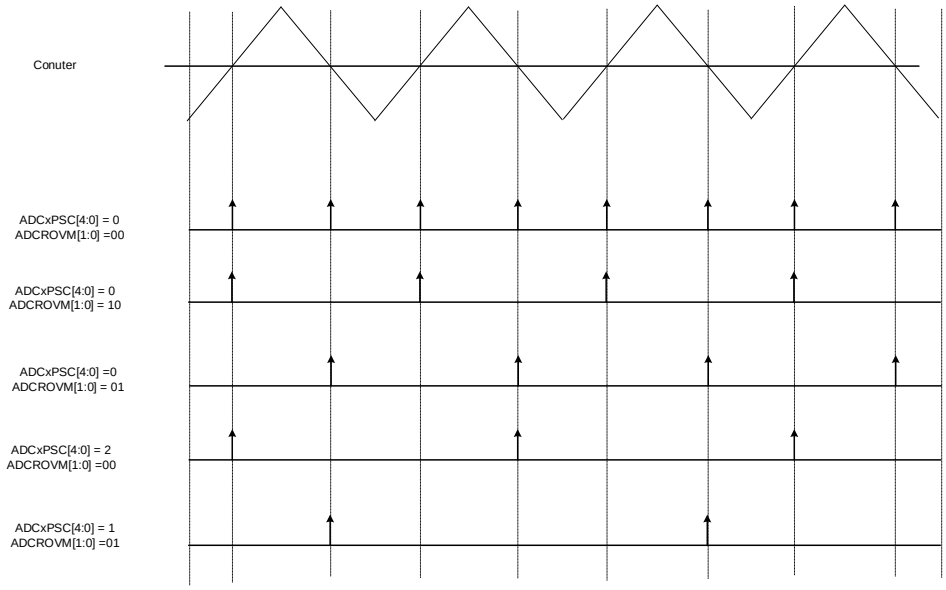


Figure 25-56. ADC trigger division in center aligned mode



25.4.9. Trigger to DAC

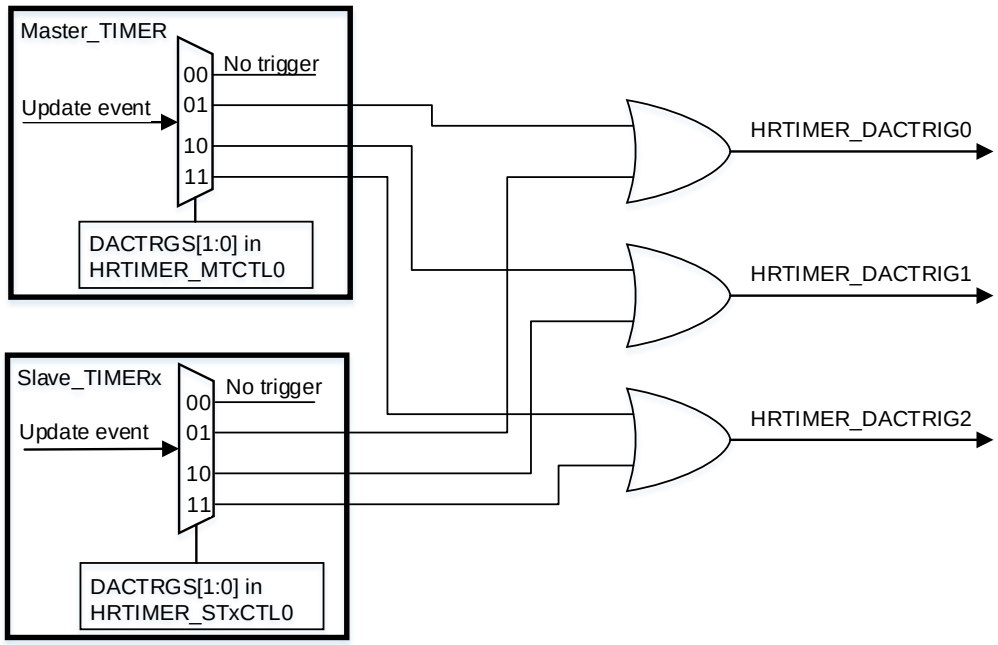
The HRTIMER allows to have the embedded DACs updated synchronously with the timer updates. The update events from the Master_TIMER and Slave_TIMERx can generate DAC update triggers on HRTIMER_DACTRIGy(y=0..2).

DACTRGS[1:0] bit-field of the HRTIMER_MTCTL0 and HRTIMER_STxCTL0 registers are programmed as follows:

- 00: No DAC trigger event generated
- 01: DAC trigger event generated on HRTIMER_DACTRIG0
- 10: DAC trigger event generated on HRTIMER_DACTRIG1
- 11: DAC trigger event generated on HRTIMER_DACTRIG2

When DACTRGS[1:0] bit-field are enabled in multiple timers, the HRTIMER_DACTRIGy(y=0..2) will consist of an OR of all timers' update events. Refer to ***Figure 25-57. Trigger to DAC selection overview.***

Figure 25-57. Trigger to DAC selection overview



An output pulse of 32 f_{APB2} clock periods is generated on the DAC triggers output. The synchronization pulse is followed by an idle level of 32 APB2 clock cycles during which any new DAC update request is ignored. Consequently, the maximum synchronization frequency is f_{APB2} / 64.

HRTIMER_DACTRIG0- HRTIMER_DACTRIG2 are connected to TRIGSEL, and TRIGSEL decides what module to trigger.

25.4.10. Double source trigger

Enabling double source trigger mode requires setting the TRIGEN bit in the HRTIMER_STxCTL1 register. Once the timer is running (with the STxCEN bit set), the TRIGEN bit cannot be changed. At this time, the TRIG0M and TRIG1M in the HRTIMER_STxCTL1 register configure the trigger signals TRIG0 and TRIG1 output by the Slave_TIMERx to be connected to TRIGSEL (in the TRIGSEL module, they are respectively HRTIMER_STx_TRIG0 and HRTIMER_STx_TRIG1). By configuring the TRIG0M and TRIG1M bits in the HRTIMER_STxCTL1 register, user can select the trigger signal source.

The TRIG0 signal is configured through the TRIG0M

- TRIG0M = 0 : Compare 1 event generate TRIG0 trigger.
- TRIG0M = 1 : Output 0 reset event generate TRIG0 trigger.

The TRIG1 signal is configured through the TRIG1M

- TRIG1M = 0 : Counter reset or roll-over event generate TRIG1 trigger.
- TRIG1M = 1 : Output 0 set event generate TRIG1 trigger.

Below is an example using the DAC to illustrate the use of double-source trigger mode. With

this mode, it is easy to implement slope compensation techniques and hysteresis control. In this mode, the DAC outputs a sawtooth signal that gradually decreases, with the period of the sawtooth wave synchronized with the period of the PWM wave.

Note: In double source trigger mode, only Slave Timer 0 and Slave Timer 3 support hardware slope compensation when TRIG0M = 0 and TRIG1M = 0, but compare 1 register has the following limitations:

When CNTCKDIV[2:0] < 3'b101

- The minimum value must be greater than or equal to the count value corresponding to 8 $t_{HRTIMER_CK}$ period.

Note: The count value corresponding to each $t_{HRTIMER_CK}$ period = $f_{HRTIMER_PSSCK}$ / $f_{HRTIMER_CK}$.

When CNTCKDIV[2:0] >= 3'b101

- The minimum value must be greater than or equal to 0x0008.

The DAC module can customize which signal to trigger the DAC to produce a step or reset signal based on the trigger signal from TRIGSEL. This example describes using the TRIG0 from HRTIMER to generate a step signal and TRIG1 to generate a reset signal. It should be noted that the signals that trigger the DAC step or reset are not limited to these two signals; any signal from TRIGSEL can trigger it.

Figure 25-58. Trigger to DAC when TRIG0M = 0 and TRIG1M = 0

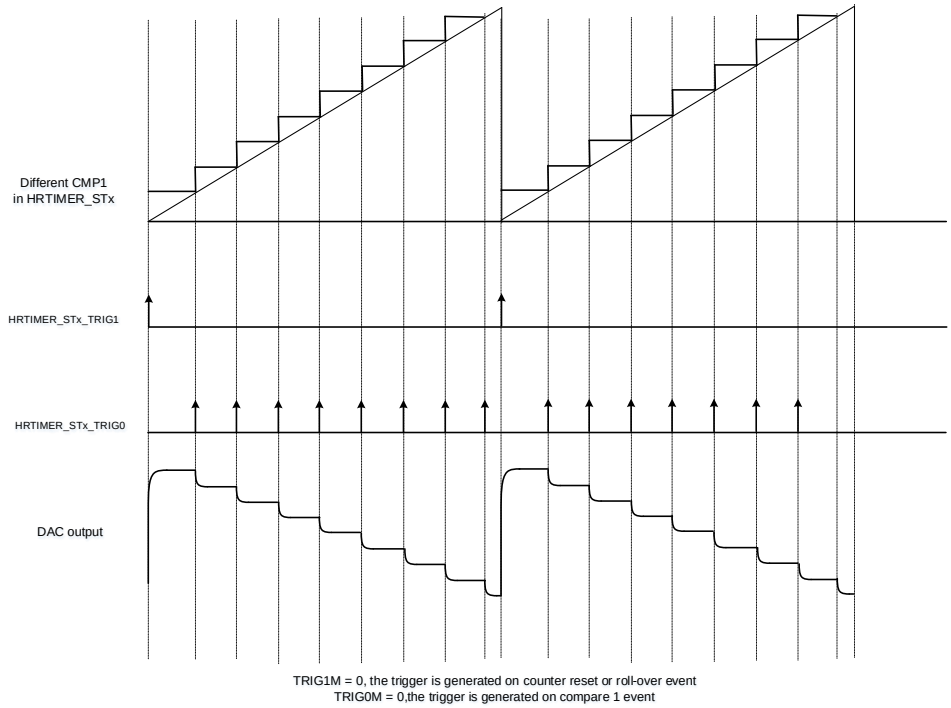
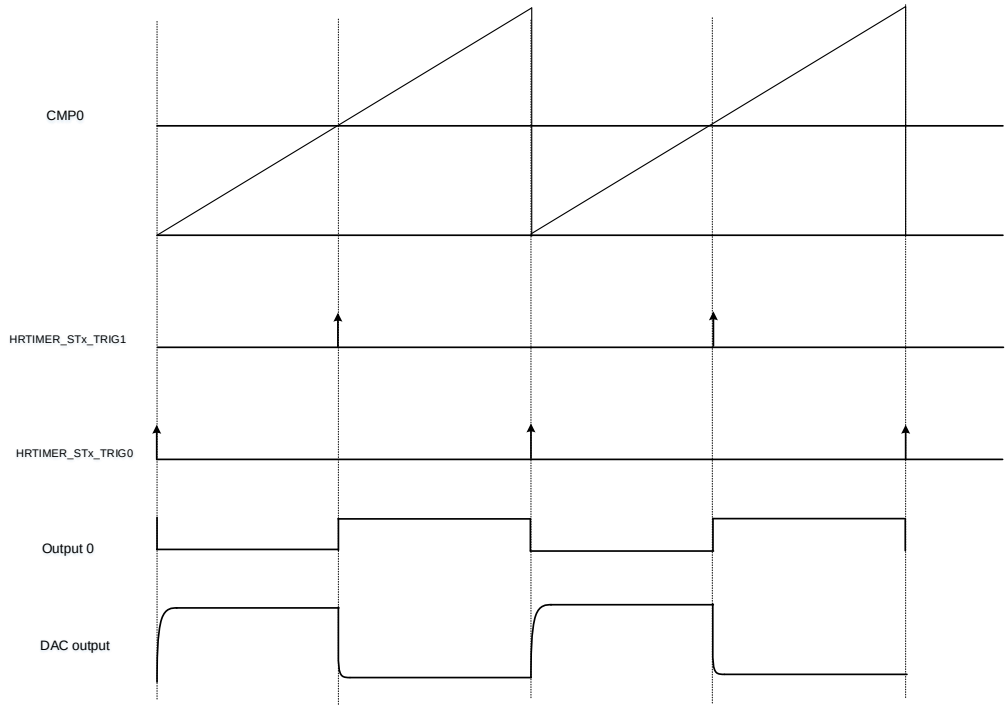


Figure 25-59. Trigger to DAC when TRIG0M = 1 and TRIG1M = 1



25.4.11. Interrupt

Most events can generate interrupt requests. All interrupt requests are grouped in 10 vectors (HRTIMER_IRQy,y=0..9).Refer to ***Table 25-24. Interrupt mapping*** for details

Table 25-24. Interrupt mapping

Interrupt Number	Event	Control bit
Master_TIMER: HRTIMER_IRQ0	Update event	UPIE in HRTIMER_MTDMAINTEN
	Synchronization input event	SYNIIE in HRTIMER_MTDMAINTEN
	Repetition event	REPIE in HRTIMER_MTDMAINTEN
	Compare 0 event	CMP0IE in HRTIMER_MTDMAINTEN
	Compare 1 event	CMP1IE in HRTIMER_MTDMAINTEN
	Compare 2 event	CMP2IE in HRTIMER_MTDMAINTEN
	Compare 3 event	CMP3IE in HRTIMER_MTDMAINTEN
Slave_TIMER0: HRTIMER_IRQ1	Delayed IDLE mode entry	DLYIIE in HRTIMER_STxDMAINTEN
	Counter reset event	RSTIE in HRTIMER_STxDMAINTEN
Slave_TIMER1: HRTIMER_IRQ2	C1OPRE goes from active to inactive	CH1ONAIE in HRTIMER_STxDMAINTEN
	C1OPRE goes from inactive to active	CH1OAIE in HRTIMER_STxDMAINTEN
Slave_TIMER2: HRTIMER_IRQ3	C0OPRE goes from active to inactive	CH0ONAIE in HRTIMER_STxDMAINTEN
	C0OPRE goes from inactive to active	CH0OAIE in HRTIMER_STxDMAINTEN
Slave_TIMER3:	Capture 1 event	CAP1IE in HRTIMER_STxDMAINTEN

Interrupt Number	Event	Control bit
HRTIMER_IRQ4 Slave_TIMER4: HRTIMER_IRQ5 Slave_TIMER5: HRTIMER_IRQ6 Slave_TIMER6: HRTIMER_IRQ7 Slave_TIMER7: HRTIMER_IRQ8	Capture 0 event	CAP0IE in HRTIMER_STxDMAINTEN
	Update event	UPIE in HRTIMER_STxDMAINTEN
	Repetition event	REPIE in HRTIMER_STxDMAINTEN
	Compare 3 event	CMP3IE in HRTIMER_STxDMAINTEN
	Compare 2 event	CMP2IE in HRTIMER_STxDMAINTEN
	Compare 1 event	CMP1IE in HRTIMER_STxDMAINTEN
	Compare 0 event	CMP0IE in HRTIMER_STxDMAINTEN
HRTIMER_IRQ0	Bunch mode period event	BMPERIE in HRTIMER_INTEN
	DLL calibration completed	DLLCALIE in HRTIMER_INTEN
HRTIMER_IRQ9	System fault	SYSFLTIE in HRTIMER_INTEN
	fault x (x = 0..7)	FLTxIE (x = 0..7) in HRTIMER_INTEN

25.4.12. DMA request

Most events can generate DMA requests and each timer corresponds to a DMA channel. Refer to ***Table 25-25. DMA request mapping*** for details.

Table 25-25. DMA request mapping

DMA channel	Event	Control bit
Master_TIMER: DMA0_Channel1	Update event	UPDEN in HRTIMER_MTDMAINTEN
	Synchronization input event	SYNIDEN in HRTIMER_MTDMAINTEN
	Repetition event	REPDEN in HRTIMER_MTDMAINTEN
	Compare 0 event	CMP0DEN in HRTIMER_MTDMAINTEN
	Compare 1 event	CMP1DEN in HRTIMER_MTDMAINTEN
	Compare 2 event	CMP2DEN in HRTIMER_MTDMAINTEN
	Compare 3 event	CMP3DEN in HRTIMER_MTDMAINTEN
Slave_TIMER0: DMA0_Channel2	Delayed IDLE mode entry	DLYIDEN in HRTIMER_STxDMAINTEN
	Counter reset event	RSTDEN in HRTIMER_STxDMAINTEN
Slave_TIMER1: DMA0_Channel3	C1OPRE goes from active to inactive	CH1ONADEN in HRTIMER_STxDMAINTEN
	C1OPRE goes from inactive to active	CH1OADEN in HRTIMER_STxDMAINTEN
Slave_TIMER2: DMA0_Channel4	C0OPRE goes from active to inactive	CH0ONADEN in HRTIMER_STxDMAINTEN
	C0OPRE goes from inactive to active	CH0OADEN in HRTIMER_STxDMAINTEN
Slave_TIMER3: DMA0_Channel5	Capture 1 event	CAP1DEN in HRTIMER_STxDMAINTEN
	Capture 0 event	CAP0DEN in HRTIMER_STxDMAINTEN
Slave_TIMER4: DMA0_Channel6	Update event	UPDEN in HRTIMER_STxDMAINTEN
	Repetition event	REPDEN in HRTIMER_STxDMAINTEN
Slave_TIMER5: DMA0_Channel7	Compare 3 event	CMP3DEN in HRTIMER_STxDMAINTEN
	Compare 2 event	CMP2DEN in HRTIMER_STxDMAINTEN
Slave_TIMER6: DMA0_Channel8	Compare 1 event	CMP1DEN in HRTIMER_STxDMAINTEN
	Compare 0 event	CMP0DEN in HRTIMER_STxDMAINTEN

DMA channel	Event	Control bit
Slave_TIMER7: DMA0_Channel9		

Note: It is necessary to disable first the DMA controller before disable a DMA request.

25.4.13. DMA mode

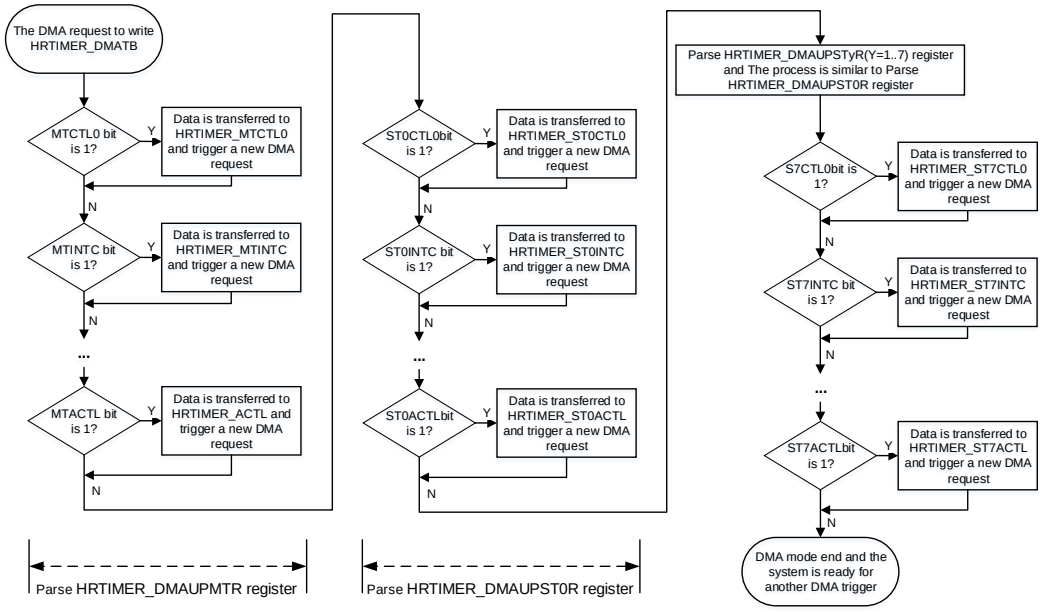
Timer's DMA mode is the function that configures HRTIMER's multiple registers by DMA module with a single DMA request. The relative registers (10 registers in total) are as follows:

- HRTIMER_DMAUPMTR: Defines which registers in the Master_TIMER are updated. Most of Master_TIMER control and data registers are associated with a selection bit. If the selection bit is set, the write access is redirected to the associated register.
- HRTIMER_DMAUPSTxR(x=0..7): Defines which registers in the Slave_TIMERx are updated. Most of Slave_TIMERx control and data registers are associated with a selection bit. If the selection bit is set, the write access is redirected to the associated register.
- HRTIMER_DMATB: DMA transfer buffer register. It is only necessary to have the DMA module pointing to the HRTIMER_DMATB register as the destination, to the peripheral configuration with the peripheral increment mode disabled. All write accesses to this register will be internally re-routed to the final destination register by a redirection mechanism.

The DMA mode is permanently enabled (there is no enable bit). A DMA operation is started by the first write access into the HRTIMER_DMATB register.

When the DMA request occurs, the HRTIMER generates multiple 32-bit DMA requests and parses the register to be updated (defined in HRTIMER_DMAUPMTR and HRTIMER_DMAUPSTxR registers). If the selection bit is set to 1, the write access is redirected to the associated register. If the bit is 0, the register update is skipped and the register parsing is resumed until a new bit set is detected, to trigger a new DMA request. Once the 9 registers (HRTIMER_DMAUPMTR and HRTIMER_DMAUPSTxR registers) are all parsed, the DMA mode is completed and a DMA mode end event is generated. The system is ready for another DMA trigger. If one more DMA request event coming, HRTIMER will repeat the process as above. Refer to ***Figure 25-60. DMA mode operation flowchart.***

Figure 25-60. DMA mode operation flowchart



25.4.14. Debug mode

When the Cortex®-M33 halted, the HRTIMER_HOLD bit in DBG_CTL0 register determines whether the counter stops running.

HRTIMER_HOLD = 0

If HRTIMER_HOLD = 0, the HRTIMER continues to work normally.

HRTIMER_HOLD = 1

If HRTIMER_HOLD = 1, the counter in Master_TIMER and all Slave_TIMERx is stopped.

The outputs enter the Fault state if CHyFLTOS[1:0] = 2'b01, 02'b10, 2'b11. The outputs can be enabled again by settings STxCHyEN bit in HRTIMER_CHOUTEN register and clearing HRTIMER_HOLD. If CHyFLTOS[1:0] = 2'b00, the outputs keep their current state. When exit debug mode, the output returns to its original state.

All counter reset/start and capture triggers are disabled. All triggers from external events are disabled, except for triggers to ADCs. The update events are discarded. The bunch mode circuit is frozen: the triggers are ignored and the bunch mode counter stopped.

DLL calibration is run normally. These units which drive the normal output in RUN mode are not affected by debug, for instance the dead-time unit, the carrier-signal, the set/reset crossbar and so on.

25.5. Register definition

HRTIMER base address: 0x4001 5800

The registers can be segmented for ease of addressing：

HRTIMER Master_TIMER registers base address: 0x4001 5800

HRTIMER Slave_TIMER0 registers base address: 0x4001 5880

HRTIMER Slave_TIMER1 registers base address: 0x4001 5900

HRTIMER Slave_TIMER2 registers base address: 0x4001 5980

HRTIMER Slave_TIMER3 registers base address: 0x4001 5A00

HRTIMER Slave_TIMER4 registers base address: 0x4001 5A80

HRTIMER Slave_TIMER5 registers base address: 0x4001 5B00

HRTIMER Slave_TIMER6 registers base address: 0x4001 6000

HRTIMER Slave_TIMER7 registers base address: 0x4001 6080

HRTIMER Common registers base address: 0x4001 5B80

25.5.1. Master_TIMER registers

HRTIMER Master_TIMER registers base address: 0x4001 5800

HRTIMER Master_TIMER control register 0 (HRTIMER_MTCTL0)

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
UPSEL[1:0]		UPREP	Reserved	SHWEN	DACTRGS[1:0]		ST7CEN	ST6CEN	ST5CEN	ST4CEN	ST3CEN	ST2CEN	ST1CEN	ST0CEN	MTCEN
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
SYNOSRC[1:0]		SYNOPLS[1:0]		SYNISTRT	SYNIRST	SYNISRC[1:0]		ALTM[1:0]		HALFM	CNTRSTM	CTNM	CNTCKDIV[2:0]		
rw		rw		rw	rw	rw		rw		rw	rw	rw	rw		

Bits	Fields	Descriptions
31:30	UPSEL[1:0]	Update event selection This bit-field specifies which the relationship between update events and the DMA mode. 00: Update event generated independently from DMA mode.

		01: Update event generated when the DMA transfer is completed in DMA mode. 10: Update event generated on counter roll-over following a DMA transfer completion in DMA mode. This configuration is only applicable in continuous mode. 11: Reserved
29	UPREP	Update event generated by repetition event This bit specifies whether repetition event can generate update event. 0: Update event generated by repetition event disable 1: Update event generated by repetition event enable **Note:** UPREP can be set only if UPSEL[1:0] = 2'b00 or 2'b01.
28	Reserved	Must be kept at reset value
27	SHWEN	Shadow registers enable 0: The shadow registers are disabled 1: The shadow registers are enabled
26:25	DACTRGS[1:0]	Trigger source to DAC The timer can also generate a DAC trigger event when an update event occurs. This bit-field specifies which trigger source generates the DAC trigger event. 00:No DAC trigger event generated 01: DAC trigger event generated on HRTIMER_DACTRIG0 10: DAC trigger event generated on HRTIMER_DACTRIG1 11: DAC trigger event generated on HRTIMER_DACTRIG2
24	ST7CEN	The counter of Slave_TIMER7 enable 0: The counter of Slave_TIMER7 disable 1: The counter of Slave_TIMER7 enable **Note:** This bit must not be modified within a minimum of 8 $t_{HRTIMER_CK}$ clock.
23	ST6CEN	The counter of Slave_TIMER6 enable 0: The counter of Slave_TIMER6 disable 1: The counter of Slave_TIMER6 enable **Note:** This bit must not be modified within a minimum of 8 $t_{HRTIMER_CK}$ clock.
22	ST5CEN	The counter of Slave_TIMER5 enable 0: The counter of Slave_TIMER5 disable 1: The counter of Slave_TIMER5 enable **Note:** This bit must not be modified within a minimum of 8 $t_{HRTIMER_CK}$ clock.
21	ST4CEN	The counter of Slave_TIMER4 enable 0: The counter of Slave_TIMER4 disable 1: The counter of Slave_TIMER4 enable **Note:** This bit must not be modified within a minimum of 8 $t_{HRTIMER_CK}$ clock.
20	ST3CEN	The counter of Slave_TIMER3 enable 0: The counter of Slave_TIMER3 disable 1: The counter of Slave_TIMER3 enable

		Note: This bit must not be modified within a minimum of 8 $t_{HRTIMER_CK}$ clock.
19	ST2CEN	The counter of Slave_TIMER2 enable 0: The counter of Slave_TIMER2 disable 1: The counter of Slave_TIMER2 enable **Note:** This bit must not be modified within a minimum of 8 $t_{HRTIMER_CK}$ clock.
18	ST1CEN	The counter of Slave_TIMER1 enable 0: The counter of Slave_TIMER1 disable 1: The counter of Slave_TIMER1 enable **Note:** This bit must not be modified within a minimum of 8 $t_{HRTIMER_CK}$ clock.
17	ST0CEN	The counter of Slave_TIMER0 enable 0: The counter of Slave_TIMER0 disable 1: The counter of Slave_TIMER0 enable **Note:** This bit must not be modified within a minimum of 8 $t_{HRTIMER_CK}$ clock.
16	MTCEN	The counter of Master_TIMER enable 0: The counter of Master_TIMER disable 1: The counter of Master_TIMER enable **Note:** This bit must not be modified within a minimum of 8 $t_{HRTIMER_CK}$ clock.
15:14	SYNOSRC[1:0]	Synchronization output source This bit-field specifies the event to be sent to the synchronization output pad HRTIMER_SCOUT. 00: Master_TIMER start event. 01: Master_TIMER compare 0 event 10: Slave_TIMER0 reset and start event 11: Slave_TIMER0 compare 0 event
13:12	SYNOPLS[1:0]	Synchronization output pulse This bit-field specifies the pulse on the synchronization output pad HRTIMER_SCOUT. 00: Pulse generated disable. No pulse on HRTIMER_SCOUT. 01: Reserved. 10: Positive pulse generated on the HRTIMER_SCOUT. The length of it is 16 $t_{HRTIMER_CK}$ cycles. 11: Negative pulse generated on the HRTIMER_SCOUT. The length of it is 16 $t_{HRTIMER_CK}$ cycles.
11	SYNISTRT	Synchronization input start counter This bit specifies whether the synchronous input can start the counter. 0: Synchronization input cannot start counter. 1: Synchronization input can start counter.
10	SYNIRST	Synchronization input reset counter This bit specifies whether the synchronous input can reset the counter.

0: Synchronization input cannot reset counter.
1: Synchronization input can reset counter.

9:8	SYNISRC[1:0]	Synchronization input source This bit-field specifies the synchronization input source. 00: Synchronization input disable. 01: Reserved. 10: Internal signal: TIMER0_TRGO in the advanced timer TIMER0 11: External signal: a positive pulse on the HRTIMER_SCIN pin triggers the Master_TIMER. **Note:** This bit-field cannot be modified once the impacted timers are enabled
7:6	ALTM[1:0]	Alternate mode This bitfield is significant only when the HALFM bit is reset. It enables the alternate mode. 00: Alternate mode disabled 01: Triple alternate mode: when HRTIMER_MTCAR register is written, the HRTIMER_MTCMP0V active register is automatically updated with HRTIMER_MTCAR/3 value, and the HRTIMER_MTCMP1V active register is automatically updated with 2x (HRTIMER_MTCAR/3) value. 10: Quad alternate mode: when HRTIMER_MTCAR register is written, the HRTIMER_MTCMP0V active register is automatically updated with HRTIMER_MTCAR/4 value, the HRTIMER_MTCMP1V active register is automatically updated with HRTIMER_MTCAR/2 value and the HRTIMER_MTCMP2V active register is automatically updated with 3x (HRTIMER_MTCAR/4) value. 11: Alternate mode disabled
5	HALFM	Half mode When the bit is set, HRTIMER_MTCMP0V active register is always the half of counter auto-reload value (HRTIMER_MTCAR). 0: Half mode disable. 1: Half mode enable.
4	CNTRSTM	Counter reset mode This bit defines the behavior of the timer counter in single pulse mode. 0: The counter can be reset only if it stops (period elapsed) 1: The counter can be reset at any time (running or stopped).
3	CTNM	Continuous mode 0: Single pulse mode. The counter stops by hardware when it reaches the HRTIMER_MTCAR value. 1: Continuous mode. The counter rolls over to zero and count continuously when it reaches the HRTIMER_MTCAR value
2:0	CNTCKDIV[2:0]	Counter clock division

This bit-field can be configured by software to specify division ratio between the high resolution clock (HRTIMER_HPCK) and the counter clock (HRTIMER_PSCCK). $f_{HRTIMER_PSCCK} = f_{HRTIMER_HPCK}/2^{CNTCKDIV[2:0]}$.

000: $f_{HRTIMER_PSCCK} = f_{HRTIMER_HPCK}$
001: $f_{HRTIMER_PSCCK} = f_{HRTIMER_HPCK}/2$
010: $f_{HRTIMER_PSCCK} = f_{HRTIMER_HPCK}/4$
011: $f_{HRTIMER_PSCCK} = f_{HRTIMER_HPCK}/8$
100: $f_{HRTIMER_PSCCK} = f_{HRTIMER_HPCK}/16$
101: $f_{HRTIMER_PSCCK} = f_{HRTIMER_HPCK}/32$
110: $f_{HRTIMER_PSCCK} = f_{HRTIMER_HPCK}/64$
111: $f_{HRTIMER_PSCCK} = f_{HRTIMER_HPCK}/128$

Note: The CNTCKDIV[2:0] bit-field cannot be modified once the timer is enabled.

HRTIMER Master_TIMER interrupt flag register (HRTIMER_MTINTF)

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									UPIF	SYNIIF	REPIF	CMP3IF	CMP2IF	CMP1IF	CMP0IF
									r	r	r	r	r	r	r

Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value
6	UPIF	Update interrupt flag This flag is set by hardware when an update event occurs. 0: No update interrupt occurred 1: Update interrupt occurred
5	SYNIIF	Synchronization input interrupt flag This flag is set by hardware when synchronization input occurs. 0: No synchronization input interrupt occurred 1: Synchronization input interrupt occurred
4	REPIF	Repetition interrupt flag This flag is set by hardware when a repetition event occurs. 0: No repetition interrupt occurred 1: Repetition interrupt occurred

3	CMP3IF	Compare 3 interrupt flag This flag is set by hardware when a compare 3 event occurs. 0: No compare 3 interrupt occurred 1: Compare 3 interrupt occurred
2	CMP2IF	Compare 2 interrupt flag This flag is set by hardware when a compare 2 event occurs. 0: No compare 2 interrupt occurred 1: Compare 2 interrupt occurred
1	CMP1IF	Compare 1 interrupt flag This flag is set by hardware when a compare 1 event occurs. 0: No compare 1 interrupt occurred 1: Compare 1 interrupt occurred
0	CMP0IF	Compare 0 interrupt flag This flag is set by hardware when a compare 0 event occurs. 0: No compare 0 interrupt occurred 1: Compare 0 interrupt occurred

HRTIMER Master_TIMER interrupt flag clear register (HRTIMER_MTINTC)

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									UPIFC	SYNIIFC	REPIFC	CMP3IFC	CMP2IFC	CMP1IFC	CMP0IFC
									w	w	w	w	w	w	w

Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value
6	UPIFC	Clear update interrupt flag 0: No effect 1: Clear update interrupt flag
5	SYNIIFC	Clear synchronization input interrupt flag 0: No effect 1: Clear synchronization input interrupt flag
4	REPIFC	Clear repetition interrupt flag 0: No effect

		1: Clear repetition interrupt flag
3	CMP3IFC	Clear compare 3 interrupt flag 0: No effect 1: Clear compare 3 interrupt flag
2	CMP2IFC	Clear compare 2 interrupt flag 0: No effect 1: Clear compare 2 interrupt flag
1	CMP1IFC	Clear compare 1 interrupt flag 0: No effect 1: Clear compare 1 interrupt flag
0	CMP0IFC	Clear compare 0 interrupt flag 0: No effect 1: Clear compare 0 interrupt flag

HRTIMER Master_TIMER DMA and interrupt enable register (HRTIMER_MTDMAINTEN)

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									UPDEN	SYNIDEN	REPDEN	CMP3DEN	CMP2DEN	CMP1DEN	CMP0DEN
									rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									UPIE	SYNIIE	REPIE	CMP3IE	CMP2IE	CMP1IE	CMP0IE
									rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:23	Reserved	Must be kept at reset value
22	UPDEN	Update DMA request enable 0: disabled 1: enabled
21	SYNIDEN	Synchronization input DMA request enable 0: disabled 1: enabled
20	REPDEN	Repetition DMA request enable 0: disabled

		1: enabled
19	CMP3DEN	Compare 3 DMA request enable 0: disabled 1: enabled
18	CMP2DEN	Compare 2 DMA request enable 0: disabled 1: enabled
17	CMP1DEN	Compare 1 DMA request enable 0: disabled 1: enabled
16	CMP0DEN	Compare 0 DMA request enable 0: disabled 1: enabled
15:7	Reserved	Must be kept at reset value
6	UPIE	Update interrupt enable 0: disabled 1: enabled
5	SYNIIE	Synchronization input interrupt enable 0: disabled 1: enabled
4	REPIE	Repetition interrupt enable 0: disabled 1: enabled
3	CMP3IE	Compare 3 interrupt enable 0: disabled 1: enabled
2	CMP2IE	Compare 2 interrupt enable 0: disabled 1: enabled
1	CMP1IE	Compare 1 interrupt enable 0: disabled 1: enabled
0	CMP0IE	Compare 0 interrupt enable 0: disabled 1: enabled

HRTIMER Master_TIMER counter register (HRTIMER_MTCNT)

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
CNT[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CNT[15:0]	The current counter value. Writing to it can change the value of the counter only when the Master_TIMER is stopped (MTCEN = 0 in HRTIMER_MTCTL0 register). **Note:** (1) For counter clock division below 64 (CNTCKDIV[3:0] < 5), the least significant bits of the counter are not significant. They cannot be written and read 0. (2) If the value written to this bit-field is above the HRTIM_MPER register value, the behavior of the timer is unpredictable.

HRTIMER Master_TIMER counter auto reload register (HRTIMER_MTCAR)

Address offset: 0x14

Reset value: 0x0000 FFDF

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

31	30	29	28	27	26	25	24	23	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. This register has a shadow register. If the shadow register is disabled (SHWEN = 0), it holds the content of the active register; otherwise, it holds the content of the shadow register.

Note:

(1) The minimum value must be greater than or equal to 3 $t_{HRTIMER_CK}$. For example: CARL[15:0] >= 0x60 when CNTCKDIV[3:0] = 4'b0000.

(2) The maximum value must be less than or equal to 0xFFFF – (1 $t_{HRTIMER_CK}$). For example: CARL[15:0] <= 0xFFDF when CNTCKDIV[3:0] = 4'b0000.

HRTIMER Master_TIMER counter repetition register (HRTIMER_MTCREP)

Address offset: 0x18

Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	CREP[7:0]	Counter repetition value This bit-field specifies the repetition event generation rate. When the repetition counter had count down to zero, the coming roll-over event in continuous mode or reset event will generate a repetition event. This register has a shadow register. If the shadow register is disabled (SHWEN = 0), it holds the content of the active register; otherwise, it holds the content of the shadow register.

HRTIMER Master_TIMER compare 0 value register (HRTIMER_MTCMP0V)

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
CMP0VAL[15:0]															
rw															

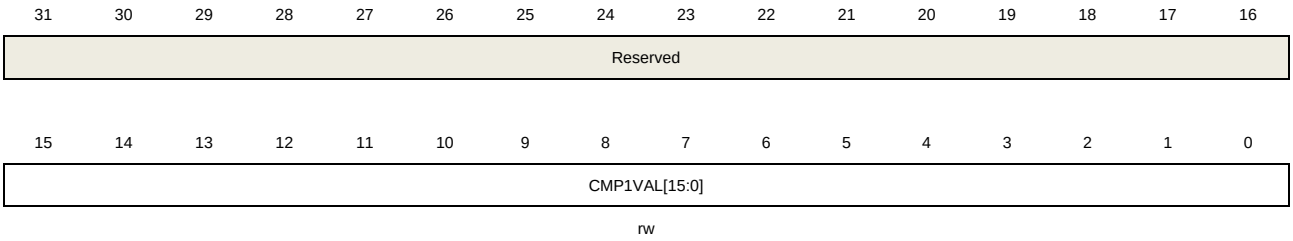
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:0	CMP0VAL[15:0]	Compare 0 value This bit-field contains value to be compared to the counter. This register has a shadow register. If the shadow register is disabled (SHWEN = 0), it holds the content of the active register; otherwise, it holds the content of the shadow register. **Note:** (1) The minimum value must be greater than or equal to 3 $t_{HRTIMER_CK}$. For example: CARL[15:0] >= 0x60 when CNTCKDIV[3:0] = 4'b0000. (2) The maximum value must be less than or equal to 0xFFFF – (1 $t_{HRTIMER_CK}$). For example: CARL[15:0] <= 0xFFDF when CNTCKDIV[3:0] = 4'b0000.

HRTIMER Master_TIMER compare 1 value register (HRTIMER_MTCMP1V)

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	CMP1VAL[15:0]	Compare 1 value This bit-field contains value to be compared to the counter. This register has a shadow register. If the shadow register is disabled (SHWEN = 0), it holds the content of the active register; otherwise, it holds the content of the shadow register. **Note:** (1) The minimum value must be greater than or equal to 3 $t_{HRTIMER_CK}$. For example: CARL[15:0] >= 0x60 when CNTCKDIV[3:0] = 4'b0000. (2) The maximum value must be less than or equal to 0xFFFF – (1 $t_{HRTIMER_CK}$). For example: CARL[15:0] <= 0xFFDF when CNTCKDIV[3:0] = 4'b0000.

HRTIMER Master_TIMER compare 2 value register (HRTIMER_MTCMP2V)

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
CMP2VAL[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:0	CMP2VAL[15:0]	Compare 2 value This bit-field contains value to be compared to the counter. This register has a shadow register. If the shadow register is disabled (SHWEN = 0), it holds the content of the active register; otherwise, it holds the content of the shadow register. **Note:** (1) The minimum value must be greater than or equal to 3 $t_{HRTIMER_CK}$. For example: CARL[15:0] >= 0x60 when CNTCKDIV[3:0] = 4'b0000. (2) The maximum value must be less than or equal to 0xFFFF – (1 $t_{HRTIMER_CK}$). For example: CARL[15:0] <= 0xFFDF when CNTCKDIV[3:0] = 4'b0000.

HRTIMER Master_TIMER compare 3 value register (HRTIMER_MTCMP3V)

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
CMP3VAL[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:0	CMP3VAL[15:0]	Compare 3 value This bit-field contains value to be compared to the counter. This register has a shadow register. If the shadow register is disabled (SHWEN = 0), it holds the content of the active register; otherwise, it holds the content of the shadow register. **Note:** (1) The minimum value must be greater than or equal to 3 $t_{HRTIMER_CK}$. For example:

CARL[15:0] >= 0x60 when CNTCKDIV[3:0] = 4'b0000.

(2) The maximum value must be less than or equal to 0xFFFF – (1 $t_{HRTIMER_CK}$). For example: CARL[15:0] <= 0xFFDF when CNTCKDIV[3:0] = 4'b0000.

25.5.2. Slave_TIMERx registers(x=0..7)

HRTIMER Slave_TIMER0 registers base address: 0x4001 5880

HRTIMER Slave_TIMER1 registers base address: 0x4001 5900

HRTIMER Slave_TIMER2 registers base address: 0x4001 5980

HRTIMER Slave_TIMER3 registers base address: 0x4001 5A00

HRTIMER Slave_TIMER4 registers base address: 0x4001 5A80

HRTIMER Slave_TIMER5 registers base address: 0x4001 5B00

HRTIMER Slave_TIMER6 registers base address: 0x4001 6000

HRTIMER Slave_TIMER7 registers base address: 0x4001 6080

HRTIMER Slave_TIMERx control register 0 (HRTIMER_STxCTL0)

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
UPSEL[3:0]				SHWEN	DACTRGS[1:0]		UPBMT	UPBST4	UPBST3	UPBST2	UPBST1	UPBST0	UPRST	UPREP	UPBST5
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
DELCMP3M[1:0]		DELCMP1M[1:0]		SYNISTRT	SYNIRST	RSYNUPD	ALTM[1:0]		BLNMEN	HALFM	CNTRSTM	CTNM	CNTCKDIV[2:0]		
rw		rw		rw	rw	rw	rw		rw	rw	rw	rw	rw		

Bits	Fields	Descriptions
31:28	UPSEL[3:0]	Update event selection This bit-field specifies which the relationship between update events and the DMA mode. 0000: Update event generated independently from DMA mode. 0001: Update event generated when the DMA transfer completed in DMA mode. 0010: Update event generated on the update event following the DMA transfer completed in DMA mode. 0011: Update event generated on the rising edge of STxUPIN0. 0100: Update event generated on the rising edge of STxUPIN1.

		0101: Update event generated on the rising edge of STxUPIN2. 0110: Update event generated on the update event following the rising edge of STxUPIN0. 0111: Update event generated on the update event following the rising edge of STxUPIN1. 1000: Update event generated on the update event following the rising edge of STxUPIN2. Other values are reserved **Note:** (1) The bit-field must reset before writing new value. (2) When UPSEL[3:0] = 4'b0001, 4'b0011, 4'b0100, 4'b0101, it is possible to have multiple concurrent update source. For instance, update by Master_TIMER (UPBMT = 1) and DMA mode.
27	SHWEN	Shadow registers enable 0: The shadow registers are disabled 1: The shadow registers are enabled
26:25	DACTRGS[1:0]	Trigger source to DAC The timer can also generate a DAC trigger event when an update event occurs. This bit-field specifies which trigger source generates the DAC trigger event. 00:No DAC trigger event generated 01: DAC trigger event generated on HRTIMER_DACTRIG0 10: DAC trigger event generated on HRTIMER_DACTRIG1 11: DAC trigger event generated on HRTIMER_DACTRIG2
24	UPBMT	Update by Master_TIMER update event When the bit is set, the Slave_TIMERx(x=0..7) update event are synchronized with Master_TIMER update event and the active registers of them are updated by the Master_TIMER update event 0: The active registers is not update by Master_TIMER. 1: The active registers is update by Master_TIMER.
23	UPBST4	Update by Slave_TIMER4 update event When the bit is set, the Slave_TIMERx(x=0..3,5..7) update event are synchronized with Slave_TIMER4 update event and the active registers of them are updated by the Slave_TIMER4 update event 0: The active registers is not update by Slave_TIMER4. 1: The active registers is update by Slave_TIMER4. **Note:** This bit does not exist in Slave_TIMER4.
22	UPBST3	Update by Slave_TIMER3 update event When the bit is set, the Slave_TIMERx(x=0..2,4..7) update event are synchronized with Slave_TIMER3 update event and the active registers of them are updated by the Slave_TIMER3 update event 0: The active registers is not update by Slave_TIMER3.

		1: The active registers is update by Slave_TIMER3. **Note:** This bit does not exist in Slave_TIMER3.
21	UPBST2	Update by Slave_TIMER2 update event When the bit is set, the Slave_TIMERx(x=0,1,3..7) update event are synchronized with Slave_TIMER2 update event and the active registers of them are updated by the Slave_TIMER2 update event 0: The active registers is not update by Slave_TIMER2. 1: The active registers is update by Slave_TIMER2. **Note:** This bit does not exist in Slave_TIMER2.
20	UPBST1	Update by Slave_TIMER1 update event When the bit is set, the Slave_TIMERx(x=0,2..7) update event are synchronized with Slave_TIMER1 update event and the active registers of them are updated by the Slave_TIMER1 update event 0: The active registers is not update by Slave_TIMER1. 1: The active registers is update by Slave_TIMER1. **Note:** This bit does not exist in Slave_TIMER1.
19	UPBST0	Update by Slave_TIMER0 update event When the bit is set, the Slave_TIMERx(x=1..7) update event are synchronized with Slave_TIMER0 update event and the active registers of them are updated by the Slave_TIMER0 update event 0: The active registers is not update by Slave_TIMER0. 1: The active registers is update by Slave_TIMER0. **Note:** This bit does not exist in Slave_TIMER0.
18	UPRST	Update event generated by reset event This bit specifies whether counter reset event or roll-over event can generate update event. 0: Update event generated by reset event or roll-over event disable 1: Update event generated by reset event or roll-over event enable
17	UPREP	Update event generated by repetition event This bit specifies whether repetition event can generate update event. 0: Update event generated by repetition event disable 1: Update event generated by repetition event enable
16	UPBST5	Update by Slave_TIMER5 update event When the bit is set, the Slave_TIMERx(x=0..4,6,7) update event are synchronized with Slave_TIMER5 update event and the active registers of them are updated by the Slave_TIMER5 update event 0: The active registers is not update by Slave_TIMER5. 1: The active registers is update by Slave_TIMER5. Note: This bit does not exist in Slave_TIMER5.
15:14	DELCMP3M[1:0]	Compare 3 delayed mode

		00: Compare 3 delayed mode disable. Compare match occurs as soon as the counter equals the value of compare 3 active register. 01: Compare 3 delayed mode 0. After a capture 1 event, the recalculated value of compare 3 is: (compare 3 active register value + capture 1 value). Compare match occurs as soon as the counter equals the recalculated value. 10: Compare 3 delayed mode 1. After a capture 1 event or compare 0 event, the recalculated value of compare 3 is: (compare 3 active register value + capture 1 value for capture 1 event, or compare 3 active register value + compare 0 value for compare 0 event). Compare match occurs as soon as the counter equals the recalculated value. 11: Compare 3 delayed mode 2. After a capture 1 event or compare 2 event, the recalculated value of compare 3 is: (compare 3 active register value + capture 1 value for capture 1 event, or compare 3 active register value + compare 2 value for compare 2 event). Compare match occurs as soon as the counter equals the recalculated value. **Note:** This bit-field must not be modified once the counter is enabled (STxCEN =1 in HRTIMER_MTCTL0 register).
13:12	DELCMP1M[1:0]	Compare 1 delayed mode 00: Compare 1 delayed mode disable. Compare match occurs as soon as the counter equals the value of compare 1 active register. 01: Compare 1 delayed mode 0. After a capture 0 event, the recalculated value of compare 1 is: (compare 1 active register value + capture 0 value). Compare match occurs as soon as the counter equals the recalculated value. 10: Compare 1 delayed mode 1. After a capture 0 event or compare 0 event, the recalculated value of compare 1 is: (compare 1 active register value + capture 0 value for capture 0 event, or compare 1 active register value + compare 0 active register value for compare 0 event). Compare match occurs as soon as the counter equals the recalculated value. 11: Compare 1 delayed mode 2. After a capture 0 event or compare 2 event, the recalculated value of compare 1 is:(compare 1 active register value + capture 0 value for capture 0 event, or compare 1 active register value + compare 2 value for compare 2 event). Compare match occurs as soon as the counter equals the recalculated value. **Note:** This bit-field must not be modified once the counter is enabled (STxCEN =1 in HRTIMER_MTCTL0 register).
11	SYNISTRT	Synchronous input start timer This bit specifies whether the synchronous input signal can start the counter 0: The synchronous input signal cannot start the counter 1: The synchronous input signal can start the counter
10	SYNIRST	Synchronous input reset timer This bit specifies whether the synchronous input signal can reset the counter 0: The synchronous input signal cannot reset the counter

		1: The synchronous input signal can reset the counter
9	RSYNUPD	Re-synchronized update This bit specifies whether update source coming outside from the timing unit must be synchronized: 0: The update coming from other timers (when UPBMT, UPBST0, UPBST1, UPBST2, UPBST3, UPBST4, UPBST5, UPBST6, UPBST7 bit is set) or from a software update (STxSUP bit) is taken into account immediately 1: The update coming from other timers (when UPBMT, UPBST0, UPBST1, UPBST2, UPBST3, UPBST4, UPBST5, UPBST6, UPBST7 bit is set) or from a software update (STxSUP bit) is taken into account on the following reset/roll-over event. Note: This bit is significant only when UPSEL [3:0] = 0000, it is ignored otherwise
8:7	ALTM	Alternate mode This bitfield is significant only when the HALFM bit is reset. It enables the alternate mode. 00: Alternate mode disabled 01: Triple alternate mode: when HRTIMER_STxCAR register is written, the HRTIMER_STxCMP0V active register is automatically updated with HRTIMER_STxCAR/3 value, and the HRTIMER_STxCMP1V active register is automatically updated with 2x (HRTIMER_STxCAR/3) value. 10: Quad alternate mode: when HRTIMER_STxCAR register is written, the HRTIMER_STxCMP0V active register is automatically updated with HRTIMER_STxCAR/4 value, the HRTIMER_STxCMP1V active register is automatically updated with HRTIMER_STxCAR/2 value and the HRTIMER_STxCMP2V active register is automatically updated with 3x (HRTIMER_STxCAR/4) value. 11: Alternate mode disabled
6	BLNMEN	Balanced mode enable 0: Balanced mode disable 1: Balanced mode enable **Note:** This bit-field must not be modified once the counter is enabled (STxCEN =1 in HRTIMER_MTCTL0 register).
5	HALFM	Half mode When the bit is set, HRTIMER_STxCMP0V active register is always the half of counter auto-reload value (HRTIMER_STxCAR). 0: Half mode disable. 1: Half mode enable.
4	CNTRSTM	Counter reset mode This bit defines the behavior of the timer counter in single pulse mode. 0: The counter can be reset only if it stops (period elapsed)

1: The counter can be reset at any time (running or stopped).

3	CTNM	Continuous mode. 0: Single pulse mode. The counter stops by hardware when it reaches the HRTIMER_STxCAR value. 1: Continuous mode. The counter rolls over to zero and count continuously when it reaches the HRTIMER_STxCAR value
2:0	CNTCKDIV[2:0]	Counter clock division This bit-field can be configured by software to specify division ratio between the high resolution clock (HRTIMER_HPCK) and the counter clock (HRTIMER_PSCCK). $f_{HRTIMER_PSCCK} = f_{HRTIMER_HPCK}/2^{CNTCKDIV[2:0]}$. 000: $f_{HRTIMER_PSCCK} = f_{HRTIMER_HPCK}$ 001: $f_{HRTIMER_PSCCK} = f_{HRTIMER_HPCK}/2$ 010: $f_{HRTIMER_PSCCK} = f_{HRTIMER_HPCK}/4$ 011: $f_{HRTIMER_PSCCK} = f_{HRTIMER_HPCK}/8$ 100: $f_{HRTIMER_PSCCK} = f_{HRTIMER_HPCK}/16$ 101: $f_{HRTIMER_PSCCK} = f_{HRTIMER_HPCK}/32$ 110: $f_{HRTIMER_PSCCK} = f_{HRTIMER_HPCK}/64$ 111: $f_{HRTIMER_PSCCK} = f_{HRTIMER_HPCK}/128$ **Note:** This bit-field must not be modified once the counter is enabled (STxCEN =1 in HRTIMER_MTCTL0 register).

HRTIMER Slave_TIMERx interrupt flag register (HRTIMER_STxINTF)

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
DIR	Reserved									CH1F	CH0F	CH1SF	CH0SF	BLNIF	CBLNF
r										r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	DLYIIF	RSTIF	CH1ONAIF	CH1OAIF	CH0ONAIF	CH0OAIF	CAP1IF	CAP0IF	UPIF	Reserved	REPIF	CMP3IF	CMP2IF	CMP1IF	CMP0IF
	r	r	r	r	r	r	r	r	r		r	r	r	r	r

Bits	Fields	Descriptions
31	DIR	Direction 0: Count up 1: Count down
30:22	Reserved	Must be kept at reset value
21	CH1F	Channel 1 output flag

This bit indicates the output level state of channel 1.
0: Channel 1 outputs inactive level.
1: Channel 1 outputs active level.

20 CH0F
Channel 0 output flag
This bit indicates the output level state of channel 0.
0: Channel 0 outputs inactive level.
1: Channel 0 outputs active level.

19 CH1SF
Channel 1 output state flag
This bit indicates the output state of channel 1. when the delayed IDLE mode entry occurred, this bit is updated only when any new delayed protection entry.
0: Channel 1 outputs inactive level.
1: Channel 1 outputs active level.

18 CH0SF
Channel 0 output state flag
This bit indicates the output state of channel 0. when the delayed IDLE mode entry occurred, this bit is updated only when any new delayed protection entry.
0: Channel 0 outputs inactive level.
1: Channel 0 outputs active level.

17 BLNIF
Balanced IDLE flag
This bit indicates which channel is outputting the signal when balanced IDLE entry occurred.
0: Channel 0 outputs the signal and channel 1 outputs inactive level when balanced IDLE entry occurred.
1: Channel 1 outputs the signal and channel 0 outputs inactive level when balanced IDLE entry occurred.

16 CBLNF
Current balanced flag
This bit is only valid in balanced mode. This bit indicates which channel is currently outputting the signal.
0: Channel 0 outputs the signal and channel 1 outputs inactive level.
1: Channel 1 outputs the signal and channel 0 outputs inactive level.

15 Reserved
Must be kept at reset value

14 DLYIIF
Delayed IDLE mode entry interrupt flag
This flag is set by hardware when delayed IDLE or balanced IDLE mode entry occurred.
0: No counter delayed IDLE mode entry interrupt occurred
1: Delayed IDLE mode entry interrupt occurred

13 RSTIF
Counter reset interrupt flag
This flag is set by hardware when counter reset or roll-over event occurred.
0: No counter reset or roll-over interrupt occurred
1: Counter reset or roll-over interrupt occurred

12	CH1ONAIF	Channel 1 output inactive interrupt flag Refer to CH0ONAIF description.
11	CH1OAIF	Channel 1 output active interrupt flag Refer to CH0OAIF description.
10	CH0ONAIF	Channel 0 output inactive interrupt flag This flag is set by hardware when channel 0 output inactive (C0OPRE from active to inactive) occurs. 0: No channel 0 output inactive interrupt occurred 1: Channel 0 output inactive interrupt occurred
9	CH0OAIF	Channel 0 output active interrupt flag This flag is set by hardware when channel 0 output active (C1OPRE from inactive to active) occurs. 0: No channel 0 output active interrupt occurred 1: Channel 0 output active interrupt occurred
8	CAP1IF	Capture 1 interrupt flag This flag is set by hardware when capture 1 event occurs. 0: No capture 1 interrupt occurred 1: Capture 1 interrupt occurred
7	CAP0IF	Capture 0 interrupt flag This flag is set by hardware when capture 0 event occurs. 0: No capture 0 interrupt occurred 1: Capture 0 interrupt occurred
6	UPIF	Update interrupt flag This flag is set by hardware when an update event occurs. 0: No update interrupt occurred 1: Update interrupt occurred
5	Reserved	Must be kept at reset value
4	REPIF	Repetition interrupt flag This flag is set by hardware when a repetition event occurs. 0: No repetition interrupt occurred 1: Repetition interrupt occurred
3	CMP3IF	Compare 3 interrupt flag This flag is set by hardware when a compare 3 event occurs. 0: No compare 3 interrupt occurred 1: Compare 3 interrupt occurred
2	CMP2IF	Compare 2 interrupt flag This flag is set by hardware when a compare 2 event occurs. 0: No compare 2 interrupt occurred

		1: Compare 2 interrupt occurred
1	CMP1IF	Compare 1 interrupt flag This flag is set by hardware when a compare 1 event occurs. 0: No compare 1 interrupt occurred 1: Compare 1 interrupt occurred
0	CMP0IF	Compare 0 interrupt flag This flag is set by hardware when a compare 0 event occurs. 0: No compare 0 interrupt occurred 1: Compare 0 interrupt occurred

HRTIMER Slave_TIMERx interrupt flag clear register (HRTIMER_STxINTC)

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	DLYIIFC	RSTIFC	CH1ONAIFC	CH1OAIFC	CH0ONAIFC	CH0OAIFC	CAP1IFC	CAP0IFC	UPIFC	Reserved	REPIFC	CMP3IFC	CMP2IFC	CMP1IFC	CMP0IFC
	w	w	w	w	w	w	w	w	w		w	w	w	w	w

Bits	Fields	Descriptions
31:15	Reserved	Must be kept at reset value
14	DLYIIFC	Clear delayed IDLE mode entry interrupt flag 0: No effect 1: Clear delayed IDLE state mode entry interrupt flag (DLYIIF in HRTIMER_STxINTF register)
13	RSTIFC	Clear counter reset interrupt flag 0: No effect 1: Clear counter reset interrupt flag (RSTIF in HRTIMER_STxINTF register)
12	CH1ONAIFC	Clear channel 1 output inactive interrupt flag Clear CH1ONAIF in HRTIMER_STxINTF register. Refer to CH0ONAIFC description.
11	CH1OAIFC	Clear channel 1 output active interrupt flag Clear CH1OAIF in HRTIMER_STxINTF register. Refer to CH0OAIFC description.

10	CH0ONAIFC	Clear channel 0 output inactive interrupt flag 0: No effect 1: Clear channel 0 output inactive interrupt flag (CH0ONAIF in HRTIMER_STxINTF register)
9	CH0OAIFC	Clear channel 0 output active interrupt flag 0: No effect 1: Clear channel 0 output inactive interrupt flag (CH0OAIF in HRTIMER_STxINTF register)
8	CAP1IFC	Clear capture 1 interrupt flag 0: No effect 1: Clear capture 1 interrupt flag (CAP1IF in HRTIMER_STxINTF register)
7	CAP0IFC	Clear capture 0 interrupt flag 0: No effect 1: Clear capture 0 interrupt flag (CAP0IF in HRTIMER_STxINTF register)
6	UPIFC	Clear update interrupt flag 0: No effect 1: Clear update interrupt flag (UPIF in HRTIMER_STxINTF register)
5	Reserved	Must be kept at reset value
4	REPIFC	Clear repetition interrupt flag 0: No effect 1: Clear repetition interrupt flag (REPIF in HRTIMER_STxINTF register)
3	CMP3IFC	Clear compare 3 interrupt flag 0: No effect 1: Clear compare 3 interrupt flag (CMP3IF in HRTIMER_STxINTF register)
2	CMP2IFC	Clear compare 2 interrupt flag 0: No effect 1: Clear compare 2 interrupt flag (CMP2IF in HRTIMER_STxINTF register)
1	CMP1IFC	Clear compare 1 interrupt flag 0: No effect 1: Clear compare 1 interrupt flag (CMP1IF in HRTIMER_STxINTF register)
0	CMP0IFC	Clear compare 0 interrupt flag 0: No effect 1: Clear compare 0 interrupt flag (CMP0IF in HRTIMER_STxINTF register)

HRTIMER Slave_TIMERx DMA and interrupt enable register (HRTIMER_STxDMAINTEN)

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	DLYIDEN	RSTDEN	CH1ONADEN	CH1OADEN	CH0ONADEN	CH0OADEN	CAP1DEN	CAP0DEN	UPDEN	Reserved	REPDEN	CMP3DEN	CMP2DEN	CMP1DEN	CMP0DEN
	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
Reserved	DLYIIE	RSTIE	CH1ONAIE	CH1OAIE	CH0ONAIE	CH0OAIE	CAP1IE	CAP0IE	UPIE	Reserved	REPIE	CMP3IE	CMP2IE	CMP1IE	CMP0IE
	rw	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	DLYIDEN	Delayed IDLE mode entry DMA request enable 0: disabled 1: enabled
29	RSTDEN	Counter reset DMA request enable 0: disabled 1: enabled
28	CH1ONADEN	Channel 1 output inactive DMA request enable Refer to CH0ONADEN description.
27	CH1OADEN	Channel 1 output active DMA request enable Refer to CH0OADEN description.
26	CH0ONADEN	Channel 0 output inactive DMA request enable 0: disabled 1: enabled
25	CH0ADEN	Channel 0 output active DMA request enable 0: disabled 1: enabled
24	CAP1DEN	Capture 1 DMA request enable 0: disabled 1: enabled
23	CAP0DEN	Capture 0 DMA request enable 0: disabled 1: enabled
22	UPDEN	Update DMA request enable 0: disabled 1: enabled

21	Reserved	Must be kept at reset value
20	REPDEN	Repetition DMA request enable 0: disabled 1: enabled
19	CMP3DEN	Compare 3 DMA request enable 0: disabled 1: enabled
18	CMP2DEN	Compare 2 DMA request enable 0: disabled 1: enabled
17	CMP1DEN	Compare 1 DMA request enable 0: disabled 1: enabled
16	CMP0DEN	Compare 0 DMA request enable 0: disabled 1: enabled
15	Reserved	Must be kept at reset value
14	DLYIIE	Delayed IDLE mode entry interrupt enable 0: disabled 1: enabled
13	RSTIE	Counter reset interrupt enable 0: disabled 1: enabled
12	CH1ONAIE	Channel 1 output inactive interrupt enable Refer to CH0ONAIE description.
11	CH1OAIE	Channel 1 output active interrupt enable Refer to CH0OAIE description.
10	CH0ONAIE	Channel 0 output inactive interrupt enable 0: disabled 1: enabled
9	CH0OAIE	Channel 0 output active interrupt enable 0: disabled 1: enabled
8	CAP1IE	Capture 1 interrupt enable 0: disabled 1: enabled

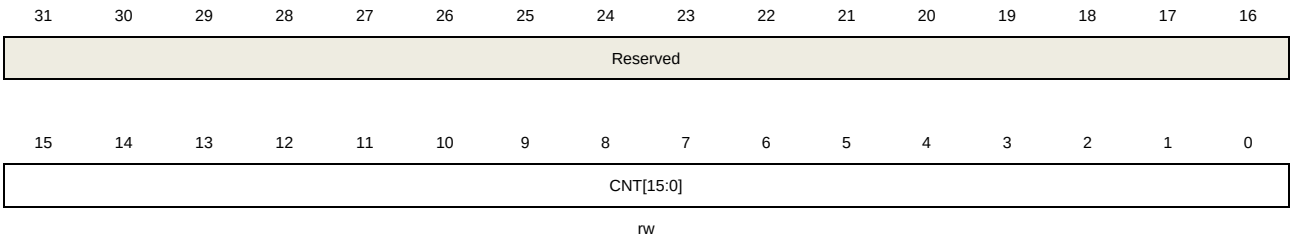
7	CAP0IE	Capture 0 interrupt enable 0: disabled 1: enabled
6	UPIE	Update interrupt enable 0: disabled 1: enabled
5	Reserved	Must be kept at reset value
4	REPIE	Repetition interrupt enable 0: disabled 1: enabled
3	CMP3IE	Compare 3 interrupt enable 0: disabled 1: enabled
2	CMP2IE	Compare 2 interrupt enable 0: disabled 1: enabled
1	CMP1IE	Compare 1 interrupt enable 0: disabled 1: enabled
0	CMP0IE	Compare 0 interrupt enable 0: disabled 1: enabled

HRTIMER Slave_TIMERx counter register (HRTIMER_STxCNT)

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.
15:0	CNT[15:0]	The current counter value.

Writing to it can change the value of the counter only when the Slave_TIMERx is stopped (STxCEN = 0 in HRTIMER_MTCTL0 register).

Note:

(1) For counter clock division below 64 (CNTCKDIV[3:0] < 5), the least significant bits of the counter are not significant. They cannot be written and read 0.

(2) If the value written to this bit-field is above the HRTIM_MPER register value, the behavior of the timer is unpredictable.

HRTIMER Slave_TIMERx counter auto reload register (HRTIMER_STxCAR)

Address offset: 0x14

Reset value: 0x0000 FFDF

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

31	30	29	28	27	26	25	24	23	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. This register has a shadow register. If the shadow register is disabled (SHWEN = 0), it holds the content of the active register; otherwise, it holds the content of the shadow register. **Note:** (1) The minimum value must be greater than or equal to 3 $t_{HRTIMER_CK}$. For example: CARL[15:0] >= 0x60 when CNTCKDIV[3:0] = 4'b0000. (2) The maximum value must be less than or equal to 0xFFFF – (1 $t_{HRTIMER_CK}$). For example: CARL[15:0] <= 0xFFDF when CNTCKDIV[3:0] = 4'b0000.

HRTIMER Slave_TIMERx counter repetition register (HRTIMER_STxCREP)

Address offset: 0x18

Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	CREP[7:0]	Counter repetition value This bit-field specifies the repetition event generation rate. When the repetition counter had count down to zero, the coming roll-over event in continuous mode or reset event will generate a repetition event. This register has a shadow register. If the shadow register is disabled (SHWEN = 0), it holds the content of the active register; otherwise, it holds the content of the shadow register.

HRTIMER Slave_TIMERx compare 0 value register (HRTIMER_STxCMP0V)

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
CMP0VAL[15:0]															
rw															

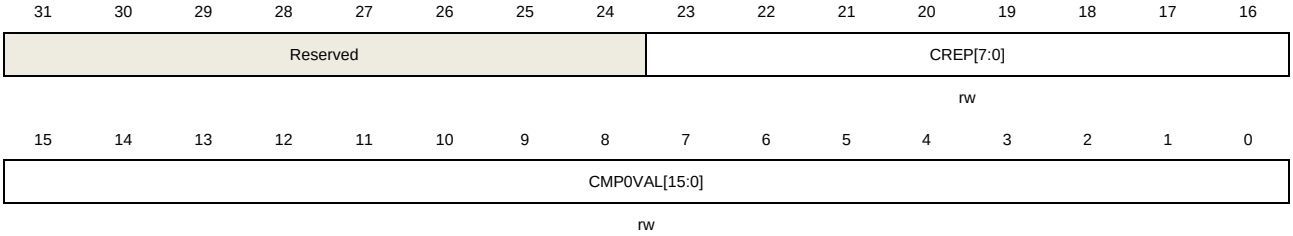
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:0	CMP0VAL[15:0]	Compare 0 value This bit-field contains value to be compared to the counter. This register has a shadow register. If the shadow register is disabled (SHWEN = 0), it holds the content of the active register; otherwise, it holds the content of the shadow register. **Note:** (1) The minimum value must be greater than or equal to 3 $t_{HRTIMER_CK}$. For example: CARL[15:0] >= 0x60 when CNTCKDIV[3:0] = 4'b0000. (2) The maximum value must be less than or equal to 0xFFFF – (1 $t_{HRTIMER_CK}$). For example: CARL[15:0] <= 0xFFDF when CNTCKDIV[3:0] = 4'b0000.

HRTIMER Slave_TIMERx compare 0 composite register (HRTIMER_STxCMP0CP)

Address offset: 0x20

Reset value: 0x0000 0000

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



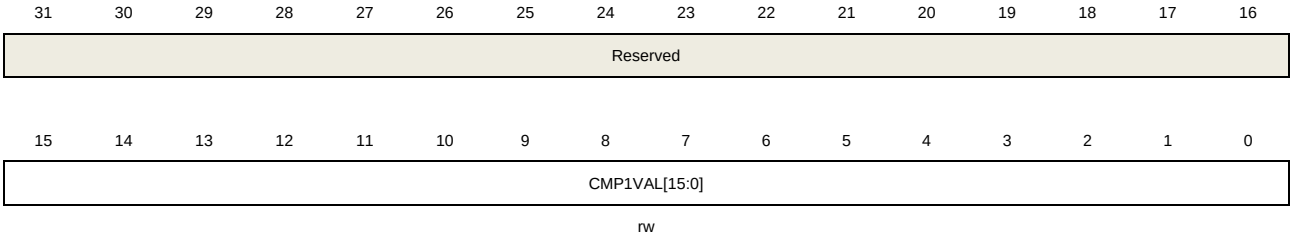
Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value
23:16	CREP[7:0]	Counter repetition value This bit-field is an alias from the CREP[7:0] in the HRTIMER_STxCREP
15:0	CMP0VAL[15:0]	Compare 0 value This bit-field is an alias from the CMP0VAL[15:0] in the HRTIMER_STxCMP0V register

HRTIMER Slave_TIMERx compare 1 value register (HRTIMER_STxCMP1V)

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	CMP1VAL[15:0]	Compare 1 value This bit-field contains value to be compared to the counter. This register has a shadow register. If the shadow register is disabled (SHWEN = 0), it holds the content of the active register; otherwise, it holds the content of the shadow register. In delayed mode, the active register can be recalculated.

Note:

(1) The minimum value must be greater than or equal to 3 $t_{HRTIMER_CK}$. For example: CARL[15:0] >= 0x60 when CNTCKDIV[3:0] = 4'b0000.

(2) The maximum value must be less than or equal to 0xFFFF – (1 $t_{HRTIMER_CK}$). For example: CARL[15:0] <= 0xFFDF when CNTCKDIV[3:0] = 4'b0000.

HRTIMER Slave_TIMERx compare 2 value register (HRTIMER_STxCMP2V)

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
CMP2VAL[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:0	CMP2VAL[15:0]	Compare 2 value This bit-field contains value to be compared to the counter. This register has a shadow register. If the shadow register is disabled (SHWEN = 0), it holds the content of the active register; otherwise, it holds the content of the shadow register. **Note:** (1) The minimum value must be greater than or equal to 3 $t_{HRTIMER_CK}$. For example: CARL[15:0] >= 0x60 when CNTCKDIV[3:0] = 4'b0000. (2) The maximum value must be less than or equal to 0xFFFF – (1 $t_{HRTIMER_CK}$). For example: CARL[15:0] <= 0xFFDF when CNTCKDIV[3:0] = 4'b0000.

HRTIMER Slave_TIMERx compare 3 value register (HRTIMER_STxCMP3V)

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
CMP3VAL[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:0	CMP3VAL[15:0]	Compare 3 value This bit-field contains value to be compared to the counter. This register has a shadow register. If the shadow register is disabled (SHWEN = 0), it holds the content of the active register; otherwise, it holds the content of the shadow register. In delayed mode, the active register can be recalculated. **Note:** (1) The minimum value must be greater than or equal to 3 $t_{HRTIMER_CK}$. For example: CARL[15:0] >= 0x60 when CNTCKDIV[3:0] = 4'b0000. (2) The maximum value must be less than or equal to 0xFFFF – (1 $t_{HRTIMER_CK}$). For example: CARL[15:0] <= 0xFFDF when CNTCKDIV[3:0] = 4'b0000.

HRTIMER Slave_TIMERx capture 0 value register (HRTIMER_STxCAP0V)

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															DIR
															r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAP0VAL[15:0]															
r															

Bits	Fields	Descriptions
31:17	Reserved	Must be kept at reset value
16	DIR	Slave_TIMERx capture 0 direction status This register holds the counting direction value when the capture 0 event occurred: 0: timer is up-counting 1: timer is down-counting In up-counting mode (CAM bit reset), the DIR bit is always read as 0.
15:0	CAP0VAL[15:0]	Capture 0 value This bit-field indicates the counter value corresponding to the last capture event. And this bit-field is read-only. **Note:** For counter clock division below 64 (CNTCKDIV[3:0] < 5), the least significant bits of the counter are not significant. They cannot be written and read 0.

HRTIMER Slave_TIMERx capture 1 value register (HRTIMER_STxCAP1V)

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															DIR
															r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAP1VAL[15:0]															
r															

Bits	Fields	Descriptions
31:17	Reserved	Must be kept at reset value
16	DIR	Slave_TIMERx capture 1 direction status This register holds the counting direction value when the capture 1 event occurred: 0: timer is up-counting 1: timer is down-counting In up-counting mode (CAM bit reset), the DIR bit is always read as 0.
15:0	CAP1VAL[15:0]	Capture 1 value This bit-field indicates the counter value corresponding to the last capture event. And this bit-field is read-only. **Note:** For counter clock division below 32 (CNTCKDIV[2:0] < 5), the least significant bits of the counter are not significant. They cannot be written and read 0.

HRTIMER Slave_TIMERx dead-time control register (HRTIMER_STxDTCTL)

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
DTFSVPROT	DTFSPROT	Reserved				DTFS	DTFCFG[8:0]								
rwo	rwo					rw	rw								

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DTRSVPROT	DTRSPROT	DTGCKDIV[3:0]				DTRS	DTRCFG[8:0]								
rwo	rwo	rw				rw	rw								

Bits	Fields	Descriptions
31	DTFSVPROT	Dead-time falling edge protection for value and sign

		This bit-field specifies the write protection for dead-time falling edge (value and sign). 0: Protect disable. DTFS and DTFCFG[15:0] are writable. 1: Protect enable. DTFS and DTFCFG[15:0] are read-only. **Note:** (1) The bit-field DTFCFG[15:9] is in HRTIMER_STxACTL register.
30	DTFSPROT	Dead-time falling edge protection for sign This bit-field specifies the write protection for dead-time falling edge (only sign). 0: protect disable. DTFS in HRTIMER_STxDTCTL register is writable. 1: protect enable. DTFS in HRTIMER_STxDTCTL register is read-only.
29:26	Reserved	Must be kept at reset value
25	DTFS	The sign of falling edge dead-time value 0: The sign of falling edge dead-time value is positive. 1: The sign of falling edge dead-time value is negative. **Note:** This bit cannot be modified when DTFSPROT or DTFSVPROT bit in HRTIMER_STxDTCTL register is set.
24:16	DTFCFG[8:0]	Falling edge dead-time value This bit-field controls the dead-time value of the following a falling edge of output prepare signal (OyPRE,y=0,1). DTFvalue = DTFCFG[15:0]x $t_{HRTIMER_DTGCK}$, $t_{HRTIMER_DTGCK}$ = 1/ $f_{HRTIMER_DTGCK}$. Writing this bit-field can change the low 9-bits of DTFCFG[15:0]. **Note:** (1) The bit-field DTFCFG[15:9] is in HRTIMER_STxACTL register. (2) This bit-field cannot be modified when DTFSVPROT bit in HRTIMER_STxDTCTL register is set.
15	DTRSVPROT	Dead-time rising edge protection for value and sign This bit-field specifies the write protection for dead-time rising edge (value and sign). 0: Protect disable. DTRS and DTRCFG[15:0] register are writable. 1: Protect enable. DTRS and DTRCFG[15:0] are read-only. **Note:** (1) The bit-field DTRCFG[15:9] is in HRTIMER_STxACTL register.
14	DTRSPROT	Dead-time rising edge protection for sign This bit-field specifies the write protection for dead-time rising edge (only sign). 0: protect disable. DTRS in HRTIMER_STxDTCTL register is writable. 1: protect enable. DTRS in HRTIMER_STxDTCTL register is read-only.
13:10	DTGCKDIV[3:0]	Dead time generator clock division This bit-field can be configured by software to specify division ratio between the HRTIMER clock (HRTIMER_CK) and the dead-time generator clock (HRTIMER_DTGCK). When DTGCKDIV[3] is '0', $f_{HRTIMER_DTGCK} = (8*f_{HRTIMER_CK})/2^{DTGCKDIV[2:0]}$.

When DTGCKDIV[3] is '1', $f_{HRTIMER_DTGCK} = 2^{(DTGCKDIV[2:0]+4)} * f_{HRTIMER_CK}$

0000: $f_{HRTIMER_DTGCK} = 8 * f_{HRTIMER_CK}$

0001: $f_{HRTIMER_DTGCK} = (8 * f_{HRTIMER_CK})/2$

0010: $f_{HRTIMER_DTGCK} = (8 * f_{HRTIMER_CK})/4$

0011: $f_{HRTIMER_DTGCK} = (8 * f_{HRTIMER_CK})/8$

0100: $f_{HRTIMER_DTGCK} = (8 * f_{HRTIMER_CK})/16$

0101: $f_{HRTIMER_DTGCK} = (8 * f_{HRTIMER_CK})/32$

0110: $f_{HRTIMER_DTGCK} = (8 * f_{HRTIMER_CK})/64$

0111: $f_{HRTIMER_DTGCK} = (8 * f_{HRTIMER_CK})/128$

1000: $f_{HRTIMER_DTGCK} = 16 * f_{HRTIMER_CK}$

1001: $f_{HRTIMER_DTGCK} = 32 * f_{HRTIMER_CK}$

Note: This bit-field cannot be modified when any of the protect bits is set (DTFSPROT , DTFSVPROT, DTRSPROT and DTRSVPROT).

9 DTRS

The sign of rising edge dead-time value

0: The sign of rising edge dead-time value is positive.

1: The sign of rising edge dead-time value is negative.

Note: This bit cannot be modified when DTRSPROT or DTRSVPROT bit in HRTIMER_STxDTCTL register is set.

8:0 DTRCFG[8:0]

Rising edge dead-time value

This bit-field controls the dead-time value of the following a rising edge of output prepare signal (OyPRE,y=0,1).

DTRvalue = DTRCFG[15:0]x $t_{HRTIMER_DTGCK}$, $t_{HRTIMER_DTGCK} = 1/ f_{HRTIMER_DTGCK}$.

Writing this bit-field can change the low 9-bits of DTRCFG[15:0].

Note:

(1) The bit-field DTRCFG[15:9] is in HRTIMER_STxACTL register.

(2) This bit-field cannot be modified when DTRSVPROT bit in HRTIMER_STxDTCTL register is set.

HRTIMER Slave_TIMERx channel 0 set request register (HRTIMER_STxCH0SET)

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
CH0SUP	CH0SEXEV9	CH0SEXEV8	CH0SEXEV7	CH0SEXEV6	CH0SEXEV5	CH0SEXEV4	CH0SEXEV3	CH0SEXEV2	CH0SEXEV1	CH0SEXEV0	CH0SSTEV8	CH0SSTEV7	CH0SSTEV6	CH0SSTEV5	CH0SSTEV4
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
CH0SSTEV3	CH0SSTEV2	CH0SSTEV1	CH0SSTEV0	CH0SMTCMP3	CH0SMTCMP2	CH0SMTCMP1	CH0SMTCMP0	CH0SMTPER	CH0SCMP3	CH0SCMP2	CH0SCMP1	CH0SCMP0	CH0SPER	CH0SRST	CH0SSEV

rw rw rw rw rw rw rw rw rw rw rw rw rw rw rw rw

Bits	Fields	Descriptions
31	CH0SUP	Update event generates channel 0 “set request” When this bit is set, update event can generate “set request”. 0: The event cannot generate “set request”. 1: The event can generate “set request”.
30	CH0SEXEV9	External event 9 generates channel 0 “set request” Refer to CH0SEXEV0 description.
29	CH0SEXEV8	External event 8 generates channel 0 “set request” Refer to CH0SEXEV0 description.
28	CH0SEXEV7	External event 7 generates channel 0 “set request” Refer to CH0SEXEV0 description.
27	CH0SEXEV6	External event 6 generates channel 0 “set request” Refer to CH0SEXEV0 description.
26	CH0SEXEV5	External event 5 generates channel 0 “set request” Refer to CH0SEXEV0 description.
25	CH0SEXEV4	External event 4 generates channel 0 “set request” Refer to CH0SEXEV0 description.
24	CH0SEXEV3	External event 3 generates channel 0 “set request” Refer to CH0SEXEV0 description.
23	CH0SEXEV2	External event 2 generates channel 0 “set request” Refer to CH0SEXEV0 description.
22	CH0SEXEV1	External event 1 generates channel 0 “set request” Refer to CH0SEXEV0 description.
21	CH0SEXEV0	External event 0 generates channel 0 “set request” When this bit is set, external event 0 can generate “set request”. 0: The event cannot generate “set request”. 1: The event can generate “set request”.
20	CH0SSTEV8	Slave_TIMERx interconnection event 8 generates channel 0 “set request” Refer to CH0SSTEV0 description.
19	CH0SSTEV7	Slave_TIMERx interconnection event 7 generates channel 0 “set request” Refer to CH0SSTEV0 description.
18	CH0SSTEV6	Slave_TIMERx interconnection event 6 generates channel 0 “set request” Refer to CH0SSTEV0 description.
17	CH0SSTEV5	Slave_TIMERx interconnection event 5 generates channel 0 “set request”

		Refer to CH0SSTEV0 description.
16	CH0SSTEV4	Slave_TIMERx interconnection event 4 generates channel 0 “set request” Refer to CH0SSTEV0 description.
15	CH0SSTEV3	Slave_TIMERx interconnection event 3 generates channel 0 “set request” Refer to CH0SSTEV0 description.
14	CH0SSTEV2	Slave_TIMERx interconnection event 2 generates channel 0 “set request” Refer to CH0SSTEV0 description.
13	CH0SSTEV1	Slave_TIMERx interconnection event 1 generates channel 0 “set request” Refer to CH0SSTEV0 description.
12	CH0SSTEV0	Slave_TIMERx interconnection event 0 generates channel 0 “set request” When this bit is set, Slave_TIMERx interconnection event 0 can generate “set request”. 0: The event cannot generate “set request”. 1: The event can generate “set request”.
11	CH0SMTCMP3	Master_TIMER compare 3 event generates channel 0 “set request” When this bit is set, Master_TIMER compare 3 event can generate “set request”. 0: The event cannot generate “set request”. 1: The event can generate “set request”.
10	CH0SMTCMP2	Master_TIMER compare 2 event generates channel 0 “set request” When this bit is set, Master_TIMER compare 2 event can generate “set request”. 0: The event cannot generate “set request”. 1: The event can generate “set request”.
9	CH0SMTCMP1	Master_TIMER compare 1 event generates channel 0 “set request” When this bit is set, Master_TIMER compare 1 event can generate “set request”. 0: The event cannot generate “set request”. 1: The event can generate “set request”.
8	CH0SMTCMP0	Master_TIMER compare 0 event generates channel 0 “set request” When this bit is set, Master_TIMER compare 0 event can generate “set request”. 0: The event cannot generate “set request”. 1: The event can generate “set request”.
7	CH0SMTPER	Master_TIMER period event generates channel 0 “set request” In continuous mode, the Master_TIMER counter roll-over event can generate “set request”. In single pulse mode, the Master_TIMER reset event can generate “set request”. 0: The event cannot generate “set request”. 1: The event can generates “set request”.
6	CH0SCMP3	Slave_TIMERx compare 3 event generates channel 0 “set request” When this bit is set, Slave_TIMERx compare 3 event can generate “set request”.

		0: The event cannot generate “set request”. 1: The event can generate “set request”.
5	CH0SCMP2	Slave_TIMERx compare 2 event generates channel 0 “set request” When this bit is set, Slave_TIMERx compare 2 event can generate “set request”. 0: The event cannot generate “set request”. 1: The event can generate “set request”.
4	CH0SCMP1	Slave_TIMERx compare 1 event generates channel 0 “set request” When this bit is set, Slave_TIMERx compare 1 event can generate “set request”. 0: The event cannot generate “set request”. 1: The event can generate “set request”.
3	CH0SCMP0	Slave_TIMERx compare 0 event generates channel 0 “set request” When this bit is set, Slave_TIMERx compare 0 event can generate “set request”. 0: The event cannot generate “set request”. 1: The event can generate “set request”.
2	CH0SPER	Slave_TIMERx period event generates channel 0 “set request” When this bit is set, Slave_TIMERx period event can generate “set request”. 0: The event cannot generate “set request”. 1: The event can generate “set request”.
1	CH0SRST	Slave_TIMERx reset event generates channel 0 “set request” When this bit is set, Slave_TIMERx reset event from synchronous input and software can generate channel 0 “set request”. 0: The event cannot generate “set request”. 1: The event can generate “set request”. **Note:** When this bit is set, the reset of other timers does not affect the output.
0	CH0SSEV	Software event generates channel 0 “set request” This bit is set by software and cleared by hardware automatically. When this bit is set, it can generate channel 0 “set request”. 0: The event cannot generate “set request”. 1: The event can generate “set request”. **Note:** This bit is not preloaded

HRTIMER Slave_TIMERx channel 0 reset request register (HRTIMER_STxCH0RST)

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

CH0RSUP	CH0RSEXEV9	CH0RSEXEV8	CH0RSEXEV7	CH0RSEXEV6	CH0RSEXEV5	CH0RSEXEV4	CH0RSEXEV3	CH0RSEXEV2	CH0RSEXEV1	CH0RSEXEV0	CH0RSSTEV8	CH0RSSTEV7	CH0RSSTEV6	CH0RSSTEV5	CH0RSSTEV4
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
CH0RSSTEV3	CH0RSSTEV2	CH0RSSTEV1	CH0RSSTEV0	CH0RSMTCMP3	CH0RSMTCMP2	CH0RSMTCMP1	CH0RSMTCMP0	CH0RSMTPER	CH0RSCMP3	CH0RSCMP2	CH0RSCMP1	CH0RSCMP0	CH0RSPER	CH0RSRST	CH0RSSEV
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31	CH0RSUP	Update event generates channel 0 “reset request” When this bit is set, update event can generate “reset request”. 0: The event cannot generate “reset request”. 1: The event can generate “reset request”.
30	CH0RSEXEV9	External event 9 generates channel 0 “reset request” Refer to CH0RSEXEV0 description.
29	CH0RSEXEV8	External event 8 generates channel 0 “reset request” Refer to CH0RSEXEV0 description.
28	CH0RSEXEV7	External event 7 generates channel 0 “reset request” Refer to CH0RSEXEV0 description.
27	CH0RSEXEV6	External event 6 generates channel 0 “reset request” Refer to CH0RSEXEV0 description.
26	CH0RSEXEV5	External event 5 generates channel 0 “reset request” Refer to CH0RSEXEV0 description.
25	CH0RSEXEV4	External event 4 generates channel 0 “reset request” Refer to CH0RSEXEV0 description.
24	CH0RSEXEV3	External event 3 generates channel 0 “reset request” Refer to CH0RSEXEV0 description.
23	CH0RSEXEV2	External event 2 generates channel 0 “reset request” Refer to CH0RSEXEV0 description.
22	CH0RSEXEV1	External event 1 generates channel 0 “reset request” Refer to CH0RSEXEV0 description.
21	CH0RSEXEV0	External event 0 generates channel 0 “reset request” When this bit is set, external event 0 can generate channel “reset request”. 0: The event cannot generate “reset request”. 1: The event can generate “reset request”.
20	CH0RSSTEV8	Slave_TIMERx interconnection event 8 generates channel 0 “reset request” Refer to CH0RSSTEV0 description.

19	CH0RSSTEV7	Slave_TIMERx interconnection event 7 generates channel 0 “reset request” Refer to CH0RSSTEV0 description.
18	CH0RSSTEV6	Slave_TIMERx interconnection event 6 generates channel 0 “reset request” Refer to CH0RSSTEV0 description.
17	CH0RSSTEV5	Slave_TIMERx interconnection event 5 generates channel 0 “reset request” Refer to CH0RSSTEV0 description.
16	CH0RSSTEV4	Slave_TIMERx interconnection event 4 generates channel 0 “reset request” Refer to CH0RSSTEV0 description.
15	CH0RSSTEV3	Slave_TIMERx interconnection event 3 generates channel 0 “reset request” Refer to CH0RSSTEV0 description.
14	CH0RSSTEV2	Slave_TIMERx interconnection event 2 generates channel 0 “reset request” Refer to CH0RSSTEV0 description.
13	CH0RSSTEV1	Slave_TIMERx interconnection event 1 generates channel 0 “reset request” Refer to CH0RSSTEV0 description.
12	CH0RSSTEV0	Slave_TIMERx interconnection event 0 generates channel 0 “reset request” When this bit is set, Slave_TIMERx interconnection event 0 can generate channel “reset request”. 0: The event cannot generate “reset request”. 1: The event can generate “reset request”.
11	CH0RSMTCMP3	Master_TIMER compare 3 event generates channel 0 “reset request” When this bit is set, Master_TIMER compare 3 event can generate channel “reset request”. 0: The event cannot generate “reset request”. 1: The event can generate “reset request”.
10	CH0RSMTCMP2	Master_TIMER compare 2 event generates channel 0 “reset request” When this bit is set, Master_TIMER compare 2 event can generate channel “reset request”. 0: The event cannot generate “reset request”. 1: The event can generate “reset request”.
9	CH0RSMTCMP1	Master_TIMER compare 1 event generates channel 0 “reset request” When this bit is set, Master_TIMER compare 1 event can generate channel “reset request”. 0: The event cannot generate “reset request”. 1: The event can generate “reset request”.
8	CH0RSMTCMP0	Master_TIMER compare 0 event generates channel 0 “reset request” When this bit is set, Master_TIMER compare 0 event can generate channel “reset request”. 0: The event cannot generate “reset request”.

		1: The event can generate “reset request”.
7	CH0RSMTPER	Master_TIMER period event generates channel 0 “reset request” In continuous mode, the Master_TIMER counter roll-over event can generate channel “reset request”. In single pulse mode, the Master_TIMER reset event can generate channel “reset request”. 0: The event cannot generate “reset request”. 1: The event can generate “reset request”.
6	CH0RSCMP3	Slave_TIMERx compare 3 event generates channel 0 “reset request” When this bit is set, Slave_TIMERx compare 3 event can generate channel “reset request”. 0: The event cannot generate “reset request”. 1: The event can generate “reset request”.
5	CH0RSCMP2	Slave_TIMERx compare 2 event generates channel 0 “reset request” When this bit is set, Slave_TIMERx compare 2 event can generate channel “reset request”. 0: The event cannot generate “reset request”. 1: The event can generate “reset request”.
4	CH0RSCMP1	Slave_TIMERx compare 1 event generates channel 0 “reset request” When this bit is set, Slave_TIMERx compare 1 event can generate channel “reset request”. 0: The event cannot generate “reset request”. 1: The event can generate “reset request”.
3	CH0RSCMP0	Slave_TIMERx compare 0 event generates channel 0 “reset request” When this bit is set, Slave_TIMERx compare 0 event can generate channel “reset request”. 0: The event cannot generate “reset request”. 1: The event can generate “reset request”.
2	CH0RSPER	Slave_TIMERx period event generates channel 0 “reset request” When this bit is set, Slave_TIMERx period event can generate channel “reset request”. 0: The event cannot generate “reset request”. 1: The event can generate “reset request”.
1	CH0RSRST	Slave_TIMERx reset event generates channel 0 “reset request” When this bit is set, Slave_TIMERx reset event from synchronous input and software can generate channel “reset request”. 0: The event cannot generate “reset request”. 1: The event can generate “reset request”. **Note:** When this bit is set, the reset of other timers does not affect the output.
0	CH0RSSEV	Software event generates channel 0 “reset request”

This bit is set by software and cleared by hardware automatically. When this bit is set, it will generate “reset request”.

0: The event cannot generate “reset request”.

1: The event can generate “reset request”.

Note: This bit is not preloaded

HRTIMER Slave_TIMERx channel 1 set request register (HRTIMER_STxCH1SET)

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
CH1SUP	CH1SEXEV9	CH1SEXEV8	CH1SEXEV7	CH1SEXEV6	CH1SEXEV5	CH1SEXEV4	CH1SEXEV3	CH1SEXEV2	CH1SEXEV1	CH1SEXEV0	CH1SSTEV8	CH1SSTEV7	CH1SSTEV6	CH1SSTEV5	CH1SSTEV4
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
CH1SSTEV3	CH1SSTEV2	CH1SSTEV1	CH1SSTEV0	CH1SMTCMP3	CH1SMTCMP2	CH1SMTCMP1	CH1SMTCMP0	CH1SMTPER	CH1SCMP3	CH1SCMP2	CH1SCMP1	CH1SCMP0	CH1SPER	CH1SRST	CH1SSEV
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31	CH1SUP	Update event generates channel 1 “set request” When this bit is set, update event can generate “set request”. 0: The event cannot generate “set request”. 1: The event can generate “set request”.
30	CH1SEXEV9	External event 9 generates channel 1 “set request” Refer to CH1SEXEV0 description.
29	CH1SEXEV8	External event 8 generates channel 1 “set request” Refer to CH1SEXEV0 description.
28	CH1SEXEV7	External event 7 generates channel 1 “set request” Refer to CH1SEXEV0 description.
27	CH1SEXEV6	External event 6 generates channel 1 “set request” Refer to CH1SEXEV0 description.
26	CH1SEXEV5	External event 5 generates channel 1 “set request” Refer to CH1SEXEV0 description.
25	CH1SEXEV4	External event 4 generates channel 1 “set request” Refer to CH1SEXEV0 description.

24	CH1SEXEV3	External event 3 generates channel 1 “set request” Refer to CH1SEXEV0 description.
23	CH1SEXEV2	External event 2 generates channel 1 “set request” Refer to CH1SEXEV0 description.
22	CH1SEXEV1	External event 1 generates channel 1 “set request” Refer to CH1SEXEV0 description.
21	CH1SEXEV0	External event 0 generates channel 1 “set request” When this bit is set, external event 0 can generate “set request”. 0: The event cannot generate “set request”. 1: The event can generate “set request”.
20	CH1SSTEV8	Slave_TIMERx interconnection event 8 generates channel 1 “set request” Refer to CH1SSTEV0 description.
19	CH1SSTEV7	Slave_TIMERx interconnection event 7 generates channel 1 “set request” Refer to CH1SSTEV0 description.
18	CH1SSTEV6	Slave_TIMERx interconnection event 6 generates channel 1 “set request” Refer to CH1SSTEV0 description.
17	CH1SSTEV5	Slave_TIMERx interconnection event 5 generates channel 1 “set request” Refer to CH1SSTEV0 description.
16	CH1SSTEV4	Slave_TIMERx interconnection event 4 generates channel 1 “set request” Refer to CH1SSTEV0 description.
15	CH1SSTEV3	Slave_TIMERx interconnection event 3 generates channel 1 “set request” Refer to CH1SSTEV0 description.
14	CH1SSTEV2	Slave_TIMERx interconnection event 2 generates channel 1 “set request” Refer to CH1SSTEV0 description.
13	CH1SSTEV1	Slave_TIMERx interconnection event 1 generates channel 1 “set request” Refer to CH1SSTEV0 description.
12	CH1SSTEV0	Slave_TIMERx interconnection event 0 generates channel 1 “set request” When this bit is set, Slave_TIMERx interconnection event 0 can generate “set request”. 0: The event cannot generate “set request”. 1: The event can generate “set request”.
11	CH1SMTCMP3	Master_TIMER compare 3 event generates channel 1 “set request” When this bit is set, Master_TIMER compare 3 event can generate “set request”. 0: The event cannot generate “set request”. 1: The event can generate “set request”.
10	CH1SMTCMP2	Master_TIMER compare 2 event generates channel 1 “set request”

When this bit is set, Master_TIMER compare 2 event can generate “set request”.
0: The event cannot generate “set request”.
1: The event can generate “set request”.

9 CH1SMTCMP1 Master_TIMER compare 1 event generates channel 1 “set request”
When this bit is set, Master_TIMER compare 1 event can generate “set request”.
0: The event cannot generate “set request”.
1: The event can generate “set request”.

8 CH1SMTCMP0 Master_TIMER compare 0 event generates channel 1 “set request”
When this bit is set, Master_TIMER compare 0 event can generate “set request”.
0: The event cannot generate “set request”.
1: The event can generate “set request”.

7 CH1SMTPER Master_TIMER period event generates channel 1 “set request”
In continuous mode, the Master_TIMER counter roll-over event can generate “set request”. In single pulse mode, the Master_TIMER reset event can generate “set request”.
0: The event cannot generate “set request”.
1: The event can generates “set request”.

6 CH1SCMP3 Slave_TIMERx compare 3 event generates channel 1 “set request”
When this bit is set, Slave_TIMERx compare 3 event can generate “set request”.
0: The event cannot generate “set request”.
1: The event can generate “set request”.

5 CH1SCMP2 Slave_TIMERx compare 2 event generates channel 1 “set request”
When this bit is set, Slave_TIMERx compare 2 event can generate “set request”.
0: The event cannot generate “set request”.
1: The event can generate “set request”.

4 CH1SCMP1 Slave_TIMERx compare 1 event generates channel 1 “set request”
When this bit is set, Slave_TIMERx compare 1 event can generate “set request”.
0: The event cannot generate “set request”.
1: The event can generate “set request”.

3 CH1SCMP0 Slave_TIMERx compare 0 event generates channel 1 “set request”
When this bit is set, Slave_TIMERx compare 0 event can generate “set request”.
0: The event cannot generate “set request”.
1: The event can generate “set request”.

2 CH1SPER Slave_TIMERx period event generates channel 1 “set request”
When this bit is set, Slave_TIMERx period event can generate “set request”.
0: The event cannot generate “set request”.
1: The event can generate “set request”.

1 CH1SRST Slave_TIMERx reset event generates channel 1 “set request”
When this bit is set, Slave_TIMERx reset event from synchronous input and

		software can generate channel 1 “set request”. 0: The event cannot generate “set request”. 1: The event can generate “set request”. **Note:** When this bit is set, the reset of other timers does not affect the output.
0	CH1SSEV	Software event generates channel 1 “set request” This bit is set by software and cleared by hardware automatically. When this bit is set, it can generate channel 1 “set request”. 0: The event cannot generate “set request”. 1: The event can generate “set request”. **Note:** This bit is not preloaded

HRTIMER Slave_TIMERx channel 1 reset request register (HRTIMER_STxCH1RST)

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
CH1RSUP	CH1RSEXEV9	CH1RSEXEV8	CH1RSEXEV7	CH1RSEXEV6	CH1RSEXEV5	CH1RSEXEV4	CH1RSEXEV3	CH1RSEXEV2	CH1RSEXEV1	CH1RSEXEV0	CH1RSSTEV8	CH1RSSTEV7	CH1RSSTEV6	CH1RSSTEV5	CH1RSSTEV4
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
CH1RSSTEV3	CH1RSSTEV2	CH1RSSTEV1	CH1RSSTEV0	CH1RSMTCMP3	CH1RSMTCMP2	CH1RSMTCMP1	CH1RSMTCMP0	CH1RSMTPER	CH1RSCMP3	CH1RSCMP2	CH1RSCMP1	CH1RSCMP0	CH1RSPER	CH1RSRST	CH1RSSEV
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31	CH1RSUP	Update event generates channel 1 “reset request” When this bit is set, update event can generate “reset request”. 0: The event cannot generate “reset request”. 1: The event can generate “reset request”.
30	CH1RSEXEV9	External event 9 generates channel 1 “reset request” Refer to CH1RSEXEV0 description.
29	CH1RSEXEV8	External event 8 generates channel 1 “reset request” Refer to CH1RSEXEV0 description.
28	CH1RSEXEV7	External event 7 generates channel 1 “reset request” Refer to CH1RSEXEV0 description.
27	CH1RSEXEV6	External event 6 generates channel 1 “reset request” Refer to CH1RSEXEV0 description.

26	CH1RSEXEV5	External event 5 generates channel 1 “reset request” Refer to CH1RSEXEV0 description.
25	CH1RSEXEV4	External event 4 generates channel 1 “reset request” Refer to CH1RSEXEV0 description.
24	CH1RSEXEV3	External event 3 generates channel 1 “reset request” Refer to CH1RSEXEV0 description.
23	CH1RSEXEV2	External event 2 generates channel 1 “reset request” Refer to CH1RSEXEV0 description.
22	CH1RSEXEV1	External event 1 generates channel 1 “reset request” Refer to CH1RSEXEV0 description.
21	CH1RSEXEV0	External event 0 generates channel 1 “reset request” When this bit is set, external event 0 can generate channel “reset request”. 0: The event cannot generate “reset request”. 1: The event can generate “reset request”.
20	CH1RSSTEV8	Slave_TIMERx interconnection event 8 generates channel 1 “reset request” Refer to CH1RSSTEV0 description.
19	CH1RSSTEV7	Slave_TIMERx interconnection event 7 generates channel 1 “reset request” Refer to CH1RSSTEV0 description.
18	CH1RSSTEV6	Slave_TIMERx interconnection event 6 generates channel 1 “reset request” Refer to CH1RSSTEV0 description.
17	CH1RSSTEV5	Slave_TIMERx interconnection event 5 generates channel 1 “reset request” Refer to CH1RSSTEV0 description.
16	CH1RSSTEV4	Slave_TIMERx interconnection event 4 generates channel 1 “reset request” Refer to CH1RSSTEV0 description.
15	CH1RSSTEV3	Slave_TIMERx interconnection event 3 generates channel 1 “reset request” Refer to CH1RSSTEV0 description.
14	CH1RSSTEV2	Slave_TIMERx interconnection event 2 generates channel 1 “reset request” Refer to CH1RSSTEV0 description.
13	CH1RSSTEV1	Slave_TIMERx interconnection event 1 generates channel 1 “reset request” Refer to CH1RSSTEV0 description.
12	CH1RSSTEV0	Slave_TIMERx interconnection event 0 generates channel 1 “reset request” When this bit is set, Slave_TIMERx interconnection event 0 can generate channel “reset request”. 0: The event cannot generate “reset request”. 1: The event can generate “reset request”.
11	CH1RSMTCMP3	Master_TIMER compare 3 event generates channel 1 “reset request”

When this bit is set, Master_TIMER compare 3 event can generate channel “reset request”.
0: The event cannot generate “reset request”.
1: The event can generate “reset request”.

10	CH1RSMTCMP2	Master_TIMER compare 2 event generates channel 1 “reset request” When this bit is set, Master_TIMER compare 2 event can generate channel “reset request”. 0: The event cannot generate “reset request”. 1: The event can generate “reset request”.
9	CH1RSMTCMP1	Master_TIMER compare 1 event generates channel 1 “reset request” When this bit is set, Master_TIMER compare 1 event can generate channel “reset request”. 0: The event cannot generate “reset request”. 1: The event can generate “reset request”.
8	CH1RSMTCMP0	Master_TIMER compare 0 event generates channel 1 “reset request” When this bit is set, Master_TIMER compare 0 event can generate channel “reset request”. 0: The event cannot generate “reset request”. 1: The event can generate “reset request”.
7	CH1RSMTPER	Master_TIMER period event generates channel 1 “reset request” In continuous mode, the Master_TIMER counter roll-over event can generate channel “reset request”. In single pulse mode, the Master_TIMER reset event can generate channel “reset request”. 0: The event cannot generate “reset request”. 1: The event can generate “reset request”.
6	CH1RSCMP3	Slave_TIMERx compare 3 event generates channel 1 “reset request” When this bit is set, Slave_TIMERx compare 3 event can generate channel “reset request”. 0: The event cannot generate “reset request”. 1: The event can generate “reset request”.
5	CH1RSCMP2	Slave_TIMERx compare 2 event generates channel 1 “reset request” When this bit is set, Slave_TIMERx compare 2 event can generate channel “reset request”. 0: The event cannot generate “reset request”. 1: The event can generate “reset request”.
4	CH1RSCMP1	Slave_TIMERx compare 1 event generates channel 1 “reset request” When this bit is set, Slave_TIMERx compare 1 event can generate channel “reset request”. 0: The event cannot generate “reset request”.

1: The event can generate “reset request”.

3	CH1RSCMP0	Slave_TIMERx compare 0 event generates channel 1 “reset request” When this bit is set, Slave_TIMERx compare 0 event can generate channel “reset request”. 0: The event cannot generate “reset request”. 1: The event can generate “reset request”.
2	CH1RSPER	Slave_TIMERx period event generates channel 1 “reset request” When this bit is set, Slave_TIMERx period event can generate channel “reset request”. 0: The event cannot generate “reset request”. 1: The event can generate “reset request”.
1	CH1RSRST	Slave_TIMERx reset event generates channel 1 “reset request” When this bit is set, Slave_TIMERx reset event from synchronous input and software can generate channel “reset request”. 0: The event cannot generate “reset request”. 1: The event can generate “reset request”. **Note:** When this bit is set, the reset of other timers does not affect the output.
0	CH1RSSEV	Software event generates channel 1 “reset request” This bit is set by software and cleared by hardware automatically. When this bit is set, it will generate “reset request”. 0: The event cannot generate “reset request”. 1: The event can generate “reset request”. **Note:** This bit is not preloaded

HRTIMER Slave_TIMERx external event filter configuration register 0 (HRTIMER_STxEXEVFCFG0)

Address offset: 0x4C

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		EXEV4FM[4:0]					EXEV4MEEN	EXEV3FM[4:0]					EXEV3MEEN	EXEV2FM[4:3]	
		rw					rw	rw					rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXEV2FM[2:0]			EXEV0MEEN	EXEV1FM[4:0]					EXEV1MEEN	EXEV0FM[4:0]					EXEV0MEEN
rw			rw	rw					rw	rw					rw

Bits	Fields	Descriptions

31:30	Reserved	Must be kept at reset value
29:25	EXEV4FM[4:0]	External event 4 filter mode Refer to EXEV0FM[4:0] description.
24	EXEV4MEEN	External event 4 memorized enable Refer to EXEV0MEEN description.
23:19	EXEV3FM[4:0]	External event 3 filter mode Refer to EXEV0FM[4:0] description.
18	EXEV3MEEN	External event 3 memorized enable Refer to EXEV0MEEN description.
17:13	EXEV2FM[3:0]	External event 2 filter mode Refer to EXEV0FM[4:0] description.
12	EXEV2MEEN	External event 2 memorized enable Refer to EXEV0MEEN description.
11:7	EXEV1FM[4:0]	External event 1 filter mode Refer to EXEV0FM[4:0] description.
6	EXEV1MEEN	External event 1 memorized enable Refer to EXEV0MEEN description.
5:1	EXEV0FM[4:0]	External event 0 filter mode In blanking mode, the external event is ignored if it occurs during a blank. In windowing mode, the external event is taken into account only if it occurs within a given time window. 00000: filter mode disable. 00001: Blanking mode. The blank is from counter reset to HRTIMER_STxCMP0V. 00010: Blanking mode. The blank is from counter reset to HRTIMER_STxCMP1V. 00011: Blanking mode. The blank is from counter reset to HRTIMER_STxCMP2V. 00100: Blanking mode. The blank is from counter reset to HRTIMER_STxCMP3V. 00101: Blanking mode. The blank is from other Slave_TIMERy(not Slave_TIMERx): STBLKSRC0 00110: Blanking mode. The blank is from other Slave_TIMERy(not Slave_TIMERx): STBLKSRC1 00111: Blanking mode. The blank is from other Slave_TIMERy(not Slave_TIMERx): STBLKSRC2 01000: Blanking mode. The blank is from other Slave_TIMERy(not Slave_TIMERx): STBLKSRC3 01001: Blanking mode. The blank is from other Slave_TIMERy(not Slave_TIMERx): STBLKSRC4 01010: Blanking mode. The blank is from other Slave_TIMERy(not Slave_TIMERx): STBLKSRC5 01011: Blanking mode. The blank is from other Slave_TIMERy(not Slave_TIMERx):

		STBLKSRC6 01100: Blanking mode. The blank is from other Slave_TIMERy(not Slave_TIMERx): STBLKSRC7 01101: Windowing mode. The windowing is from counter reset to HRTIMER_STxCMP1V. 01110: Windowing mode. The windowing is from counter reset to HRTIMER_STxCMP2V. 01111: Windowing mode. The windowing is from other Slave_TIMERy(not Slave_TIMERx):STWDSRC 10000: Blanking mode. The blank is from other Slave_TIMERy(not Slave_TIMERx): STBLKSRC8 10001: Blanking mode. The blank is from other Slave_TIMERy(not Slave_TIMERx): STBLKSRC9 **Note:** (1) This bit-field must not be modified once the counter is enabled (STxCEN bit set) (2) The value of the compare register which used for filter must be above 0.
0	EXEV0MEEN	External event 0 memory 0: External event memory disable. 1: External event memory enable. The memorized event is generated as soon as the blanking period or windowing period is completed. **Note:** (1) This bit-field must not be modified once the counter is enabled (STxCEN bit set) (2) When this bit is set, a timeout event can be generated in window mode.

HRTIMER Slave_TIMERx external event filter configuration register 1 (HRTIMER_STxEXEVFCFG1)

Address offset: 0x50

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		EXEV9FM[4:0]					EXEV9MEEN	EXEV8FM[4:0]					EXEV8MEEN	EXEV7FM[4:3]	
		rw					rw	rw					rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXEV7FM[2:0]			EXEV7MEEN	EXEV6FM[4:0]					EXEV6MEEN	EXEV5FM[4:0]					EXEV5MEEN
rw			rw	rw					rw	rw					rw

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value
29:25	EXEV9FM[4:0]	External event 9 filter mode

		Refer to EXEV0FM[4:0] in HRTIMER_STxEXEVFCFG0 description.
24	EXEV9MEEN	External event 9 memorized enable Refer to EXEV0MEEN in HRTIMER_STxEXEVFCFG0 description.
23:19	EXEV8FM[3:0]	External event 8 filter mode Refer to EXEV0FM[4:0] in HRTIMER_STxEXEVFCFG0 description.
18	EXEV8MEEN	External event 8 memorized enable Refer to EXEV0MEEN in HRTIMER_STxEXEVFCFG0 description.
17:13	EXEV7FM[4:0]	External event 7 filter mode Refer to EXEV0FM[4:0] in HRTIMER_STxEXEVFCFG0 description.
12	EXEV7MEEN	External event 7 memorized enable Refer to EXEV0MEEN in HRTIMER_STxEXEVFCFG0 description.
11:7	EXEV6FM[4:0]	External event 6 filter mode Refer to EXEV0FM[4:0] in HRTIMER_STxEXEVFCFG0 description.
6	EXEV6MEEN	External event 6 memorized enable Refer to EXEV0MEEN in HRTIMER_STxEXEVFCFG0 description.
5:1	EXEV5FM[4:0]	External event 5 filter mode Refer to EXEV0FM[4:0] in HRTIMER_STxEXEVFCFG0 description.
0	EXEV5MEEN	External event 0 memorized enable Refer to EXEV0MEEN in HRTIMER_STxEXEVFCFG0 description.

HRTIMER Slave_TIMERx counter reset register (HRTIMER_STxCNTRST)

Address offset: 0x54

Reset value: 0x0000 0000

For Slave_TIMER0

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ST5CMP1RST	ST4CMP3RST	ST4CMP1RST	ST4CMP0RST	ST3CMP3RST	ST3CMP1RST	ST3CMP0RST	ST2CMP3RST	ST2CMP1RST	ST2CMP0RST	ST1CMP3RST	ST1CMP1RST	ST1CMP0RST	EXEV9RST	EXEV8RST	EXEV7RST
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
EXEV6RST	EXEV5RST	EXEV4RST	EXEV3RST	EXEV2RST	EXEV1RST	EXEV0RST	MTCMP3RST	MTCMP2RST	MTCMP1RST	MTCMP0RST	MTPERRST	CMP3RST	CMP1RST	UPRST	ST5CMP0RST
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31	ST5CMP1RST	Slave_TIMER5 compare 1 event resets counter

		Refer to ST5CMP1RST description.
30	ST4CMP3RST	Slave_TIMER4 compare 3 event resets counter Refer to ST4CMP0RST description.
29	ST4CMP1RST	Slave_TIMER4 compare 1 event resets counter Refer to ST4CMP0RST description.
28	ST4CMP0RST	Slave_TIMER4 compare 0 event resets counter This bit specifies whether the Slave_TIMER4 compare 0 event can reset the counter. 0: Slave_TIMER4 compare 0 event do not reset counter 1: Slave_TIMER4 compare 0 event resets counter
27	ST3CMP3RST	Slave_TIMER3 compare 3 event resets counter Refer to ST3CMP0RST description.
26	ST3CMP1RST	Slave_TIMER3 compare 1 event resets counter Refer to ST3CMP0RST description.
25	ST3CMP0RST	Slave_TIMER3 compare 0 event resets counter This bit specifies whether the Slave_TIMER3 compare 0 event can reset the counter. 0: Slave_TIMER3 compare 0 event do not reset counter 1: Slave_TIMER3 compare 0 event resets counter
24	ST2CMP3RST	Slave_TIMER2 compare 3 event resets counter Refer to ST2CMP0RST description.
23	ST2CMP1RST	Slave_TIMER2 compare 1 event resets counter Refer to ST2CMP0RST description.
22	ST2CMP0RST	Slave_TIMER2 compare 0 event resets counter This bit specifies whether the Slave_TIMER2 compare 0 event can reset the counter. 0: Slave_TIMER2 compare 0 event do not reset counter 1: Slave_TIMER2 compare 0 event resets counter
21	ST1CMP3RST	Slave_TIMER1 compare 3 event resets counter Refer to ST1CMP0RST description.
20	ST1CMP1RST	Slave_TIMER1 compare 1 event resets counter Refer to ST1CMP0RST description.
19	ST1CMP0RST	Slave_TIMER1 compare 0 event resets counter This bit specifies whether the Slave_TIMER1 compare 0 event can reset the counter. 0: Slave_TIMER1 compare 0 event do not reset counter 1: Slave_TIMER1 compare 0 event resets counter

18	EXEV9RST	External event 9 resets counter Refer to EXEV0RST description.
17	EXEV8RST	External event 8 resets counter Refer to EXEV0RST description.
16	EXEV7RST	External event 7 resets counter Refer to EXEV0RST description.
15	EXEV6RST	External event 6 resets counter Refer to EXEV0RST description.
14	EXEV5RST	External event 5 resets counter Refer to EXEV0RST description.
13	EXEV4RST	External event 4 resets counter Refer to EXEV0RST description.
12	EXEV3RST	External event 3 resets counter Refer to EXEV0RST description.
11	EXEV2RST	External event 2 resets counter Refer to EXEV0RST description.
10	EXEV1RST	External event 1 resets counter Refer to EXEV0RST description.
9	EXEV0RST	External event 0 resets counter This bit specifies whether the External event 0 can reset the counter. 0: External event 0 do not reset counter. 1: External event 0 resets counter.
8	MTCMP3RST	Master_TIMER compare 3 event resets counter Refer to MTCMP0RST description
7	MTCMP2RST	Master_TIMER compare 2 event resets counter Refer to MTCMP0RST description
6	MTCMP1RST	Master_TIMER compare 1 event resets counter Refer to MTCMP0RST description
5	MTCMP0RST	Master_TIMER compare 0 event resets counter This bit specifies whether the Master_TIMER compare 0 event can reset the counter. 0: Master_TIMER compare 0 event do not reset counter 1: Master_TIMER compare 0 event resets counter
4	MTPERRST	Master_TIMER period event resets counter This bit specifies whether the Master_TIMER period event can reset the counter. 0: Master_TIMER period event do not reset counter

		1: Master_TIMER period event resets counter
3	CMP3RST	Slave_TIMER0 compare 3 event resets counter Refer to CMP1RST description
2	CMP1RST	Slave_TIMER0 compare 1 event resets counter This bit specifies whether the compare 1 event can reset the counter. 0: Compare 1 event do not reset counter 1: Compare 1 event resets counter
1	UPRST	Slave_TIMER0 update event resets counter This bit specifies whether the update event can reset the counter. 0: Update event do not reset counter 1: Update event resets counter
0	ST5CMP0RST	Slave_TIMER5 compare 0 event resets counter Refer to ST5CMP0RST description.

For Slave_TIMER1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ST5CMP1RST	ST4CMP3RST	ST4CMP1RST	ST4CMP0RST	ST3CMP3RST	ST3CMP1RST	ST3CMP0RST	ST2CMP3RST	ST2CMP1RST	ST2CMP0RST	ST0CMP3RST	ST0CMP1RST	ST0CMP0RST	EXEV9RST	EXEV8RST	EXEV7RST
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
EXEV6RST	EXEV5RST	EXEV4RST	EXEV3RST	EXEV2RST	EXEV1RST	EXEV0RST	MTCMP3RST	MTCMP2RST	MTCMP1RST	MTCMP0RST	MTPERRST	CMP3RST	CMP1RST	UPRST	ST5CMP0RST
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31	ST5CMP1RST	Slave_TIMER5 compare 1 event resets counter Refer to ST5CMP1RST description.
30	ST4CMP3RST	Slave_TIMER4 compare 3 event resets counter Refer to ST4CMP0RST description.
29	ST4CMP1RST	Slave_TIMER4 compare 1 event resets counter Refer to ST4CMP0RST description.
28	ST4CMP0RST	Slave_TIMER4 compare 0 event resets counter This bit specifies whether the Slave_TIMER4 compare 0 event can reset the counter. 0: Slave_TIMER4 compare 0 event do not reset counter 1: Slave_TIMER4 compare 0 event resets counter
27	ST3CMP3RST	Slave_TIMER3 compare 3 event resets counter Refer to ST3CMP0RST description.

26	ST3CMP1RST	Slave_TIMER3 compare 1 event resets counter Refer to ST3CMP0RST description.
25	ST3CMP0RST	Slave_TIMER3 compare 0 event resets counter This bit specifies whether the Slave_TIMER3 compare 0 event can reset the counter. 0: Slave_TIMER3 compare 0 event do not reset counter 1: Slave_TIMER3 compare 0 event resets counter
24	ST2CMP3RST	Slave_TIMER2 compare 3 event resets counter Refer to ST2CMP0RST description.
23	ST2CMP1RST	Slave_TIMER2 compare 1 event resets counter Refer to ST2CMP0RST description.
22	ST2CMP0RST	Slave_TIMER2 compare 0 event resets counter This bit specifies whether the Slave_TIMER2 compare 0 event can reset the counter. 0: Slave_TIMER2 compare 0 event do not reset counter 1: Slave_TIMER2 compare 0 event resets counter
21	ST0CMP3RST	Slave_TIMER0 compare 3 event resets counter Refer to ST0CMP0RST description.
20	ST0CMP1RST	Slave_TIMER0 compare 1 event resets counter Refer to ST0CMP0RST description.
19	ST0CMP0RST	Slave_TIMER0 compare 0 event resets counter This bit specifies whether the Slave_TIMER0 compare 0 event can reset the counter. 0: Slave_TIMER0 compare 0 event do not reset counter 1: Slave_TIMER0 compare 0 event resets counter
18	EXEV9RST	External event 9 resets counter Refer to EXEV0RST description.
17	EXEV8RST	External event 8 resets counter Refer to EXEV0RST description.
16	EXEV7RST	External event 7 resets counter Refer to EXEV0RST description.
15	EXEV6RST	External event 6 resets counter Refer to EXEV0RST description.
14	EXEV5RST	External event 5 resets counter Refer to EXEV0RST description.
13	EXEV4RST	External event 4 resets counter

		Refer to EXEV0RST description.
12	EXEV3RST	External event 3 resets counter Refer to EXEV0RST description.
11	EXEV2RST	External event 2 resets counter Refer to EXEV0RST description.
10	EXEV1RST	External event 1 resets counter Refer to EXEV0RST description.
9	EXEV0RST	External event 0 resets counter This bit specifies whether the External event 0 can reset the counter. 0: External event 0 do not reset counter. 1: External event 0 resets counter.
8	MTCMP3RST	Master_TIMER compare 3 event resets counter Refer to MTCMP0RST description
7	MTCMP2RST	Master_TIMER compare 2 event resets counter Refer to MTCMP0RST description
6	MTCMP1RST	Master_TIMER compare 1 event resets counter Refer to MTCMP0RST description
5	MTCMP0RST	Master_TIMER compare 0 event resets counter This bit specifies whether the Master_TIMER compare 0 event can reset the counter. 0: Master_TIMER compare 0 event do not reset counter 1: Master_TIMER compare 0 event resets counter
4	MTPERRST	Master_TIMER period event resets counter This bit specifies whether the Master_TIMER period event can reset the counter. 0: Master_TIMER period event do not reset counter 1: Master_TIMER period event resets counter
3	CMP3RST	Slave_TIMER1 compare 3 event resets counter Refer to CMP1RST description
2	CMP1RST	Slave_TIMER1 compare 1 event resets counter This bit specifies whether the compare 1 event can reset the counter. 0: Compare 1 event do not reset counter 1: Compare 1 event resets counter
1	UPRST	Slave_TIMER1 update event resets counter This bit specifies whether the update event can reset the counter. 0: Update event do not reset counter 1: Update event resets counter
0	ST5CMP0RST	Slave_TIMER5 compare 0 event resets counter

Refer to ST5CMP0RST description.

For Slave_TIMER2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ST5CMP1RST	ST4CMP3RST	ST4CMP1RST	ST4CMP0RST	ST3CMP3RST	ST3CMP1RST	ST3CMP0RST	ST1CMP3RST	ST1CMP1RST	ST1CMP0RST	ST0CMP3RST	ST0CMP1RST	ST0CMP0RST	EXEV9RST	EXEV8RST	EXEV7RST
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
EXEV6RST	EXEV5RST	EXEV4RST	EXEV3RST	EXEV2RST	EXEV1RST	EXEV0RST	MTCMP3RST	MTCMP2RST	MTCMP1RST	MTCMP0RST	MTPERRST	CMP3RST	CMP1RST	UPRST	ST5CMP0RST
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31	ST5CMP1RST	Slave_TIMER5 compare 1 event resets counter Refer to ST5CMP1RST description.
30	ST4CMP3RST	Slave_TIMER4 compare 3 event resets counter Refer to ST4CMP0RST description.
29	ST4CMP1RST	Slave_TIMER4 compare 1 event resets counter Refer to ST4CMP0RST description.
28	ST4CMP0RST	Slave_TIMER4 compare 0 event resets counter This bit specifies whether the Slave_TIMER4 compare 0 event can reset the counter. 0: Slave_TIMER4 compare 0 event do not reset counter 1: Slave_TIMER4 compare 0 event resets counter
27	ST3CMP3RST	Slave_TIMER3 compare 3 event resets counter Refer to ST3CMP0RST description.
26	ST3CMP1RST	Slave_TIMER3 compare 1 event resets counter Refer to ST3CMP0RST description.
25	ST3CMP0RST	Slave_TIMER3 compare 0 event resets counter This bit specifies whether the Slave_TIMER3 compare 0 event can reset the counter. 0: Slave_TIMER3 compare 0 event do not reset counter 1: Slave_TIMER3 compare 0 event resets counter
24	ST1CMP3RST	Slave_TIMER1 compare 3 event resets counter Refer to ST1CMP0RST description.
23	ST1CMP1RST	Slave_TIMER1 compare 1 event resets counter Refer to ST1CMP0RST description.
22	ST1CMP0RST	Slave_TIMER1 compare 0 event resets counter

		This bit specifies whether the Slave_TIMER1 compare 0 event can reset the counter. 0: Slave_TIMER1 compare 0 event do not reset counter 1: Slave_TIMER1 compare 0 event resets counter
21	ST0CMP3RST	Slave_TIMER0 compare 3 event resets counter Refer to ST0CMP0RST description.
20	ST0CMP1RST	Slave_TIMER0 compare 1 event resets counter Refer to ST0CMP0RST description.
19	ST0CMP0RST	Slave_TIMER0 compare 0 event resets counter This bit specifies whether the Slave_TIMER0 compare 0 event can reset the counter. 0: Slave_TIMER0 compare 0 event do not reset counter 1: Slave_TIMER0 compare 0 event resets counter
18	EXEV9RST	External event 9 resets counter Refer to EXEV0RST description.
17	EXEV8RST	External event 8 resets counter Refer to EXEV0RST description.
16	EXEV7RST	External event 7 resets counter Refer to EXEV0RST description.
15	EXEV6RST	External event 6 resets counter Refer to EXEV0RST description.
14	EXEV5RST	External event 5 resets counter Refer to EXEV0RST description.
13	EXEV4RST	External event 4 resets counter Refer to EXEV0RST description.
12	EXEV3RST	External event 3 resets counter Refer to EXEV0RST description.
11	EXEV2RST	External event 2 resets counter Refer to EXEV0RST description.
10	EXEV1RST	External event 1 resets counter Refer to EXEV0RST description.
9	EXEV0RST	External event 0 resets counter This bit specifies whether the External event 0 can reset the counter. 0: External event 0 do not reset counter. 1: External event 0 resets counter.
8	MTCMP3RST	Master_TIMER compare 3 event resets counter

		Refer to MTCMP0RST description
7	MTCMP2RST	Master_TIMER compare 2 event resets counter Refer to MTCMP0RST description
6	MTCMP1RST	Master_TIMER compare 1 event resets counter Refer to MTCMP0RST description
5	MTCMP0RST	Master_TIMER compare 0 event resets counter This bit specifies whether the Master_TIMER compare 0 event can reset the counter. 0: Master_TIMER compare 0 event do not reset counter 1: Master_TIMER compare 0 event resets counter
4	MTPERRST	Master_TIMER period event resets counter This bit specifies whether the Master_TIMER period event can reset the counter. 0: Master_TIMER period event do not reset counter 1: Master_TIMER period event resets counter
3	CMP3RST	Slave_TIMER2 compare 3 event resets counter Refer to CMP1RST description
2	CMP1RST	Slave_TIMER2 compare 1 event resets counter This bit specifies whether the compare 1 event can reset the counter. 0: Compare 1 event do not reset counter 1: Compare 1 event resets counter
1	UPRST	Slave_TIMER2 update event resets counter This bit specifies whether the update event can reset the counter. 0: Update event do not reset counter 1: Update event resets counter
0	ST5CMP0RST	Slave_TIMER5 compare 0 event resets counter Refer to ST5CMP0RST description.

For Slave_TIMER3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ST5CMP1RST	ST4CMP3RST	ST4CMP1RST	ST4CMP0RST	ST2CMP3RST	ST2CMP1RST	ST2CMP0RST	ST1CMP3RST	ST1CMP1RST	ST1CMP0RST	ST0CMP3RST	ST0CMP1RST	ST0CMP0RST	EXEV9RST	EXEV8RST	EXEV7RST
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
EXEV6RST	EXEV5RST	EXEV4RST	EXEV3RST	EXEV2RST	EXEV1RST	EXEV0RST	MTCMP3RST	MTCMP2RST	MTCMP1RST	MTCMP0RST	MTPERRST	CMP3RST	CMP1RST	UPRST	ST5CMP0RST
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31	ST5CMP1RST	Slave_TIMER5 compare 1 event resets counter

		Refer to ST5CMP1RST description.
30	ST4CMP3RST	Slave_TIMER4 compare 3 event resets counter Refer to ST4CMP0RST description.
29	ST4CMP1RST	Slave_TIMER4 compare 1 event resets counter Refer to ST4CMP0RST description.
28	ST4CMP0RST	Slave_TIMER4 compare 0 event resets counter This bit specifies whether the Slave_TIMER4 compare 0 event can reset the counter. 0: Slave_TIMER4 compare 0 event do not reset counter 1: Slave_TIMER4 compare 0 event resets counter
27	ST2CMP3RST	Slave_TIMER2 compare 3 event resets counter Refer to ST2CMP0RST description.
26	ST2CMP1RST	Slave_TIMER2 compare 1 event resets counter Refer to ST2CMP0RST description.
25	ST2CMP0RST	Slave_TIMER2 compare 0 event resets counter This bit specifies whether the Slave_TIMER2 compare 0 event can reset the counter. 0: Slave_TIMER2 compare 0 event do not reset counter 1: Slave_TIMER2 compare 0 event resets counter
24	ST1CMP3RST	Slave_TIMER1 compare 3 event resets counter Refer to ST1CMP0RST description.
23	ST1CMP1RST	Slave_TIMER1 compare 1 event resets counter Refer to ST1CMP0RST description.
22	ST1CMP0RST	Slave_TIMER1 compare 0 event resets counter This bit specifies whether the Slave_TIMER1 compare 0 event can reset the counter. 0: Slave_TIMER1 compare 0 event do not reset counter 1: Slave_TIMER1 compare 0 event resets counter
21	ST0CMP3RST	Slave_TIMER0 compare 3 event resets counter Refer to ST0CMP0RST description.
20	ST0CMP1RST	Slave_TIMER0 compare 1 event resets counter Refer to ST0CMP0RST description.
19	ST0CMP0RST	Slave_TIMER0 compare 0 event resets counter This bit specifies whether the Slave_TIMER0 compare 0 event can reset the counter. 0: Slave_TIMER0 compare 0 event do not reset counter 1: Slave_TIMER0 compare 0 event resets counter

18	EXEV9RST	External event 9 resets counter Refer to EXEV0RST description.
17	EXEV8RST	External event 8 resets counter Refer to EXEV0RST description.
16	EXEV7RST	External event 7 resets counter Refer to EXEV0RST description.
15	EXEV6RST	External event 6 resets counter Refer to EXEV0RST description.
14	EXEV5RST	External event 5 resets counter Refer to EXEV0RST description.
13	EXEV4RST	External event 4 resets counter Refer to EXEV0RST description.
12	EXEV3RST	External event 3 resets counter Refer to EXEV0RST description.
11	EXEV2RST	External event 2 resets counter Refer to EXEV0RST description.
10	EXEV1RST	External event 1 resets counter Refer to EXEV0RST description.
9	EXEV0RST	External event 0 resets counter This bit specifies whether the External event 0 can reset the counter. 0: External event 0 do not reset counter. 1: External event 0 resets counter.
8	MTCMP3RST	Master_TIMER compare 3 event resets counter Refer to MTCMP0RST description
7	MTCMP2RST	Master_TIMER compare 2 event resets counter Refer to MTCMP0RST description
6	MTCMP1RST	Master_TIMER compare 1 event resets counter Refer to MTCMP0RST description
5	MTCMP0RST	Master_TIMER compare 0 event resets counter This bit specifies whether the Master_TIMER compare 0 event can reset the counter. 0: Master_TIMER compare 0 event do not reset counter 1: Master_TIMER compare 0 event resets counter
4	MTPERRST	Master_TIMER period event resets counter This bit specifies whether the Master_TIMER period event can reset the counter. 0: Master_TIMER period event do not reset counter

		1: Master_TIMER period event resets counter
3	CMP3RST	Slave_TIMER3 compare 3 event resets counter Refer to CMP1RST description
2	CMP1RST	Slave_TIMER3 compare 1 event resets counter This bit specifies whether the compare 1 event can reset the counter. 0: Compare 1 event do not reset counter 1: Compare 1 event resets counter
1	UPRST	Slave_TIMER3 update event resets counter This bit specifies whether the update event can reset the counter. 0: Update event do not reset counter 1: Update event resets counter
0	ST5CMP0RST	Slave_TIMER5 compare 0 event resets counter Refer to ST5CMP0RST description.

For Slave_TIMER4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ST5CMP1RST	ST3CMP3RST	ST3CMP1RST	ST3CMP0RST	ST2CMP3RST	ST2CMP1RST	ST2CMP0RST	ST1CMP3RST	ST1CMP1RST	ST1CMP0RST	ST0CMP3RST	ST0CMP1RST	ST0CMP0RST	EXEV9RST	EXEV8RST	EXEV7RST
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
EXEV6RST	EXEV5RST	EXEV4RST	EXEV3RST	EXEV2RST	EXEV1RST	EXEV0RST	MTCMP3RST	MTCMP2RST	MTCMP1RST	MTCMP0RST	MTPERRST	CMP3RST	CMP1RST	UPRST	ST5CMP0RST
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31	ST5CMP1RST	Slave_TIMER5 compare 1 event resets counter Refer to ST5CMP1RST description.
30	ST3CMP3RST	Slave_TIMER3 compare 3 event resets counter Refer to ST3CMP0RST description.
29	ST3CMP1RST	Slave_TIMER3 compare 1 event resets counter Refer to ST3CMP0RST description.
28	ST3CMP0RST	Slave_TIMER3 compare 0 event resets counter This bit specifies whether the Slave_TIMER3 compare 0 event can reset the counter. 0: Slave_TIMER3 compare 0 event do not reset counter 1: Slave_TIMER3 compare 0 event resets counter
27	ST2CMP3RST	Slave_TIMER2 compare 3 event resets counter Refer to ST2CMP0RST description.

26	ST2CMP1RST	Slave_TIMER2 compare 1 event resets counter Refer to ST2CMP0RST description.
25	ST2CMP0RST	Slave_TIMER2 compare 0 event resets counter This bit specifies whether the Slave_TIMER2 compare 0 event can reset the counter. 0: Slave_TIMER2 compare 0 event do not reset counter 1: Slave_TIMER2 compare 0 event resets counter
24	ST1CMP3RST	Slave_TIMER1 compare 3 event resets counter Refer to ST1CMP0RST description.
23	ST1CMP1RST	Slave_TIMER1 compare 1 event resets counter Refer to ST1CMP0RST description.
22	ST1CMP0RST	Slave_TIMER1 compare 0 event resets counter This bit specifies whether the Slave_TIMER1 compare 0 event can reset the counter. 0: Slave_TIMER1 compare 0 event do not reset counter 1: Slave_TIMER1 compare 0 event resets counter
21	ST0CMP3RST	Slave_TIMER0 compare 3 event resets counter Refer to ST0CMP0RST description.
20	ST0CMP1RST	Slave_TIMER0 compare 1 event resets counter Refer to ST0CMP0RST description.
19	ST0CMP0RST	Slave_TIMER0 compare 0 event resets counter This bit specifies whether the Slave_TIMER0 compare 0 event can reset the counter. 0: Slave_TIMER0 compare 0 event do not reset counter 1: Slave_TIMER0 compare 0 event resets counter
18	EXEV9RST	External event 9 resets counter Refer to EXEV0RST description.
17	EXEV8RST	External event 8 resets counter Refer to EXEV0RST description.
16	EXEV7RST	External event 7 resets counter Refer to EXEV0RST description.
15	EXEV6RST	External event 6 resets counter Refer to EXEV0RST description.
14	EXEV5RST	External event 5 resets counter Refer to EXEV0RST description.
13	EXEV4RST	External event 4 resets counter

		Refer to EXEV0RST description.
12	EXEV3RST	External event 3 resets counter Refer to EXEV0RST description.
11	EXEV2RST	External event 2 resets counter Refer to EXEV0RST description.
10	EXEV1RST	External event 1 resets counter Refer to EXEV0RST description.
9	EXEV0RST	External event 0 resets counter This bit specifies whether the External event 0 can reset the counter. 0: External event 0 do not reset counter. 1: External event 0 resets counter.
8	MTCMP3RST	Master_TIMER compare 3 event resets counter Refer to MTCMP0RST description
7	MTCMP2RST	Master_TIMER compare 2 event resets counter Refer to MTCMP0RST description
6	MTCMP1RST	Master_TIMER compare 1 event resets counter Refer to MTCMP0RST description
5	MTCMP0RST	Master_TIMER compare 0 event resets counter This bit specifies whether the Master_TIMER compare 0 event can reset the counter. 0: Master_TIMER compare 0 event do not reset counter 1: Master_TIMER compare 0 event resets counter
4	MTPERRST	Master_TIMER period event resets counter This bit specifies whether the Master_TIMER period event can reset the counter. 0: Master_TIMER period event do not reset counter 1: Master_TIMER period event resets counter
3	CMP3RST	Slave_TIMER4 compare 3 event resets counter Refer to CMP1RST description
2	CMP1RST	Slave_TIMER4 compare 1 event resets counter This bit specifies whether the compare 1 event can reset the counter. 0: Compare 1 event do not reset counter 1: Compare 1 event resets counter
1	UPRST	Slave_TIMER4 update event resets counter This bit specifies whether the update event can reset the counter. 0: Update event do not reset counter 1: Update event resets counter
0	ST5CMP0RST	Slave_TIMER5 compare 0 event resets counter

Refer to ST5CMP0RST description.

For Slave_TIMER5 - For Slave_TIMER7

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ST4CMP1RST	ST3CMP3RST	ST3CMP1RST	ST3CMP0RST	ST2CMP3RST	ST2CMP1RST	ST2CMP0RST	ST1CMP3RST	ST1CMP1RST	ST1CMP0RST	ST0CMP3RST	ST0CMP1RST	ST0CMP0RST	EXEV9RST	EXEV8RST	EXEV7RST
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
EXEV6RST	EXEV5RST	EXEV4RST	EXEV3RST	EXEV2RST	EXEV1RST	EXEV0RST	MTCMP3RST	MTCMP2RST	MTCMP1RST	MTCMP0RST	MTPERRST	CMP3RST	CMP1RST	UPRST	ST4CMP0RST
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31	ST4CMP1RST	Slave_TIMER4 compare 1 event resets counter Refer to ST4CMP1RST description.
30	ST3CMP3RST	Slave_TIMER3 compare 3 event resets counter Refer to ST3CMP0RST description.
29	ST3CMP1RST	Slave_TIMER3 compare 1 event resets counter Refer to ST3CMP0RST description.
28	ST3CMP0RST	Slave_TIMER3 compare 0 event resets counter This bit specifies whether the Slave_TIMER3 compare 0 event can reset the counter. 0: Slave_TIMER3 compare 0 event do not reset counter 1: Slave_TIMER3 compare 0 event resets counter
27	ST2CMP3RST	Slave_TIMER2 compare 3 event resets counter Refer to ST2CMP0RST description.
26	ST2CMP1RST	Slave_TIMER2 compare 1 event resets counter Refer to ST2CMP0RST description.
25	ST2CMP0RST	Slave_TIMER2 compare 0 event resets counter This bit specifies whether the Slave_TIMER2 compare 0 event can reset the counter. 0: Slave_TIMER2 compare 0 event do not reset counter 1: Slave_TIMER2 compare 0 event resets counter
24	ST1CMP3RST	Slave_TIMER1 compare 3 event resets counter Refer to ST1CMP0RST description.
23	ST1CMP1RST	Slave_TIMER1 compare 1 event resets counter Refer to ST1CMP0RST description.
22	ST1CMP0RST	Slave_TIMER1 compare 0 event resets counter

		This bit specifies whether the Slave_TIMER1 compare 0 event can reset the counter. 0: Slave_TIMER1 compare 0 event do not reset counter 1: Slave_TIMER1 compare 0 event resets counter
21	ST0CMP3RST	Slave_TIMER0 compare 3 event resets counter Refer to ST0CMP0RST description.
20	ST0CMP1RST	Slave_TIMER0 compare 1 event resets counter Refer to ST0CMP0RST description.
19	ST0CMP0RST	Slave_TIMER0 compare 0 event resets counter This bit specifies whether the Slave_TIMER0 compare 0 event can reset the counter. 0: Slave_TIMER0 compare 0 event do not reset counter 1: Slave_TIMER0 compare 0 event resets counter
18	EXEV9RST	External event 9 resets counter Refer to EXEV0RST description.
17	EXEV8RST	External event 8 resets counter Refer to EXEV0RST description.
16	EXEV7RST	External event 7 resets counter Refer to EXEV0RST description.
15	EXEV6RST	External event 6 resets counter Refer to EXEV0RST description.
14	EXEV5RST	External event 5 resets counter Refer to EXEV0RST description.
13	EXEV4RST	External event 4 resets counter Refer to EXEV0RST description.
12	EXEV3RST	External event 3 resets counter Refer to EXEV0RST description.
11	EXEV2RST	External event 2 resets counter Refer to EXEV0RST description.
10	EXEV1RST	External event 1 resets counter Refer to EXEV0RST description.
9	EXEV0RST	External event 0 resets counter This bit specifies whether the External event 0 can reset the counter. 0: External event 0 do not reset counter. 1: External event 0 resets counter.
8	MTCMP3RST	Master_TIMER compare 3 event resets counter

		Refer to MTCMP0RST description
7	MTCMP2RST	Master_TIMER compare 2 event resets counter Refer to MTCMP0RST description
6	MTCMP1RST	Master_TIMER compare 1 event resets counter Refer to MTCMP0RST description
5	MTCMP0RST	Master_TIMER compare 0 event resets counter This bit specifies whether the Master_TIMER compare 0 event can reset the counter. 0: Master_TIMER compare 0 event do not reset counter 1: Master_TIMER compare 0 event resets counter
4	MTPERRST	Master_TIMER period event resets counter This bit specifies whether the Master_TIMER period event can reset the counter. 0: Master_TIMER period event do not reset counter 1: Master_TIMER period event resets counter
3	CMP3RST	Slave_TIMERy (y = 5 - 7) compare 3 event resets counter Refer to CMP1RST description
2	CMP1RST	Slave_TIMERy (y = 5 - 7) compare 1 event resets counter This bit specifies whether the compare 1 event can reset the counter. 0: Compare 1 event do not reset counter 1: Compare 1 event resets counter
1	UPRST	Slave_TIMERy (y = 5 - 7) update event resets counter This bit specifies whether the update event can reset the counter. 0: Update event do not reset counter 1: Update event resets counter
0	ST4CMP0RST	Slave_TIMER4 compare 0 event resets counter Refer to ST4CMP0RST description.

HRTIMER Slave_TIMERx carrier-signal control register (HRTIMER_STxCSCTL)

Address offset: 0x58

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	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					CSFSTPW[3:0]				CSDTY[2:0]			CSPRD[3:0]			
					rw				rw			rw			

Bits	Fields	Descriptions
31:11	Reserved	Must be kept at reset value
10:7	CSFSTPW[3:0]	First carrier-signal pulse width This bit-field defines the first carrier-signal pulse width following a rising edge on channel output prepare signal(CHxOPRE). $t_{CSFSTPW}$ = (CSFSTPW[3:0]+1) x $t_{HRTIMER_CSGCK}$, $t_{HRTIMER_CSGCK}$ =16 x $t_{HRTIMER_CK}$. 0000: $t_{CSFSTPW}$ = $t_{HRTIMER_CSGCK}$ 0001: $t_{CSFSTPW}$ = 2*$t_{HRTIMER_CSGCK}$ … 1110: $t_{CSFSTPW}$ = 15*$t_{HRTIMER_CSGCK}$ 1111: $t_{CSFSTPW}$ = 16*$t_{HRTIMER_CSGCK}$
6:4	CSDTY[2:0]	Carrier signal duty cycle This bit-field defines the duty cycle of carrier signal (except the first pulse) which is equal to CSDTY[2:0]/8. 000: 0%(only the first pulse is present). 001: 12.5% 010: 25.0% 011: 37.5% 100: 50.0% 101: 62.5% 110: 75.0% 111: 87.5%
3:0	CSPRD[3:0]	Carrier signal period This bit-field defines the period of carrier signal (except the first pulse). t_{CSPRD} = (CSPRD[3:0]+1) x $t_{HRTIMER_CSGCK}$, $t_{HRTIMER_CSGCK}$ =16 x $t_{HRTIMER_CK}$. 0000: 16 x $t_{HRTIMER_CK}$ 0001: 32 x $t_{HRTIMER_CK}$ … 1111: 256 x $t_{HRTIMER_CK}$

HRTIMER Slave_TIMERx capture 0 trigger register (HRTIMER_STxCAP0TRG)

Address offset: 0x5C

Reset value: 0x0000 0000

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

For Slave_TIMER0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CP0BST4 CMP1	CP0BST4 CMP0	CP0BST4 NA	CP0BST4 A	CP0BST3 CMP1	CP0BST3 CMP0	CP0BST3 NA	CP0BST3 A	CP0BST2 MP1	CP0BST2 CMP0	CP0BST2 NA	CP0BST2 A	CP0BST1 CMP1	CP0BST1 CMP0	CP0BST1 NA	CP0BST1 A
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

Reserved	CP0BEXEV9	CP0BEXEV8	CP0BEXEV7	CP0BEXEV6	CP0BEXEV5	CP0BEXEV4	CP0BEXEV3	CP0BEXEV2	CP0BEXEV1	CP0BEXEV0	CP0BUP	CP0BSW
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31	CP0BST4CMP1	Capture 0 triggered by compare 1 event of Slave_TIMER4 This bit reserved only in Slave_TIMER4. Refer to CP0BST1CMP1 description.
30	CP0BST4CMP0	Capture 0 triggered by compare 0 event of Slave_TIMER4 This bit reserved only in Slave_TIMER4. Refer to CP0BST1CMP0 description.
29	CP0BST4NA	Capture 0 triggered by ST4CH0_O output active to inactive transition This bit reserved only in Slave_TIMER4. Refer to CP0BST1NA description.
28	CP0BST4A	Capture 0 triggered by ST4CH0_O output inactive to active transition This bit reserved only in Slave_TIMER4. Refer to CP0BST1A description.
27	CP0BST3CMP1	Capture 0 triggered by compare 1 event of Slave_TIMER3 This bit reserved only in Slave_TIMER3. Refer to CP0BST1CMP1 description.
26	CP0BST3CMP0	Capture 0 triggered by compare 0 event of Slave_TIMER3 This bit reserved only in Slave_TIMER3. Refer to CP0BST1CMP0 description.
25	CP0BST3NA	Capture 0 triggered by ST3CH0_O output active to inactive transition This bit reserved only in Slave_TIMER3. Refer to CP0BST1NA description.
24	CP0BST3A	Capture 0 triggered by ST3CH0_O output inactive to active transition This bit reserved only in Slave_TIMER3. Refer to CP0BST1A description.
23	CP0BST2CMP1	Capture 0 triggered by compare 1 event of Slave_TIMER2 This bit reserved only in Slave_TIMER2. Refer to CP0BST1CMP1 description.
22	CP0BST2CMP0	Capture 0 triggered by compare 0 event of Slave_TIMER2 This bit reserved only in Slave_TIMER2. Refer to CP0BST1CMP0 description.
21	CP0BST2NA	Capture 0 triggered by ST2CH0_O output active to inactive transition This bit reserved only in Slave_TIMER2. Refer to CP0BST1NA description.

20	CP0BST2A	Capture 0 triggered by ST2CH0_O output inactive to active transition This bit reserved only in Slave_TIMER2. Refer to CP0BST1A description.
19	CP0BST1CMP1	Capture 0 triggered by compare 1 event of Slave_TIMER1 This bit reserved only in Slave_TIMER1. 0: Capture 0 is not triggered by compare 1 event of Slave_TIMER1. 1: Capture 0 is triggered by compare 1 event of Slave_TIMER1.
18	CP0BST1CMP0	Capture 0 triggered by compare 0 event of Slave_TIMER1 This bit reserved only in Slave_TIMER1. 0: Capture 0 is not triggered by compare 0 event of Slave_TIMER1. 1: Capture 0 is triggered by compare 0 event of Slave_TIMER1.
17	CP0BST1NA	Capture 0 triggered by ST1CH0_O output active to inactive transition When the Slave_TIMER1 channel 0 output is converted from the active level to the inactive level, capture 0 is triggered by ST1CH0_O. This bit reserved only in Slave_TIMER1. 0: Capture 0 is not triggered by ST1CH0_O output active to inactive transition. 1: Capture 0 is triggered by ST1CH0_O output active to inactive transition.
16	CP0BST1A	Capture 0 triggered by ST1CH0_O output inactive to active transition When the Slave_TIMER1 channel 0 output is converted from the inactive level to the active, capture 0 is triggered by ST1CH0_O. This bit reserved only in Slave_TIMER1. 0: Capture 0 is not triggered by ST1CH0_O output inactive to active transition. 1: Capture 0 is triggered by ST1CH0_O output inactive to active transition.
15:12	Reserved	Must be kept at reset value
11	CP0BEXEV9	Capture 0 triggered by external event 9 Refer to CP0BEXEV0 description.
10	CP0BEXEV8	Capture 0 triggered by external event 8 Refer to CP0BEXEV0 description.
9	CP0BEXEV7	Capture 0 triggered by external event 7 Refer to CP0BEXEV0 description.
8	CP0BEXEV6	Capture 0 triggered by external event 6 Refer to CP0BEXEV0 description.
7	CP0BEXEV5	Capture 0 triggered by external event 5 Refer to CP0BEXEV0 description.
6	CP0BEXEV4	Capture 0 triggered by external event 4 Refer to CP0BEXEV0 description.
5	CP0BEXEV3	Capture 0 triggered by external event 3

		Refer to CP0BEXEV0 description.
4	CP0BEXEV2	Capture 0 triggered by external event 2 Refer to CP0BEXEV0 description.
3	CP0BEXEV1	Capture 0 triggered by external event 1 Refer to CP0BEXEV0 description.
2	CP0BEXEV0	Capture 0 triggered by external event 0 When the bit is set, capture 0 is triggered by external event 0 0: Capture 0 is not triggered by external event 0 1: Capture 0 is triggered by external event 0
1	CP0BUP	Capture 0 triggered by update event When the bit is set, capture 0 is triggered by update event 0: Capture 0 is not triggered by update event 1: Capture 0 is triggered by update event
0	CP0BSW	Capture 0 triggered by software This bit can be set by software, and cleared by hardware automatically. When the bit is set, it forces the capture 0 by software. 0: No capture 0 is triggered by software 1: Capture 0 is triggered by software

For Slave_TIMER1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CP0BST4CMP1	CP0BST4CMP0	CP0BST4NA	CP0BST4A	CP0BST3CMP1	CP0BST3CMP0	CP0BST3NA	CP0BST3A	CP0BST2MP1	CP0BST2CMP0	CP0BST2NA	CP0BST2A	Reserved			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CP0BST0CMP1	CP0BST0CMP0	CP0BST0NA	CP0BST0A	CP0BEXEV9	CP0BEXEV8	CP0BEXEV7	CP0BEXEV6	CP0BEXEV5	CP0BEXEV4	CP0BEXEV3	CP0BEXEV2	CP0BEXEV1	CP0BEXEV0	CP0BUP	CP0BSW
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31	CP0BST4CMP1	Capture 0 triggered by compare 1 event of Slave_TIMER4 This bit reserved only in Slave_TIMER4. Refer to CP0BST0CMP1 description.
30	CP0BST4CMP0	Capture 0 triggered by compare 0 event of Slave_TIMER4 This bit reserved only in Slave_TIMER4. Refer to CP0BST0CMP0 description.
29	CP0BST4NA	Capture 0 triggered by ST4CH0_O output active to inactive transition This bit reserved only in Slave_TIMER4. Refer to CP0BST0NA description.

28	CP0BST4A	Capture 0 triggered by ST4CH0_O output inactive to active transition This bit reserved only in Slave_TIMER4. Refer to CP0BST0A description.
27	CP0BST3CMP1	Capture 0 triggered by compare 1 event of Slave_TIMER3 This bit reserved only in Slave_TIMER3. Refer to CP0BST0CMP1 description.
26	CP0BST3CMP0	Capture 0 triggered by compare 0 event of Slave_TIMER3 This bit reserved only in Slave_TIMER3. Refer to CP0BST0CMP0 description.
25	CP0BST3NA	Capture 0 triggered by ST3CH0_O output active to inactive transition This bit reserved only in Slave_TIMER3. Refer to CP0BST0NA description.
24	CP0BST3A	Capture 0 triggered by ST3CH0_O output inactive to active transition This bit reserved only in Slave_TIMER3. Refer to CP0BST0A description.
23	CP0BST2CMP1	Capture 0 triggered by compare 1 event of Slave_TIMER2 This bit reserved only in Slave_TIMER2. Refer to CP0BST0CMP1 description.
22	CP0BST2CMP0	Capture 0 triggered by compare 0 event of Slave_TIMER2 This bit reserved only in Slave_TIMER2. Refer to CP0BST0CMP0 description.
21	CP0BST2NA	Capture 0 triggered by ST2CH0_O output active to inactive transition This bit reserved only in Slave_TIMER2. Refer to CP0BST0NA description.
20	CP0BST2A	Capture 0 triggered by ST2CH0_O output inactive to active transition This bit reserved only in Slave_TIMER2. Refer to CP0BST0A description.
19:16	Reserved	Must be kept at reset value
15	CP0BST0CMP1	Capture 0 triggered by compare 1 event of Slave_TIMER0 This bit reserved only in Slave_TIMER0. 0: Capture 0 is not triggered by compare 1 event of Slave_TIMER0. 1: Capture 0 is triggered by compare 1 event of Slave_TIMER0.
14	CP0BST0CMP0	Capture 0 triggered by compare 0 event of Slave_TIMER0 This bit reserved only in Slave_TIMER0. 0: Capture 0 is not triggered by compare 0 event of Slave_TIMER0. 1: Capture 0 is triggered by compare 0 event of Slave_TIMER0.
13	CP0BST0NA	Capture 0 triggered by ST0CH0_O output active to inactive transition

		When the Slave_TIMER0 channel 0 output is converted from the active level to the inactive level, capture 0 is triggered by ST0CH0_O. This bit reserved only in Slave_TIMER0. 0: Capture 0 is not triggered by ST0CH0_O output active to inactive transition. 1: Capture 0 is triggered by ST0CH0_O output active to inactive transition.
12	CP0BST0A	Capture 0 triggered by ST0CH0_O output inactive to active transition When the Slave_TIMER0 channel 0 output is converted from the inactive level to the active, capture 0 is triggered by ST0CH0_O. This bit reserved only in Slave_TIMER0. 0: Capture 0 is not triggered by ST0CH0_O output inactive to active transition. 1: Capture 0 is triggered by ST0CH0_O output inactive to active transition.
11	CP0BEXEV9	Capture 0 triggered by external event 9 Refer to CP0BEXEV0 description.
10	CP0BEXEV8	Capture 0 triggered by external event 8 Refer to CP0BEXEV0 description.
9	CP0BEXEV7	Capture 0 triggered by external event 7 Refer to CP0BEXEV0 description.
8	CP0BEXEV6	Capture 0 triggered by external event 6 Refer to CP0BEXEV0 description.
7	CP0BEXEV5	Capture 0 triggered by external event 5 Refer to CP0BEXEV0 description.
6	CP0BEXEV4	Capture 0 triggered by external event 4 Refer to CP0BEXEV0 description.
5	CP0BEXEV3	Capture 0 triggered by external event 3 Refer to CP0BEXEV0 description.
4	CP0BEXEV2	Capture 0 triggered by external event 2 Refer to CP0BEXEV0 description.
3	CP0BEXEV1	Capture 0 triggered by external event 1 Refer to CP0BEXEV0 description.
2	CP0BEXEV0	Capture 0 triggered by external event 0 When the bit is set, capture 0 is triggered by external event 0 0: Capture 0 is not triggered by external event 0 1: Capture 0 is triggered by external event 0
1	CP0BUP	Capture 0 triggered by update event When the bit is set, capture 0 is triggered by update event 0: Capture 0 is not triggered by update event 1: Capture 0 is triggered by update event

0	CP0BSW	Capture 0 triggered by software This bit can be set by software, and cleared by hardware automatically. When the bit is set, it forces the capture 0 by software. 0: No capture 0 is triggered by software 1: Capture 0 is triggered by software

For Slave_TIMER2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CP0BST4CMP1	CP0BST4CMP0	CP0BST4NA	CP0BST4A	CP0BST3CMP1	CP0BST3CMP0	CP0BST3NA	CP0BST3A	Reserved				CP0BST1CMP1	CP0BST1CMP0	CP0BST1NA	CP0BST1A
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
CP0BST0CMP1	CP0BST0CMP0	CP0BST0NA	CP0BST0A	CP0BEXEV9	CP0BEXEV8	CP0BEXEV7	CP0BEXEV6	CP0BEXEV5	CP0BEXEV4	CP0BEXEV3	CP0BEXEV2	CP0BEXEV1	CP0BEXEV0	CP0BUP	CP0BSW
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31	CP0BST4CMP1	Capture 0 triggered by compare 1 event of Slave_TIMER4 This bit reserved only in Slave_TIMER4. Refer to CP0BST0CMP1 description.
30	CP0BST4CMP0	Capture 0 triggered by compare 0 event of Slave_TIMER4 This bit reserved only in Slave_TIMER4. Refer to CP0BST0CMP0 description.
29	CP0BST4NA	Capture 0 triggered by ST4CH0_O output active to inactive transition This bit reserved only in Slave_TIMER4. Refer to CP0BST0NA description.
28	CP0BST4A	Capture 0 triggered by ST4CH0_O output inactive to active transition This bit reserved only in Slave_TIMER4. Refer to CP0BST0A description.
27	CP0BST3CMP1	Capture 0 triggered by compare 1 event of Slave_TIMER3 This bit reserved only in Slave_TIMER3. Refer to CP0BST0CMP1 description.
26	CP0BST3CMP0	Capture 0 triggered by compare 0 event of Slave_TIMER3 This bit reserved only in Slave_TIMER3. Refer to CP0BST0CMP0 description.
25	CP0BST3NA	Capture 0 triggered by ST3CH0_O output active to inactive transition This bit reserved only in Slave_TIMER3. Refer to CP0BST0NA description.
24	CP0BST3A	Capture 0 triggered by ST3CH0_O output inactive to active transition

		This bit reserved only in Slave_TIMER3. Refer to CP0BST0A description.
23:20	Reserved	Must be kept at reset value
19	CP0BST1CMP1	Capture 0 triggered by compare 1 event of Slave_TIMER1 This bit reserved only in Slave_TIMER1. Refer to CP0BST0CMP1 description.
18	CP0BST1CMP0	Capture 0 triggered by compare 0 event of Slave_TIMER1 This bit reserved only in Slave_TIMER1. Refer to CP0BST0CMP0 description.
17	CP0BST1NA	Capture 0 triggered by ST1CH0_O output active to inactive transition This bit reserved only in Slave_TIMER1. Refer to CP0BST0NA description.
16	CP0BST1A	Capture 0 triggered by ST1CH0_O output inactive to active transition This bit reserved only in Slave_TIMER1. Refer to CP0BST0A description.
15	CP0BST0CMP1	Capture 0 triggered by compare 1 event of Slave_TIMER0 This bit reserved only in Slave_TIMER0. 0: Capture 0 is not triggered by compare 1 event of Slave_TIMER0. 1: Capture 0 is triggered by compare 1 event of Slave_TIMER0.
14	CP0BST0CMP0	Capture 0 triggered by compare 0 event of Slave_TIMER0 This bit reserved only in Slave_TIMER0. 0: Capture 0 is not triggered by compare 0 event of Slave_TIMER0. 1: Capture 0 is triggered by compare 0 event of Slave_TIMER0.
13	CP0BST0NA	Capture 0 triggered by ST0CH0_O output active to inactive transition When the Slave_TIMER0 channel 0 output is converted from the active level to the inactive level, capture 0 is triggered by ST0CH0_O. This bit reserved only in Slave_TIMER0. 0: Capture 0 is not triggered by ST0CH0_O output active to inactive transition. 1: Capture 0 is triggered by ST0CH0_O output active to inactive transition.
12	CP0BST0A	Capture 0 triggered by ST0CH0_O output inactive to active transition When the Slave_TIMER0 channel 0 output is converted from the inactive level to the active, capture 0 is triggered by ST0CH0_O. This bit reserved only in Slave_TIMER0. 0: Capture 0 is not triggered by ST0CH0_O output inactive to active transition. 1: Capture 0 is triggered by ST0CH0_O output inactive to active transition.
11	CP0BEXEV9	Capture 0 triggered by external event 9 Refer to CP0BEXEV0 description.
10	CP0BEXEV8	Capture 0 triggered by external event 8

		Refer to CP0BEXEV0 description.
9	CP0BEXEV7	Capture 0 triggered by external event 7 Refer to CP0BEXEV0 description.
8	CP0BEXEV6	Capture 0 triggered by external event 6 Refer to CP0BEXEV0 description.
7	CP0BEXEV5	Capture 0 triggered by external event 5 Refer to CP0BEXEV0 description.
6	CP0BEXEV4	Capture 0 triggered by external event 4 Refer to CP0BEXEV0 description.
5	CP0BEXEV3	Capture 0 triggered by external event 3 Refer to CP0BEXEV0 description.
4	CP0BEXEV2	Capture 0 triggered by external event 2 Refer to CP0BEXEV0 description.
3	CP0BEXEV1	Capture 0 triggered by external event 1 Refer to CP0BEXEV0 description.
2	CP0BEXEV0	Capture 0 triggered by external event 0 When the bit is set, capture 0 is triggered by external event 0 0: Capture 0 is not triggered by external event 0 1: Capture 0 is triggered by external event 0
1	CP0BUP	Capture 0 triggered by update event When the bit is set, capture 0 is triggered by update event 0: Capture 0 is not triggered by update event 1: Capture 0 is triggered by update event
0	CP0BSW	Capture 0 triggered by software This bit can be set by software, and cleared by hardware automatically. When the bit is set, it forces the capture 0 by software. 0: No capture 0 is triggered by software 1: Capture 0 is triggered by software

For Slave_TIMER3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CP0BST4 CMP1	CP0BST4 CMP0	CP0BST4 NA	CP0BST4 A	Reserved				CP0BST2 MP1	CP0BST2 CMP0	CP0BST2 NA	CP0BST2 A	CP0BST1 CMP1	CP0BST1 CMP0	CP0BST1 NA	CP0BST1 A
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
CP0BST0 CMP1	CP0BST0 CMP0	CP0BST0 NA	CP0BST0 A	CP0BEX EV9	CP0BEX EV8	CP0BEX EV7	CP0BEX EV6	CP0BEX EV5	CP0BEX EV4	CP0BEX EV3	CP0BEX EV2	CP0BEX EV1	CP0BEX EV0	CP0BUP	CP0BSW
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31	CP0BST4CMP1	Capture 0 triggered by compare 1 event of Slave_TIMER4 This bit reserved only in Slave_TIMER4. Refer to CP0BST0CMP1 description.
30	CP0BST4CMP0	Capture 0 triggered by compare 0 event of Slave_TIMER4 This bit reserved only in Slave_TIMER4. Refer to CP0BST0CMP0 description.
29	CP0BST4NA	Capture 0 triggered by ST4CH0_O output active to inactive transition This bit reserved only in Slave_TIMER4. Refer to CP0BST0NA description.
28	CP0BST4A	Capture 0 triggered by ST4CH0_O output inactive to active transition This bit reserved only in Slave_TIMER4. Refer to CP0BST0A description.
27:24	Reserved	Must be kept at reset value
23	CP0BST2CMP1	Capture 0 triggered by compare 1 event of Slave_TIMER2 This bit reserved only in Slave_TIMER2. Refer to CP0BST0CMP1 description.
22	CP0BST2CMP0	Capture 0 triggered by compare 0 event of Slave_TIMER2 This bit reserved only in Slave_TIMER2. Refer to CP0BST0CMP0 description.
21	CP0BST2NA	Capture 0 triggered by ST2CH0_O output active to inactive transition This bit reserved only in Slave_TIMER2. Refer to CP0BST0NA description.
20	CP0BST2A	Capture 0 triggered by ST2CH0_O output inactive to active transition This bit reserved only in Slave_TIMER2. Refer to CP0BST0A description.
19	CP0BST1CMP1	Capture 0 triggered by compare 1 event of Slave_TIMER1 This bit reserved only in Slave_TIMER1. Refer to CP0BST0CMP1 description.
18	CP0BST1CMP0	Capture 0 triggered by compare 0 event of Slave_TIMER1 This bit reserved only in Slave_TIMER1. Refer to CP0BST0CMP0 description.
17	CP0BST1NA	Capture 0 triggered by ST1CH0_O output active to inactive transition This bit reserved only in Slave_TIMER1. Refer to CP0BST0NA description.
16	CP0BST1A	Capture 0 triggered by ST1CH0_O output inactive to active transition

		This bit reserved only in Slave_TIMER1. Refer to CP0BST0A description.
15	CP0BST0CMP1	Capture 0 triggered by compare 1 event of Slave_TIMER0 This bit reserved only in Slave_TIMER0. 0: Capture 0 is not triggered by compare 1 event of Slave_TIMER0. 1: Capture 0 is triggered by compare 1 event of Slave_TIMER0.
14	CP0BST0CMP0	Capture 0 triggered by compare 0 event of Slave_TIMER0 This bit reserved only in Slave_TIMER0. 0: Capture 0 is not triggered by compare 0 event of Slave_TIMER0. 1: Capture 0 is triggered by compare 0 event of Slave_TIMER0.
13	CP0BST0NA	Capture 0 triggered by ST0CH0_O output active to inactive transition When the Slave_TIMER0 channel 0 output is converted from the active level to the inactive level, capture 0 is triggered by ST0CH0_O. This bit reserved only in Slave_TIMER0. 0: Capture 0 is not triggered by ST0CH0_O output active to inactive transition. 1: Capture 0 is triggered by ST0CH0_O output active to inactive transition.
12	CP0BST0A	Capture 0 triggered by ST0CH0_O output inactive to active transition When the Slave_TIMER0 channel 0 output is converted from the inactive level to the active, capture 0 is triggered by ST0CH0_O. This bit reserved only in Slave_TIMER0. 0: Capture 0 is not triggered by ST0CH0_O output inactive to active transition. 1: Capture 0 is triggered by ST0CH0_O output inactive to active transition.
11	CP0BEXEV9	Capture 0 triggered by external event 9 Refer to CP0BEXEV0 description.
10	CP0BEXEV8	Capture 0 triggered by external event 8 Refer to CP0BEXEV0 description.
9	CP0BEXEV7	Capture 0 triggered by external event 7 Refer to CP0BEXEV0 description.
8	CP0BEXEV6	Capture 0 triggered by external event 6 Refer to CP0BEXEV0 description.
7	CP0BEXEV5	Capture 0 triggered by external event 5 Refer to CP0BEXEV0 description.
6	CP0BEXEV4	Capture 0 triggered by external event 4 Refer to CP0BEXEV0 description.
5	CP0BEXEV3	Capture 0 triggered by external event 3 Refer to CP0BEXEV0 description.
4	CP0BEXEV2	Capture 0 triggered by external event 2

		Refer to CP0BEXEV0 description.
3	CP0BEXEV1	Capture 0 triggered by external event 1 Refer to CP0BEXEV0 description.
2	CP0BEXEV0	Capture 0 triggered by external event 0 When the bit is set, capture 0 is triggered by external event 0 0: Capture 0 is not triggered by external event 0 1: Capture 0 is triggered by external event 0
1	CP0BUP	Capture 0 triggered by update event When the bit is set, capture 0 is triggered by update event 0: Capture 0 is not triggered by update event 1: Capture 0 is triggered by update event
0	CP0BSW	Capture 0 triggered by software This bit can be set by software, and cleared by hardware automatically. When the bit is set, it forces the capture 0 by software. 0: No capture 0 is triggered by software 1: Capture 0 is triggered by software

For Slave_TIMER4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				CP0BST3CMP1	CP0BST3CMP0	CP0BST3NA	CP0BST3A	CP0BST2MP1	CP0BST2CMP0	CP0BST2NA	CP0BST2A	CP0BST1CMP1	CP0BST1CMP0	CP0BST1NA	CP0BST1A
				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
CP0BST0CMP1	CP0BST0CMP0	CP0BST0NA	CP0BST0A	CP0BEXEV9	CP0BEXEV8	CP0BEXEV7	CP0BEXEV6	CP0BEXEV5	CP0BEXEV4	CP0BEXEV3	CP0BEXEV2	CP0BEXEV1	CP0BEXEV0	CP0BUP	CP0BSW
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	CP0BST3CMP1	Capture 0 triggered by compare 1 event of Slave_TIMER3 This bit reserved only in Slave_TIMER3. Refer to CP0BST0CMP1 description.
26	CP0BST3CMP0	Capture 0 triggered by compare 0 event of Slave_TIMER3 This bit reserved only in Slave_TIMER3. Refer to CP0BST0CMP0 description.
25	CP0BST3NA	Capture 0 triggered by ST3CH0_O output active to inactive transition This bit reserved only in Slave_TIMER3. Refer to CP0BST0NA description.
24	CP0BST3A	Capture 0 triggered by ST3CH0_O output inactive to active transition

This bit reserved only in Slave_TIMER3.
Refer to CP0BST0A description.

23	CP0BST2CMP1	Capture 0 triggered by compare 1 event of Slave_TIMER2 This bit reserved only in Slave_TIMER2. Refer to CP0BST0CMP1 description.
22	CP0BST2CMP0	Capture 0 triggered by compare 0 event of Slave_TIMER2 This bit reserved only in Slave_TIMER2. Refer to CP0BST0CMP0 description.
21	CP0BST2NA	Capture 0 triggered by ST2CH0_O output active to inactive transition This bit reserved only in Slave_TIMER2. Refer to CP0BST0NA description.
20	CP0BST2A	Capture 0 triggered by ST2CH0_O output inactive to active transition This bit reserved only in Slave_TIMER2. Refer to CP0BST0A description.
19	CP0BST1CMP1	Capture 0 triggered by compare 1 event of Slave_TIMER1 This bit reserved only in Slave_TIMER1. Refer to CP0BST0CMP1 description.
18	CP0BST1CMP0	Capture 0 triggered by compare 0 event of Slave_TIMER1 This bit reserved only in Slave_TIMER1. Refer to CP0BST0CMP0 description.
17	CP0BST1NA	Capture 0 triggered by ST1CH0_O output active to inactive transition This bit reserved only in Slave_TIMER1. Refer to CP0BST0NA description.
16	CP0BST1A	Capture 0 triggered by ST1CH0_O output inactive to active transition This bit reserved only in Slave_TIMER1. Refer to CP0BST0A description.
15	CP0BST0CMP1	Capture 0 triggered by compare 1 event of Slave_TIMER0 This bit reserved only in Slave_TIMER0. 0: Capture 0 is not triggered by compare 1 event of Slave_TIMER0. 1: Capture 0 is triggered by compare 1 event of Slave_TIMER0.
14	CP0BST0CMP0	Capture 0 triggered by compare 0 event of Slave_TIMER0 This bit reserved only in Slave_TIMER0. 0: Capture 0 is not triggered by compare 0 event of Slave_TIMER0. 1: Capture 0 is triggered by compare 0 event of Slave_TIMER0.
13	CP0BST0NA	Capture 0 triggered by ST0CH0_O output active to inactive transition When the Slave_TIMER0 channel 0 output is converted from the active level to the inactive level, capture 0 is triggered by ST0CH0_O. This bit reserved only in Slave_TIMER0.

		0: Capture 0 is not triggered by ST0CH0_O output active to inactive transition. 1: Capture 0 is triggered by ST0CH0_O output active to inactive transition.
12	CP0BST0A	Capture 0 triggered by ST0CH0_O output inactive to active transition When the Slave_TIMER0 channel 0 output is converted from the inactive level to the active, capture 0 is triggered by ST0CH0_O. This bit reserved only in Slave_TIMER0. 0: Capture 0 is not triggered by ST0CH0_O output inactive to active transition. 1: Capture 0 is triggered by ST0CH0_O output inactive to active transition.
11	CP0BEXEV9	Capture 0 triggered by external event 9 Refer to CP0BEXEV0 description.
10	CP0BEXEV8	Capture 0 triggered by external event 8 Refer to CP0BEXEV0 description.
9	CP0BEXEV7	Capture 0 triggered by external event 7 Refer to CP0BEXEV0 description.
8	CP0BEXEV6	Capture 0 triggered by external event 6 Refer to CP0BEXEV0 description.
7	CP0BEXEV5	Capture 0 triggered by external event 5 Refer to CP0BEXEV0 description.
6	CP0BEXEV4	Capture 0 triggered by external event 4 Refer to CP0BEXEV0 description.
5	CP0BEXEV3	Capture 0 triggered by external event 3 Refer to CP0BEXEV0 description.
4	CP0BEXEV2	Capture 0 triggered by external event 2 Refer to CP0BEXEV0 description.
3	CP0BEXEV1	Capture 0 triggered by external event 1 Refer to CP0BEXEV0 description.
2	CP0BEXEV0	Capture 0 triggered by external event 0 When the bit is set, capture 0 is triggered by external event 0 0: Capture 0 is not triggered by external event 0 1: Capture 0 is triggered by external event 0
1	CP0BUP	Capture 0 triggered by update event When the bit is set, capture 0 is triggered by update event 0: Capture 0 is not triggered by update event 1: Capture 0 is triggered by update event
0	CP0BSW	Capture 0 triggered by software This bit can be set by software, and cleared by hardware automatically. When the bit is set, it forces the capture 0 by software.

0: No capture 0 is triggered by software

1: Capture 0 is triggered by software

For Slave_TIMER5 - For Slave_TIMER7

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CP0BST4CMP1	CP0BST4CMP0	CP0BST4NA	CP0BST4A	CP0BST3CMP1	CP0BST3CMP0	CP0BST3NA	CP0BST3A	CP0BST2MP1	CP0BST2CMP0	CP0BST2NA	CP0BST2A	CP0BST1CMP1	CP0BST1CMP0	CP0BST1NA	CP0BST1A
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
CP0BST0CMP1	CP0BST0CMP0	CP0BST0NA	CP0BST0A	CP0BEXEV9	CP0BEXEV8	CP0BEXEV7	CP0BEXEV6	CP0BEXEV5	CP0BEXEV4	CP0BEXEV3	CP0BEXEV2	CP0BEXEV1	CP0BEXEV0	CP0BUP	CP0BSW
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31	CP0BST4CMP1	Capture 0 triggered by compare 1 event of Slave_TIMER4 This bit reserved only in Slave_TIMER4. Refer to CP0BST0CMP1 description.
30	CP0BST4CMP0	Capture 0 triggered by compare 0 event of Slave_TIMER4 This bit reserved only in Slave_TIMER4. Refer to CP0BST0CMP0 description.
29	CP0BST4NA	Capture 0 triggered by ST4CH0_O output active to inactive transition This bit reserved only in Slave_TIMER4. Refer to CP0BST0NA description.
28	CP0BST4A	Capture 0 triggered by ST4CH0_O output inactive to active transition This bit reserved only in Slave_TIMER4. Refer to CP0BST0A description.
27	CP0BST3CMP1	Capture 0 triggered by compare 1 event of Slave_TIMER3 This bit reserved only in Slave_TIMER3. Refer to CP0BST0CMP1 description.
26	CP0BST3CMP0	Capture 0 triggered by compare 0 event of Slave_TIMER3 This bit reserved only in Slave_TIMER3. Refer to CP0BST0CMP0 description.
25	CP0BST3NA	Capture 0 triggered by ST3CH0_O output active to inactive transition This bit reserved only in Slave_TIMER3. Refer to CP0BST0NA description.
24	CP0BST3A	Capture 0 triggered by ST3CH0_O output inactive to active transition This bit reserved only in Slave_TIMER3. Refer to CP0BST0A description.
23	CP0BST2CMP1	Capture 0 triggered by compare 1 event of Slave_TIMER2

		This bit reserved only in Slave_TIMER2. Refer to CP0BST0CMP1 description.
22	CP0BST2CMP0	Capture 0 triggered by compare 0 event of Slave_TIMER2 This bit reserved only in Slave_TIMER2. Refer to CP0BST0CMP0 description.
21	CP0BST2NA	Capture 0 triggered by ST2CH0_O output active to inactive transition This bit reserved only in Slave_TIMER2. Refer to CP0BST0NA description.
20	CP0BST2A	Capture 0 triggered by ST2CH0_O output inactive to active transition This bit reserved only in Slave_TIMER2. Refer to CP0BST0A description.
19	CP0BST1CMP1	Capture 0 triggered by compare 1 event of Slave_TIMER1 This bit reserved only in Slave_TIMER1. Refer to CP0BST0CMP1 description.
18	CP0BST1CMP0	Capture 0 triggered by compare 0 event of Slave_TIMER1 This bit reserved only in Slave_TIMER1. Refer to CP0BST0CMP0 description.
17	CP0BST1NA	Capture 0 triggered by ST1CH0_O output active to inactive transition This bit reserved only in Slave_TIMER1. Refer to CP0BST0NA description.
16	CP0BST1A	Capture 0 triggered by ST1CH0_O output inactive to active transition This bit reserved only in Slave_TIMER1. Refer to CP0BST0A description.
15	CP0BST0CMP1	Capture 0 triggered by compare 1 event of Slave_TIMER0 This bit reserved only in Slave_TIMER0. 0: Capture 0 is not triggered by compare 1 event of Slave_TIMER0. 1: Capture 0 is triggered by compare 1 event of Slave_TIMER0.
14	CP0BST0CMP0	Capture 0 triggered by compare 0 event of Slave_TIMER0 This bit reserved only in Slave_TIMER0. 0: Capture 0 is not triggered by compare 0 event of Slave_TIMER0. 1: Capture 0 is triggered by compare 0 event of Slave_TIMER0.
13	CP0BST0NA	Capture 0 triggered by ST0CH0_O output active to inactive transition When the Slave_TIMER0 channel 0 output is converted from the active level to the inactive level, capture 0 is triggered by ST0CH0_O. This bit reserved only in Slave_TIMER0. 0: Capture 0 is not triggered by ST0CH0_O output active to inactive transition. 1: Capture 0 is triggered by ST0CH0_O output active to inactive transition.
12	CP0BST0A	Capture 0 triggered by ST0CH0_O output inactive to active transition

		When the Slave_TIMER0 channel 0 output is converted from the inactive level to the active, capture 0 is triggered by ST0CH0_O. This bit reserved only in Slave_TIMER0. 0: Capture 0 is not triggered by ST0CH0_O output inactive to active transition. 1: Capture 0 is triggered by ST0CH0_O output inactive to active transition.
11	CP0BEXEV9	Capture 0 triggered by external event 9 Refer to CP0BEXEV0 description.
10	CP0BEXEV8	Capture 0 triggered by external event 8 Refer to CP0BEXEV0 description.
9	CP0BEXEV7	Capture 0 triggered by external event 7 Refer to CP0BEXEV0 description.
8	CP0BEXEV6	Capture 0 triggered by external event 6 Refer to CP0BEXEV0 description.
7	CP0BEXEV5	Capture 0 triggered by external event 5 Refer to CP0BEXEV0 description.
6	CP0BEXEV4	Capture 0 triggered by external event 4 Refer to CP0BEXEV0 description.
5	CP0BEXEV3	Capture 0 triggered by external event 3 Refer to CP0BEXEV0 description.
4	CP0BEXEV2	Capture 0 triggered by external event 2 Refer to CP0BEXEV0 description.
3	CP0BEXEV1	Capture 0 triggered by external event 1 Refer to CP0BEXEV0 description.
2	CP0BEXEV0	Capture 0 triggered by external event 0 When the bit is set, capture 0 is triggered by external event 0 0: Capture 0 is not triggered by external event 0 1: Capture 0 is triggered by external event 0
1	CP0BUP	Capture 0 triggered by update event When the bit is set, capture 0 is triggered by update event 0: Capture 0 is not triggered by update event 1: Capture 0 is triggered by update event
0	CP0BSW	Capture 0 triggered by software This bit can be set by software, and cleared by hardware automatically. When the bit is set, it forces the capture 0 by software. 0: No capture 0 is triggered by software 1: Capture 0 is triggered by software

HRTIMER Slave_TIMERx capture 1 trigger register (HRTIMER_STxCAP1TRG)

Address offset: 0x60

Reset value: 0x0000 0000

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

For Slave_TIMER0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CP1BST4CMP1	CP1BST4CMP0	CP1BST4NA	CP1BST4A	CP1BST3CMP1	CP1BST3CMP0	CP1BST3NA	CP1BST3A	CP1BST2MP1	CP1BST2CMP0	CP1BST2NA	CP1BST2A	CP1BST1CMP1	CP1BST1CMP0	CP1BST1NA	CP1BST1A
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
Reserved				CP1BEXEV9	CP1BEXEV8	CP1BEXEV7	CP1BEXEV6	CP1BEXEV5	CP1BEXEV4	CP1BEXEV3	CP1BEXEV2	CP1BEXEV1	CP1BEXEV0	CP1BUP	CP1BSW
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31	CP1BST4CMP1	Capture 1 triggered by compare 1 event of Slave_TIMER4 This bit reserved only in Slave_TIMER4. Refer to CP1BST0CMP1 description.
30	CP1BST4CMP0	Capture 1 triggered by compare 0 event of Slave_TIMER4 This bit reserved only in Slave_TIMER4. Refer to CP1BST0CMP0 description.
29	CP1BST4NA	Capture 1 triggered by ST4CH0_O output active to inactive transition. This bit reserved only in Slave_TIMER4. Refer to CP1BST0NA description.
28	CP1BST4A	Capture 1 triggered by ST4CH0_O output inactive to active transition This bit reserved only in Slave_TIMER4. Refer to CP1BST0A description.
27	CP1BST3CMP1	Capture 1 triggered by compare 1 event of Slave_TIMER3 This bit reserved only in Slave_TIMER3. Refer to CP1BST0CMP1 description.
26	CP1BST3CMP0	Capture 1 triggered by compare 0 event of Slave_TIMER3 This bit reserved only in Slave_TIMER3. Refer to CP1BST0CMP0 description.
25	CP1BST3NA	Capture 1 triggered by ST3CH0_O output active to inactive transition This bit reserved only in Slave_TIMER3. Refer to CP1BST0NA description.
24	CP1BST3A	Capture 1 triggered by ST3CH0_O output inactive to active transition This bit reserved only in Slave_TIMER3.

		Refer to CP1BST0A description.
23	CP1BST2CMP1	Capture 1 triggered by compare 1 event of Slave_TIMER2 This bit reserved only in Slave_TIMER2. Refer to CP1BST0CMP1 description.
22	CP1BST2CMP0	Capture 1 triggered by compare 0 event of Slave_TIMER2 This bit reserved only in Slave_TIMER2. Refer to CP1BST0CMP0 description.
21	CP1BST2NA	Capture 1 triggered by ST2CH0_O output active to inactive transition This bit reserved only in Slave_TIMER2. Refer to CP1BST0NA description.
20	CP1BST2A	Capture 1 triggered by ST2CH0_O output inactive to active transition This bit reserved only in Slave_TIMER2. Refer to CP1BST0A description.
19	CP1BST1CMP1	Capture 1 triggered by compare 1 event of Slave_TIMER1 This bit reserved only in Slave_TIMER1. Refer to CP1BST0CMP1 description.
18	CP1BST1CMP0	Capture 1 triggered by compare 0 event of Slave_TIMER1 This bit reserved only in Slave_TIMER1. Refer to CP1BST0CMP0 description.
17	CP1BST1NA	Capture 1 triggered by ST1CH0_O output active to inactive transition This bit reserved only in Slave_TIMER1. Refer to CP1BST0NA description.
16	CP1BST1A	Capture 1 triggered by ST1CH0_O output inactive to active transition This bit reserved only in Slave_TIMER1. Refer to CP1BST0A description.
15:12	Reserved	Must be kept at reset value
11	CP1BEXEV9	Capture 1 triggered by external event 9 Refer to CP1BEXEV0 description.
10	CP1BEXEV8	Capture 1 triggered by external event 8 Refer to CP1BEXEV0 description.
9	CP1BEXEV7	Capture 1 triggered by external event 7 Refer to CP1BEXEV0 description.
8	CP1BEXEV6	Capture 1 triggered by external event 6 Refer to CP1BEXEV0 description.
7	CP1BEXEV5	Capture 1 triggered by external event 5 Refer to CP1BEXEV0 description.

6	CP1BEXEV4	Capture 1 triggered by external event 4 Refer to CP1BEXEV0 description.
5	CP1BEXEV3	Capture 1 triggered by external event 3 Refer to CP1BEXEV0 description.
4	CP1BEXEV2	Capture 1 triggered by external event 2 Refer to CP1BEXEV0 description.
3	CP1BEXEV1	Capture 1 triggered by external event 1 Refer to CP1BEXEV0 description.
2	CP1BEXEV0	Capture 1 triggered by external event 0 When the bit is set, capture 1 is triggered by update external event 0 0: Capture 1 is not triggered by external event 0 1: Capture 1 is triggered by external event 0
1	CP1BUP	Capture 1 triggered by update event When the bit is set, capture 1 is triggered by update event 0: Capture 1 is not triggered by update event 1: Capture 1 is triggered by update event
0	CP1BSW	Capture 1 triggered by software This bit can be set by software, and cleared by hardware automatically. When the bit is set, it forces the capture 1 by software. 0: No capture 1 triggered by software 1: Capture 1 triggered by software

For Slave_TIMER1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CP1BST4CMP1	CP1BST4CMP0	CP1BST4NA	CP1BST4A	CP1BST3CMP1	CP1BST3CMP0	CP1BST3NA	CP1BST3A	CP1BST2MP1	CP1BST2CMP0	CP1BST2NA	CP1BST2A	Reserved			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CP1BST0CMP1	CP1BST0CMP0	CP1BST0NA	CP1BST0A	CP1BEXEV9	CP1BEXEV8	CP1BEXEV7	CP1BEXEV6	CP1BEXEV5	CP1BEXEV4	CP1BEXEV3	CP1BEXEV2	CP1BEXEV1	CP1BEXEV0	CP1BUP	CP1BSW
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31	CP1BST4CMP1	Capture 1 triggered by compare 1 event of Slave_TIMER4 This bit reserved only in Slave_TIMER4. Refer to CP1BST0CMP1 description.
30	CP1BST4CMP0	Capture 1 triggered by compare 0 event of Slave_TIMER4 This bit reserved only in Slave_TIMER4. Refer to CP1BST0CMP0 description.

29	CP1BST4NA	Capture 1 triggered by ST4CH0_O output active to inactive transition. This bit reserved only in Slave_TIMER4. Refer to CP1BST0NA description.
28	CP1BST4A	Capture 1 triggered by ST4CH0_O output inactive to active transition This bit reserved only in Slave_TIMER4. Refer to CP1BST0A description.
27	CP1BST3CMP1	Capture 1 triggered by compare 1 event of Slave_TIMER3 This bit reserved only in Slave_TIMER3. Refer to CP1BST0CMP1 description.
26	CP1BST3CMP0	Capture 1 triggered by compare 0 event of Slave_TIMER3 This bit reserved only in Slave_TIMER3. Refer to CP1BST0CMP0 description.
25	CP1BST3NA	Capture 1 triggered by ST3CH0_O output active to inactive transition This bit reserved only in Slave_TIMER3. Refer to CP1BST0NA description.
24	CP1BST3A	Capture 1 triggered by ST3CH0_O output inactive to active transition This bit reserved only in Slave_TIMER3. Refer to CP1BST0A description.
23	CP1BST2CMP1	Capture 1 triggered by compare 1 event of Slave_TIMER2 This bit reserved only in Slave_TIMER2. Refer to CP1BST0CMP1 description.
22	CP1BST2CMP0	Capture 1 triggered by compare 0 event of Slave_TIMER2 This bit reserved only in Slave_TIMER2. Refer to CP1BST0CMP0 description.
21	CP1BST2NA	Capture 1 triggered by ST2CH0_O output active to inactive transition This bit reserved only in Slave_TIMER2. Refer to CP1BST0NA description.
20	CP1BST2A	Capture 1 triggered by ST2CH0_O output inactive to active transition This bit reserved only in Slave_TIMER2. Refer to CP1BST0A description.
19:16	Reserved	Must be kept at reset value
15	CP1BST0CMP1	Capture 1 triggered by compare 1 event of Slave_TIMER0 This bit reserved only in Slave_TIMER0. 0: Capture 1 is not triggered by compare 1 event of Slave_TIMER0. 1: Capture 1 is triggered by compare 1 event of Slave_TIMER0.
14	CP1BST0CMP0	Capture 1 triggered by compare 0 event of Slave_TIMER0 This bit reserved only in Slave_TIMER0.

		0: Capture 1 is not triggered by compare 0 event of Slave_TIMER0. 1: Capture 1 is triggered by compare 0 event of Slave_TIMER0.
13	CP1BST0NA	Capture 1 triggered by ST0CH0_O output active to inactive transition When the Slave_TIMER0 channel 0 output is converted from the active level to the inactive level, capture 1 is triggered by ST0CH0_O. This bit reserved only in Slave_TIMER0. 0: Capture 1 is not triggered by ST0CH0_O output active to inactive transition. 1: Capture 1 is triggered by ST0CH0_O output active to inactive transition.
12	CP1BST0A	Capture 1 triggered by ST0CH0_O output inactive to active transition When the Slave_TIMER0 channel 0 output is converted from the inactive level to the active, capture 1 is triggered by ST0CH0_O. This bit reserved only in Slave_TIMER0. 0: Capture 1 is not triggered by ST0CH0_O output inactive to active transition. 1: Capture 1 is triggered by ST0CH0_O output inactive to active transition.
11	CP1BEXEV9	Capture 1 triggered by external event 9 Refer to CP1BEXEV0 description.
10	CP1BEXEV8	Capture 1 triggered by external event 8 Refer to CP1BEXEV0 description.
9	CP1BEXEV7	Capture 1 triggered by external event 7 Refer to CP1BEXEV0 description.
8	CP1BEXEV6	Capture 1 triggered by external event 6 Refer to CP1BEXEV0 description.
7	CP1BEXEV5	Capture 1 triggered by external event 5 Refer to CP1BEXEV0 description.
6	CP1BEXEV4	Capture 1 triggered by external event 4 Refer to CP1BEXEV0 description.
5	CP1BEXEV3	Capture 1 triggered by external event 3 Refer to CP1BEXEV0 description.
4	CP1BEXEV2	Capture 1 triggered by external event 2 Refer to CP1BEXEV0 description.
3	CP1BEXEV1	Capture 1 triggered by external event 1 Refer to CP1BEXEV0 description.
2	CP1BEXEV0	Capture 1 triggered by external event 0 When the bit is set, capture 1 is triggered by update external event 0 0: Capture 1 is not triggered by external event 0 1: Capture 1 is triggered by external event 0
1	CP1BUP	Capture 1 triggered by update event

When the bit is set, capture 1 is triggered by update event
0: Capture 1 is not triggered by update event
1: Capture 1 is triggered by update event

0 CP1BSW Capture 1 triggered by software
This bit can be set by software, and cleared by hardware automatically. When the bit is set, it forces the capture 1 by software.
0: No capture 1 triggered by software
1: Capture 1 triggered by software

For Slave_TIMER2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CP1BST4CMP1	CP1BST4CMP0	CP1BST4NA	CP1BST4A	CP1BST3CMP1	CP1BST3CMP0	CP1BST3NA	CP1BST3A	Reserved				CP1BST1CMP1	CP1BST1CMP0	CP1BST1NA	CP1BST1A
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
CP1BST0CMP1	CP1BST0CMP0	CP1BST0NA	CP1BST0A	CP1BEXEV9	CP1BEXEV8	CP1BEXEV7	CP1BEXEV6	CP1BEXEV5	CP1BEXEV4	CP1BEXEV3	CP1BEXEV2	CP1BEXEV1	CP1BEXEV0	CP1BUP	CP1BSW
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31	CP1BST4CMP1	Capture 1 triggered by compare 1 event of Slave_TIMER4 This bit reserved only in Slave_TIMER4. Refer to CP1BST0CMP1 description.
30	CP1BST4CMP0	Capture 1 triggered by compare 0 event of Slave_TIMER4 This bit reserved only in Slave_TIMER4. Refer to CP1BST0CMP0 description.
29	CP1BST4NA	Capture 1 triggered by ST4CH0_O output active to inactive transition. This bit reserved only in Slave_TIMER4. Refer to CP1BST0NA description.
28	CP1BST4A	Capture 1 triggered by ST4CH0_O output inactive to active transition This bit reserved only in Slave_TIMER4. Refer to CP1BST0A description.
27	CP1BST3CMP1	Capture 1 triggered by compare 1 event of Slave_TIMER3 This bit reserved only in Slave_TIMER3. Refer to CP1BST0CMP1 description.
26	CP1BST3CMP0	Capture 1 triggered by compare 0 event of Slave_TIMER3 This bit reserved only in Slave_TIMER3. Refer to CP1BST0CMP0 description.
25	CP1BST3NA	Capture 1 triggered by ST3CH0_O output active to inactive transition

		This bit reserved only in Slave_TIMER3. Refer to CP1BST0NA description.
24	CP1BST3A	Capture 1 triggered by ST3CH0_O output inactive to active transition This bit reserved only in Slave_TIMER3. Refer to CP1BST0A description.
23:20	Reserved	Must be kept at reset value
19	CP1BST1CMP1	Capture 1 triggered by compare 1 event of Slave_TIMER1 This bit reserved only in Slave_TIMER1. Refer to CP1BST0CMP1 description.
18	CP1BST1CMP0	Capture 1 triggered by compare 0 event of Slave_TIMER1 This bit reserved only in Slave_TIMER1. Refer to CP1BST0CMP0 description.
17	CP1BST1NA	Capture 1 triggered by ST1CH0_O output active to inactive transition This bit reserved only in Slave_TIMER1. Refer to CP1BST0NA description.
16	CP1BST1A	Capture 1 triggered by ST1CH0_O output inactive to active transition This bit reserved only in Slave_TIMER1. Refer to CP1BST0A description.
15	CP1BST0CMP1	Capture 1 triggered by compare 1 event of Slave_TIMER0 This bit reserved only in Slave_TIMER0. 0: Capture 1 is not triggered by compare 1 event of Slave_TIMER0. 1: Capture 1 is triggered by compare 1 event of Slave_TIMER0.
14	CP1BST0CMP0	Capture 1 triggered by compare 0 event of Slave_TIMER0 This bit reserved only in Slave_TIMER0. 0: Capture 1 is not triggered by compare 0 event of Slave_TIMER0. 1: Capture 1 is triggered by compare 0 event of Slave_TIMER0.
13	CP1BST0NA	Capture 1 triggered by ST0CH0_O output active to inactive transition When the Slave_TIMER0 channel 0 output is converted from the active level to the inactive level, capture 1 is triggered by ST0CH0_O. This bit reserved only in Slave_TIMER0. 0: Capture 1 is not triggered by ST0CH0_O output active to inactive transition. 1: Capture 1 is triggered by ST0CH0_O output active to inactive transition.
12	CP1BST0A	Capture 1 triggered by ST0CH0_O output inactive to active transition When the Slave_TIMER0 channel 0 output is converted from the inactive level to the active, capture 1 is triggered by ST0CH0_O. This bit reserved only in Slave_TIMER0. 0: Capture 1 is not triggered by ST0CH0_O output inactive to active transition. 1: Capture 1 is triggered by ST0CH0_O output inactive to active transition.

11	CP1BEXEV9	Capture 1 triggered by external event 9 Refer to CP1BEXEV0 description.
10	CP1BEXEV8	Capture 1 triggered by external event 8 Refer to CP1BEXEV0 description.
9	CP1BEXEV7	Capture 1 triggered by external event 7 Refer to CP1BEXEV0 description.
8	CP1BEXEV6	Capture 1 triggered by external event 6 Refer to CP1BEXEV0 description.
7	CP1BEXEV5	Capture 1 triggered by external event 5 Refer to CP1BEXEV0 description.
6	CP1BEXEV4	Capture 1 triggered by external event 4 Refer to CP1BEXEV0 description.
5	CP1BEXEV3	Capture 1 triggered by external event 3 Refer to CP1BEXEV0 description.
4	CP1BEXEV2	Capture 1 triggered by external event 2 Refer to CP1BEXEV0 description.
3	CP1BEXEV1	Capture 1 triggered by external event 1 Refer to CP1BEXEV0 description.
2	CP1BEXEV0	Capture 1 triggered by external event 0 When the bit is set, capture 1 is triggered by update external event 0 0: Capture 1 is not triggered by external event 0 1: Capture 1 is triggered by external event 0
1	CP1BUP	Capture 1 triggered by update event When the bit is set, capture 1 is triggered by update event 0: Capture 1 is not triggered by update event 1: Capture 1 is triggered by update event
0	CP1BSW	Capture 1 triggered by software This bit can be set by software, and cleared by hardware automatically. When the bit is set, it forces the capture 1 by software. 0: No capture 1 triggered by software 1: Capture 1 triggered by software

For Slave_TIMER3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CP1BST4 CMP1	CP1BST4 CMP0	CP1BST4 NA	CP1BST4 A	Reserved				CP1BST2 MP1	CP1BST2 CMP0	CP1BST2 NA	CP1BST2 A	CP1BST1 CMP1	CP1BST1 CMP0	CP1BST1 NA	CP1BST1 A
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
CP1BST0CMP1	CP1BST0CMP0	CP1BST0NA	CP1BST0A	CP1BEXEV9	CP1BEXEV8	CP1BEXEV7	CP1BEXEV6	CP1BEXEV5	CP1BEXEV4	CP1BEXEV3	CP1BEXEV2	CP1BEXEV1	CP1BEXEV0	CP1BUP	CP1BSW
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31	CP1BST4CMP1	Capture 1 triggered by compare 1 event of Slave_TIMER4 This bit reserved only in Slave_TIMER4. Refer to CP1BST0CMP1 description.
30	CP1BST4CMP0	Capture 1 triggered by compare 0 event of Slave_TIMER4 This bit reserved only in Slave_TIMER4. Refer to CP1BST0CMP0 description.
29	CP1BST4NA	Capture 1 triggered by ST4CH0_O output active to inactive transition. This bit reserved only in Slave_TIMER4. Refer to CP1BST0NA description.
28	CP1BST4A	Capture 1 triggered by ST4CH0_O output inactive to active transition This bit reserved only in Slave_TIMER4. Refer to CP1BST0A description.
27:24	Reserved	Must be kept at reset value
23	CP1BST2CMP1	Capture 1 triggered by compare 1 event of Slave_TIMER2 This bit reserved only in Slave_TIMER2. Refer to CP1BST0CMP1 description.
22	CP1BST2CMP0	Capture 1 triggered by compare 0 event of Slave_TIMER2 This bit reserved only in Slave_TIMER2. Refer to CP1BST0CMP0 description.
21	CP1BST2NA	Capture 1 triggered by ST2CH0_O output active to inactive transition This bit reserved only in Slave_TIMER2. Refer to CP1BST0NA description.
20	CP1BST2A	Capture 1 triggered by ST2CH0_O output inactive to active transition This bit reserved only in Slave_TIMER2. Refer to CP1BST0A description.
19	CP1BST1CMP1	Capture 1 triggered by compare 1 event of Slave_TIMER1 This bit reserved only in Slave_TIMER1. Refer to CP1BST0CMP1 description.
18	CP1BST1CMP0	Capture 1 triggered by compare 0 event of Slave_TIMER1 This bit reserved only in Slave_TIMER1. Refer to CP1BST0CMP0 description.
17	CP1BST1NA	Capture 1 triggered by ST1CH0_O output active to inactive transition

		This bit reserved only in Slave_TIMER1. Refer to CP1BST0NA description.
16	CP1BST1A	Capture 1 triggered by ST1CH0_O output inactive to active transition This bit reserved only in Slave_TIMER1. Refer to CP1BST0A description.
15	CP1BST0CMP1	Capture 1 triggered by compare 1 event of Slave_TIMER0 This bit reserved only in Slave_TIMER0. 0: Capture 1 is not triggered by compare 1 event of Slave_TIMER0. 1: Capture 1 is triggered by compare 1 event of Slave_TIMER0.
14	CP1BST0CMP0	Capture 1 triggered by compare 0 event of Slave_TIMER0 This bit reserved only in Slave_TIMER0. 0: Capture 1 is not triggered by compare 0 event of Slave_TIMER0. 1: Capture 1 is triggered by compare 0 event of Slave_TIMER0.
13	CP1BST0NA	Capture 1 triggered by ST0CH0_O output active to inactive transition When the Slave_TIMER0 channel 0 output is converted from the active level to the inactive level, capture 1 is triggered by ST0CH0_O. This bit reserved only in Slave_TIMER0. 0: Capture 1 is not triggered by ST0CH0_O output active to inactive transition. 1: Capture 1 is triggered by ST0CH0_O output active to inactive transition.
12	CP1BST0A	Capture 1 triggered by ST0CH0_O output inactive to active transition When the Slave_TIMER0 channel 0 output is converted from the inactive level to the active, capture 1 is triggered by ST0CH0_O. This bit reserved only in Slave_TIMER0. 0: Capture 1 is not triggered by ST0CH0_O output inactive to active transition. 1: Capture 1 is triggered by ST0CH0_O output inactive to active transition.
11	CP1BEXEV9	Capture 1 triggered by external event 9 Refer to CP1BEXEV0 description.
10	CP1BEXEV8	Capture 1 triggered by external event 8 Refer to CP1BEXEV0 description.
9	CP1BEXEV7	Capture 1 triggered by external event 7 Refer to CP1BEXEV0 description.
8	CP1BEXEV6	Capture 1 triggered by external event 6 Refer to CP1BEXEV0 description.
7	CP1BEXEV5	Capture 1 triggered by external event 5 Refer to CP1BEXEV0 description.
6	CP1BEXEV4	Capture 1 triggered by external event 4 Refer to CP1BEXEV0 description.

5	CP1BEXEV3	Capture 1 triggered by external event 3 Refer to CP1BEXEV0 description.
4	CP1BEXEV2	Capture 1 triggered by external event 2 Refer to CP1BEXEV0 description.
3	CP1BEXEV1	Capture 1 triggered by external event 1 Refer to CP1BEXEV0 description.
2	CP1BEXEV0	Capture 1 triggered by external event 0 When the bit is set, capture 1 is triggered by update external event 0 0: Capture 1 is not triggered by external event 0 1: Capture 1 is triggered by external event 0
1	CP1BUP	Capture 1 triggered by update event When the bit is set, capture 1 is triggered by update event 0: Capture 1 is not triggered by update event 1: Capture 1 is triggered by update event
0	CP1BSW	Capture 1 triggered by software This bit can be set by software, and cleared by hardware automatically. When the bit is set, it forces the capture 1 by software. 0: No capture 1 triggered by software 1: Capture 1 triggered by software

For Slave_TIMER4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				CP1BST3CMP1	CP1BST3CMP0	CP1BST3NA	CP1BST3A	CP1BST2MP1	CP1BST2CMP0	CP1BST2NA	CP1BST2A	CP1BST1CMP1	CP1BST1CMP0	CP1BST1NA	CP1BST1A
				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
CP1BST0CMP1	CP1BST0CMP0	CP1BST0NA	CP1BST0A	CP1BEXEV9	CP1BEXEV8	CP1BEXEV7	CP1BEXEV6	CP1BEXEV5	CP1BEXEV4	CP1BEXEV3	CP1BEXEV2	CP1BEXEV1	CP1BEXEV0	CP1BUP	CP1BSW
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	CP1BST3CMP1	Capture 1 triggered by compare 1 event of Slave_TIMER3 This bit reserved only in Slave_TIMER3. Refer to CP1BST0CMP1 description.
26	CP1BST3CMP0	Capture 1 triggered by compare 0 event of Slave_TIMER3 This bit reserved only in Slave_TIMER3. Refer to CP1BST0CMP0 description.
25	CP1BST3NA	Capture 1 triggered by ST3CH0_O output active to inactive transition

		This bit reserved only in Slave_TIMER3. Refer to CP1BST0NA description.
24	CP1BST3A	Capture 1 triggered by ST3CH0_O output inactive to active transition This bit reserved only in Slave_TIMER3. Refer to CP1BST0A description.
23	CP1BST2CMP1	Capture 1 triggered by compare 1 event of Slave_TIMER2 This bit reserved only in Slave_TIMER2. Refer to CP1BST0CMP1 description.
22	CP1BST2CMP0	Capture 1 triggered by compare 0 event of Slave_TIMER2 This bit reserved only in Slave_TIMER2. Refer to CP1BST0CMP0 description.
21	CP1BST2NA	Capture 1 triggered by ST2CH0_O output active to inactive transition This bit reserved only in Slave_TIMER2. Refer to CP1BST0NA description.
20	CP1BST2A	Capture 1 triggered by ST2CH0_O output inactive to active transition This bit reserved only in Slave_TIMER2. Refer to CP1BST0A description.
19	CP1BST1CMP1	Capture 1 triggered by compare 1 event of Slave_TIMER1 This bit reserved only in Slave_TIMER1. Refer to CP1BST0CMP1 description.
18	CP1BST1CMP0	Capture 1 triggered by compare 0 event of Slave_TIMER1 This bit reserved only in Slave_TIMER1. Refer to CP1BST0CMP0 description.
17	CP1BST1NA	Capture 1 triggered by ST1CH0_O output active to inactive transition This bit reserved only in Slave_TIMER1. Refer to CP1BST0NA description.
16	CP1BST1A	Capture 1 triggered by ST1CH0_O output inactive to active transition This bit reserved only in Slave_TIMER1. Refer to CP1BST0A description.
15	CP1BST0CMP1	Capture 1 triggered by compare 1 event of Slave_TIMER0 This bit reserved only in Slave_TIMER0. 0: Capture 1 is not triggered by compare 1 event of Slave_TIMER0. 1: Capture 1 is triggered by compare 1 event of Slave_TIMER0.
14	CP1BST0CMP0	Capture 1 triggered by compare 0 event of Slave_TIMER0 This bit reserved only in Slave_TIMER0. 0: Capture 1 is not triggered by compare 0 event of Slave_TIMER0. 1: Capture 1 is triggered by compare 0 event of Slave_TIMER0.

13	CP1BST0NA	Capture 1 triggered by ST0CH0_O output active to inactive transition When the Slave_TIMER0 channel 0 output is converted from the active level to the inactive level, capture 1 is triggered by ST0CH0_O. This bit reserved only in Slave_TIMER0. 0: Capture 1 is not triggered by ST0CH0_O output active to inactive transition. 1: Capture 1 is triggered by ST0CH0_O output active to inactive transition.
12	CP1BST0A	Capture 1 triggered by ST0CH0_O output inactive to active transition When the Slave_TIMER0 channel 0 output is converted from the inactive level to the active, capture 1 is triggered by ST0CH0_O. This bit reserved only in Slave_TIMER0. 0: Capture 1 is not triggered by ST0CH0_O output inactive to active transition. 1: Capture 1 is triggered by ST0CH0_O output inactive to active transition.
11	CP1BEXEV9	Capture 1 triggered by external event 9 Refer to CP1BEXEV0 description.
10	CP1BEXEV8	Capture 1 triggered by external event 8 Refer to CP1BEXEV0 description.
9	CP1BEXEV7	Capture 1 triggered by external event 7 Refer to CP1BEXEV0 description.
8	CP1BEXEV6	Capture 1 triggered by external event 6 Refer to CP1BEXEV0 description.
7	CP1BEXEV5	Capture 1 triggered by external event 5 Refer to CP1BEXEV0 description.
6	CP1BEXEV4	Capture 1 triggered by external event 4 Refer to CP1BEXEV0 description.
5	CP1BEXEV3	Capture 1 triggered by external event 3 Refer to CP1BEXEV0 description.
4	CP1BEXEV2	Capture 1 triggered by external event 2 Refer to CP1BEXEV0 description.
3	CP1BEXEV1	Capture 1 triggered by external event 1 Refer to CP1BEXEV0 description.
2	CP1BEXEV0	Capture 1 triggered by external event 0 When the bit is set, capture 1 is triggered by update external event 0 0: Capture 1 is not triggered by external event 0 1: Capture 1 is triggered by external event 0
1	CP1BUP	Capture 1 triggered by update event When the bit is set, capture 1 is triggered by update event 0: Capture 1 is not triggered by update event

		1: Capture 1 is triggered by update event
0	CP1BSW	Capture 1 triggered by software This bit can be set by software, and cleared by hardware automatically. When the bit is set, it forces the capture 1 by software. 0: No capture 1 triggered by software 1: Capture 1 triggered by software

For Slave_TIMER5 - Slave_TIMER7

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CP1BST4CMP1	CP1BST4CMP0	CP1BST4NA	CP1BST4A	CP1BST3CMP1	CP1BST3CMP0	CP1BST3NA	CP1BST3A	CP1BST2MP1	CP1BST2CMP0	CP1BST2NA	CP1BST2A	CP1BST1CMP1	CP1BST1CMP0	CP1BST1NA	CP1BST1A
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
CP1BST0CMP1	CP1BST0CMP0	CP1BST0NA	CP1BST0A	CP1BEXEV9	CP1BEXEV8	CP1BEXEV7	CP1BEXEV6	CP1BEXEV5	CP1BEXEV4	CP1BEXEV3	CP1BEXEV2	CP1BEXEV1	CP1BEXEV0	CP1BUP	CP1BSW
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31	CP1BST4CMP1	Capture 1 triggered by compare 1 event of Slave_TIMER4 This bit reserved only in Slave_TIMER4. Refer to CP1BST0CMP1 description.
30	CP1BST4CMP0	Capture 1 triggered by compare 0 event of Slave_TIMER4 This bit reserved only in Slave_TIMER4. Refer to CP1BST0CMP0 description.
29	CP1BST4NA	Capture 1 triggered by ST4CH0_O output active to inactive transition. This bit reserved only in Slave_TIMER4. Refer to CP1BST0NA description.
28	CP1BST4A	Capture 1 triggered by ST4CH0_O output inactive to active transition This bit reserved only in Slave_TIMER4. Refer to CP1BST0A description.
27	CP1BST3CMP1	Capture 1 triggered by compare 1 event of Slave_TIMER3 This bit reserved only in Slave_TIMER3. Refer to CP1BST0CMP1 description.
26	CP1BST3CMP0	Capture 1 triggered by compare 0 event of Slave_TIMER3 This bit reserved only in Slave_TIMER3. Refer to CP1BST0CMP0 description.
25	CP1BST3NA	Capture 1 triggered by ST3CH0_O output active to inactive transition This bit reserved only in Slave_TIMER3. Refer to CP1BST0NA description.

24	CP1BST3A	Capture 1 triggered by ST3CH0_O output inactive to active transition This bit reserved only in Slave_TIMER3. Refer to CP1BST0A description.
23	CP1BST2CMP1	Capture 1 triggered by compare 1 event of Slave_TIMER2 This bit reserved only in Slave_TIMER2. Refer to CP1BST0CMP1 description.
22	CP1BST2CMP0	Capture 1 triggered by compare 0 event of Slave_TIMER2 This bit reserved only in Slave_TIMER2. Refer to CP1BST0CMP0 description.
21	CP1BST2NA	Capture 1 triggered by ST2CH0_O output active to inactive transition This bit reserved only in Slave_TIMER2. Refer to CP1BST0NA description.
20	CP1BST2A	Capture 1 triggered by ST2CH0_O output inactive to active transition This bit reserved only in Slave_TIMER2. Refer to CP1BST0A description.
19	CP1BST1CMP1	Capture 1 triggered by compare 1 event of Slave_TIMER1 This bit reserved only in Slave_TIMER1. Refer to CP1BST0CMP1 description.
18	CP1BST1CMP0	Capture 1 triggered by compare 0 event of Slave_TIMER1 This bit reserved only in Slave_TIMER1. Refer to CP1BST0CMP0 description.
17	CP1BST1NA	Capture 1 triggered by ST1CH0_O output active to inactive transition This bit reserved only in Slave_TIMER1. Refer to CP1BST0NA description.
16	CP1BST1A	Capture 1 triggered by ST1CH0_O output inactive to active transition This bit reserved only in Slave_TIMER1. Refer to CP1BST0A description.
15	CP1BST0CMP1	Capture 1 triggered by compare 1 event of Slave_TIMER0 This bit reserved only in Slave_TIMER0. 0: Capture 1 is not triggered by compare 1 event of Slave_TIMER0. 1: Capture 1 is triggered by compare 1 event of Slave_TIMER0.
14	CP1BST0CMP0	Capture 1 triggered by compare 0 event of Slave_TIMER0 This bit reserved only in Slave_TIMER0. 0: Capture 1 is not triggered by compare 0 event of Slave_TIMER0. 1: Capture 1 is triggered by compare 0 event of Slave_TIMER0.
13	CP1BST0NA	Capture 1 triggered by ST0CH0_O output active to inactive transition When the Slave_TIMER0 channel 0 output is converted from the active level to the inactive level, capture 1 is triggered by ST0CH0_O.

		This bit reserved only in Slave_TIMER0. 0: Capture 1 is not triggered by ST0CH0_O output active to inactive transition. 1: Capture 1 is triggered by ST0CH0_O output active to inactive transition.
12	CP1BST0A	Capture 1 triggered by ST0CH0_O output inactive to active transition When the Slave_TIMER0 channel 0 output is converted from the inactive level to the active, capture 1 is triggered by ST0CH0_O. This bit reserved only in Slave_TIMER0. 0: Capture 1 is not triggered by ST0CH0_O output inactive to active transition. 1: Capture 1 is triggered by ST0CH0_O output inactive to active transition.
11	CP1BEXEV9	Capture 1 triggered by external event 9 Refer to CP1BEXEV0 description.
10	CP1BEXEV8	Capture 1 triggered by external event 8 Refer to CP1BEXEV0 description.
9	CP1BEXEV7	Capture 1 triggered by external event 7 Refer to CP1BEXEV0 description.
8	CP1BEXEV6	Capture 1 triggered by external event 6 Refer to CP1BEXEV0 description.
7	CP1BEXEV5	Capture 1 triggered by external event 5 Refer to CP1BEXEV0 description.
6	CP1BEXEV4	Capture 1 triggered by external event 4 Refer to CP1BEXEV0 description.
5	CP1BEXEV3	Capture 1 triggered by external event 3 Refer to CP1BEXEV0 description.
4	CP1BEXEV2	Capture 1 triggered by external event 2 Refer to CP1BEXEV0 description.
3	CP1BEXEV1	Capture 1 triggered by external event 1 Refer to CP1BEXEV0 description.
2	CP1BEXEV0	Capture 1 triggered by external event 0 When the bit is set, capture 1 is triggered by update external event 0 0: Capture 1 is not triggered by external event 0 1: Capture 1 is triggered by external event 0
1	CP1BUP	Capture 1 triggered by update event When the bit is set, capture 1 is triggered by update event 0: Capture 1 is not triggered by update event 1: Capture 1 is triggered by update event
0	CP1BSW	Capture 1 triggered by software This bit can be set by software, and cleared by hardware automatically. When the

bit is set, it forces the capture 1 by software.

0: No capture 1 triggered by software

1: Capture 1 triggered by software

HRTIMER Slave_TIMERx channel output control register (HRTIMER_STxCHOCTL)

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								BMCH1DTI	CH1CSEN	CH1FLTOS[1:0]		ISO1	BMCH1IEN	CH1P	Reserved
								rw	rw	rw		rw	rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	BALIAR	Reserved	DLYISCH[2:0]			DLYISMEN	DTEN	BMCH0DTI	CH0CSEN	CH0FLTOS[1:0]		ISO0	BMCH0IEN	CH0P	Reserved
	rw		rw			rw	rw	rw	rw	rw		rw	rw	rw	

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value
23	BMCH1DTI	Channel 1 dead-time insert in bunch mode In bunch mode, a dead-time can be inserted before output entering the IDLE state. 0: The output enter IDLE immediately. 1: Dead-time is inserted before entering the IDLE state. The deadtime value is set by DTFCFG[15:0]. **Note:** (1) This bit must not be modified once the counter is enabled (STxCEN bit set). (2) This bit can be set only if one of the output idle state is active (ISOy = 1, y=0,1) during IDLE in bunch mode, and the dead-time value is positive (DTFSPROT / DTRSPROT set to 0).
22	CH1CSEN	Channel 1 carrier-signal mode enable 0: Carrier-signal mode of channel 1 is disable. 1: Carrier-signal mode of channel 1 is enable. **Note:** This bit must not be modified once the counter is enabled (STxCEN bit is set).
21:20	CH1FLTOS[1:0]	Channel 1 Fault output state This bit-field specifies the output state when a fault event happened. 00: No effect. The output is normally in Run mode when a fault event happened. 01: The output is active level.

		10: The output is inactive level 11: The output is Hi-Z. **Note:** If FLTyEN (y=0..4) in HRTIMER_STxFLTCTL register is set or the output is in Fault state, this bit-field cannot be modified once the counter is enabled (STxCEN bit is set)
19	ISO1	channel 1 output idle state 0: channel 1 idle state output is inactive. 1: channel 1 idle state output is active. **Note:** This bit must be configured before HRTIMER control the outputs.
18	BMCH1IEN	Channel 1 IDLE state enable in bunch mode This bit specifies whether channel 1 output can be IDLE state in bunch mode 0: Channel 1 output is not affected by the bunch mode. 1: Channel 1 output can be IDLE state in bunch mode. **Note:** This bit is preloaded and can be changed during run-time, but must not be changed during the bunch mode.
17	CH1P	Channel 1 output polarity This bit specifies the channel 1 output signal polarity. 0: Channel 1 active high 1: Channel 1 active low **Note:** This bit must not be modified once the counter is enabled (STxCEN bit is set)
16:15	Reserved	Must be kept at reset value
14	BALIAR	Balanced Idle Automatic Resumption This bit determines whether the outputs are automatically reactivated following a balanced idle event. This bit holds significance only when DLYISCH [2:0] is set to 011 or 111; otherwise, it is disregarded. 0: Disable 1: Enable Note: This bit must not be modified once the counter is enabled (STxCEN bit is set)
13	Reserved	Must be kept at reset value
12:10	DLYISCH[2:0]	Delayed IDLE source and channel This bit-field specifies the source and channel on which the delayed IDLE state mode are enabled (DLYISMEN = 1). In HRTIMER_STyCHOCTL(y=0,1,2) register: 000: channel 0 output delayed IDLE on external event 5 001: channel 1 output delayed IDLE on external event 5 010: channel 0 and channel 1 output delayed IDLE on external event 5 011: channel 0 and channel 1 output balanced IDLE on external event 5 in balanced mode (BLNMEN = 1 in HRTIMER_STyCTL0(y=0,1,2) register) 100: channel 0 output delayed IDLE on external event 6 101: channel 1 output delayed IDLE on external event 6

		110: channel 0 and channel 1 delayed IDLE on external event 6 111: channel 0 and channel 1 output balanced IDLE on external event 6 in balanced mode (BLNMEN = 1 in HRTIMER_STyCTL0(y=0,1,2) register) In HRTIMER_STyCHOCTL(y=3,4,5,6,7) register: 000: channel 0 output delayed IDLE on external event 7 001: channel 1 output delayed IDLE on external event 7 010: channel 0 and channel 1 output delayed IDLE on external event 7 011: channel 0 and channel 1 output balanced IDLE on external event 7 in balanced mode (BLNMEN = 1 in HRTIMER_STyCTL0(y=3,4,5,6,7) register) 100: channel 0 output delayed IDLE on external event 8 101: channel 1 output delayed IDLE on external event 8 110: channel 0 and channel 1 output delayed IDLE on external event 8 111: channel 0 and channel 1 output balanced IDLE on external event 8 in balanced mode (BLNMEN = 1 in HRTIMER_STyCTL0(y=3,4,5,6,7) register) **Note:** This bit-field must not be modified once the delayed IDLE mode is enabled (DLYISMEN bit is set)
9	DLYISMEN	Delayed IDLE state mode enable 0: Delayed IDLE state disable. 1: Delayed IDLE state enable. **Note:** This bit must not be modified once the counter is enabled (STxCEN bit is set)
8	DTEN	Dead time enable 0: Channel 0 and channel 1 outputs are independent. 1: Channel 0 and channel 1 outputs are complementary and dead-time is inserted between channel 0 and channel 1 outputs. **No**te: This bit must not be modified once the counter is enabled (STxCEN bit is set) or its outputs are enabled and controlled by another timer.
7	BMCH0DTI	Channel 0 dead-time insert in bunch mode In bunch mode, a dead-time can be inserted before output entering the IDLE state. 0: The output enter IDLE immediately. 1: Dead-time is inserted before entering the IDLE state. The deadtime value is set by DTRCFG[15:0] **Note:** (1) This bit must not be modified once the counter is enabled (STxCEN bit set). (2) This bit can be set only if one of the output idle state is active (ISO0 = 1, y=0,1) during IDLE in bunch mode, and the dead-time value is positive (DTFSPROT / DTRSPROT set to 0).
6	CH0CSEN	Channel 0 carrier-signal mode enable 0: Carrier-signal mode of channel 0 is disable. 1: Carrier-signal mode of channel 0 is enable. **Note:** This bit must not be modified once the counter is enabled (STxCEN bit is set).
5:4	CH0FLTOS[1:0]	Channel 0 Fault output state

		This bit-field specifies the output state when a fault event happened. 00: No effect. The output is normally in run mode when a fault event happened. 01: The output is active level. 10: The output is inactive level 11: The output is Hi-Z. **Note:** If FLTyEN (y=0..4) in HRTIMER_STxFLTCTL register is set or the output is in Fault state, this bit-field cannot be modified once the counter is enabled (STxCEN bit is set)
3	ISO0	Channel 0 output idle state 0: Channel 0 idle state output is inactive. 1: Channel 0 idle state output is active. **Note:** This bit must be configured before HRTIMER control the outputs.
2	BMCH0IEN	Channel 0 IDLE state enable in bunch mode This bit specifies whether channel 0 output can be IDLE state in bunch mode 0: Channel 0 output is not affected by the bunch mode. 1: Channel 0 output can be IDLE state in bunch mode. **Note:** This bit is preloaded and can be changed during run-time, but must not be changed during the bunch mode.
1	CH0P	Channel 0 output polarity This bit specifies the channel 0 output signal polarity. 0: Channel 0 active high 1: Channel 0 active low **Note:** This bit must not be modified once the counter is enabled (STxCEN bit is set)
0	Reserved	Must be kept at reset value

HRTIMER Slave_TIMERx fault control register (HRTIMER_STxFLTCTL)

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
FLTENPROT	Reserved														
rwo															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								FLT7EN	FLT6EN	FLT5EN	FLT4EN	FLT3EN	FLT2EN	FLT1EN	FLT0EN
								rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31	FLTENPROT	Protect fault enable This bit-field specifies whether the write protection function is enable or not. This bit

		is write-once. It can only be cleared by a system reset once It is set by software. 0: Protect disable. FLTyEN (y=0..7) is write-able. 1: Protect enable. FLTyEN (y=0..7) is read-only.
30:8	Reserved	Must be kept at reset value
7	FLT7EN	Fault 7 enable 0: Fault 7 disable 1: Fault 7 enable
6	FLT6EN	Fault 6 enable 0: Fault 6 disable 1: Fault 6 enable
5	FLT5EN	Fault 5 enable 0: Fault 5 disable 1: Fault 5 enable
4	FLT4EN	Fault 4 enable 0: Fault 4 disable 1: Fault 4 enable
3	FLT3EN	Fault 3 enable 0: Fault 3 disable 1: Fault 3 enable
2	FLT2EN	Fault 2 enable 0: Fault 2 disable 1: Fault 2 enable
1	FLT1EN	Fault 1 enable 0: Fault 1 disable 1: Fault 1 enable
0	FLT0EN	Fault 0 enable 0: Fault 0 disable 1: Fault 0 enable

HRTIMER Slave_TIMERx control register 1 (HRTIMER_STxCTL1)

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											TRGHALFM	Reserved		IMUPD2V	IMUPD0V
											rw			rw	rw

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

FEROVM[1:0]	BMROVM[1:0]	ADCROVM[1:0]	OUTROVM[1:0]	ROVM[1:0]	Reserved	CAM	Reserved	TRIG1M	TRIG0M	TRIGEN
rw	rw	rw	rw	rw		rw		rw	rw	rw

Bits	Fields	Descriptions
31:21	Reserved	Must be kept at reset value
20	TRGHALFM	Variable frequency half mode This bitfield determines the operating mode of the compare 1 register, indicating whether it operates in the standard mode (where a compare match occurs as soon as the counter equals the compare value) or in the variable frequency half mode (refer to Section:***Variable frequency half mode***). 0: Only written by the user in CMP1 register (standard compare mode). 1: Set by hardware immediately upon the occurrence of a capture 0 event. It loads with the value of (capture 0 divided by 2). The initial value can be written by the user (as long as TRGHALFM is reset), but it is disregarded once the first capture is triggered (the preload mechanism is disabled for CMP1 when the TRGHALFM bit is set).
19:18	Reserved	Must be kept at reset value
17	IMUPD2V	Immediately update compare 2 value PWM mode This bit determines the operational mode for compare 2. 0: The compare 2 event occurs when the counter equals the compare value. 1: The compare 2 event occurs when the counter surpasses the compare value. In the case of on-the-fly changes to the compare value, the new value is compared to the current counter value, potentially leading to the generation of an output SET or RESET
16	IMUPD0V	Immediately update compare 0 value PWM mode This bit determines the operational mode for compare 0. 0: The compare 0 event occurs when the counter equals the compare value. 1: The compare 0 event occurs when the counter surpasses the compare value. In the case of on-the-fly changes to the compare value, the new value is compared to the current counter value, potentially leading to the generation of an output SET or RESET
15:14	FEROVM[1:0]	Fault and event roll-over mode This bit determines the conditions under which the roll-over event is generated in Center-aligned mode, specifically for the roll-over event utilized by the fault and event counters. 00: When the counter equals 0 or the HRTIMER_STxCAR value, the event is generated. 01: When the counter equals 0, the event is generated. 10: When the counter equals HRTIMER_STxCAR, the event is generated. 11: Reserved Note: This configuration is applicable only when the CAM bit is set; otherwise, it

		holds no significance. Once the timer is operational (STxCEN bit is set), this bitfield cannot be modified.
13:12	BMROVM[1:0]	Bunch mode roll-over mode This bit determines the conditions under which the roll-over event is generated in Center-aligned mode, specifically for the roll-over event in bunch mode. 00: When the counter equals 0 or the HRTIMER_STxCAR value, the event is generated. 01: When the counter equals 0, the event is generated. 10: When the counter equals HRTIMER_STxCAR, the event is generated. 11: Reserved Note: This configuration is applicable only when the CAM bit is set; otherwise, it holds no significance. Once the timer is operational (STxCEN bit is set), this bitfield cannot be modified.
11:10	ADCROVM[1:0]	ADC center-aligned mode This bit determines the conditions under which the ADC trigger event is generated in center-aligned mode. 00: ADC trigger event generated when counter in up direction and in down direction. 01: ADC trigger event generated when counter in down direction. 10: ADC trigger event generated when counter in up direction. 11: Reserved Note: This configuration is applicable only when the CAM bit is set; otherwise, it holds no significance. Once the timer is operational (STxCEN bit is set), this bitfield cannot be modified.
9:8	OUTROVM[1:0]	Output roll-over mode This bit determines the conditions under which the roll-over event is generated in Center-aligned mode, setting and/or resetting the outputs according to the HRTIMER_STxCHySET and HRTIMER_STxCHyRST settings. 00: When the counter equals 0 or the HRTIMER_STxCAR value, the event is generated. 01: When the counter equals 0, the event is generated. 10: When the counter equals HRTIMER_STxCAR, the event is generated. 11: Reserved Note: This configuration is applicable only when the CAM bit is set; otherwise, it holds no significance. Once the timer is operational (STxCEN bit is set), this bitfield cannot be modified.
7:6	ROVM[1:0]	Roll-over mode This bit defines when the roll-over event is generated in Center-aligned mode. The roll-over event is used for update triggers (to transfer content from preload to active registers), IRQ and DMA requests, decrementing the repetition counter, and external event filtering. 00: When the counter equals 0 or the HRTIMER_STxCAR value, the event is

generated.
01: When the counter equals 0, the event is generated.
10: When the counter equals HRTIMER_STxCAR, the event is generated.
11: Reserved
Note: This configuration is applicable only when the CAM bit is set; otherwise, it holds no significance. Once the timer is operational (STxCEN bit is set), this bitfield cannot be modified.

5	Reserved	Must be kept at reset value
4	CAM	Center-aligned counting mode This bit defines if the counter is operating in up or center-aligned counting mode. 0: The counter is operating in up counting mode 1: The counter is operating in center-aligned counting mode Note: This bit cannot be changed once the timer is operating (STxCEN bit set).
3	Reserved	Must be kept at reset value
2	TRIG1M	This bit defines when TRIG1 is generated. 0: The trigger is generated on counter reset or roll-over event 1: The trigger is generated on output 0 set event Note: This bit is not significant when the TRIGEN bit is reset.
1	TRIG0M	This bit defines when TRIG0 is generated. 0: The trigger is generated on compare 1 event 1: The trigger is generated on output 0 reset event Note: This bit is not significant when the TRIGEN bit is reset.
0	TRIGEN	Two trigger enable This bit enables the two triggering mechanism. 0: Two trigger disabled 1: Two trigger enabled Note: This bit cannot be changed once the timer is operating (STxCEN bit set).

HRTIMER Slave_TIMERx external event filter configuration register 2 (HRTIMER_STxEXEVFCFG2)

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
EXEVXCNT[5:0]						Reserved									
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		EXEVXCNTTHR[5:0]						EXEVXSEL[3:0]				Reserved	EXEVXRSTM	EXEVXCRST	EXEVXCEN

rw rw rw rw rw

Bits	Fields	Descriptions
31:26	EXEVXCNTT[5:0]	External event X counter value This bitfield recordsthe counter value associated with external event X.
25:14	Reserved	Must be kept at reset value
13:8	EXEVXCNTTHR[5:0]	External event X counter threshold This bitfield determines the threshold for the external event X counter. An event occurs when the number of events equals the value of (EXEVXCNTTHR[5:0]+1).
7:4	EXEVXSEL[3:0]	External event X selection This bit determines the source of external event X. 0: External event 0 serves as the source for external event X. 1: External event 1 serves as the source for external event X . ... 9: External event 9 serves as the source for external event X. Others: Reserved.
3	Reserved	Must be kept at reset value
2	EXEVXRSTM	External event X reset mode This bit determines the reset mode for external event X counter. 0: The external event counter X is reset on every reset/roll-over event. 1: The external event counter X is reset on every reset/roll-over event only if no event occurs during the last counting period.
1	EXEVXCRST	External event X counter reset This bit initiates a reset for the external event X counter. It is set by software and reset by hardware. 0: No action 1: Reset the external event counter X
0	EXEVXCEN	External event X counter enable This bit enables the external event x counter. 0: External event X counter disabled 1: External event X counter enabled

HRTIMER Slave_TIMERx capture trigger combination register (HRTIMER_STxCAPTRGCOM)

Address offset: 0x74
Reset value: 0x0000 0000

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

For Slave_TIMER0 - Slave_TIMER4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				CP1BST7CMP1	CP1BST7CMP0	CP1BST7NA	CP1BST7A	CP1BST6CMP1	CP1BST6CMP0	CP1BST6NA	CP1BST6A	CP1BST5CMP1	CP1BST5CMP0	CP1BST5NA	CP1BST5A
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				CP0BST7CMP1	CP0BST7CMP0	CP0BST7NA	CP0BST7A	CP0BST6CMP1	CP0BST6CMP0	CP0BST6NA	CP0BST6A	CP0BST5CMP1	CP0BST5CMP0	CP0BST5NA	CP0BST5A
				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	CP1BST7CMP1	Capture 1 triggered by compare 1 event of Slave_TIMER7 This bit reserved only in Slave_TIMER7. 0: Capture 1 is not triggered by compare 1 event of Slave_TIMER7. 1: Capture 1 is triggered by compare 1 event of Slave_TIMER7.
26	CP1BST7CMP0	Capture 1 triggered by compare 0 event of Slave_TIMER7 This bit reserved only in Slave_TIMER7. 0: Capture 1 is not triggered by compare 0 event of Slave_TIMER7. 1: Capture 1 is triggered by compare 0 event of Slave_TIMER7.
25	CP1BST7NA	Capture 1 triggered by ST7CH0_O output active to inactive transition When the Slave_TIMER7 channel 0 output is converted from the active level to the inactive level, capture 0 is triggered by ST7CH0_O. This bit reserved only in Slave_TIMER7. 0: Capture 1 is not triggered by ST7CH0_O output active to inactive transition. 1: Capture 1 is triggered by ST7CH0_O output active to inactive transition.
24	CP1BST7A	Capture 1 triggered by ST7CH0_O output inactive to active transition When the Slave_TIMER7 channel 0 output is converted from the inactive level to the active, capture 0 is triggered by ST7CH0_O. This bit reserved only in Slave_TIMER7. 0: Capture 1 is not triggered by ST7CH0_O output inactive to active transition. 1: Capture 1 is triggered by ST7CH0_O output inactive to active transition.
23	CP1BST6CMP1	Capture 1 triggered by compare 1 event of Slave_TIMER6 This bit reserved only in Slave_TIMER6. 0: Capture 1 is not triggered by compare 1 event of Slave_TIMER6. 1: Capture 1 is triggered by compare 1 event of Slave_TIMER6.
22	CP1BST6CMP0	Capture 1 triggered by compare 0 event of Slave_TIMER6 This bit reserved only in Slave_TIMER6. 0: Capture 1 is not triggered by compare 0 event of Slave_TIMER6. 1: Capture 1 is triggered by compare 0 event of Slave_TIMER6.

21	CP1BST6NA	Capture 1 triggered by ST6CH0_O output active to inactive transition When the Slave_TIMER0 channel 0 output is converted from the active level to the inactive level, capture 1 is triggered by ST6CH0_O. This bit reserved only in Slave_TIMER6. 0: Capture 1 is not triggered by ST6CH0_O output active to inactive transition. 1: Capture 1 is triggered by ST6CH0_O output active to inactive transition.
20	CP1BST6A	Capture 1 triggered by ST6CH0_O output inactive to active transition When the Slave_TIMER6 channel 0 output is converted from the inactive level to the active, capture 1 is triggered by ST6CH0_O. This bit reserved only in Slave_TIMER6. 0: Capture 1 is not triggered by ST6CH0_O output inactive to active transition. 1: Capture 1 is triggered by ST6CH0_O output inactive to active transition.
19	CP1BST5CMP1	Capture 1 triggered by compare 1 event of Slave_TIMER5 This bit reserved only in Slave_TIMER5. 0: Capture 1 is not triggered by compare 1 event of Slave_TIMER5. 1: Capture 1 is triggered by compare 1 event of Slave_TIMER5.
18	CP1BST5CMP0	Capture 1 triggered by compare 0 event of Slave_TIMER5 This bit reserved only in Slave_TIMER5. 0: Capture 1 is not triggered by compare 0 event of Slave_TIMER0. 1: Capture 1 is triggered by compare 0 event of Slave_TIMER0.
17	CP1BST5NA	Capture 1 triggered by ST5CH0_O output active to inactive transition When the Slave_TIMER5 channel 0 output is converted from the active level to the inactive level, capture 1 is triggered by ST5CH0_O. This bit reserved only in Slave_TIMER5. 0: Capture 1 is not triggered by ST5CH0_O output active to inactive transition. 1: Capture 1 is triggered by ST5CH0_O output active to inactive transition.
16	CP1BST5A	Capture 1 triggered by ST5CH0_O output inactive to active transition When the Slave_TIMER5 channel 0 output is converted from the inactive level to the active, capture 1 is triggered by ST5CH0_O. This bit reserved only in Slave_TIMER5. 0: Capture 1 is not triggered by ST5CH0_O output inactive to active transition. 1: Capture 1 is triggered by ST5CH0_O output inactive to active transition.
15:12	Reserved	Must kept as reserved value.
11	CP0BST7CMP1	Capture 0 triggered by compare 1 event of Slave_TIMER7 This bit reserved only in Slave_TIMER7. 0: Capture 0 is not triggered by compare 1 event of Slave_TIMER7. 1: Capture 0 is triggered by compare 1 event of Slave_TIMER7.
10	CP0BST7CMP0	Capture 0 triggered by compare 0 event of Slave_TIMER7 This bit reserved only in Slave_TIMER7.

		0: Capture 0 is not triggered by compare 0 event of Slave_TIMER7. 1: Capture 0 is triggered by compare 0 event of Slave_TIMER7.
9	CP0BST7NA	Capture 0 triggered by ST7CH0_O output active to inactive transition When the Slave_TIMER7 channel 0 output is converted from the active level to the inactive level, capture 0 is triggered by ST7CH0_O. This bit reserved only in Slave_TIMER7. 0: Capture 0 is not triggered by ST7CH0_O output active to inactive transition. 1: Capture 0 is triggered by ST7CH0_O output active to inactive transition.
8	CP0BST7A	Capture 0 triggered by ST7CH0_O output inactive to active transition When the Slave_TIMER7 channel 0 output is converted from the inactive level to the active, capture 0 is triggered by ST7CH0_O. This bit reserved only in Slave_TIMER7. 0: Capture 0 is not triggered by ST7CH0_O output inactive to active transition. 1: Capture 0 is triggered by ST7CH0_O output inactive to active transition.
7	CP0BST6CMP1	Capture 0 triggered by compare 1 event of Slave_TIMER6 This bit reserved only in Slave_TIMER6. 0: Capture 0 is not triggered by compare 1 event of Slave_TIMER6. 1: Capture 0 is triggered by compare 1 event of Slave_TIMER6.
6	CP0BST6CMP0	Capture 0 triggered by compare 0 event of Slave_TIMER6 This bit reserved only in Slave_TIMER6. 0: Capture 0 is not triggered by compare 0 event of Slave_TIMER6. 1: Capture 0 is triggered by compare 0 event of Slave_TIMER6.
5	CP0BST6NA	Capture 0 triggered by ST6CH0_O output active to inactive transition When the Slave_TIMER0 channel 0 output is converted from the active level to the inactive level, capture 0 is triggered by ST6CH0_O. This bit reserved only in Slave_TIMER6. 0: Capture 0 is not triggered by ST6CH0_O output active to inactive transition. 1: Capture 0 is triggered by ST6CH0_O output active to inactive transition.
4	CP0BST6A	Capture 0 triggered by ST6CH0_O output inactive to active transition When the Slave_TIMER6 channel 0 output is converted from the inactive level to the active, capture 0 is triggered by ST6CH0_O. This bit reserved only in Slave_TIMER6. 0: Capture 0 is not triggered by ST6CH0_O output inactive to active transition. 1: Capture 0 is triggered by ST6CH0_O output inactive to active transition.
3	CP0BST5CMP1	Capture 0 triggered by compare 1 event of Slave_TIMER5 This bit reserved only in Slave_TIMER5. 0: Capture 0 is not triggered by compare 1 event of Slave_TIMER5. 1: Capture 0 is triggered by compare 1 event of Slave_TIMER5.
2	CP0BST5CMP0	Capture 0 triggered by compare 0 event of Slave_TIMER5

This bit reserved only in Slave_TIMER5.
0: Capture 0 is not triggered by compare 0 event of Slave_TIMER0.
1: Capture 0 is triggered by compare 0 event of Slave_TIMER0.

1 CP0BST5NA Capture 0 triggered by ST5CH0_O output active to inactive transition
When the Slave_TIMER5 channel 0 output is converted from the active level to the inactive level, capture 0 is triggered by ST5CH0_O.
This bit reserved only in Slave_TIMER5.
0: Capture 0 is not triggered by ST5CH0_O output active to inactive transition.
1: Capture 0 is triggered by ST5CH0_O output active to inactive transition.

0 CP0BST5A Capture 0 triggered by ST5CH0_O output inactive to active transition
When the Slave_TIMER5 channel 0 output is converted from the inactive level to the active, capture 0 is triggered by ST5CH0_O.
This bit reserved only in Slave_TIMER5.
0: Capture 0 is not triggered by ST5CH0_O output inactive to active transition.
1: Capture 0 is triggered by ST5CH0_O output inactive to active transition.

For Slave_TIMER5

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				CP1BST7CMP1	CP1BST7CMP0	CP1BST7NA	CP1BST7A	CP1BST6CMP1	CP1BST6CMP0	CP1BST6NA	CP1BST6A	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				CP0BST7CMP1	CP0BST7CMP0	CP0BST7NA	CP0BST7A	CP0BST6CMP1	CP0BST6CMP0	CP0BST6NA	CP0BST6A	Reserved			
				rw	rw	rw	rw	rw	rw	rw	rw				

Bits	Fields	Descriptions
31:28	Reserved	Must be kept at reset value
27	CP1BST7CMP1	Capture 1 triggered by compare 1 event of Slave_TIMER7 This bit reserved only in Slave_TIMER7. 0: Capture 1 is not triggered by compare 1 event of Slave_TIMER7. 1: Capture 1 is triggered by compare 1 event of Slave_TIMER7.
26	CP1BST7CMP0	Capture 1 triggered by compare 0 event of Slave_TIMER7 This bit reserved only in Slave_TIMER7. 0: Capture 1 is not triggered by compare 0 event of Slave_TIMER7. 1: Capture 1 is triggered by compare 0 event of Slave_TIMER7.
25	CP1BST7NA	Capture 1 triggered by ST7CH0_O output active to inactive transition When the Slave_TIMER7 channel 0 output is converted from the active level to the inactive level, capture 0 is triggered by ST7CH0_O. This bit reserved only in Slave_TIMER7.

		0: Capture 1 is not triggered by ST7CH0_O output active to inactive transition. 1: Capture 1 is triggered by ST7CH0_O output active to inactive transition.
24	CP1BST7A	Capture 1 triggered by ST7CH0_O output inactive to active transition When the Slave_TIMER7 channel 0 output is converted from the inactive level to the active, capture 0 is triggered by ST7CH0_O. This bit reserved only in Slave_TIMER7. 0: Capture 1 is not triggered by ST7CH0_O output inactive to active transition. 1: Capture 1 is triggered by ST7CH0_O output inactive to active transition.
23	CP1BST6CMP1	Capture 1 triggered by compare 1 event of Slave_TIMER6 This bit reserved only in Slave_TIMER6. 0: Capture 1 is not triggered by compare 1 event of Slave_TIMER6. 1: Capture 1 is triggered by compare 1 event of Slave_TIMER6.
22	CP1BST6CMP0	Capture 1 triggered by compare 0 event of Slave_TIMER6 This bit reserved only in Slave_TIMER6. 0: Capture 1 is not triggered by compare 0 event of Slave_TIMER6. 1: Capture 1 is triggered by compare 0 event of Slave_TIMER6.
21	CP1BST6NA	Capture 1 triggered by ST6CH0_O output active to inactive transition When the Slave_TIMER0 channel 0 output is converted from the active level to the inactive level, capture 1 is triggered by ST6CH0_O. This bit reserved only in Slave_TIMER6. 0: Capture 1 is not triggered by ST6CH0_O output active to inactive transition. 1: Capture 1 is triggered by ST6CH0_O output active to inactive transition.
20	CP1BST6A	Capture 1 triggered by ST6CH0_O output inactive to active transition When the Slave_TIMER6 channel 0 output is converted from the inactive level to the active, capture 1 is triggered by ST6CH0_O. This bit reserved only in Slave_TIMER6. 0: Capture 1 is not triggered by ST6CH0_O output inactive to active transition. 1: Capture 1 is triggered by ST6CH0_O output inactive to active transition.
19:12	Reserved	Must kept as reserved value.
11	CP0BST7CMP1	Capture 0 triggered by compare 1 event of Slave_TIMER7 This bit reserved only in Slave_TIMER7. 0: Capture 0 is not triggered by compare 1 event of Slave_TIMER7. 1: Capture 0 is triggered by compare 1 event of Slave_TIMER7.
10	CP0BST7CMP0	Capture 0 triggered by compare 0 event of Slave_TIMER7 This bit reserved only in Slave_TIMER7. 0: Capture 0 is not triggered by compare 0 event of Slave_TIMER7. 1: Capture 0 is triggered by compare 0 event of Slave_TIMER7.
9	CP0BST7NA	Capture 0 triggered by ST7CH0_O output active to inactive transition When the Slave_TIMER7 channel 0 output is converted from the active level to the

		inactive level, capture 0 is triggered by ST7CH0_O. This bit reserved only in Slave_TIMER7. 0: Capture 0 is not triggered by ST7CH0_O output active to inactive transition. 1: Capture 0 is triggered by ST7CH0_O output active to inactive transition.
8	CP0BST7A	Capture 0 triggered by ST7CH0_O output inactive to active transition When the Slave_TIMER7 channel 0 output is converted from the inactive level to the active, capture 0 is triggered by ST7CH0_O. This bit reserved only in Slave_TIMER7. 0: Capture 0 is not triggered by ST7CH0_O output inactive to active transition. 1: Capture 0 is triggered by ST7CH0_O output inactive to active transition.
7	CP0BST6CMP1	Capture 0 triggered by compare 1 event of Slave_TIMER6 This bit reserved only in Slave_TIMER6. 0: Capture 0 is not triggered by compare 1 event of Slave_TIMER6. 1: Capture 0 is triggered by compare 1 event of Slave_TIMER6.
6	CP0BST6CMP0	Capture 0 triggered by compare 0 event of Slave_TIMER6 This bit reserved only in Slave_TIMER6. 0: Capture 0 is not triggered by compare 0 event of Slave_TIMER6. 1: Capture 0 is triggered by compare 0 event of Slave_TIMER6.
5	CP0BST6NA	Capture 0 triggered by ST6CH0_O output active to inactive transition When the Slave_TIMER0 channel 0 output is converted from the active level to the inactive level, capture 0 is triggered by ST6CH0_O. This bit reserved only in Slave_TIMER6. 0: Capture 0 is not triggered by ST6CH0_O output active to inactive transition. 1: Capture 0 is triggered by ST6CH0_O output active to inactive transition.
4	CP0BST6A	Capture 0 triggered by ST6CH0_O output inactive to active transition When the Slave_TIMER6 channel 0 output is converted from the inactive level to the active, capture 0 is triggered by ST6CH0_O. This bit reserved only in Slave_TIMER6. 0: Capture 0 is not triggered by ST6CH0_O output inactive to active transition. 1: Capture 0 is triggered by ST6CH0_O output inactive to active transition.
3:0	Reserved	Must kept as reserved value.

For Slave_TIMER6

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				CP1BST7 CMP1	CP1BST7 CMP0	CP1BST7 NA	CP1BST7 A	Reserved				CP1BST5 CMP1	CP1BST5 CMP0	CP1BST5 NA	CP1BST5 A
				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				CP0BST7 CMP1	CP0BST7 CMP0	CP0BST7 NA	CP0BST7 A	Reserved				CP0BST5 CMP1	CP0BST5 CMP0	CP0BST5 NA	CP0BST5 A

rw rw rw rw rw rw rw rw

Bits	Fields	Descriptions
31:28	Reserved	Must be kept at reset value
27	CP1BST7CMP1	Capture 1 triggered by compare 1 event of Slave_TIMER7 This bit reserved only in Slave_TIMER7. 0: Capture 1 is not triggered by compare 1 event of Slave_TIMER7. 1: Capture 1 is triggered by compare 1 event of Slave_TIMER7.
26	CP1BST7CMP0	Capture 1 triggered by compare 0 event of Slave_TIMER7 This bit reserved only in Slave_TIMER7. 0: Capture 1 is not triggered by compare 0 event of Slave_TIMER7. 1: Capture 1 is triggered by compare 0 event of Slave_TIMER7.
25	CP1BST7NA	Capture 1 triggered by ST7CH0_O output active to inactive transition When the Slave_TIMER7 channel 0 output is converted from the active level to the inactive level, capture 0 is triggered by ST7CH0_O. This bit reserved only in Slave_TIMER7. 0: Capture 1 is not triggered by ST7CH0_O output active to inactive transition. 1: Capture 1 is triggered by ST7CH0_O output active to inactive transition.
24	CP1BST7A	Capture 1 triggered by ST7CH0_O output inactive to active transition When the Slave_TIMER7 channel 0 output is converted from the inactive level to the active, capture 0 is triggered by ST7CH0_O. This bit reserved only in Slave_TIMER7. 0: Capture 1 is not triggered by ST7CH0_O output inactive to active transition. 1: Capture 1 is triggered by ST7CH0_O output inactive to active transition.
23:20	Reserved	Must be kept at reset value
19	CP1BST5CMP1	Capture 1 triggered by compare 1 event of Slave_TIMER5 This bit reserved only in Slave_TIMER5. 0: Capture 1 is not triggered by compare 1 event of Slave_TIMER5. 1: Capture 1 is triggered by compare 1 event of Slave_TIMER5.
18	CP1BST5CMP0	Capture 1 triggered by compare 0 event of Slave_TIMER5 This bit reserved only in Slave_TIMER5. 0: Capture 1 is not triggered by compare 0 event of Slave_TIMER0. 1: Capture 1 is triggered by compare 0 event of Slave_TIMER0.
17	CP1BST5NA	Capture 1 triggered by ST5CH0_O output active to inactive transition When the Slave_TIMER5 channel 0 output is converted from the active level to the inactive level, capture 1 is triggered by ST5CH0_O. This bit reserved only in Slave_TIMER5. 0: Capture 1 is not triggered by ST5CH0_O output active to inactive transition. 1: Capture 1 is triggered by ST5CH0_O output active to inactive transition.

16	CP1BST5A	Capture 1 triggered by ST5CH0_O output inactive to active transition When the Slave_TIMER5 channel 0 output is converted from the inactive level to the active, capture 1 is triggered by ST5CH0_O. This bit reserved only in Slave_TIMER5. 0: Capture 1 is not triggered by ST5CH0_O output inactive to active transition. 1: Capture 1 is triggered by ST5CH0_O output inactive to active transition.
15:12	Reserved	Must kept as reserved value.
11	CP0BST7CMP1	Capture 0 triggered by compare 1 event of Slave_TIMER7 This bit reserved only in Slave_TIMER7. 0: Capture 0 is not triggered by compare 1 event of Slave_TIMER7. 1: Capture 0 is triggered by compare 1 event of Slave_TIMER7.
10	CP0BST7CMP0	Capture 0 triggered by compare 0 event of Slave_TIMER7 This bit reserved only in Slave_TIMER7. 0: Capture 0 is not triggered by compare 0 event of Slave_TIMER7. 1: Capture 0 is triggered by compare 0 event of Slave_TIMER7.
9	CP0BST7NA	Capture 0 triggered by ST7CH0_O output active to inactive transition When the Slave_TIMER7 channel 0 output is converted from the active level to the inactive level, capture 0 is triggered by ST7CH0_O. This bit reserved only in Slave_TIMER7. 0: Capture 0 is not triggered by ST7CH0_O output active to inactive transition. 1: Capture 0 is triggered by ST7CH0_O output active to inactive transition.
8	CP0BST7A	Capture 0 triggered by ST7CH0_O output inactive to active transition When the Slave_TIMER7 channel 0 output is converted from the inactive level to the active, capture 0 is triggered by ST7CH0_O. This bit reserved only in Slave_TIMER7. 0: Capture 0 is not triggered by ST7CH0_O output inactive to active transition. 1: Capture 0 is triggered by ST7CH0_O output inactive to active transition.
7:4	Reserved	Must be kept at reset value
3	CP0BST5CMP1	Capture 0 triggered by compare 1 event of Slave_TIMER5 This bit reserved only in Slave_TIMER5. 0: Capture 0 is not triggered by compare 1 event of Slave_TIMER5. 1: Capture 0 is triggered by compare 1 event of Slave_TIMER5.
2	CP0BST5CMP0	Capture 0 triggered by compare 0 event of Slave_TIMER5 This bit reserved only in Slave_TIMER5. 0: Capture 0 is not triggered by compare 0 event of Slave_TIMER0. 1: Capture 0 is triggered by compare 0 event of Slave_TIMER0.
1	CP0BST5NA	Capture 0 triggered by ST5CH0_O output active to inactive transition When the Slave_TIMER5 channel 0 output is converted from the active level to the inactive level, capture 0 is triggered by ST5CH0_O.

		This bit reserved only in Slave_TIMER5. 0: Capture 0 is not triggered by ST5CH0_O output active to inactive transition. 1: Capture 0 is triggered by ST5CH0_O output active to inactive transition.
0	CP0BST5A	Capture 0 triggered by ST5CH0_O output inactive to active transition When the Slave_TIMER5 channel 0 output is converted from the inactive level to the active, capture 0 is triggered by ST5CH0_O. This bit reserved only in Slave_TIMER5. 0: Capture 0 is not triggered by ST5CH0_O output inactive to active transition. 1: Capture 0 is triggered by ST5CH0_O output inactive to active transition.

For Slave_TIMER7

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								CP1BST6CMP1	CP1BST6CMP0	CP1BST6NA	CP1BST6A	CP1BST5CMP1	CP1BST5CMP0	CP1BST5NA	CP1BST5A
								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								CP0BST6CMP1	CP0BST6CMP0	CP0BST6NA	CP0BST6A	CP0BST5CMP1	CP0BST5CMP0	CP0BST5NA	CP0BST5A
								rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value
23	CP1BST6CMP1	Capture 1 triggered by compare 1 event of Slave_TIMER6 This bit reserved only in Slave_TIMER6. 0: Capture 1 is not triggered by compare 1 event of Slave_TIMER6. 1: Capture 1 is triggered by compare 1 event of Slave_TIMER6.
22	CP1BST6CMP0	Capture 1 triggered by compare 0 event of Slave_TIMER6 This bit reserved only in Slave_TIMER6. 0: Capture 1 is not triggered by compare 0 event of Slave_TIMER6. 1: Capture 1 is triggered by compare 0 event of Slave_TIMER6.
21	CP1BST6NA	Capture 1 triggered by ST6CH0_O output active to inactive transition When the Slave_TIMER0 channel 0 output is converted from the active level to the inactive level, capture 1 is triggered by ST6CH0_O. This bit reserved only in Slave_TIMER6. 0: Capture 1 is not triggered by ST6CH0_O output active to inactive transition. 1: Capture 1 is triggered by ST6CH0_O output active to inactive transition.
20	CP1BST6A	Capture 1 triggered by ST6CH0_O output inactive to active transition When the Slave_TIMER6 channel 0 output is converted from the inactive level to the active, capture 1 is triggered by ST6CH0_O. This bit reserved only in Slave_TIMER6.

		0: Capture 1 is not triggered by ST6CH0_O output inactive to active transition. 1: Capture 1 is triggered by ST6CH0_O output inactive to active transition.
19	CP1BST5CMP1	Capture 1 triggered by compare 1 event of Slave_TIMER5 This bit reserved only in Slave_TIMER5. 0: Capture 1 is not triggered by compare 1 event of Slave_TIMER5. 1: Capture 1 is triggered by compare 1 event of Slave_TIMER5.
18	CP1BST5CMP0	Capture 1 triggered by compare 0 event of Slave_TIMER5 This bit reserved only in Slave_TIMER5. 0: Capture 1 is not triggered by compare 0 event of Slave_TIMER0. 1: Capture 1 is triggered by compare 0 event of Slave_TIMER0.
17	CP1BST5NA	Capture 1 triggered by ST5CH0_O output active to inactive transition When the Slave_TIMER5 channel 0 output is converted from the active level to the inactive level, capture 1 is triggered by ST5CH0_O. This bit reserved only in Slave_TIMER5. 0: Capture 1 is not triggered by ST5CH0_O output active to inactive transition. 1: Capture 1 is triggered by ST5CH0_O output active to inactive transition.
16	CP1BST5A	Capture 1 triggered by ST5CH0_O output inactive to active transition When the Slave_TIMER5 channel 0 output is converted from the inactive level to the active, capture 1 is triggered by ST5CH0_O. This bit reserved only in Slave_TIMER5. 0: Capture 1 is not triggered by ST5CH0_O output inactive to active transition. 1: Capture 1 is triggered by ST5CH0_O output inactive to active transition.
15:8	Reserved	Must kept as reserved value.
7	CP0BST6CMP1	Capture 0 triggered by compare 1 event of Slave_TIMER6 This bit reserved only in Slave_TIMER6. 0: Capture 0 is not triggered by compare 1 event of Slave_TIMER6. 1: Capture 0 is triggered by compare 1 event of Slave_TIMER6.
6	CP0BST6CMP0	Capture 0 triggered by compare 0 event of Slave_TIMER6 This bit reserved only in Slave_TIMER6. 0: Capture 0 is not triggered by compare 0 event of Slave_TIMER6. 1: Capture 0 is triggered by compare 0 event of Slave_TIMER6.
5	CP0BST6NA	Capture 0 triggered by ST6CH0_O output active to inactive transition When the Slave_TIMER0 channel 0 output is converted from the active level to the inactive level, capture 0 is triggered by ST6CH0_O. This bit reserved only in Slave_TIMER6. 0: Capture 0 is not triggered by ST6CH0_O output active to inactive transition. 1: Capture 0 is triggered by ST6CH0_O output active to inactive transition.
4	CP0BST6A	Capture 0 triggered by ST6CH0_O output inactive to active transition When the Slave_TIMER6 channel 0 output is converted from the inactive level to

		the active, capture 0 is triggered by ST6CH0_O. This bit reserved only in Slave_TIMER6. 0: Capture 0 is not triggered by ST6CH0_O output inactive to active transition. 1: Capture 0 is triggered by ST6CH0_O output inactive to active transition.
3	CP0BST5CMP1	Capture 0 triggered by compare 1 event of Slave_TIMER5 This bit reserved only in Slave_TIMER5. 0: Capture 0 is not triggered by compare 1 event of Slave_TIMER5. 1: Capture 0 is triggered by compare 1 event of Slave_TIMER5.
2	CP0BST5CMP0	Capture 0 triggered by compare 0 event of Slave_TIMER5 This bit reserved only in Slave_TIMER5. 0: Capture 0 is not triggered by compare 0 event of Slave_TIMER0. 1: Capture 0 is triggered by compare 0 event of Slave_TIMER0.
1	CP0BST5NA	Capture 0 triggered by ST5CH0_O output active to inactive transition When the Slave_TIMER5 channel 0 output is converted from the active level to the inactive level, capture 0 is triggered by ST5CH0_O. This bit reserved only in Slave_TIMER5. 0: Capture 0 is not triggered by ST5CH0_O output active to inactive transition. 1: Capture 0 is triggered by ST5CH0_O output active to inactive transition.
0	CP0BST5A	Capture 0 triggered by ST5CH0_O output inactive to active transition When the Slave_TIMER5 channel 0 output is converted from the inactive level to the active, capture 0 is triggered by ST5CH0_O. This bit reserved only in Slave_TIMER5. 0: Capture 0 is not triggered by ST5CH0_O output inactive to active transition. 1: Capture 0 is triggered by ST5CH0_O output inactive to active transition.

HRTIMER Slave_TIMERx counter reset and channel 0/1 set/reset request add register (HRTIMER_STxCNTRSTA)

Address offset: 0x78

Reset value: 0x0000 0000

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

For Slave_TIMER0 - Slave_TIMER4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		CH1RSSTCNTRST	CH0RSSTCNTRST	CH1RSSTEV10	CH1RSSTEV9	CH0RSSTEV10	CH0RSSTEV9	Reserved		CH1SSTCNTRST	CH0SSTCNTRST	CH1SSTEV10	CH1SSTEV9	CH0SSTEV10	CH0SSTEV9
		rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									ST7CMP3RST	ST7CMP1RST	ST7CMP0RST	ST6CMP3RST	ST6CMP1RST	ST6CMP0RST	ST5CMP3RST

rw rw rw rw rw rw

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value
29	CH1RSSTCNTRST	Slave_TIMERx counter reset event generates channel 1 “reset request” Refer to CH0SSTCNTRST description.
28	CH0RSSTCNTRST	Slave_TIMERx counter reset event generates channel 0 “reset request” Refer to CH0SSTCNTRST description.
27	CH1RSSTEV10	Slave_TIMERx interconnection event 10 generates channel 1 “reset request” Refer to CH0RSSTEV9 description.
26	CH1RSSTEV9	Slave_TIMERx interconnection event 9 generates channel 1 “reset request” Refer to CH0RSSTEV9 description.
25	CH0RSSTEV10	Slave_TIMERx interconnection event 10 generates channel 0 “reset request” Refer to CH0RSSTEV9 description.
24	CH0RSSTEV9	Slave_TIMERx interconnection event 9 generates channel 0 “reset request” When this bit is set, Slave_TIMERx interconnection event 0 can generate channel 0 “reset request”. 0: The event cannot generate channel 0 “reset request”. 1: The event can generate channel 0 “reset request”.
23:22	Reserved	Must be kept at reset value
21	CH1SSTCNTRST	Slave_TIMERx counter reset event generates channel 1 “set request” Refer to CH0SSTCNTRST description.
20	CH0SSTCNTRST	Slave_TIMERx counter reset event generates channel 0 “set request” 0: The counter reset event cannot generate channel 0 “set request”. 1: The counter reset event can generate channel 0 “set request”.
19	CH1SSTEV10	Slave_TIMERx interconnection event 10 generates channel 1 “set request” Refer to CH0SSTEV8 description.
18	CH1SSTEV9	Slave_TIMERx interconnection event 9 generates channel 1 “set request” Refer to CH0SSTEV8 description.
17	CH0SSTEV10	Slave_TIMERx interconnection event 10 generates channel 0 “set request” Refer to CH0SSTEV8 description.
16	CH0SSTEV9	Slave_TIMERx interconnection event 9 generates channel 0 “set request” When this bit is set, Slave_TIMERx interconnection event 0 can generate “set request”. 0: The event cannot generate “set request”. 1: The event can generate “set request”.

15:7	Reserved	Must be kept at reset value
6	ST7CMP3RST	Slave_TIMER7 compare 3 event resets counter Refer to ST5CMP3RST description
5	ST7CMP1RST	Slave_TIMER7 compare 1 event resets counter Refer to ST5CMP3RST description
4	ST7CMP0RST	Slave_TIMER7 compare 0 event resets counter Refer to ST5CMP3RST description
3	ST6CMP3RST	Slave_TIMER6 compare 3 event resets counter Refer to ST5CMP3RST description
2	ST6CMP1RST	Slave_TIMER6 compare 1 event resets counter Refer to ST5CMP3RST description
1	ST6CMP0RST	Slave_TIMER6 compare 0 event resets counter Refer to ST5CMP3RST description
0	ST5CMP3RST	Slave_TIMER5 compare 3 event resets counter This bit specifies whether the Slave_TIMER5 compare 3 event can reset the counter. 0: Slave_TIMER5 compare 3 event do not reset counter 1: Slave_TIMER5 compare 3 event resets counter

For Slave_TIMER5

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		CH1RSSTCNTRST	CH0RSSTCNTRST	CH1RSSTEV10	CH1RSSTEV9	CH0RSSTEV10	CH0RSSTEV9	Reserved		CH1SSTCNTRST	CH0SSTCNTRST	CH1SSTEV10	CH1SSTEV9	CH0SSTEV10	CH0SSTEV9
		rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									ST7CMP3RST	ST7CMP1RST	ST7CMP0RST	ST6CMP3RST	ST6CMP1RST	ST6CMP0RST	ST4CMP3RST
									rw	rw	rw	rw	rw	rw	

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value
29	CH1RSSTCNTRST	Slave_TIMERx counter reset event generates channel 1 “reset request” Refer to CH0SSTCNTRST description.
28	CH0RSSTCNTRST	Slave_TIMERx counter reset event generates channel 0 “reset request” Refer to CH0SSTCNTRST description.
27	CH1RSSTEV10	Slave_TIMERx interconnection event 10 generates channel 1 “reset request”

		Refer to CH0RSSTEV9 description.
26	CH1RSSTEV9	Slave_TIMERx interconnection event 9 generates channel 1 “reset request” Refer to CH0RSSTEV9 description.
25	CH0RSSTEV10	Slave_TIMERx interconnection event 10 generates channel 0 “reset request” Refer to CH0RSSTEV9 description.
24	CH0RSSTEV9	Slave_TIMERx interconnection event 9 generates channel 0 “reset request” When this bit is set, Slave_TIMERx interconnection event 0 can generate channel “reset request”. 0: The event cannot generate “reset request”. 1: The event can generate “reset request”.
23:22	Reserved	Must be kept at reset value
21	CH1SSTCNTRST	Slave_TIMERx counter reset event generates channel 1 “set request” Refer to CH0SSTCNTRST description.
20	CH0SSTCNTRST	Slave_TIMERx counter reset event generates channel 0 “set request” 0: The counter reset event cannot generate channel 0 “set request”. 1: The counter reset event can generate channel 0 “set request”.
19	CH1SSTEV10	Slave_TIMERx interconnection event 10 generates channel 1 “set request” Refer to CH0SSTEV9 description.
18	CH1SSTEV9	Slave_TIMERx interconnection event 9 generates channel 1 “set request” Refer to CH0SSTEV9 description.
17	CH0SSTEV10	Slave_TIMERx interconnection event 10 generates channel 0 “set request” Refer to CH0SSTEV9 description.
16	CH0SSTEV9	Slave_TIMERx interconnection event 9 generates channel 1 “set request” When this bit is set, Slave_TIMERx interconnection event 0 can generate “set request”. 0: The event cannot generate “set request”. 1: The event can generate “set request”.
15:7	Reserved	Must be kept at reset value
6	ST7CMP3RST	Slave_TIMER7 compare 3 event resets counter This bit specifies whether the Slave_TIMER7 compare 3 event can reset the counter. 0: Slave_TIMER7 compare 3 event do not reset counter 1: Slave_TIMER7 compare 3 event resets counter
5	ST7CMP1RST	Slave_TIMER7 compare 1 event resets counter This bit specifies whether the Slave_TIMER7 compare 1 event can reset the counter. 0: Slave_TIMER7 compare 1 event do not reset counter

		1: Slave_TIMER7 compare 1 event resets counter
4	ST7CMP0RST	Slave_TIMER7 compare 0 event resets counter This bit specifies whether the Slave_TIMER7 compare 0 event can reset the counter. 0: Slave_TIMER7 compare 0 event do not reset counter 1: Slave_TIMER7 compare 0 event resets counter
3	ST6CMP3RST	Slave_TIMER6 compare 3 event resets counter This bit specifies whether the Slave_TIMER6 compare 3 event can reset the counter. 0: Slave_TIMER6 compare 3 event do not reset counter 1: Slave_TIMER6 compare 3 event resets counter
2	ST6CMP1RST	Slave_TIMER6 compare 1 event resets counter This bit specifies whether the Slave_TIMER6 compare 1 event can reset the counter. 0: Slave_TIMER6 compare 1 event do not reset counter 1: Slave_TIMER6 compare 1 event resets counter
1	ST6CMP0RST	Slave_TIMER6 compare 0 event resets counter This bit specifies whether the Slave_TIMER6 compare 0 event can reset the counter. 0: Slave_TIMER6 compare 0 event do not reset counter 1: Slave_TIMER6 compare 0 event resets counter
0	ST4CMP3RST	Slave_TIMER4 compare 3 event resets counter This bit specifies whether the Slave_TIMER5 compare 3 event can reset the counter. 0: Slave_TIMER4 compare 3 event do not reset counter 1: Slave_TIMER4 compare 3 event resets counter

For Slave_TIMER6

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		CH1RSSTCNTRST	CH0RSSTCNTRST	CH1RSSTEV10	CH1RSSTEV9	CH0RSSTEV10	CH0RSSTEV9	Reserved		CH1SSTCNTRST	CH0SSTCNTRST	CH1SSTEV10	CH1SSTEV9	CH0SSTEV10	CH0SSTEV9
		rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									ST7CMP3RST	ST7CMP1RST	ST7CMP0RST	ST5CMP3RST	ST5CMP1RST	ST5CMP0RST	ST4CMP3RST
									rw	rw	rw	rw	rw	rw	

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value

29	CH1RSSTCNTRST	Slave_TIMERx counter reset event generates channel 1 “reset request” Refer to CH0SSTCNTRST description.
28	CH0RSSTCNTRST	Slave_TIMERx counter reset event generates channel 0 “reset request” Refer to CH0SSTCNTRST description.
27	CH1RSSTEV10	Slave_TIMERx interconnection event 10 generates channel 1 “reset request” Refer to CH0RSSTEV9 description.
26	CH1RSSTEV9	Slave_TIMERx interconnection event 9 generates channel 1 “reset request” Refer to CH0RSSTEV9 description.
25	CH0RSSTEV10	Slave_TIMERx interconnection event 10 generates channel 0 “reset request” Refer to CH0RSSTEV9 description.
24	CH0RSSTEV9	Slave_TIMERx interconnection event 9 generates channel 0 “reset request” When this bit is set, Slave_TIMERx interconnection event 0 can generate channel “reset request”. 0: The event cannot generate “reset request”. 1: The event can generate “reset request”.
23:22	Reserved	Must be kept at reset value
21	CH1SSTCNTRST	Slave_TIMERx counter reset event generates channel 1 “set request” Refer to CH0SSTCNTRST description.
20	CH0SSTCNTRST	Slave_TIMERx counter reset event generates channel 0 “set request” 0: The counter reset event cannot generate channel 0 “set request”. 1: The counter reset event can generate channel 0 “set request”.
19	CH1SSTEV10	Slave_TIMERx interconnection event 10 generates channel 1 “set request” Refer to CH0SSTEV9 description.
18	CH1SSTEV9	Slave_TIMERx interconnection event 9 generates channel 1 “set request” Refer to CH0SSTEV9 description.
17	CH0SSTEV10	Slave_TIMERx interconnection event 10 generates channel 0 “set request” Refer to CH0SSTEV9 description.
16	CH0SSTEV9	Slave_TIMERx interconnection event 9 generates channel 1 “set request” When this bit is set, Slave_TIMERx interconnection event 0 can generate “set request”. 0: The event cannot generate “set request”. 1: The event can generate “set request”.
15:7	Reserved	Must be kept at reset value
6	ST7CMP3RST	Slave_TIMER7 compare 3 event resets counter This bit specifies whether the Slave_TIMER7 compare 3 event can reset the counter.

0: Slave_TIMER7 compare 3 event do not reset counter
1: Slave_TIMER7 compare 3 event resets counter

5 ST7CMP1RST Slave_TIMER7 compare 1 event resets counter
This bit specifies whether the Slave_TIMER7 compare 1 event can reset the counter.
0: Slave_TIMER7 compare 1 event do not reset counter
1: Slave_TIMER7 compare 1 event resets counter

4 ST7CMP0RST Slave_TIMER7 compare 0 event resets counter
This bit specifies whether the Slave_TIMER7 compare 0 event can reset the counter.
0: Slave_TIMER7 compare 0 event do not reset counter
1: Slave_TIMER7 compare 0 event resets counter

3 ST5CMP3RST Slave_TIMER5 compare 3 event resets counter
This bit specifies whether the Slave_TIMER6 compare 3 event can reset the counter.
0: Slave_TIMER5 compare 3 event do not reset counter
1: Slave_TIMER5 compare 3 event resets counter

2 ST5CMP1RST Slave_TIMER5 compare 1 event resets counter
This bit specifies whether the Slave_TIMER6 compare 1 event can reset the counter.
0: Slave_TIMER5 compare 1 event do not reset counter
1: Slave_TIMER5 compare 1 event resets counter

1 ST5CMP0RST Slave_TIMER5 compare 0 event resets counter
This bit specifies whether the Slave_TIMER6 compare 0 event can reset the counter.
0: Slave_TIMER5 compare 0 event do not reset counter
1: Slave_TIMER5 compare 0 event resets counter

0 ST4CMP3RST Slave_TIMER4 compare 3 event resets counter
This bit specifies whether the Slave_TIMER5 compare 3 event can reset the counter.
0: Slave_TIMER4 compare 3 event do not reset counter
1: Slave_TIMER4 compare 3 event resets counter

For Slave_TIMER7

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		CH1RSSTCNTRST	CH0RSSTCNTRST	CH1RSSTEV10	CH1RSSTEV9	CH0RSSTEV10	CH0RSSTEV9	Reserved		CH1SSTCNTRST	CH0SSTCNTRST	CH1SSTEV10	CH1SSTEV9	CH0SSTEV10	CH0SSTEV9
		rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Reserved	ST6CMP 3RST	ST6CMP 1RST	ST6CMP 0RST	ST5CMP 3RST	ST5CMP 1RST	ST5CMP 0RST	ST4CMP 3RST
	rw	rw	rw	rw	rw	rw	

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value
29	CH1RSSTCNTRST	Slave_TIMERx counter reset event generates channel 1 “reset request” Refer to CH0SSTCNTRST description.
28	CH0RSSTCNTRST	Slave_TIMERx counter reset event generates channel 0 “reset request” Refer to CH0SSTCNTRST description.
27	CH1RSSTEV10	Slave_TIMERx interconnection event 10 generates channel 1 “reset request” Refer to CH0RSSTEV9 description.
26	CH1RSSTEV9	Slave_TIMERx interconnection event 9 generates channel 1 “reset request” Refer to CH0RSSTEV9 description.
25	CH0RSSTEV10	Slave_TIMERx interconnection event 10 generates channel 0 “reset request” Refer to CH0RSSTEV9 description.
24	CH0RSSTEV9	Slave_TIMERx interconnection event 9 generates channel 0 “reset request” When this bit is set, Slave_TIMERx interconnection event 0 can generate channel “reset request”. 0: The event cannot generate “reset request”. 1: The event can generate “reset request”.
23:22	Reserved	Must be kept at reset value
21	CH1SSTCNTRST	Slave_TIMERx counter reset event generates channel 1 “set request” Refer to CH0SSTCNTRST description.
20	CH0SSTCNTRST	Slave_TIMERx counter reset event generates channel 0 “set request” 0: The counter reset event cannot generate channel 0 “set request”. 1: The counter reset event can generate channel 0 “set request”.
19	CH1SSTEV10	Slave_TIMERx interconnection event 10 generates channel 1 “set request” Refer to CH0SSTEV9 description.
18	CH1SSTEV9	Slave_TIMERx interconnection event 9 generates channel 1 “set request” Refer to CH0SSTEV9 description.
17	CH0SSTEV10	Slave_TIMERx interconnection event 10 generates channel 0 “set request” Refer to CH0SSTEV9 description.
16	CH0SSTEV9	Slave_TIMERx interconnection event 9 generates channel 1 “set request” When this bit is set, Slave_TIMERx interconnection event 0 can generate “set request”. 0: The event cannot generate “set request”.

		1: The event can generate “set request”.
15:7	Reserved	Must be kept at reset value
6	ST6CMP3RST	Slave_TIMER6 compare 3 event resets counter This bit specifies whether the Slave_TIMER6 compare 3 event can reset the counter. 0: Slave_TIMER6 compare 3 event do not reset counter 1: Slave_TIMER6 compare 3 event resets counter
5	ST6CMP1RST	Slave_TIMER6 compare 1 event resets counter This bit specifies whether the Slave_TIMER6 compare 1 event can reset the counter. 0: Slave_TIMER6 compare 1 event do not reset counter 1: Slave_TIMER6 compare 1 event resets counter
4	ST6CMP0RST	Slave_TIMER6 compare 0 event resets counter This bit specifies whether the Slave_TIMER6 compare 0 event can reset the counter. 0: Slave_TIMER6 compare 0 event do not reset counter 1: Slave_TIMER6 compare 0 event resets counter
3	ST5CMP3RST	Slave_TIMER5 compare 3 event resets counter This bit specifies whether the Slave_TIMER5 compare 3 event can reset the counter. 0: Slave_TIMER5 compare 3 event do not reset counter 1: Slave_TIMER5 compare 3 event resets counter
2	ST5CMP1RST	Slave_TIMER5 compare 1 event resets counter This bit specifies whether the Slave_TIMER5 compare 1 event can reset the counter. 0: Slave_TIMER5 compare 1 event do not reset counter 1: Slave_TIMER5 compare 1 event resets counter
1	ST5CMP0RST	Slave_TIMER5 compare 0 event resets counter This bit specifies whether the Slave_TIMER5 compare 0 event can reset the counter. 0: Slave_TIMER5 compare 0 event do not reset counter 1: Slave_TIMER5 compare 0 event resets counter
0	ST4CMP3RST	Slave_TIMER4 compare 3 event resets counter This bit specifies whether the Slave_TIMER4 compare 3 event can reset the counter. 0: Slave_TIMER4 compare 3 event do not reset counter 1: Slave_TIMER4 compare 3 event resets counter

HRTIMER Slave_TIMERx additional control register (HRTIMER_STxACTL)

Address offset: 0x7C

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DTFCFG[15:9]							Reserved								
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DTRCFG[15:9]							FLTAR	Reserved						UPBST7	UPBST6
rw							rw							rw	rw

Bits	Fields	Descriptions
31:25	DTFCFG[15:9]	Falling edge dead-time value configure This bit-field controls the value of the dead-time following a falling edge of output prepare signal (OyPRE,y=0,1): DTFvalue = DTFCFG[15:0]x $t_{HRTIMER_DTGCK}$, $t_{HRTIMER_DTGCK}$ = 1/ $f_{HRTIMER_DTGCK}$. Writing this register can change the high 7-bit of DTFCFG[15:0]. **Note:** (1) This bit-filed cannot be modified when DTFSVPROT bit in HRTIMER_STxDTCTL register is set. (2) This bit-filed is preloaded.
24:16	Reserved	Must be kept at reset value
15:9	DTRCFG[15:9]	Rising edge dead-time value configure This bit-field controls the value of the dead-time following a rising edge of output prepare signal (OyPRE,y=0,1): DTRvalue = DTRCFG[15:0]x $t_{HRTIMER_DTGCK}$, $t_{HRTIMER_DTGCK}$ = 1/ $f_{HRTIMER_DTGCK}$. Writing this register can change the high 7-bit of DTRCFG[15:0]. **Note:** (1) This bit-filed cannot be modified when DTRSVPROT bit in HRTIMER_STxDTCTL register is set. (2) This bit-filed is preloaded.
8	FLTAR	Fault automatic resume This bit selects is the outputs are automatically re-enabled when all falut sources are inactive. 0: Disabled 1: Enabled **Note:** This bit cannot be changed once the timer is operating (STxCEN bit set).
7:2	Reserved	Must be kept at reset value
1	UPBST7	Update by Slave_TIMER7 update event When the bit is set, the Slave_TIMERx(x=0,1,2,3,4,5,6) update event are

		synchronized with Slave_TIMER7 update event and the active registers of them are updated by the Slave_TIMER7 update event 0: The active registers is not update by Slave_TIMER7. 1: The active registers is update by Slave_TIMER7. **Note:** This bit does not exist in Slave_TIMER7.
0	UPBST6	Update by Slave_TIMER6 update event When the bit is set, the Slave_TIMERx(x=0,1,2,3,4,5,7) update event are synchronized with Slave_TIMER6 update event and the active registers of them are updated by the Slave_TIMER6 update event 0: The active registers is not update by Slave_TIMER6. 1: The active registers is update by Slave_TIMER6. **Note:** This bit does not exist in Slave_TIMER6.

25.5.3. Common registers

HRTIMER Common registers base address: 0x4001 5B80

HRTIMER control register 0 (HRTIMER_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
ADTG3USRC[3]	ADCTG2USRC[3]	ADCTG1USRC[3]	ADCTG0USRC[3]	ADTG3USRC[2:0]			ADTG2USRC[2:0]			ADTG1USRC[2:0]			ADTG0USRC[2:0]		
rw	rw	rw	rw	rw			rw			rw			rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							ST7UPDIS	ST6UPDIS	ST5UPDIS	ST4UPDIS	ST3UPDIS	ST2UPDIS	ST1UPDIS	ST0UPDIS	MTUPDIS
							rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31	ADTG3USRC[3]	Refer to ADTG0USRC[2:0].
30	ADTG2USRC[3]	Refer to ADTG0USRC[2:0].
29	ADTG1USRC[3]	Refer to ADTG0USRC[2:0].
28	ADTG0USRC[3]	Refer to ADTG0USRC[2:0].
27:25	ADTG3USRC[2:0]	HRTIMER_ADCTRIG3 update source This bit-field can be configured by software to specify the source to update the HRTIMER_ADCTRIGS3 register. 0000: Master_TIMER update event

0001: Slaver_TIMER0 update event
0010: Slaver_TIMER1 update event
0011: Slaver_TIMER2 update event
0100: Slaver_TIMER3 update event
0101: Slaver_TIMER4 update event
0110: Slaver_TIMER5 update event
0111: Slaver_TIMER6 update event
1000: Slaver_TIMER7 update event
Other values are reserved

24:22	ADTG2USRC[2:0]	HRTIMER_ADCTRIG2 update source This bit-field can be configured by software to specify the the source to update the HRTIMER_ADCTRIGS2 register. 0000: Master_TIMER update event 0001: Slaver_TIMER0 update event 0010: Slaver_TIMER1 update event 0011: Slaver_TIMER2 update event 0100: Slaver_TIMER3 update event 0101: Slaver_TIMER4 update event 0110: Slaver_TIMER5 update event 0111: Slaver_TIMER6 update event 1000: Slaver_TIMER7 update event Other values are reserved
21:19	ADTG1USRC[2:0]	HRTIMER_ADCTRIG1 update source This bit-field can be configured by software to specify the the source to update the HRTIMER_ADCTRIGS1 register. 0000: Master_TIMER update event 0001: Slaver_TIMER0 update event 0010: Slaver_TIMER1 update event 0011: Slaver_TIMER2 update event 0100: Slaver_TIMER3 update event 0101: Slaver_TIMER4 update event 0110: Slaver_TIMER5 update event 0111: Slaver_TIMER6 update event 1000: Slaver_TIMER7 update event Other values are reserved
18:16	ADTG0USRC[2:0]	HRTIMER_ADCTRIG0 update source This bit-field can be configured by software to specify the the source to update the HRTIMER_ADCTRIGS0 register. 0000: Master_TIMER update event 0001: Slaver_TIMER0 update event 0010: Slaver_TIMER1 update event 0011: Slaver_TIMER2 update event

		0100: Slaver_TIMER3 update event 0101: Slaver_TIMER4 update event 0110: Slaver_TIMER5 update event 0111: Slaver_TIMER6 update event 1000: Slaver_TIMER7 update event Other values are reserved
15:9	Reserved	Must be kept at reset value
8	ST7UPDIS	Slave_TIMER7 update disable This bit is used to enable or disable the update event generation. 0: Update event enable. 1: Update event disable.
7	ST6UPDIS	Slave_TIMER6 update disable This bit is used to enable or disable the update event generation. 0: Update event enable. 1: Update event disable.
6	ST5UPDIS	Slave_TIMER5 update disable This bit is used to enable or disable the update event generation. 0: Update event enable. 1: Update event disable.
5	ST4UPDIS	Slave_TIMER4 update disable This bit is used to enable or disable the update event generation. 0: Update event enable. 1: Update event disable.
4	ST3UPDIS	Slave_TIMER3 update disable This bit is used to enable or disable the update event generation. 0: Update event enable. 1: Update event disable.
3	ST2UPDIS	Slave_TIMER2 update disable This bit is used to enable or disable the update event generation. 0: Update event enable. 1: Update event disable.
2	ST1UPDIS	Slave_TIMER1 update disable This bit is used to enable or disable the update event generation. 0: Update event enable. 1: Update event disable.
1	ST0UPDIS	Slave_TIMER0 update disable This bit is used to enable or disable the update event generation. 0: Update event enable.

		1: Update event disable.
0	MTUPDIS	Master_TIMER update disable This bit is used to enable or disable the update event generation. 0: Update event enable. 1: Update event disable.

HRTIMER control register 1 (HRTIMER_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		ST7SRST	ST7SUP	Reserved				EXC7	EXC6	EXC5	EXC4	EXC3	EXC2	EXC1	EXC0
		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
ST6SRST	ST5SRST	ST4SRST	ST3SRST	ST2SRST	ST1SRST	ST0SRST	MTSRST	ST6SUP	ST5SUP	ST4SUP	ST3SUP	ST2SUP	ST1SUP	ST0SUP	MTSUP
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	ST7SRST	Slave_TIMER7 software reset Refer to ST0SRST.
28	ST7SUP	Slave_TIMER7 software update Refer to ST0SUP.
27:24	Reserved	Must be kept at reset value
23	EXC7	Exchange Slave_TIMER7 outputs Refer to EXC0.
22	EXC6	Exchange Slave_TIMER6 outputs Refer to EXC0.
21	EXC5	Exchange Slave_TIMER5 outputs Refer to EXC0.
20	EXC4	Exchange Slave_TIMER4 outputs Refer to EXC0.
19	EXC3	Exchange Slave_TIMER3 outputs Refer to EXC0.
18	EXC2	Exchange Slave_TIMER2 outputs Refer to EXC0.

17	EXC1	Exchange Slave_TIMER1 outputs Refer to EXC0.
16	EXC0	Exchange Slave_TIMER0 outputs This bit enables the exchange of Slave_TIMER7 outputs. 0: HRTIMER_STxCH0SET and HRTIMER_STxCH0RST encode the CHO output, while HRTIMER_STxCH1SET and HRTIMER_STxCH1RST encode the CH1 output. 1: HRTIMER_STxCH0SET and HRTIMER_STxCH0RST encode the CH1 output, while HRTIMER_STxCH1SET and HRTIMER_STxCH1RST encode the CH0 output.
15	ST6SRST	Slave_TIMER6 software reset Refer to ST0SRST.
14	ST5SRST	Slave_TIMER5 software reset Refer to ST0SRST.
13	ST4SRST	Slave_TIMER4 software reset Refer to ST0SRST.
12	ST3SRST	Slave_TIMER3 software reset Refer to ST0SRST.
11	ST2SRST	Slave_TIMER2 software reset Refer to ST0SRST.
10	ST1SRST	Slave_TIMER1 software reset Refer to ST0SRST.
9	ST0SRST	Slave_TIMER0 software reset This bit can be set by software, and cleared by hardware automatically. When this bit is set, the counter is reset. 0: No effect. 1: The counter is reset.
8	MTSRST	Master_TIMER software reset This bit can be set by software, and cleared by hardware automatically. When this bit is set, the counter is reset. 0: No effect. 1: The counter is reset.
7	ST6SUP	Slave_TIMER6 software update Refer to ST0SUP.
6	ST5SUP	Slave_TIMER5 software update Refer to ST0SUP.
5	ST4SUP	Slave_TIMER4 software update

		Refer to ST0SUP.
4	ST3SUP	Slave_TIMER3 software update Refer to ST0SUP.
3	ST2SUP	Slave_TIMER2 software update Refer to ST0SUP.
2	ST1SUP	Slave_TIMER1 software update Refer to ST0SUP.
1	ST0SUP	Slave_TIMER0 software update This bit can be set by software, and cleared by hardware automatically. When this bit is set, the content of shadow register is transferred to the active register and any pending update request is cancelled. 0: No effect. 1: update generated.
0	MTSUP	Master_TIMER software update This bit can be set by software, and cleared by hardware automatically. When this bit is set, the content of shadow register is transferred to the active register and any pending update request is cancelled. 0: No effect. 1: update generated.

HRTIMER interrupt flag register (HRTIMER_INTF)

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														BMPERIF	DLLCALIF
														r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							FLT7IF	FLT6IF	FLT5IF	SYSFLTIF	FLT4IF	FLT3IF	FLT2IF	FLT1IF	FLT0IF
							r	r	r	r	r	r	r	r	r

Bits	Fields	Descriptions
31:18	Reserved	Must be kept at reset value
17	BMPERIF	Bunch mode period interrupt flag This flag is set by hardware when bunch mode period has reached. It is cleared by software writing it at 1. 0: No bunch mode period interrupt occurred

		1: Bunch mode period interrupt occurred
16	DLLCALIF	DLL calibration completed interrupt flag This flag is set by hardware when the DLL calibration is completed. It is cleared by software writing it at 1. 0: No DLL calibration completed interrupt occurred 1: DLL calibration completed interrupt occurred
15:9	Reserved	Must be kept at reset value
8	FLT7IF	Fault 7 interrupt flag Refer to FLT0IF description.
7	FLT6IF	Fault 6 interrupt flag Refer to FLT0IF description.
6	FLT5IF	Fault 5 interrupt flag Refer to FLT0IF description.
5	SYSFLTIF	System fault interrupt flag This flag is set by hardware when the system fault occurred. It is cleared by software writing it at 1. 0: No system fault interrupt occurred 1: System fault completed interrupt occurred
4	FLT4IF	Fault 4 interrupt flag Refer to FLT0IF description.
3	FLT3IF	Fault 3 interrupt flag Refer to FLT0IF description.
2	FLT2IF	Fault 2 interrupt flag Refer to FLT0IF description.
1	FLT1IF	Fault 1 interrupt flag Refer to FLT0IF description.
0	FLT0IF	Fault 0 interrupt flag This flag is set by hardware when the fault 0 occurred. It is cleared by software writing it at 1. 0: No fault 0 interrupt occurred 1: Fault 0 completed interrupt occurred

HRTIMER interrupt flag clear register (HRTIMER_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														BMPERIFC	DLLCALIFC
														w	w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							FLT7IFC	FLT6IFC	FLT5IFC	SYSFLTIFC	FLT4IFC	FLT3IFC	FLT2IFC	FLT1IFC	FLT0IFC
							w	w	w	w	w	w	w	w	w

Bits	Fields	Descriptions
31:18	Reserved	Must be kept at reset value
17	BMPERIFC	Clear bunch mode period interrupt flag Writing 1 to this bit clears the BMPERIF in HRTIMER_INTF register. 0: No effect 1: Clear bunch mode period interrupt flag
16	DLLCALIFC	Clear DLL calibration completed interrupt flag Writing 1 to this bit clears the DLLCALIF in HRTIMER_INTF register. 0: No effect 1: Clear DLL calibration completed interrupt flag
15:9	Reserved	Must be kept at reset value
8	FLT7IFC	Clear fault 7 interrupt flag Writing 1 to this bit clears the FLT7IF in HRTIMER_INTF register. Refer to FLT0IF description.
7	FLT6IFC	Clear fault 6 interrupt flag Writing 1 to this bit clears the FLT6IF in HRTIMER_INTF register. Refer to FLT0IF description.
6	FLT5IFC	Clear fault 5 interrupt flag Writing 1 to this bit clears the FLT5IF in HRTIMER_INTF register. Refer to FLT0IF description.
5	SYSFLTIFC	Clear system fault interrupt flag Writing 1 to this bit clears the SYSFLTIF in HRTIMER_INTF register. 0: No effect 1: Clear system fault completed interrupt flag
4	FLT4IFC	Clear fault 4 interrupt flag Writing 1 to this bit clears the FLT4IF in HRTIMER_INTF register. Refer to FLT0IF description.
3	FLT3IFC	Clear fault 3 interrupt flag Writing 1 to this bit clears the FLT3IF in HRTIMER_INTF register.

		Refer to FLT0IF description.
2	FLT2IFC	Clear fault 2 interrupt flag Writing 1 to this bit clears the FLT2IF in HRTIMER_INTF register. Refer to FLT0IF description.
1	FLT1IFC	Clear fault 1 interrupt flag Writing 1 to this bit clears the FLT1IF in HRTIMER_INTF register. Refer to FLT0IF description.
0	FLT0IFC	Clear fault 0 interrupt flag Writing 1 to this bit clears the FLT0IF in HRTIMER_INTF register. 0: No effect 1: Clear fault 0 completed interrupt flag

HRTIMER interrupt enable register (HRTIMER_INTEN)

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														BMPERIE	DLLCALIE
														rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							FLT7IE	FLT6IE	FLT5IE	SYSFLTIE	FLT4IE	FLT3IE	FLT2IE	FLT1IE	FLT0IE
							rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:18	Reserved	Must be kept at reset value
17	BMPERIE	Bunch mode period interrupt enable 0: Disabled 1: Enabled
16	DLLCALIE	DLL calibration completed interrupt enable 0: disabled 1: enabled
15:9	Reserved	Must be kept at reset value
8	FLT7IE	fault 7 interrupt enable 0: disabled 1: enabled

7	FLT6IE	fault 6 interrupt enable 0: disabled 1: enabled
6	FLT5IE	fault 5 interrupt enable 0: disabled 1: enabled
5	SYSFLTIE	System fault interrupt enable 0: disabled 1: enabled
4	FLT4IE	fault 4 interrupt enable 0: disabled 1: enabled
3	FLT3IE	fault 3 interrupt enable 0: disabled 1: enabled
2	FLT2IE	fault 2 interrupt enable 0: disabled 1: enabled
1	FLT1IE	fault 1 interrupt enable 0: disabled 1: enabled
0	FLT0IE	fault 0 interrupt enable 0: disabled 1: enabled

HRTIMER channel output enable register (HRTIMER_CHOUTEN)

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
ST7CH1EN	ST7CH0EN	ST6CH1EN	ST6CH0EN	ST5CH1EN	ST5CH0EN	ST4CH1EN	ST4CH0EN	ST3CH1EN	ST3CH0EN	ST2CH1EN	ST2CH0EN	ST1CH1EN	ST1CH0EN	ST0CH1EN	ST0CH0EN
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	ST7CH1EN	Slave_TIMER7 channel 1 output (ST7CH1_O) enable Refer to ST0CH0EN description. **Note:** The disable status corresponds to both Idle and Fault states which is given by ST7CH1DISF bit in the HRTIMER_CHOUTDISF register.
14	ST7CH0EN	Slave_TIMER7 channel 0 output (ST7CH0_O) enable Refer to ST0CH0EN description. **Note:** The disable status corresponds to both Idle and Fault states which is given by ST7CH0DISF bit in the HRTIMER_CHOUTDISF register.
13	ST6CH1EN	Slave_TIMER6 channel 1 output (ST6CH1_O) enable Refer to ST0CH0EN description. **Note:** The disable status corresponds to both Idle and Fault states which is given by ST6CH1DISF bit in the HRTIMER_CHOUTDISF register.
12	ST6CH0EN	Slave_TIMER6 channel 0 output (ST6CH0_O) enable Refer to ST0CH0EN description. **Note:** The disable status corresponds to both Idle and Fault states which is given by ST6CH0DISF bit in the HRTIMER_CHOUTDISF register.
11	ST5CH1EN	Slave_TIMER5 channel 1 output (ST5CH1_O) enable Refer to ST0CH0EN description. **Note:** The disable status corresponds to both Idle and Fault states which is given by ST5CH1DISF bit in the HRTIMER_CHOUTDISF register.
10	ST5CH0EN	Slave_TIMER5 channel 0 output (ST5CH0_O) enable Refer to ST0CH0EN description. **Note:** The disable status corresponds to both Idle and Fault states which is given by ST5CH0DISF bit in the HRTIMER_CHOUTDISF register.
9	ST4CH1EN	Slave_TIMER4 channel 1 output (ST4CH1_O) enable Refer to ST0CH0EN description. **Note:** The disable status corresponds to both Idle and Fault states which is given by ST4CH1DISF bit in the HRTIMER_CHOUTDISF register.
8	ST4CH0EN	Slave_TIMER4 channel 0 output (ST4CH0_O) enable Refer to ST0CH0EN description. **Note:** The disable status corresponds to both Idle and Fault states which is given by ST4CH0DISF bit in the HRTIMER_CHOUTDISF register.
7	ST3CH1EN	Slave_TIMER3 channel 1 output (ST3CH1_O) enable Refer to ST0CH0EN description. **Note:** The disable status corresponds to both Idle and Fault states which is given by ST3CH1DISF bit in the HRTIMER_CHOUTDISF register.

6	ST3CH0EN	Slave_TIMER3 channel 0 output (ST3CH0_O) enable Refer to ST0CH0EN description. **Note:** The disable status corresponds to both Idle and Fault states which is given by ST3CH0DISF bit in the HRTIMER_CHOUTDISF register.
5	ST2CH1EN	Slave_TIMER2 channel 1 output (ST2CH1_O) enable Refer to ST0CH0EN description. **Note:** The disable status corresponds to both Idle and Fault states which is given by ST2CH1DISF bit in the HRTIMER_CHOUTDISF register.
4	ST2CH0EN	Slave_TIMER2 channel 0 output (ST2CH0_O) enable Refer to ST0CH0EN description. **Note:** The disable status corresponds to both Idle and Fault states which is given by ST2CH0DISF bit in the HRTIMER_CHOUTDISF register.
3	ST1CH1EN	Slave_TIMER1 channel 1 output (ST1CH1_O) enable Refer to ST0CH0EN description. **Note:** The disable status corresponds to both Idle and Fault states which is given by ST1CH1DISF bit in the HRTIMER_CHOUTDISF register.
2	ST1CH0EN	Slave_TIMER1 channel 0 output (ST1CH0_O) enable Refer to ST0CH0EN description. **Note:** The disable status corresponds to both Idle and Fault states which is given by ST1CH0DISF bit in the HRTIMER_CHOUTDISF register.
1	ST0CH1EN	Slave_TIMER0 channel 1 output (ST0CH1_O) enable Refer to ST0CH0EN description. **Note:** The disable status corresponds to both Idle and Fault states which is given by ST0CH1DISF bit in the HRTIMER_CHOUTDISF register.
0	ST0CH0EN	Slave_TIMER0 channel 0 output (ST0CH0_O) enable Writing 1 to the bit to enable output and writing 0 is has no effect. When fault input is active, the bit is cleared asynchronously by hardware. Reading the bit returns the output enable or disable status. 0: Slave_TIMER0 channel 0 output ST0CH0_O disabled. The output is either in Fault state or Idle state. 1: Slave_TIMER0 channel 0 output ST0CH0_O enabled **Note:** The disable status corresponds to both Idle and Fault states which is given by ST0CH0DISF bit in the HRTIMER_CHOUTDISF register.

HRTIMER channel output disable register (HRTIMER_CHOUTDIS)

Address offset: 0x18

Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ST7CH1DIS	ST7CH0DIS	ST6CH1DIS	ST6CH0DIS	ST5CH1DIS	ST5CH0DIS	ST4CH1DIS	ST4CH0DIS	ST3CH1DIS	ST3CH0DIS	ST2CH1DIS	ST2CH0DIS	ST1CH1DIS	ST1CH0DIS	ST0CH1DIS	ST0CH0DIS
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	ST7CH1DIS	Slave_TIMER7 channel 1 output (ST7CH1_O) disable. Refer to ST0CH0DIS description.
14	ST7CH0DIS	Slave_TIMER7 channel 0 output (ST7CH0_O) disable. Refer to ST0CH0DIS description.
13	ST6CH1DIS	Slave_TIMER6 channel 1 output (ST6CH1_O) disable. Refer to ST0CH0DIS description.
12	ST6CH0DIS	Slave_TIMER6 channel 0 output (ST6CH0_O) disable. Refer to ST0CH0DIS description.
11	ST5CH1DIS	Slave_TIMER5 channel 1 output (ST5CH1_O) disable. Refer to ST0CH0DIS description.
10	ST5CH0DIS	Slave_TIMER5 channel 0 output (ST5CH0_O) disable. Refer to ST0CH0DIS description.
9	ST4CH1DIS	Slave_TIMER4 channel 1 output (ST4CH1_O) disable. Refer to ST0CH0DIS description.
8	ST4CH0DIS	Slave_TIMER4 channel 0 output (ST4CH0_O) disable. Refer to ST0CH0DIS description.
7	ST3CH1DIS	Slave_TIMER3 channel 1 output (ST3CH1_O) disable. Refer to ST0CH0DIS description.
6	ST3CH0DIS	Slave_TIMER3 channel 0 output (ST3CH0_O) disable. Refer to ST0CH0DIS description.
5	ST2CH1DIS	Slave_TIMER2 channel 1 output (ST2CH1_O) disable. Refer to ST0CH0DIS description.
4	ST2CH0DIS	Slave_TIMER2 channel 0 output (ST2CH0_O) disable. Refer to ST0CH0DIS description.
3	ST1CH1DIS	Slave_TIMER1 channel 1 output (ST1CH1_O) disable. Refer to ST0CH0DIS description.

2	ST1CH0DIS	Slave_TIMER1 channel 0 output (ST1CH0_O) disable. Refer to ST0CH0DIS description.
1	ST0CH1DIS	Slave_TIMER0 channel 1 output (ST0CH1_O) disable. Refer to ST0CH0DIS description.
0	ST0CH0DIS	Slave_TIMER0 channel 0 output (ST0CH0_O) disable. Writing 1 to the bit to disable output and channel 0 enters Idle state. Writing 0 has no effect 0: No effect. 1: Slave_TIMER0 channel 0 output ST0CH0_O disabled. The output entes the Idle state, either from the Fault state or the Run state.

HRTIMER channel output disable flag register (HRTIMER_CHOUTDISF)

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															
r	r	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
ST7CH1DISF	ST7CH0DISF	ST6CH1DISF	ST6CH0DISF	ST5CH1DISF	ST5CH0DISF	ST4CH1DISF	ST4CH0DISF	ST3CH1DISF	ST3CH0DISF	ST2CH1DISF	ST2CH0DISF	ST1CH1DISF	ST1CH0DISF	ST0CH1DISF	ST0CH0DISF
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	ST7CH1DISF	Slave_TIMER7 channel 1 output (ST7CH1_O) disable flag. Refer to ST0CH0DISF description.
14	ST7CH0DISF	Slave_TIMER7 channel 0 output (ST7CH0_O) disable flag. Refer to ST0CH0DISF description.
13	ST6CH1DISF	Slave_TIMER6 channel 1 output (ST6CH1_O) disable flag. Refer to ST0CH0DISF description.
12	ST6CH0DISF	Slave_TIMER6 channel 0 output (ST6CH0_O) disable flag. Refer to ST0CH0DISF description.
11	ST5CH1DISF	Slave_TIMER5 channel 1 output (ST5CH1_O) disable flag. Refer to ST0CH0DISF description.
10	ST5CH0DISF	Slave_TIMER5 channel 0 output (ST5CH0_O) disable flag.

		Refer to ST0CH0DISF description.
9	ST4CH1DISF	Slave_TIMER4 channel 1 output (ST4CH1_O) disable flag. Refer to ST0CH0DISF description.
8	ST4CH0DISF	Slave_TIMER4 channel 0 output (ST4CH0_O) disable flag. Refer to ST0CH0DISF description.
7	ST3CH1DISF	Slave_TIMER3 channel 1 output (ST3CH1_O) disable flag. Refer to ST0CH0DISF description.
6	ST3CH0DISF	Slave_TIMER3 channel 0 output (ST3CH0_O) disable flag. Refer to ST0CH0DISF description.
5	ST2CH1DISF	Slave_TIMER2 channel 1 output (ST2CH1_O) disable flag. Refer to ST0CH0DISF description.
4	ST2CH0DISF	Slave_TIMER2 channel 0 output (ST2CH0_O) disable flag. Refer to ST0CH0DISF description.
3	ST1CH1DISF	Slave_TIMER1 channel 1 output (ST1CH1_O) disable flag. Refer to ST0CH0DISF description.
2	ST1CH0DISF	Slave_TIMER1 channel 0 output (ST1CH0_O) disable flag. Refer to ST0CH0DISF description.
1	ST0CH1DISF	Slave_TIMER0 channel 1 output (ST0CH1_O) disable flag. Refer to ST0CH0DISF description.
0	ST0CH0DISF	Slave_TIMER0 channel 0 output (ST0CH0_O) disable flag. Reading the bit returns the channel 0 output disable state. It is not significant when the output is enabled. 0: Slave_TIMER0 channel 0 output ST0CH0_O disabled, in Idle state. 1: Slave_TIMER0 channel 0 output ST0CH0_O disabled, in Fault state

HRTIMER bunch mode control register (HRTIMER_BMCTL)

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
BMOPTF	Reserved						BMST7	BMST6	BMST5	BMST4	BMST3	BMST2	BMST1	BMST0	BMMT
rc_w0							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					BMSE	BMPSC[3:0]				BMCLKS[3:0]				BMCTN	BMEN
					rw	rw				rw				rw	rw

Bits	Fields	Descriptions
31	BMOPTF	Bunch mode operating flag This flag is set by hardware when the bunch mode is on-going. Writing this bit to 0 will terminate bunch mode. 0: Normal operation. Bunch mode is not operation. 1: Bunch mode operation on-going.
30:25	Reserved	Must be kept at reset value
24	BMST7	Slave_TIMER7 bunch mode 0: Slave_TIMER7 counter clock(HRTIMER_PSCCK) is maintained and the counter operates normally 1: Slave_TIMER7 counter clock(HRTIMER_PSCCK) is stopped and the counter is reset **Note:** (1) This bit cannot be changed while the bunch mode is enabled. (2) This bit must not be set when the balanced IDLE mode is active (DLYISCH[2:0] = 3'bx11 in HRTIMER_STxCHOCTL register).
23	BMST6	Slave_TIMER6 bunch mode 0: Slave_TIMER6 counter clock(HRTIMER_PSCCK) is maintained and the counter operates normally 1: Slave_TIMER6 counter clock(HRTIMER_PSCCK) is stopped and the counter is reset **Note:** (1) This bit cannot be changed while the bunch mode is enabled. (2) This bit must not be set when the balanced IDLE mode is active (DLYISCH[2:0] = 3'bx11 in HRTIMER_STxCHOCTL register).
22	BMST5	Slave_TIMER5 bunch mode 0: Slave_TIMER5 counter clock(HRTIMER_PSCCK) is maintained and the counter operates normally 1: Slave_TIMER5 counter clock(HRTIMER_PSCCK) is stopped and the counter is reset **Note:** (1) This bit cannot be changed while the bunch mode is enabled. (2) This bit must not be set when the balanced IDLE mode is active (DLYISCH[2:0] = 3'bx11 in HRTIMER_STxCHOCTL register).
21	BMST4	Slave_TIMER4 bunch mode 0: Slave_TIMER4 counter clock(HRTIMER_PSCCK) is maintained and the counter operates normally 1: Slave_TIMER4 counter clock(HRTIMER_PSCCK) is stopped and the counter is reset **Note:** (1) This bit cannot be changed while the bunch mode is enabled.

(2) This bit must not be set when the balanced IDLE mode is active (DLYISCH[2:0] = 3'bx11 in HRTIMER_STxCHOCTL register).

20	BMST3	Slave_TIMER3 bunch mode 0: Slave_TIMER3 counter clock(HRTIMER_PSCCK) is maintained and the counter operates normally 1: Slave_TIMER3 counter clock(HRTIMER_PSCCK) is stopped and the counter is reset **Note:** (1) This bit cannot be changed while the bunch mode is enabled. (2) This bit must not be set when the balanced IDLE mode is active (DLYISCH[2:0] = 3'bx11 in HRTIMER_STxCHOCTL register).
19	BMST2	Slave_TIMER2 bunch mode 0: Slave_TIMER2 counter clock(HRTIMER_PSCCK) is maintained and the counter operates normally 1: Slave_TIMER2 counter clock(HRTIMER_PSCCK) is stopped and the counter is reset **Note:** (1) This bit cannot be changed while the bunch mode is enabled. (2) This bit must not be set when the balanced IDLE mode is active (DLYISCH[2:0] = 3'bx11 in HRTIMER_STxCHOCTL register).
18	BMST1	Slave_TIMER1 bunch mode 0: Slave_TIMER1 counter clock(HRTIMER_PSCCK) is maintained and the counter operates normally 1: Slave_TIMER1 counter clock(HRTIMER_PSCCK) is stopped and the counter is reset **Note:** (1) This bit cannot be changed while the bunch mode is enabled. (2) This bit must not be set when the balanced IDLE mode is active (DLYISCH[2:0] = 3'bx11 in HRTIMER_STxCHOCTL register).
17	BMST0	Slave_TIMER0 bunch mode 0: Slave_TIMER0 counter clock(HRTIMER_PSCCK) is maintained and the counter operates normally 1: Slave_TIMER0 counter clock(HRTIMER_PSCCK) is stopped and the counter is reset **Note:** (1) This bit cannot be changed while the bunch mode is enabled. (2) This bit must not be set when the balanced IDLE mode is active (DLYISCH[2:0] = 3'bx11 in HRTIMER_STxCHOCTL register).
16	BMMT	Master_TIMER bunch mode 0: Master_TIMER counter clock(HRTIMER_PSCCK) is maintained and the counter operates normally

		1: Master_TIMER counter clock(HRTIMER_PSCCK) is stopped and the counter is reset
15:11	Reserved	Must be kept at reset value
10	BMSE	Bunch mode shadow enable 0: The shadow registers for HRTIMER_BMCMPV and HRTIMER_BMCAR registers are disabled 1: The shadow registers for HRTIMER_BMCMPV and HRTIMER_BMCAR registers are enabled. **Note:** This bit cannot be changed while the bunch mode is enabled
9:6	BMPSC[3:0]	Bunch mode clock division This bit-field can be configured by software to specify division ratio between the HRTIMER clock (HRTIMER_CK) and bunch mode counter (HRTIMER_BMCNTCK) when BMCLKS[3:0] = 4'b1010 in HRTIMER_BMCTL register. $f_{HRTIMER_BMCNTCK} = f_{HRTIMER_CK}/2^{BMPSC[3:0]}$ 0000: $f_{HRTIMER_BMCNTCK} = f_{HRTIMER_CK}$ 0001: $f_{HRTIMER_BMCNTCK} = f_{HRTIMER_CK}/2$ 0010: $f_{HRTIMER_BMCNTCK} = f_{HRTIMER_CK}/4$ 0011: $f_{HRTIMER_BMCNTCK} = f_{HRTIMER_CK}/8$ 0100: $f_{HRTIMER_BMCNTCK} = f_{HRTIMER_CK}/16$ 0101: $f_{HRTIMER_BMCNTCK} = f_{HRTIMER_CK}/32$ 0110: $f_{HRTIMER_BMCNTCK} = f_{HRTIMER_CK}/64$ 0111: $f_{HRTIMER_BMCNTCK} = f_{HRTIMER_CK}/128$ 1000: $f_{HRTIMER_BMCNTCK} = f_{HRTIMER_CK}/256$ 1001: $f_{HRTIMER_BMCNTCK} = f_{HRTIMER_CK}/512$ 1010: $f_{HRTIMER_BMCNTCK} = f_{HRTIMER_CK}/1024$ 1011: $f_{HRTIMER_BMCNTCK} = f_{HRTIMER_CK}/2048$ 1100: $f_{HRTIMER_BMCNTCK} = f_{HRTIMER_CK}/4096$ 1101: $f_{HRTIMER_BMCNTCK} = f_{HRTIMER_CK}/8192$ 1110: $f_{HRTIMER_BMCNTCK} = f_{HRTIMER_CK}/16384$ 1111: $f_{HRTIMER_BMCNTCK} = f_{HRTIMER_CK}/32768$ **Note:** This bit cannot be changed while the bunch mode is enabled
5:2	BMCLKS[3:0]	Bunch mode clock source This bit-field defines the clock source for the bunch mode counter. 0000: Master_TIMER counter reset/roll-over event. 0001: Slave_TIMER0 counter reset/roll-over event. 0010: Slave_TIMER1 counter reset/roll-over event. 0011: Slave_TIMER2 counter reset/roll-over event. 0100: Slave_TIMER3 counter reset/roll-over event. 0101: Slave_TIMER4 counter reset/roll-over event. 0110: Chip internal signal 0 : BMCLK0 0111: Chip internal signal 1: BMCLK1

1000: Chip internal signal 2: BMCLK2
1001: Chip internal signal 3: BMCLK3
1010: Prescaled $f_{HRTIMER_CK}$ clock (as per BMPRSC[3:0] setting)
Other values are reserved
Note:
(1) This bit cannot be changed while the bunch mode is enabled
(2) BMCLKy (y=0..3):

1 BMCTN Continuous mode in bunch mode
0: Single pulse mode. The BM-counter stops by hardware when it reaches the HRTIMER_BMCAR value.
1: Continuous mode. The BM-counter rolls over to zero and counts continuously when it reaches the HRTIMER_BMCAR value

0 BMEN Bunch mode enable
The bunch mode controller is ready to receive the bunch mode start trigger when the bit is set. Writing this bit to 0 will terminate bunch mode.
0: Bunch mode disable.
1: Bunch mode enable.

HRTIMER bunch mode start trigger register (HRTIMER_BMSTRG)

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
CISGN	EXEV7	EXEV6	ST3EXEV7	ST0EXEV6	ST4CMP1	ST4CMP0	ST4REP	ST4RST	ST3CMP1	ST3CMP0	ST3REP	ST3RST	ST2CMP1	ST2CMP0	ST2REP
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
ST2RST	ST1CMP1	ST1CMP0	ST1REP	ST1RST	ST0CMP1	ST0CMP0	ST0REP	ST0RST	MTCMP3	MTCMP2	MTCMP1	MTCMP0	MTREP	MTRST	SWTRG
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31	CISGN	Chip internal signal triggers bunch mode operation Chip internal signal (TIMER6_TRGO) is starting the bunch mode operation. 0：No effect on bunch mode operation. 1：Chip internal signal is starting bunch mode operation.
30	EXEV7	External event 7 triggers bunch mode operation External event 7 is starting the bunch mode operation. 0：No effect on bunch mode operation. 1：External event 7 is starting bunch mode operation.

29	EXEV6	External event 6 triggers bunch mode operation External event 6 is starting the bunch mode operation. 0：No effect on bunch mode operation. 1：External event 6 is starting bunch mode operation.
28	ST3EXEV7	Slave_TIMER3 period event following external event 7 triggers bunch mode operation Slave_TIMER3 period event following external event 7 is starting the bunch mode operation. 0：No effect on bunch mode operation. 1：Slave_TIMER3 period event following external event 7 is starting bunch mode operation.
27	ST0EXEV6	Slave_TIMER0 period event following external event 6 triggers bunch mode operation Slave_TIMER0 period event following external event 6 is starting the bunch mode operation. 0：No effect on bunch mode operation. 1：Slave_TIMER0 period event following external event 6 is starting bunch mode operation.
26	ST4CMP1	Slave_TIMER4 compare 1 event triggers bunch mode operation Refer to MTCMP1 description.
25	ST4CMP0	Slave_TIMER4 compare 0 event triggers bunch mode operation Refer to MTCMP0 description.
24	ST4REP	Slave_TIMER4 repetition event triggers bunch mode operation Refer to MTREP description.
23	ST4RST	Slave_TIMER4 reset event triggers bunch mode operation Refer to MTRST description.
22	ST3CMP1	Slave_TIMER3 compare 1 event triggers bunch mode operation Refer to MTCMP1 description.
21	ST3CMP0	Slave_TIMER3 compare 0 event triggers bunch mode operation Refer to MTCMP0 description.
20	ST3REP	Slave_TIMER3 repetition event triggers bunch mode operation Refer to MTREP description.
19	ST3RST	Slave_TIMER3 reset event triggers bunch mode operation Refer to MTRST description.
18	ST2CMP1	Slave_TIMER2 compare 1 event triggers bunch mode operation Refer to MTCMP0 description.
17	ST2CMP0	Slave_TIMER2 compare 0 event triggers bunch mode operation

		Refer to MTCMP0 description.
16	ST2REP	Slave_TIMER2 repetition event triggers bunch mode operation Refer to MTREP description.
15	ST2RST	Slave_TIMER2 reset event triggers bunch mode operation Refer to MTRST description.
14	ST1CMP1	Slave_TIMER1 compare 1 event triggers bunch mode operation Refer to MTCMP1 description.
13	ST1CMP0	Slave_TIMER1 compare 0 event triggers bunch mode operation Refer to MTCMP0 description.
12	ST1REP	Slave_TIMER1 repetition event triggers bunch mode operation Refer to MTREP description.
11	ST1RST	Slave_TIMER1 reset event triggers bunch mode operation Refer to MTRST description.
10	ST0CMP1	Slave_TIMER0 compare 1 event triggers bunch mode operation Refer to MTCMP1 description.
9	ST0CMP0	Slave_TIMER0 compare 0 event triggers bunch mode operation Refer to MTCMP0 description.
8	ST0REP	Slave_TIMER0 repetition event triggers bunch mode operation Refer to MTREP description.
7	ST0RST	Slave_TIMER0 reset event triggers bunch mode operation Refer to MTRST description.
6	MTCMP3	Master_TIMER compare 3 event triggers bunch mode operation Refer to MTCMP0 description.
5	MTCMP2	Master_TIMER compare 2 event triggers bunch mode operation Refer to MTCMP0 description.
4	MTCMP1	Master_TIMER compare 1 event triggers bunch mode operation Refer to MTCMP0 description.
3	MTCMP0	Master_TIMER compare 0 event triggers bunch mode operation The Master_TIMER compare 0 event is starting the bunch mode operation. 0：No effect on bunch mode operation. 1：Master_TIMER compare 0 event is starting bunch mode operation.
2	MTREP	Master_TIMER repetition event triggers bunch mode operation The Master_TIMER repetition event is starting the bunch mode operation. 0：No effect on bunch mode operation. 1：Master_TIMER repetition event is starting bunch mode operation.

1	MTRST	Master_TIMER reset event triggers bunch mode operation The Master_TIMER reset and roll-over event is starting the bunch mode operation. 0：No effect on bunch mode operation. 1：Master_TIMER reset and roll-over event is starting bunch mode operation.
0	SWTRG	Software triggers bunch mode operation This bit is set by software and cleared by hardware automatically. When this bit is set, it will trigger bunch mode operation. This bit is not significant if the bunch mode is not enabled (BMEN bit in HRTIMER_BMCTL register is reset). 0：No effect on bunch mode operation. 1：Software trigger is starting bunch mode operation. **Note:** This bit is not active if the bunch mode is not enabled (BMEN bit is reset).

HRTIMER bunch mode compare value register (HRTIMER_BMCMPV)

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
BMCMPVAL[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:0	BMCMPVAL[15:0]	Bunch mode compare value This bit-field contains value to be compared to the BM-counter and defines the duration of the IDLE. This register has a shadow register and can be preloaded. **Note:** BMCMPVAL [15:0] cannot be set to 0x0000 when using the $f_{HRTIMER_CK}$ clock with a prescaler equaled to 0 as the bunch mode clock source (BMCLKS[3:0] = 4'b1010 and BMPSC[3:0] = 4'b0000).

HRTIMER bunch mode counter auto reload register (HRTIMER_BMCAR)

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
BMCARL[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:0	BMCARL[15:0]	Bunch mode counter auto reload value This bit-field specifies the auto reload value of the BM-counter and defines the bunch mode period which is the sum of the IDLE and RUN duration. This register has a shadow register and can be preloaded. **Note:** This bit-field must not be zero when the burst mode is enabled.

HRTIMER external event configuration register 0 (HRTIMER_EXEVCFG0)

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		EXEV4FAST	EXEV4EG[1:0]		EXEV4P	EXEV4SRC[1:0]		EXEV3FAST	EXEV3EG[1:0]		EXEV3P	EXEV3SRC[1:0]		EXEV2FAST	EXEV2EG[1]
		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
EXEV2EG[0]	EXEV2P	EXEV2SRC[1:0]		EXEV1FAST	EXEV1EG[1:0]		EXEV1P	EXEV1SRC[1:0]		EXEV0FAST	EXEV0EG[1:0]		EXEV0P	EXEV0SRC[1:0]	
rw	rw	rw		rw	rw		rw	rw		rw	rw		rw	rw	

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value
29	EXEV4FAST	External Event 4 fast mode Refer to EXEV0FAST description.
28:27	EXEV4EG[1:0]	External event 4 edge sensitivity Refer to EXEV0EG[1:0] description.
26	EXEV4P	External event 4 polarity Refer to EXEV0P description.
25:24	EXEV4SRC[1:0]	External event 4 source Refer to EXEV0SRC[1:0] description.
23	EXEV3FAST	External Event 3 fast mode Refer to EXEV0FAST description.

22:21	EXEV3EG[1:0]	External event 3 edge sensitivity Refer to EXEV0EG[1:0] description.
20	EXEV3P	External event 3 polarity Refer to EXEV0P description.
19:18	EXEV3SRC[1:0]	External event 3 source Refer to EXEV0SRC[1:0] description.
17	EXEV2FAST	External Event 2 fast mode Refer to EXEV0FAST description.
16:15	EXEV2EG[1:0]	External event 2 edge sensitivity Refer to EXEV0EG[1:0] description.
14	EXEV2P	External event 2 polarity Refer to EXEV0P description.
13:12	EXEV2SRC[1:0]	External event 2 source Refer to EXEV0SRC[1:0] description.
11	EXEV1FAST	External Event 1 fast mode Refer to EXEV0FAST description.
10:9	EXEV1EG[1:0]	External event 1 edge sensitivity Refer to EXEV0EG[1:0] description.
8	EXEV1P	External event 1 polarity Refer to EXEV0P description.
7:6	EXEV1SRC[1:0]	External event 1 source Refer to EXEV0SRC[1:0] description.
5	EXEV0FAST	External Event 0 fast mode 0: External Event 0 undergoes re-synchronization by the HRTIMER logic before influencing outputs, introducing a HRTIMER clock-related latency. 1: External Event 0 operates asynchronously on outputs (low-latency mode).
4:3	EXEV0EG[1:0]	External event 0 edge sensitivity This bit-field specifies the polarity of external event 0. 00: Level active. Active level is defined by EXEV0P bit 01: Rising edge active, EXEV0P bit is invalid. 10: Falling edge active, EXEV0P bit is invalid 11: Both edges active, EXEV0P bit is invalid.
2	EXEV0P	External event 0 polarity This bit specifies the active level of external event 0 when EXEV0EG[1:0] = 2'b00. 0: External event 0 active at high level. 1: External event 0 active at low level.

		Note: This bit cannot be changed once the Slave_TIMERx is enabled.
1:0	EXEV0SRC[1:0]	External event 0 source 00: External event 0 source is EXEV0SRC 0 01: External event 0 source is EXEV0SRC 1 10: External event 0 source is EXEV0SRC 2 11: External event 0 source is EXEV0SRC 3 **Note:** This bit cannot be changed once the Slave_TIMERx is enabled.

HRTIMER external event configuration register 1 (HRTIMER_EXEVCFG1)

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			EXEV9EG[1:0]		EXEV9P	EXEV9SRC[1:0]		Reserved	EXEV8EG[1:0]		EXEV8P	EXEV8SRC[1:0]		Reserved	EXEV7EG[1]
			rw		rw	rw			rw		rw	rw			rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXEV7EG[0]	EXEV7P	EXEV7SRC[1:0]		Reserved	EXEV6EG[1:0]		EXEV6P	EXEV6SRC[1:0]		Reserved	EXEV5EG[1:0]		EXEV5P	EXEV5SRC[1:0]	
rw	rw	rw			rw		rw	rw			rw		rw	rw	

Bits	Fields	Descriptions
31:29	Reserved	Must be kept at reset value
28:27	EXEV9EG[1:0]	External event 9 edge sensitivity Refer to EXEV0EG[1:0] in HRTIMER_EXEVCFG0 register description.
26	EXEV9P	External event 9 polarity Refer to EXEV0P in HRTIMER_EXEVCFG0 register description.
25:23	EXEV9SRC[1:0]	External event 9 source Refer to EXEV0SRC[1:0] in HRTIMER_EXEVCFG0 register description.
22:21	EXEV8EG[1:0]	External event 8 edge sensitivity Refer to EXEV0EG[1:0] in HRTIMER_EXEVCFG0 register description.
20	EXEV8P	External event 8 polarity Refer to EXEV0P in HRTIMER_EXEVCFG0 register description.
19:18	EXEV8SRC[1:0]	External event 8 source Refer to EXEV0SRC[1:0] in HRTIMER_EXEVCFG0 register description.
17	EXEV7FAST	External Event 7 Fast mode Refer to EXEV0FAST description.

16:15	EXEV7EG[1:0]	External event 7 edge sensitivity Refer to EXEV0EG[1:0] in HRTIMER_EXEVCFG0 register description.
14	EXEV7P	External event 7 polarity Refer to EXEV0P in HRTIMER_EXEVCFG0 register description.
13:11	EXEV7SRC[1:0]	External event 7 source Refer to EXEV0SRC[1:0] in HRTIMER_EXEVCFG0 register description.
10:9	EXEV6EG[1:0]	External event 6 edge sensitivity Refer to EXEV0EG[1:0] in HRTIMER_EXEVCFG0 register description.
8	EXEV6P	External event 6 polarity Refer to EXEV0P in HRTIMER_EXEVCFG0 register description.
7:5	EXEV6SRC[1:0]	External event 6 source Refer to EXEV0SRC[1:0] in HRTIMER_EXEVCFG0 register description.
4:3	EXEV5EG[1:0]	External event 5 edge sensitivity Refer to EXEV0EG[1:0] in HRTIMER_EXEVCFG0 register description.
2	EXEV5P	External event 5 polarity Refer to EXEV0P in HRTIMER_EXEVCFG0 register description.
1:0	EXEV5SRC[1:0]	External event 5 source Refer to EXEV0SRC[1:0] in HRTIMER_EXEVCFG0 register description.

HRTIMER external event digital filter control register (HRTIMER_EXEVDFCTL)

Address offset: 0x38

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:30	EXEVFDIV[2:0]	External event digital filter clock division This bit-field can be configured by software to specify division ratio between the HRTIMER clock (HRTIMER_CK) and the external event digital filter clock (HRTIMER_EXEVFCK). $f_{HRTIMER_EXEVFCK} = f_{HRTIMER_CK}/2^{EXEVFDIV[2:0]}$ 00: $f_{HRTIMER_EXEVFCK} = f_{HRTIMER_CK}$ 01: $f_{HRTIMER_EXEVFCK} = f_{HRTIMER_CK}/2$

		10: $f_{HRTIMER_EXEVFCK} = f_{HRTIMER_CK}/4$ 11: $f_{HRTIMER_EXEVFCK} = f_{HRTIMER_CK}/8$
29:28	Reserved	Must be kept at reset value
27:24	EXEV9FC[3:0]	External event 9 filter control Refer to EXEV5FC[3:0] description.
23:22	Reserved	Must be kept at reset value
21:18	EXEV8FC[3:0]	External event 8 filter control Refer to EXEV5FC[3:0] description.
17:16	Reserved	Must be kept at reset value
15:12	EXEV7FC[3:0]	External event 7 filter control Refer to EXEV5FC[3:0] description.
11:10	Reserved	Must be kept at reset value
9:6	EXEV6FC[3:0]	External event 6 filter control Refer to EXEV5FC[3:0] description.
5:4	Reserved	Must be kept at reset value
3:0	EXEV5FC[3:0]	External event 5 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 (f_{SAMP}) used to sample external event and the length of the digital filter applied to external event. 0000: Filter disalble. 0001: $f_{SAMP} = f_{HRTIMER_CK}$, N=2. 0010: $f_{SAMP} = f_{HRTIMER_CK}$, N=4. 0011: $f_{SAMP} = f_{HRTIMER_CK}$, N=8. 0100: $f_{SAMP} = f_{HRTIMER_EXEVFCK}$ /2, N=6. 0101: $f_{SAMP} = f_{HRTIMER_EXEVFCK}$ /2, N=8. 0110: $f_{SAMP} = f_{HRTIMER_EXEVFCK}$ /4, N=6. 0111: $f_{SAMP} = f_{HRTIMER_EXEVFCK}$ /4, N=8. 1000: $f_{SAMP} = f_{HRTIMER_EXEVFCK}$ /8, N=6. 1001: $f_{SAMP} = f_{HRTIMER_EXEVFCK}$ /8, N=8. 1010: $f_{SAMP} = f_{HRTIMER_EXEVFCK}$ /16, N=5. 1011: $f_{SAMP} = f_{HRTIMER_EXEVFCK}$ /16, N=6. 1100: $f_{SAMP} = f_{HRTIMER_EXEVFCK}$ /16, N=8. 1101: $f_{SAMP} = f_{HRTIMER_EXEVFCK}$ /32, N=5. 1110: $f_{SAMP} = f_{HRTIMER_EXEVFCK}$ /32, N=6. 1111: $f_{SAMP} = f_{HRTIMER_EXEVFCK}$ /32, N=8.

HRTIMER trigger source 0 to ADC register (HRTIMER_ADCTRIGS0)

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
TRG0ST4PER	TRG0ST4C3	TRG0ST4C2	TRG0ST4C1	TRG0ST3PER	TRG0ST3C3	TRG0ST3C2	TRG0ST3C1	TRG0ST2PER	TRG0ST2C3	TRG0ST2C2	TRG0ST2C1	TRG0ST1RST	TRG0ST1PER	TRG0ST1C3	TRG0ST1C2
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
TRG0ST1C1	TRG0ST0RST	TRG0ST0PER	TRG0ST0C3	TRG0ST0C2	TRG0ST0C1	TRG0EXEV4	TRG0EXEV3	TRG0EXEV2	TRG0EXEV1	TRG0EXEV0	TRG0MTPER	TRG0MTC3	TRG0MTC2	TRG0MTC1	TRG0MTC0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31	TRG0ST4PER	HRTIMER_ADCTRIG0 on Slave_TIMER4 period event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG0. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER4 period event. 1: ADC trigger event generated on HRTIMER Slave_TIMER4 period event.
30	TRG0ST4C3	HRTIMER_ADCTRIG0 on Slave_TIMER4 compare 3 event Refer to TRG0ST4C1 description.
29	TRG0ST4C2	HRTIMER_ADCTRIG0 on Slave_TIMER4 compare 2 event Refer to TRG0ST4C1 description.
28	TRG0ST4C1	HRTIMER_ADCTRIG0 on Slave_TIMER4 compare 1 event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG0. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER4 compare 1 event. 1: ADC trigger event generated on HRTIMER Slave_TIMER4 compare 1 event
27	TRG0ST3PER	HRTIMER_ADCTRIG0 on Slave_TIMER3 period event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG0. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER3 period event. 1: ADC trigger event generated on HRTIMER Slave_TIMER3 period event.
26	TRG0ST3C3	HRTIMER_ADCTRIG0 on Slave_TIMER3 compare 3 event Refer to TRG0ST3C1 description.
25	TRG0ST3C2	HRTIMER_ADCTRIG0 on Slave_TIMER3 compare 2 event Refer to TRG0ST3C1 description.
24	TRG0ST3C1	HRTIMER_ADCTRIG0 on Slave_TIMER3 compare 1 event

		The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG0. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER3 compare 1 event. 1: ADC trigger event generated on HRTIMER Slave_TIMER3 compare 1 event
23	TRG0ST2PER	HRTIMER_ADCTRIG0 on Slave_TIMER2 period event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG0. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER2 period event. 1: ADC trigger event generated on HRTIMER Slave_TIMER2 period event.
22	TRG0ST2C3	HRTIMER_ADCTRIG0 on Slave_TIMER2 compare 3 event Refer to TRG0ST2C1 description.
21	TRG0ST2C2	HRTIMER_ADCTRIG0 on Slave_TIMER2 compare 2 event Refer to TRG0ST2C1 description.
20	TRG0ST2C1	HRTIMER_ADCTRIG0 on Slave_TIMER2 compare 1 event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG0. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER2 compare 1 event. 1: ADC trigger event generated on HRTIMER Slave_TIMER2 compare 1 event
19	TRG0ST1RST	HRTIMER_ADCTRIG0 on Slave_TIMER1 reset The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG0. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER1 reset. 1: ADC trigger event generated on HRTIMER Slave_TIMER1 reset.
18	TRG0ST1PER	HRTIMER_ADCTRIG0 on Slave_TIMER1 period event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG0. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER1 period event. 1: ADC trigger event generated on HRTIMER Slave_TIMER1 period event.
17	TRG0ST1C3	HRTIMER_ADCTRIG0 on Slave_TIMER1 compare 3 event Refer to TRG0ST1C1 description.
16	TRG0ST1C2	HRTIMER_ADCTRIG0 on Slave_TIMER1 compare 2 event Refer to TRG0ST1C1 description.
15	TRG0ST1C1	HRTIMER_ADCTRIG0 on Slave_TIMER1 compare 1 event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG0. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER1 compare 1 event.

		1: ADC trigger event generated on HRTIMER Slave_TIMER1 compare 1 event
14	TRG0ST0RST	HRTIMER_ADCTRIG0 on Slave_TIMER0 reset The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG0. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER0 reset. 1: ADC trigger event generated on HRTIMER Slave_TIMER0 reset.
13	TRG0ST0PER	HRTIMER_ADCTRIG0 on Slave_TIMER0 period event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG0. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER0 period event. 1: ADC trigger event generated on HRTIMER Slave_TIMER0 period event.
12	TRG0ST0C3	HRTIMER_ADCTRIG0 on Slave_TIMER0 compare 3 event Refer to TRG0ST0C1 description.
11	TRG0ST0C2	HRTIMER_ADCTRIG0 on Slave_TIMER0 compare 2 event Refer to TRG0ST0C1 description.
10	TRG0ST0C1	HRTIMER_ADCTRIG0 on Slave_TIMER0 compare 1 event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG0. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER0 compare 1 event. 1: ADC trigger event generated on HRTIMER Slave_TIMER0 compare 1 event.
9	TRG0EXEV4	HRTIMER_ADCTRIG0 on external event 4 Refer to TRG0EXEV0 description.
8	TRG0EXEV3	HRTIMER_ADCTRIG0 on external event 3 Refer to TRG0EXEV0 description.
7	TRG0EXEV2	HRTIMER_ADCTRIG0 on external event 2 Refer to TRG0EXEV0 description.
6	TRG0EXEV1	HRTIMER_ADCTRIG0 on external event 1 Refer to TRG0EXEV0 description.
5	TRG0EXEV0	HRTIMER_ADCTRIG0 on external event 0 The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG0. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER external event 0(EXEV0C). 1: ADC trigger event generated on HRTIMER external event 0(EXEV0C).
4	TRG0MTPER	HRTIMER_ADCTRIG0 on Master_TIMER period event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG0. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Master_TIMER period event.

		1: ADC trigger event generated on HRTIMER Master_TIMER period event.
3	TRG0MTC3	HRTIMER_ADCTRIG0 on Master_TIMER compare 3 event Refer to TRG0MTC0 description.
2	TRG0MTC2	HRTIMER_ADCTRIG0 on Master_TIMER compare 2 event Refer to TRG0MTC0 description.
1	TRG0MTC1	HRTIMER_ADCTRIG0 on Master_TIMER compare 1 event Refer to TRG0MTC0 description.
0	TRG0MTC0	HRTIMER_ADCTRIG0 on Master_TIMER compare 0 event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG0. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Master_TIMER compare 0 event. 1: ADC trigger event generated on HRTIMER Master_TIMER compare 0 event.

HRTIMER trigger source 1 to ADC register (HRTIMER_ADCTRIGS1)

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
TRG1ST4RST	TRG1ST4C3	TRG1ST4C2	TRG1ST4C1	TRG1ST3RST	TRG1ST3PER	TRG1ST3C3	TRG1ST3C2	TRG1ST3C1	TRG1ST2RST	TRG1ST2PER	TRG1ST2C3	TRG1ST2C2	TRG1ST2C1	TRG1ST1PER	TRG1ST1C3
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
TRG1ST1C2	TRG1ST1C1	TRG1ST0PER	TRG1ST0C3	TRG1ST0C2	TRG1ST0C1	TRG1EXEV9	TRG1EXEV8	TRG1EXEV7	TRG1EXEV6	TRG1EXEV5	TRG1MTPER	TRG1MTC3	TRG1MTC2	TRG1MTC1	TRG1MTC0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31	TRG1ST4RST	HRTIMER_ADCTRIG1 on Slave_TIMER4 reset The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG1. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER4 reset . 1: ADC trigger event generated on HRTIMER Slave_TIMER4 reset.
30	TRG1ST4C3	HRTIMER_ADCTRIG1 on Slave_TIMER4 compare 3 event Refer to TRG1ST4C1 description.
29	TRG1ST4C2	HRTIMER_ADCTRIG1 on Slave_TIMER4 compare 2 event Refer to TRG1ST4C1 description.
28	TRG1ST4C1	HRTIMER_ADCTRIG1 on Slave_TIMER4 compare 1 event

		The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG1. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER4 compare 1 event. 1: ADC trigger event generated on HRTIMER Slave_TIMER4 compare 1 event.
27	TRG1ST3RST	HRTIMER_ADCTRIG1 on Slave_TIMER3 reset The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG1. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER3 reset. 1: ADC trigger event generated on HRTIMER Slave_TIMER3 reset .
26	TRG1ST3PER	HRTIMER_ADCTRIG1 on Slave_TIMER3 period event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG1. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER3 period event. 1: ADC trigger event generated on HRTIMER Slave_TIMER3 period event.
25	TRG1ST3C3	HRTIMER_ADCTRIG1 on Slave_TIMER3 compare 3 event Refer to TRG1ST3C1 description.
24	TRG1ST3C2	HRTIMER_ADCTRIG1 on Slave_TIMER3 compare2 event Refer to TRG1ST3C1 description.
23	TRG1ST3C1	HRTIMER_ADCTRIG1 on Slave_TIMER3 compare 1 event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG1. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER3 compare 1 event. 1: ADC trigger event generated on HRTIMER Slave_TIMER3 compare 1 event.
22	TRG1ST2RST	HRTIMER_ADCTRIG1 on Slave_TIMER2 reset The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG1. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER2 reset . 1: ADC trigger event generated on HRTIMER Slave_TIMER2 reset.
21	TRG1ST2PER	HRTIMER_ADCTRIG1 on Slave_TIMER2 period event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG1. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER2 period event. 1: ADC trigger event generated on HRTIMER Slave_TIMER2 period event.
20	TRG1ST2C3	HRTIMER_ADCTRIG1 on Slave_TIMER2 compare 3 event Refer to TRG1ST2C1 description.
19	TRG1ST2C2	HRTIMER_ADCTRIG1 on Slave_TIMER2 compare2 event

		Refer to TRG1ST2C1 description.
18	TRG1ST2C1	HRTIMER_ADCTRIG1 on Slave_TIMER2 compare 1 event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG1. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER2 compare 1 event. 1: ADC trigger event generated on HRTIMER Slave_TIMER2 compare 1 event.
17	TRG1ST1PER	HRTIMER_ADCTRIG1 on Slave_TIMER1 period event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG1. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER1 period event. 1: ADC trigger event generated on HRTIMER Slave_TIMER1 period event.
16	TRG1ST1C3	HRTIMER_ADCTRIG1 on Slave_TIMER1 compare 3 event Refer to TRG1ST1C1 description.
15	TRG1ST1C2	HRTIMER_ADCTRIG1 on Slave_TIMER1 compare2 event Refer to TRG1ST1C1 description.
14	TRG1ST1C1	HRTIMER_ADCTRIG1 on Slave_TIMER1 compare 1 event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG1. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER1 compare 1 event. 1: ADC trigger event generated on HRTIMER Slave_TIMER1 compare 1 event.
13	TRG1ST0PER	HRTIMER_ADCTRIG1 on Slave_TIMER0 period event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG1. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER0 period event. 1: ADC trigger event generated on HRTIMER Slave_TIMER0 period event.
12	TRG1ST0C3	HRTIMER_ADCTRIG1 on Slave_TIMER0 compare 3 event Refer to TRG1ST0C1 description.
11	TRG1ST0C2	HRTIMER_ADCTRIG1 on Slave_TIMER0 compare 2 event Refer to TRG1ST0C1 description.
10	TRG1ST0C1	HRTIMER_ADCTRIG1 on Slave_TIMER0 compare 1 event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG1. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER0 compare 1 event. 1: ADC trigger event generated on HRTIMER Slave_TIMER0 compare 1 event.
9	TRG1EXEV9	HRTIMER_ADCTRIG1 on external event 9

Refer to TRG1EXEV5 description.

8 TRG1EXEV8 HRTIMER_ADCTRIG1 on external event 8
Refer to TRG1EXEV5 description.

7 TRG1EXEV7 HRTIMER_ADCTRIG1 on external event 7
Refer to TRG1EXEV5 description.

6 TRG1EXEV6 HRTIMER_ADCTRIG1 on external event 6
Refer to TRG1EXEV5 description.

5 TRG1EXEV5 HRTIMER_ADCTRIG1 on external event 5
The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG1.
This bit specifies whether the event can generate the ADC trigger event.
0: No ADC trigger event generated on HRTIMER external event 5 (EXEV5C).
1: ADC trigger event generated on HRTIMER external event 5 (EXEV5C).

4 TRG1MTPER HRTIMER_ADCTRIG1 on Master_TIMER period event
The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG1.
This bit specifies whether the event can generate the ADC trigger event.
0: No ADC trigger event generated on HRTIMER Master_TIMER period event.
1: ADC trigger event generated on HRTIMER Master_TIMER period event.

3 TRG1MTC3 HRTIMER_ADCTRIG1 on Master_TIMER compare 3 event
Refer to TRG1MTC0 description.

2 TRG1MTC2 HRTIMER_ADCTRIG1 on Master_TIMER compare 2 event
Refer to TRG1MTC0 description.

1 TRG1MTC1 HRTIMER_ADCTRIG1 on Master_TIMER compare 1 event
Refer to TRG1MTC0 description.

0 TRG1MTC0 HRTIMER_ADCTRIG1 on Master_TIMER compare 0 event
The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG1.
This bit specifies whether the event can generate the ADC trigger event.
0: No ADC trigger event generated on HRTIMER Master_TIMER compare 1 event.
1: ADC trigger event generated on HRTIMER Master_TIMER compare 1 event.

HRTIMER trigger source 2 to ADC register (HRTIMER_ADCTRIGS2)

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
TRG2ST4 PER	TRG2ST4 C3	TRG2ST4 C2	TRG2ST4 C1	TRG2ST3 PER	TRG2ST3 C3	TRG2ST3 C2	TRG2ST3 C1	TRG2ST2 PER	TRG2ST2 C3	TRG2ST2 C2	TRG2ST2 C1	TRG2ST1 RST	TRG2ST1 PER	TRG2ST1 C3	TRG2ST1 C2

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
TRG2ST1C1	TRG2ST0RST	TRG2ST0PER	TRG2ST0C3	TRG2ST0C2	TRG2ST0C1	TRG2EXEV4	TRG2EXEV3	TRG2EXEV2	TRG2EXEV1	TRG2EXEV0	TRG2MTPER	TRG2MTC3	TRG2MTC2	TRG2MTC1	TRG2MTC0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31	TRG2ST4PER	HRTIMER_ADCTRIG2 on Slave_TIMER4 period event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG2. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER4 period event. 1: ADC trigger event generated on HRTIMER Slave_TIMER4 period event.
30	TRG2ST4C3	HRTIMER_ADCTRIG2 on Slave_TIMER4 compare 3 event Refer to TRG2ST4C1 description.
29	TRG2ST4C2	HRTIMER_ADCTRIG2 on Slave_TIMER4 compare 2 event Refer to TRG2ST4C1 description.
28	TRG2ST4C1	HRTIMER_ADCTRIG2 on Slave_TIMER4 compare 1 event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG2. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER4 compare 1 event. 1: ADC trigger event generated on HRTIMER Slave_TIMER4 compare 1 event.
27	TRG2ST3PER	HRTIMER_ADCTRIG2 on Slave_TIMER3 period event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG2. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER3 period event. 1: ADC trigger event generated on HRTIMER Slave_TIMER3 period event.
26	TRG2ST3C3	HRTIMER_ADCTRIG2 on Slave_TIMER3 compare 3 event Refer to TRG2ST3C1 description.
25	TRG2ST3C2	HRTIMER_ADCTRIG2 on Slave_TIMER3 compare 2 event Refer to TRG2ST3C1 description.
24	TRG2ST3C1	HRTIMER_ADCTRIG2 on Slave_TIMER3 compare 1 event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG2. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER3 compare 1 event. 1: ADC trigger event generated on HRTIMER Slave_TIMER3 compare 1 event.
23	TRG2ST2PER	HRTIMER_ADCTRIG2 on Slave_TIMER2 period event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG2.

		This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER2 period event. 1: ADC trigger event generated on HRTIMER Slave_TIMER2 period event.
22	TRG2ST2C3	HRTIMER_ADCTRIG2 on Slave_TIMER2 compare 3 event Refer to TRG2ST2C1 description.
21	TRG2ST2C2	HRTIMER_ADCTRIG2 on Slave_TIMER2 compare 2 event Refer to TRG2ST2C1 description.
20	TRG2ST2C1	HRTIMER_ADCTRIG2 on Slave_TIMER2 compare 1 event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG2. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER2 compare 1 event. 1: ADC trigger event generated on HRTIMER Slave_TIMER2 compare 1 event.
19	TRG2ST1RST	HRTIMER_ADCTRIG2 on Slave_TIMER1 reset The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG2. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER1 reset . 1: ADC trigger event generated on HRTIMER Slave_TIMER1 reset.
18	TRG2ST1PER	HRTIMER_ADCTRIG2 on Slave_TIMER1 period event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG2. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER1 period event. 1: ADC trigger event generated on HRTIMER Slave_TIMER1 period event.
17	TRG2ST1C3	HRTIMER_ADCTRIG2 on Slave_TIMER1 compare 3 event Refer to TRG2ST1C1 description.
16	TRG2ST1C2	HRTIMER_ADCTRIG2 on Slave_TIMER1 compare 2 event Refer to TRG2ST1C1 description.
15	TRG2ST1C1	HRTIMER_ADCTRIG2 on Slave_TIMER1 compare 1 event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG2. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER1 compare 1 event. 1: ADC trigger event generated on HRTIMER Slave_TIMER1 compare 1 event.
14	TRG2ST0RST	HRTIMER_ADCTRIG2 on Slave_TIMER0 reset The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG2. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER0 reset . 1: ADC trigger event generated on HRTIMER Slave_TIMER0 reset.

13	TRG2ST0PER	HRTIMER_ADCTRIG2 on Slave_TIMER0 period event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG2. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER0 period event. 1: ADC trigger event generated on HRTIMER Slave_TIMER0 period event.
12	TRG2ST0C3	HRTIMER_ADCTRIG2 on Slave_TIMER0 compare 3 event Refer to TRG2ST0C1 description.
11	TRG2ST0C2	HRTIMER_ADCTRIG2 on Slave_TIMER0 compare 2 event Refer to TRG2ST0C1 description.
10	TRG2ST0C1	HRTIMER_ADCTRIG2 on Slave_TIMER0 compare 1 event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG2. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER0 compare 1 event. 1: ADC trigger event generated on HRTIMER Slave_TIMER0 compare 1 event.
9	TRG2EXEV4	HRTIMER_ADCTRIG2 on external event 4 Refer to TRG2EXEV0 description.
8	TRG2EXEV3	HRTIMER_ADCTRIG2 on external event 3 Refer to TRG2EXEV0 description.
7	TRG2EXEV2	HRTIMER_ADCTRIG2 on external event 2 Refer to TRG2EXEV0 description.
6	TRG2EXEV1	HRTIMER_ADCTRIG2 on external event 1 Refer to TRG2EXEV0 description.
5	TRG2EXEV0	HRTIMER_ADCTRIG2 on external event 0 The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG2. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER external event 0 (EXEV0C). 1: ADC trigger event generated on HRTIMER external event 0 (EXEV0C).
4	TRG2MTPER	HRTIMER_ADCTRIG2 on Master_TIMER period event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG2. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Master_TIMER period event. 1: ADC trigger event generated on HRTIMER Master_TIMER period event.
3	TRG2MTC3	HRTIMER_ADCTRIG2 on Master_TIMER compare 3 event Refer to TRG2MTC0 description.
2	TRG2MTC2	HRTIMER_ADCTRIG2 on Master_TIMER compare 2 event Refer to TRG2MTC0 description.

1	TRG2MTC1	HRTIMER_ADCTRIG2 on Master_TIMER compare 1 event Refer to TRG2MTC0 description.
0	TRG2MTC0	HRTIMER_ADCTRIG2 on Master_TIMER compare 0 event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG2. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Master_TIMER compare 0 event. 1: ADC trigger event generated on HRTIMER Master_TIMER compare 0 event.

HRTIMER trigger source 3 to ADC register (HRTIMER_ADCTRIGS3)

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
TRG3ST4RST	TRG3ST4C3	TRG3ST4C2	TRG3ST4C1	TRG3ST3RST	TRG3ST3PER	TRG3ST3C3	TRG3ST3C2	TRG3ST3C1	TRG3ST2RST	TRG3ST2PER	TRG3ST2C3	TRG3ST2C2	TRG3ST2C1	TRG3ST1PER	TRG3ST1C3
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
TRG3ST1C2	TRG3ST1C1	TRG3ST0PER	TRG3ST0C3	TRG3ST0C2	TRG3ST0C1	TRG3EXEV9	TRG3EXEV8	TRG3EXEV7	TRG3EXEV6	TRG3EXEV5	TRG3MTPER	TRG3MTC3	TRG3MTC2	TRG3MTC1	TRG3MTC0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31	TRG3ST4RST	HRTIMER_ADCTRG3 on Slave_TIMER4 reset The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRG3. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER4 reset . 1: ADC trigger event generated on HRTIMER Slave_TIMER4 reset.
30	TRG3ST4C3	HRTIMER_ADCTRG3 on Slave_TIMER4 compare 3 event Refer to TRG3ST4C1 description.
29	TRG3ST4C2	HRTIMER_ADCTRG3 on Slave_TIMER4 compare 2 event Refer to TRG3ST4C1 description.
28	TRG3ST4C1	HRTIMER_ADCTRG3 on Slave_TIMER4 compare 1 event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRG3. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER4 compare 1 event. 1: ADC trigger event generated on HRTIMER Slave_TIMER4 compare 1 event.
27	TRG3ST3RST	HRTIMER_ADCTRG3 on Slave_TIMER3 reset

		The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRG3. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER3 reset . 1: ADC trigger event generated on HRTIMER Slave_TIMER3 reset.
26	TRG3ST3PER	HRTIMER_ADCTRG3 on Slave_TIMER3 period event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRG3. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER3 period event. 1: ADC trigger event generated on HRTIMER Slave_TIMER3 period event.
25	TRG3ST3C3	HRTIMER_ADCTRG3 on Slave_TIMER3 compare 3 event Refer to TRG3ST3C1 description.
24	TRG3ST3C2	HRTIMER_ADCTRG3 on Slave_TIMER3 compare 2 event Refer to TRG3ST3C1 description..
23	TRG3ST3C1	HRTIMER_ADCTRG3 on Slave_TIMER3 compare 1 event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRG3. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER3 compare 1 event. 1: ADC trigger event generated on HRTIMER Slave_TIMER3 compare 1 event.
22	TRG3ST2RST	HRTIMER_ADCTRG3 on Slave_TIMER2 reset The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRG3. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER2 reset . 1: ADC trigger event generated on HRTIMER Slave_TIMER2 reset.
21	TRG3ST2PER	HRTIMER_ADCTRG3 on Slave_TIMER2 period event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRG3. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER2 period event. 1: ADC trigger event generated on HRTIMER Slave_TIMER2 period event.
20	TRG3ST2C3	HRTIMER_ADCTRG3 on Slave_TIMER2 compare 3 event Refer to TRG3ST2C1 description.
19	TRG3ST2C2	HRTIMER_ADCTRG3 on Slave_TIMER2 compare 2 event Refer to TRG3ST2C1 description.
18	TRG3ST2C1	HRTIMER_ADCTRG3 on Slave_TIMER2 compare 1 event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRG3. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER2 compare 1 event.

		1: ADC trigger event generated on HRTIMER Slave_TIMER2 compare 1 event.
17	TRG3ST1PER	HRTIMER_ADCTRG3 on Slave_TIMER1 period event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRG3. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER1 period event. 1: ADC trigger event generated on HRTIMER Slave_TIMER1 period event.
16	TRG3ST1C3	HRTIMER_ADCTRG3 on Slave_TIMER1 compare 3 event Refer to TRG3ST1C1 description.
15	TRG3ST1C2	HRTIMER_ADCTRG3 on Slave_TIMER1 compare 2 event Refer to TRG3ST1C1 description.
14	TRG3ST1C1	HRTIMER_ADCTRG3 on Slave_TIMER1 compare 1 event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRG3. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER1 compare 1 event. 1: ADC trigger event generated on HRTIMER Slave_TIMER1 compare 1 event.
13	TRG3ST0PER	HRTIMER_ADCTRG3 on Slave_TIMER0 period event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRG3. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER0 period event. 1: ADC trigger event generated on HRTIMER Slave_TIMER0 period event.
12	TRG3ST0C3	HRTIMER_ADCTRG3 on Slave_TIMER0 compare 3 event Refer to TRG3ST0C1 description.
11	TRG3ST0C2	HRTIMER_ADCTRG3 on Slave_TIMER0 compare 2 event Refer to TRG3ST0C1 description.
10	TRG3ST0C1	HRTIMER_ADCTRG3 on Slave_TIMER0 compare 1 event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRG3. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER0 compare 1 event. 1: ADC trigger event generated on HRTIMER Slave_TIMER0 compare 1 event.
9	TRG3EXEV9	HRTIMER_ADCTRG3 on external event 9 Refer to TRG3EXEV5 description.
8	TRG3EXEV8	HRTIMER_ADCTRG3 on external event 8 Refer to TRG3EXEV5 description.
7	TRG3EXEV7	HRTIMER_ADCTRG3 on external event 7 Refer to TRG3EXEV5 description.

6	TRG3EXEV6	HRTIMER_ADCTRG3 on external event 6 Refer to TRG3EXEV5 description.
5	TRG3EXEV5	HRTIMER_ADCTRG3 on external event 5 The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRG3. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER external event 5 (EXEV0C). 1: ADC trigger event generated on HRTIMER external event 5 (EXEV0C).
4	TRG3MTPER	HRTIMER_ADCTRG3 on Master_TIMER period event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRG3. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Master_TIMER period event. 1: ADC trigger event generated on HRTIMER Master_TIMER period event.
3	TRG3MTC3	HRTIMER_ADCTRG3 on Master_TIMER compare 3 event Refer to TRG3MTC0 description.
2	TRG3MTC2	HRTIMER_ADCTRG3 on Master_TIMER compare 2 event Refer to TRG3MTC0 description.
1	TRG3MTC1	HRTIMER_ADCTRG3 on Master_TIMER compare 1 event Refer to TRG3MTC0 description.
0	TRG3MTC0	HRTIMER_ADCTRG3 on Master_TIMER compare 0 event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG3. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Master_TIMER compare 1 event. 1: ADC trigger event generated on HRTIMER Master_TIMER compare 1 event.

HRTIMER DLL calibration control register (HRTIMER_DLLCCTL)

Address offset: 0x4C

Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions

31:4	Reserved	Must be kept at reset value
3:2	CLBPER[1:0]	DLL calibration period This bit-field defines the length of the DLL calibration cycle. 00: 1048576 * $t_{HRTIMER_CK}$ 01: 131072 * $t_{HRTIMER_CK}$ 10: 16384 * $t_{HRTIMER_CK}$ 11: 2048 * $t_{HRTIMER_CK}$
1	CLBPEREN	DLL periodic calibration enable This bit enables the periodic DLL calibration. The bit-field CLBPER[1:0] sets the calibration period. 0: DLL periodic calibration disable. 1: DLL periodic calibration enable. **Note:** CLBPEREN bit and CLBSTRT bit must not be set simultaneously.
0	CLBSTRT	DLL calibration start once Writing 1 to the bit starts the DLL calibration when CLBPEREN = 0. This bit is write-only. 0: No effect. 1: DLL calibration start once. **Note:** CLBPEREN bit and CLBSTRT bit must not be set simultaneously.

HRTIMER fault input configuration register 0 (HRTIMER_FLTINCFG0)

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
FLT3INPROT		FLT3INFC[3:0]			FLT3INSRC[0]	FLT3INP	FLT3INEN	FLT2INPROT	FLT2INFC[3:0]				FLT2INSRC[0]	FLT2INP	FLT2INEN
rwo		rw			rw	rw	rw	rwo	rw				rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLT1INPROT		FLT1INFC[3:0]			FLT1INSRC[0]	FLT1INP	FLT1INEN	FLT0INPROT	FLT0INFC[3:0]				FLT0INSRC[0]	FLT0INP	FLT0INEN
rwo		rw			rw	rw	rw	rwo	rw				rw	rw	rw

Bits	Fields	Descriptions
31	FLT3INPROT	Protect fault 3 input configuration Refer to FLT0INPROT description.
30:27	FLT3INFC[3:0]	Fault 3 input filter control Refer to FLT0INFC[3:0] description.
26	FLT3INSRC[0]	Fault 3 input source

		Refer to FLT0INSRC description.
25	FLT3INP	Fault 3 input polarity Refer to FLT0INP description.
24	FLT3INEN	Fault 3 input enable Refer to FLT0INEN description.
23	FLT2INPROT	Protect fault 2 input configuration Refer to FLT0INPROT description.
22:19	FLT2INFC[3:0]	Fault 2 input filter control Refer to FLT0INFC[3:0] description.
18	FLT2INSRC[0]	Fault 2 input source Refer to FLT0INSRC description.
17	FLT2INP	Fault 2 input polarity Refer to FLT0INP description.
16	FLT2INEN	Fault 2 input enable Refer to FLT0INEN description.
15	FLT1INPROT	Protect fault 1 input configuration Refer to FLT0INPROT description.
14:11	FLT1INFC[3:0]	Fault 1 input filter control Refer to FLT0INFC[3:0] description.
10	FLT1INSRC[0]	Fault 1 input source Refer to FLT0INSRC description.
9	FLT1INP	Fault 1 input polarity Refer to FLT0INP description.
8	FLT1INEN	Fault 1 input enable Refer to FLT0INEN description.
7	FLT0INPROT	Protect fault 0 input configuration This bit-field specifies the write protection property to protect fault 0 input configuration. This bit is write-once. It can only be cleared by a system reset once It is set by software. 0: Protect disable. FLT0INEN, FLT0INP, FLT0INSRC and FLT0INFC[3:0] is writable. 1: Protect enable. FLT0INEN, FLT0INP, FLT0INSRC and FLT0INFC[3:0] is read-only.
6:3	FLT0INFC[3:0]	Fault 0 input 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(f_{SAMP}) used to sample fault

		input and the length of the digital filter applied to fault input. 0000: Filter disalble. 0001: $f_{SAMP} = f_{HRTIMER_CK}$, N=2. 0010: $f_{SAMP} = f_{HRTIMER_CK}$, N=4. 0011: $f_{SAMP} = f_{HRTIMER_CK}$, N=8. 0100: $f_{SAMP} = f_{HRTIMER_FLTFCK}$ /2, N=6. 0101: $f_{SAMP} = f_{HRTIMER_FLTFCK}$ /2, N=8. 0110: $f_{SAMP} = f_{HRTIMER_FLTFCK}$ /4, N=6. 0111: $f_{SAMP} = f_{HRTIMER_FLTFCK}$ /4, N=8. 1000: $f_{SAMP} = f_{HRTIMER_FLTFCK}$ /8, N=6. 1001: $f_{SAMP} = f_{HRTIMER_FLTFCK}$ /8, N=8. 1010: $f_{SAMP} = f_{HRTIMER_FLTFCK}$ /16, N=5. 1011: $f_{SAMP} = f_{HRTIMER_FLTFCK}$ /16, N=6. 1100: $f_{SAMP} = f_{HRTIMER_FLTFCK}$ /16, N=8. 1101: $f_{SAMP} = f_{HRTIMER_FLTFCK}$ /32, N=5. 1110: $f_{SAMP} = f_{HRTIMER_FLTFCK}$ /32, N=6. 1111: $f_{SAMP} = f_{HRTIMER_FLTFCK}$ /32, N=8. **Note:** (1) This bit-field can be written only when FLT0INEN bit is reset. (2) This bit-field cannot be modified when FLT0INPROT has been programmed.
2	FLT0INSRC[0]	Fault 0 input source, combine with FLT0INSRC[1]. 00: The source of fault 0 input is chip external pin. 01: The source of fault 0 input is chip internal signal(for example comparator) 10: The source of fault 0 input is external event 0. Others: Reserved. **Note:** This bit can be written only when FLT0INEN bit is reset.
1	FLT0INP	Fault 0 input polarity This bit specifies the polarity of fault 0 input. 0: Fault 0 input active at low level. 1: Fault 0 input active at high level. **Note:** This bit can be written only when FLT0INEN bit is reset.
0	FLT0INEN	Fault 0 input enable This bit can be set to enable fault 0 input circuitry. 0: Fault 0 input disable 1: Fault 0 input enable

HRTIMER fault input configuration register 1 (HRTIMER_FLTINCFG1)

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	FLTFDIV[2:0]	FLT7INSRC[1]	FLT6INSRC[1]	FLT5INSRC[1]	FLT4INSRC[1]	FLT3INSRC[1]	FLT2INSRC[1]	FLT1INSRC[1]	FLT0INSRC[1]
	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
FLT5INPROT	FLT5INFC[3:0]				FLT5INSRC[0]	FLT5INP	FLT5INEN	FLT4INPROT	FLT4INFC[3:0]				FLT4INSRC[0]	FLT4INP	FLT4INEN
rw	rw				rw	rw	rw	rwo	rw				rw	rw	rw

Bits	Fields	Descriptions
31:26	Reserved	Must be kept at reset value
25:24	FLTFDIV[2:0]	Fault input digital filter clock division This bit-field can be configured by software to specify division ratio between the HRTIMER clock (HRTIMER_CK) and the fault input digital filter clock (HRTIMER_FLTFCK). $f_{HRTIMER_FLTFCK} = f_{HRTIMER_CK}/2^{FLTFDIV[2:0]}$. 00: $f_{HRTIMER_FLTFCK} = f_{HRTIMER_CK}$ 01: $f_{HRTIMER_FLTFCK} = f_{HRTIMER_CK}/2$ 10: $f_{HRTIMER_FLTFCK} = f_{HRTIMER_CK}/4$ 11: $f_{HRTIMER_FLTFCK} = f_{HRTIMER_CK}/8$ **Note:** This bit must be configured before setting any FLTyINEN(y=0..4).
23	FLT7INSRC[1]	Fault 7 input source Refer to FLT0INSRC[0] description.
22	FLT6INSRC[1]	Fault 6 input source Refer to FLT0INSRC[0] description.
21	FLT5INSRC[1]	Fault 5 input source Refer to FLT0INSRC[0] description.
20	FLT4INSRC[1]	Fault 4 input source Refer to FLT0INSRC[0] description.
19	FLT3INSRC[1]	Fault 3 input source Refer to FLT0INSRC[0] description.
18	FLT2INSRC[1]	Fault 2 input source Refer to FLT0INSRC[0] description.
17	FLT1INSRC[1]	Fault 1 input source Refer to FLT0INSRC[0] description.
16	FLT0INSRC[1]	Fault 0 input source Refer to FLT0INSRC[0] description.
15	FLT5INPROT	Protect fault 5 input configuration Refer to FLT0INPROT in HRTIMER_FLTINCFG0 register description.

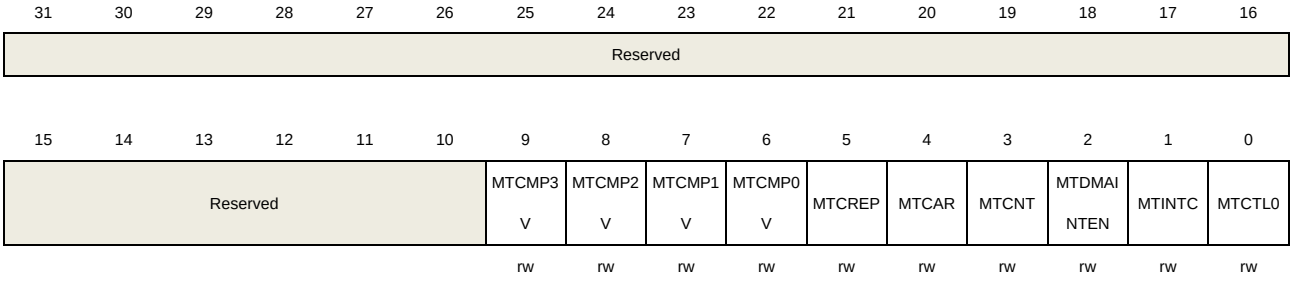
14:11	FLT5INFC[3:0]	Fault 5 input filter control Refer to FLT0INFC[3:0] in HRTIMER_FLTINCFG0 register description.
10	FLT5INSRC[0]	Fault 5 input source Refer to FLT0INSRC in HRTIMER_FLTINCFG0 register description.
9	FLT5INP	Fault 5 input polarity Refer to FLT0INP in HRTIMER_FLTINCFG0 register description.
8	FLT5INEN	Fault 5 input enable Refer to FLT0INEN in HRTIMER_FLTINCFG0 register description.
7	FLT4INPROT	Protect fault 4 input configuration Refer to FLT0INPROT in HRTIMER_FLTINCFG0 register description.
6:3	FLT4INFC[3:0]	Fault 4 input filter control Refer to FLT0INFC[3:0] in HRTIMER_FLTINCFG0 register description.
2	FLT4INSRC[0]	Fault 4 input source Refer to FLT0INSRC in HRTIMER_FLTINCFG0 register description.
1	FLT4INP	Fault 4 input polarity Refer to FLT0INP in HRTIMER_FLTINCFG0 register description.
0	FLT4INEN	Fault 4 input enable Refer to FLT0INEN in HRTIMER_FLTINCFG0 register description.

HRTIMER DMA update Master_TIMER register (HRTIMER_DMAUPMTR)

Address offset: 0x58

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
30:10	Reserved	Must be kept at reset value
9	MTCMP3V	HRTIMER_MTCMP3V update by DMA mode Refer to MTCTL0 description.

8	MTCMP2V	HRTIMER_MTCMP2V update by DMA mode Refer to MTCTL0 description.
7	MTCMP1V	HRTIMER_MTCMP1V update by DMA mode Refer to MTCTL0 description.
6	MTCMP0V	HRTIMER_MTCMP0V update by DMA mode Refer to MTCTL0 description.
5	MTCREP	HRTIMER_MTCREP update by DMA mode Refer to MTCTL0 description.
4	MTCAR	HRTIMER_MTCAR update by DMA mode Refer to MTCTL0 description.
3	MTCNT	HRTIMER_MTCNT update by DMA mode Refer to MTCTL0 description.
2	MTDMAINTEN	HRTIMER_MTDMAINTEN update by DMA mode Refer to MTCTL0 description.
1	MTINTC	HRTIMER_MTINTC update by DMA mode Refer to MTCTL0 description.
0	MTCTL0	HRTIMER_MTCTL0 update by DMA mode This bit defines if the HRTIMER_MTCTL0 register is updated by the DMA mode. 0: HRTIMER_MTCTL0 register is not updated by DMA mode. 1: HRTIMER_MTCTL0 register is updated DMA mode.

HRTIMER DMA update Slave_TIMERx register (HRTIMER_DMAUPSTxR)(x=0..7)

Address offset: Slave_TIMER0: 0x5C
Address offset: Slave_TIMER1: 0x60
Address offset: Slave_TIMER2: 0x64
Address offset: Slave_TIMER3: 0x68
Address offset: Slave_TIMER4: 0x6C
Address offset: Slave_TIMER5: 0x74
Address offset: Slave_TIMER6: 0x100
Address offset: Slave_TIMER7: 0x104
Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
STxACTL	Reserved								STxEXEVFCFG2	STxCTL1	STxFLTCTL	STxCHOCTL	STxCSCTL	STxCNTRST	STxEXEVFCFG1
rw									rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

STxEXEVFCFG0	STxCH1RST	STxCH1SET	STxCH0RST	STxCH0SET	STxDTCTL	STxCMP3V	STxCMP2V	STxCMP1V	STxCMP0V	STxCREP	STxCAR	STxCNT	STxDMAINTEN	STxINTC	STxCTL0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31	STxACTL	HRTIMER_STxACTL update by DMA mode Refer to STxCTL0 bit description.
30:23	Reserved	Must be kept at reset value
22	STxEXEVFCFG2	HRTIMER_STxEXEVFCFG2 update by DMA mode Refer to STxCTL0 bit description.
21	STxCTL1	HRTIMER_STxCTL1 update by DMA mode Refer to STxCTL0 bit description.
20	STxFLTCTL	HRTIMER_STxFLTCTL update by DMA mode Refer to STxCTL0 bit description.
19	STxCHOCTL	HRTIMER_STxCHOCTL update by DMA mode Refer to STxCTL0 bit description.
18	STxCSCTL	HRTIMER_STxCSCTL update by DMA mode Refer to STxCTL0 bit description.
17	STxCNTRST	HRTIMER_STxCNTRST update by DMA mode Refer to STxCTL0 bit description.
16	STxEXEVFCFG1	HRTIMER_STxEXEVFCFG1update by DMA mode Refer to STxCTL0 bit description.
15	STxEXEVFCFG0	HRTIMER_STxEXEVFCFG0update by DMA mode Refer to STxCTL0 bit description.
14	STxCH1RST	HRTIMER_STxCH1RST update by DMA mode Refer to STxCTL0 bit description.
13	STxCH1SET	HRTIMER_STxCH1SET update by DMA mode Refer to STxCTL0 bit description.
12	STxCH0RST	HRTIMER_STxCH0RST update by DMA mode Refer to STxCTL0 bit description.
11	STxCH0SET	HRTIMER_STxCH0SET update by DMA mode Refer to STxCTL0 bit description.
10	STxDTCTL	HRTIMER_STxDTCTL update by DMA mode Refer to STxCTL0 bit description.
9	STxCMP3V	HRTIMER_STxCMP3V update by DMA mode

		Refer to STxCTL0 bit description.
8	STxCMP2V	HRTIMER_STxCMP2V update by DMA mode Refer to STxCTL0 bit description.
7	STxCMP1V	HRTIMER_STxCMP1V update by DMA mode Refer to STxCTL0 bit description.
6	STxCMP0V	HRTIMER_STxCMP0V update by DMA mode Refer to STxCTL0 bit description.
5	STxCREP	HRTIMER_STxCREP update by DMA mode Refer to STxCTL0 bit description.
4	STxCAR	HRTIMER_STxCAR update by DMA mode Refer to STxCTL0 bit description.
3	STxCNT	HRTIMER_STxCNT update by DMA mode Refer to STxCTL0 bit description.
2	STxDMAINTEN	HRTIMER_STxDMAINTEN update by DMA mode Refer to STxCTL0 bit description.
1	STxINTC	HRTIMER_STxINTC update by DMA mode Refer to STxCTL0 bit description.
0	STxCTL0	HRTIMER_STxCTL0 update by DMA mode This bit defines if the HRTIMER_STxCTL0 register is updated by the DMA mode. 0: HRTIMER_STxCTL0 register is not updated by DMA mode. 1: HRTIMER_STxCTL0 register is updated DMA mode.

HRTIMER DMA transfer buffer register (HRTIMER_DMATB)

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
DMATB[31:16]															
wo															

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

Bits	Fields	Descriptions
31:0	DMATB[31:0]	DMA transfer buffer When a write operation is assigned to this register, the register selected in HRTIMER_DMAUPMTR and HRTIMER_DMAUPSTxR(x=0..4) registers will be

accessed.

The increment of the register pointer is calculated by hardware.

HRTIMER ADC extended trigger register (HRTIMER_ADCEXTTRG)

Address offset: 0x78

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	ADC9TRG[4:0]					ADC8TRG[4:0]					ADC7TRG[4:0]				
	rw					rw					rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	ADC6TRG[4:0]					ADC5TRG[4:0]					ADC4TRG[4:0]				
	rw					rw					rw				

Bits	Fields	Descriptions
31	Reserved	Must kept as reserved value.
30:26	ADC9TRG[4:0]	ADC trigger 9 selection This bit selects the ADC trigger 9 source. Refer to ADC5TRG[4:0] description.
25:21	ADC8TRG[4:0]	ADC trigger 8 selection This bit selects the ADC trigger 8 source. Refer to ADC4TRG[4:0] description.
20:16	ADC7TRG[4:0]	ADC trigger 7 selection This bit selects the ADC trigger 7 source. Refer to ADC5TRG[4:0] description.
15	Reserved	Must kept as reserved value.
14:10	ADC6TRG[4:0]	ADC trigger 6 selection This bit selects the ADC trigger 6 source. Refer to ADC4TRG[4:0] description.
9:5	ADC5TRG[4:0]	ADC trigger 5 selection This bit selects the ADC trigger 5 source. 000000: ADC triggered by compare 0 event of Master_TIMER 000001: ADC triggered by compare 1 event of Master_TIMER 000010: ADC triggered by compare 2 event of Master_TIMER 000011: ADC triggered by compare 3 event of Master_TIMER 000100: ADC triggered by period event of Master_TIMER 000101: ADC triggered by external event 5 000110: ADC triggered by external event 6 000111: ADC triggered by external event 7 001000: ADC triggered by external event 8 001001: ADC triggered by external event 9 001010: ADC triggered by compare 1 event of Slave_TIMER0 001011: ADC triggered by compare 3 event of Slave_TIMER0 001100: ADC triggered by period event of Slave_TIMER0

001101: ADC triggered by compare 1 event of Slave_TIMER1
001110: ADC triggered by compare 3 event of Slave_TIMER1
001111: ADC triggered by period event of Slave_TIMER1
010000: ADC triggered by compare 1 event of Slave_TIMER2
010001: ADC triggered by compare 3 event of Slave_TIMER2
010010: ADC triggered by period event of Slave_TIMER2
10011: ADC triggered by reset and counter roll-over event of Slave_TIMER2
010100: ADC triggered by compare 1 event of Slave_TIMER3
010101: ADC triggered by compare 3 event of Slave_TIMER3
010110: ADC triggered by period event of Slave_TIMER3
010111: ADC triggered by reset and counter roll-over event of Slave_TIMER3
011000: ADC triggered by compare 1 event of Slave_TIMER4
011001: ADC triggered by compare 2 event of Slave_TIMER4
011010: ADC triggered by compare 3 event of Slave_TIMER4
011011: ADC triggered by reset and counter roll-over event of Slave_TIMER5
011100: ADC triggered by compare 1 event of Slave_TIMER5
011101: ADC triggered by compare 2 event of Slave_TIMER5
011110: ADC triggered by compare 3 event of Slave_TIMER5
011111: ADC triggered by period event of Slave_TIMER5
100000: ADC triggered by compare 1 event of Slave_TIMER6
100001: ADC triggered by compare 2 event of Slave_TIMER6
100010: ADC triggered by compare 3 event of Slave_TIMER6
100011: ADC triggered by period event of Slave_TIMER6
100100: ADC triggered by reset and counter roll-over event of Slave_TIMER6
100101: ADC triggered by compare 1 event of Slave_TIMER7
100110: ADC triggered by compare 2 event of Slave_TIMER7
100111: ADC triggered by compare 3 event of Slave_TIMER7
101000: ADC triggered by period event of Slave_TIMER7
101001: ADC triggered by reset and counter roll-over event of Slave_TIMER7
others: Reserved.
Note: ADC5TRG[5] reference register HRTIMER_ADCEXTTRGA.

4:0	ADC4TRG[4:0]	ADC trigger 4 selection This bit selects the ADC trigger 4 source. 000000: ADC triggered by compare 0 event of Master_TIMER 000001: ADC triggered by compare 1 event of Master_TIMER 000010: ADC triggered by compare 2 event of Master_TIMER 000011: ADC triggered by compare 3 event of Master_TIMER 000100: ADC triggered by period event of Master_TIMER 000101: ADC triggered by external event 0 000110: ADC triggered by external event 1 000111: ADC triggered by external event 2 001000: ADC triggered by external event 3 001001: ADC triggered by external event 4 001010: ADC triggered by compare 2 event of Slave_TIMER0

001011: ADC triggered by compare 3 event of Slave_TIMER0
001100: ADC triggered by period event of Slave_TIMER0
001101: ADC triggered by reset and counter roll-over event of Slave_TIMER0
001110: ADC triggered by compare 2 event of Slave_TIMER1
001111: ADC triggered by compare 3 event of Slave_TIMER1
010000: ADC triggered by period event of Slave_TIMER1
010001: ADC triggered by reset and counter roll-over event of Slave_TIMER1
010010: ADC triggered by compare 2 event of Slave_TIMER2
010011: ADC triggered by compare 3 event of Slave_TIMER2
010100: ADC triggered by period event of Slave_TIMER2
010101: ADC triggered by compare 2 event of Slave_TIMER3
010110: ADC triggered by compare 3 event of Slave_TIMER3
010111: ADC triggered by period event of Slave_TIMER3
011000: ADC triggered by compare 2 event of Slave_TIMER4
011001: ADC triggered by compare 3 event of Slave_TIMER4
011010: ADC triggered by period event of Slave_TIMER4
011011: ADC triggered by compare 1 event of Slave_TIMER5
011100: ADC triggered by compare 2 event of Slave_TIMER5
011101: ADC triggered by compare 3 event of Slave_TIMER5
011110: ADC triggered by period event of Slave_TIMER5
011111: ADC triggered by reset and counter roll-over event of Slave_TIMER5
100000: ADC triggered by compare 1 event of Slave_TIMER6
100001: ADC triggered by compare 2 event of Slave_TIMER6
100010: ADC triggered by compare 3 event of Slave_TIMER6
100011: ADC triggered by period event of Slave_TIMER6
100100: ADC triggered by reset and counter roll-over event of Slave_TIMER6
100101: ADC triggered by compare 1 event of Slave_TIMER7
100110: ADC triggered by compare 2 event of Slave_TIMER7
100111: ADC triggered by compare 3 event of Slave_TIMER7
101000: ADC triggered by period event of Slave_TIMER7
101001: ADC triggered by reset and counter roll-over event of Slave_TIMER7
others: Reserved.

Note: ADC4TRG[5] reference register HRTIMER_ADCEXTTRGA.

HRTIMER ADC trigger update register (HRTIMER_ADCTRGUPD)

Address offset: 0x7C
Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								ADTG9USRC[3:0]				ADTG8USRC[3:0]			
								rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

ADTG7USRC[3:0]	ADTG6USRC[3:0]	ADTG5USRC[3:0]	ADTG4USRC[3:0]
rw	rw	rw	rw

Bits	Fields	Descriptions
31:24	Reserved	Must kept as reserved value.
23:20	ADTG9USRC[3:0]	ADC trigger 9 update source Refer to ADCTG4USRC[3:0] description.
19:16	ADTG8USRC[3:0]	ADC trigger 8 update source Refer to ADCTG4USRC[3:0] description.
15:12	ADTG7USRC[3:0]	ADC trigger 7 update source Refer to ADCTG4USRC[3:0] description.
11:8	ADTG6USRC[3:0]	ADC trigger 6update source Refer to ADCTG4USRC[3:0] description.
7:4	ADTG5USRC[3:0]	ADC trigger5 update source Refer to ADCTG4USRC[3:0] description.
3:0	ADTG4USRC[2:0]	ADC4TRG update source This bit-field can be configured by software to specify the source to update the ADC4TRG[5:0] in HRTIMER_ADCEXTTRG register and HRTIMER_ADCEXTTRGA register. 0000: Master_TIMER update event 0001: Slaver_TIMER0 update event 0010: Slaver_TIMER1 update event 0011: Slaver_TIMER2 update event 0100: Slaver_TIMER3 update event 0101: Slaver_TIMER4 update event 0110: Slaver_TIMER5 update event 0111: Slaver_TIMER6 update event 1000: Slaver_TIMER7 update event Other values are reserved

HRTIMER ADC post scaler register 0(HRTIMER_ADCPSCR0)

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			ADC4PSC[4:0]					Reserved	ADC3PSC[4:0]					Reserved	ADC2PSC[4]
			rw						rw						rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

ADC2PSC[3:0]	Reserved	ADC1PSC[4:0]	Reserved	ADC0PSC[4:0]
rw		rw		rw

Bits	Fields	Descriptions
31:29	Reserved	Must kept as reserved value.
28:24	ADC4PSC[4:0]	ADC trigger 4 prescaler This bit selects the ADC trigger 4 prescaler.
23	Reserved	Must kept as reserved value.
22:18	ADC3PSC[4:0]	ADC trigger 3 prescaler This bit selects the ADC trigger 3 prescaler.
17	Reserved	Must kept as reserved value.
16:12	ADC2PSC[4:0]	ADC trigger 2 prescaler This bit selects the ADC trigger 2 prescaler.
11	Reserved	Must kept as reserved value.
10:6	ADC1PSC[4:0]	ADC trigger 1 prescaler This bit selects the ADC trigger 1 prescaler.
5	Reserved	Must kept as reserved value.
4:0	ADC0PSC[4:0]	ADC trigger 0 prescaler This bit selects the ADC trigger 0 prescaler.

HRTIMER ADC post scaler register 1(HRTIMER_ADCPSCR1)

Address offset: 0x84

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			ADC9PSC[4:0]					Reserved	ADC8PSC[4:0]					Reserved	ADC7PSC[4]
			rw						rw						rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADC7PSC[4:0]				Reserved	ADC6PSC[4:0]					Reserved	ADC5PSC[4:0]				
rw					rw						rw				

Bits	Fields	Descriptions
31:29	Reserved	Must kept as reserved value.
28:24	ADC9PSC[4:0]	ADC trigger 9 prescaler This bit selects the ADC trigger 9 prescaler.

23	Reserved	Must kept as reserved value.
22:18	ADC8PSC[4:0]	ADC trigger 8 prescaler This bit selects the ADC trigger 8 prescaler.
17	Reserved	Must kept as reserved value.
16:12	ADC7PSC[4:0]	ADC trigger 7 prescaler This bit selects the ADC trigger 7 prescaler.
11	Reserved	Must kept as reserved value.
10:6	ADC6PSC[4:0]	ADC trigger 6 prescaler This bit selects the ADC trigger 6 prescaler.
5	Reserved	Must kept as reserved value.
4:0	ADC5PSC[4:0]	ADC trigger 5 prescaler This bit selects the ADC trigger 5 prescaler.

HRTIMER fault input configuration register 2 (HRTIMER_FLTINCFG2)

Address offset: 0x88

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLT3RST	FLT3CNTRST	FLT3CNT[3:0]				FLT3BLKS	FLT3BLKEN	FLT2RST	FLT2CNTRST	FLT2CNT[3:0]				FLT2BLKS	FLT2BLKEN
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
FLT1RST	FLT1CNTRST	FLT1CNT[3:0]				FLT1BLKS	FLT1BLKEN	FLT0RST	FLT0CNTRST	FLT0CNT[3:0]				FLT0BLKS	FLT0BLKEN
rw	rw	rw				rw	rw	rw	rw	rw				rw	rw

Bits	Fields	Descriptions
31	FLT3RST	Fault 3 reset mode Refer to FLT0RST description.
30	FLT3CNTRST	Fault 3 counter reset Refer to FLT0CNTRST description.
29:26	FLT3CNT[3:0]	Fault 3 counter Refer to FLT0CNT description.
25	FLT3BLKS	Fault 3 blanking source

		Refer to FLT0BLKS description.
24	FLT3BLKEN	Fault 3 blanking enable Refer to FLT0BLKEN description.
23	FLT2RST	Fault 2 reset mode Refer to FLT0RST description.
22	FLT2 CNTRST	Fault 2 counter reset Refer to FLT0CNTRST description.
21:18	FLT2CNT[3:0]	Fault 2 counter Refer to FLT0CNT description.
17	FLT2BLKS	Fault 2 blanking source Refer to FLT0BLKS description.
16	FLT2BLKEN	Fault 3 blanking enable Refer to FLT0BLKEN description.
15	FLT1RST	Fault 1 reset mode Refer to FLT0RST description.
14	FLT1CNTRST	Fault 1 counter reset Refer to FLT0CNTRST description.
13:10	FLT1CNT[3:0]	Fault 1 counter Refer to FLT0CNT description.
9	FLT1BLKS	Fault 1 blanking source Refer to FLT0BLKS description.
8	FLT1BLKEN	Fault 1 blanking enable Refer to FLT0BLKEN description.
7	FLT0RST	Fault 0 reset mode: This bit specifies the reset mode for the FAULT0 counter. 0: Reset the Fault 0 counter on each reset/ roll-over event. 1: When the FLTRECCTL bit in HRTIMER_FLTRECCTL register is set to 0, fault counter reset on each reset / roll-over event only if no fault event is active. When the FLTRECCTL bit in HRTIMER_FLTRECCTL register is set to 1, Fault counter reset on each reset / roll-over event only if no fault event happens in one period. Note: This configuration is effective only when the FLT0INEN enable bit is in a reset

		state.
6	FLT0CNTRST	Fault 0 counter reset This bit resets the FAULT0 counter. It is set by software and reset by hardware. 0: No action 1: Fault 0 counter is reset
5:2	FLT0CNT[3:0]	Fault 0 Counter: This bitfield determines the threshold for the FAULT0 counter. A fault is deemed valid when the event count matches the value of (FLT0CNT[3:0]+1).
1	FLT0BLKS	Fault 0 blanking source The FTL0BLKS bit selects the FAULT0 blanking source. 0: Fault 0 reset and compare windows 1: Fault 0 compare and compare windows Note: This bitfield is written only when FLT0INEN enable bit is reset.
0	FLT0BLKEN	Fault 0 blanking enable The FTL0BLKEN bit determines the blanking mode for FAULT0, with the blanking source specified by the FLT0BLKS bit. 0: No blanking on Fault 0 1: Fault 0 blanking mode Note: This bitfield is configurable only when the FLT0INEN enable bit is reset.

HRTIMER fault input configuration register 3 (HRTIMER_FLTINCFG3)

Address offset: 0x8C

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLT7RST	FLT7CNTRST	FLT7CNT[3:0]				FLT7BLKS	FLT7BLKEN	FLT6RST	FLT6CNTRST	FLT6CNT[3:0]				FLT6BLKS	FLT6BLKEN
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
FLT5RST	FLT5CNTRST	FLT5CNT[3:0]				FLT5BLKS	FLT5BLKEN	FLT4RST	FLT4CNTRST	FLT4CNT[3:0]				FLT4BLKS	FLT4BLKEN
rw	rw	rw				rw	rw	rw	rw	rw				rw	rw

Bits	Fields	Descriptions
31	FLT7RST	Fault 7 reset mode Refer to FLT0RST description.
30	FLT7 CNTRST	Fault 7 counter reset Refer to FLT0CNTRST description.
29:26	FLT7CNT[3:0]	Fault 7 counter

		Refer to FLT0CNT description.
25	FLT7BLKS	Fault 7 blanking source Refer to FLT0BLKS description.
24	FLT7BLKEN	Fault 7 blanking enable Refer to FLT0BLKEN description.
23	FLT6RST	Fault 6 reset mode Refer to FLT0RST description.
22	FLT6CNTRST	Fault 6 counter reset Refer to FLT0CNTRST description.
21:18	FLT6CNT[3:0]	Fault 6 counter Refer to FLT0CNT description.
17	FLT6BLKS	Fault 6 blanking source Refer to FLT0BLKS description.
16	FLT6BLKEN	Fault 6 blanking enable Refer to FLT0BLKEN description.
15	FLT5RST	Fault 5 reset mode Refer to FLT0RST description.
14	FLT5CNTRST	Fault 5 counter reset Refer to FLT0CNTRST description.
13:10	FLT5CNT[3:0]	Fault 5 counter Refer to FLT0CNT description.
9	FLT5BLKS	Fault 5 blanking source Refer to FLT0BLKS description.
8	FLT5BLKEN	Fault 5 blanking enable Refer to FLT0BLKEN description.
7	FLT4RST	Fault 4 reset mode Refer to FLT0RST description.
6	FLT4CNTRST	Fault 4 counter reset Refer to FLT0CNTRST description.
5:2	FLT4CNT[3:0]	Fault 4 counter Refer to FLT0CNT description.
1	FLT4BLKS	Fault 4 blanking source Refer to FLT0BLKS description.
0	FLT4BLKEN	Fault 4 blanking enable

Refer to FLT0BLKEN description.

HRTIMER bunch mode start trigger add register (HRTIMER_BMSTRGA)

Address offset: 0x108

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	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				ST7CMP1	ST7CMP0	ST7REP	ST7RST	ST6CMP1	ST6CMP0	ST6REP	ST6RST	ST5CMP1	ST5CMP0	ST5REP	ST5RST
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:12	Reserved	Must kept as reserved value.
11	ST7CMP1	Slave_TIMER7 compare 1 event triggers bunch mode operation Refer to MTCMP1 description.
10	ST7CMP0	Slave_TIMER7 compare 0 event triggers bunch mode operation Refer to MTCMP0 description.
9	ST7REP	Slave_TIMER7 repetition event triggers bunch mode operation Refer to MTREP description.
8	ST7RST	Slave_TIMER7 reset event triggers bunch mode operation Refer to MTRST description.
7	ST6CMP1	Slave_TIMER6 compare 1 event triggers bunch mode operation Refer to MTCMP1 description.
6	ST6CMP0	Slave_TIMER6 compare 0 event triggers bunch mode operation Refer to MTCMP6 description.
5	ST6REP	Slave_TIMER6 repetition event triggers bunch mode operation Refer to MTREP description.
4	ST6RST	Slave_TIMER6 reset event triggers bunch mode operation Refer to MTRST description.
3	ST5CMP1	Slave_TIMER5 compare 1 event triggers bunch mode operation Refer to MTCMP1 description.
2	ST5CMP0	Slave_TIMER5 compare 0 event triggers bunch mode operation

		Refer to MTCMP6 description.
1	ST5REP	Slave_TIMER5 repetition event triggers bunch mode operation Refer to MTREP description.
0	ST5RST	Slave_TIMER5 reset event triggers bunch mode operation Refer to MTRST description.

HRTIMER fault input configuration register 4 (HRTIMER_FLTINCFG4)

Address offset: 0x10C

Reset value: 0x0000 0000

This register has 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
FLT7INPROT	FLT7INFC[3:0]				FLT7INSRC[0]	FLT7INP	FLT7INEN	FLT6INPROT	FLT6INFC[3:0]				FLT6INSRC[0]	FLT6INP	FLT6INEN
rw	rw				rw	rw	rw	rw	rw				rw	rw	rw

Bits	Fields	Descriptions
31:16	Reserved	Must kept as reserved value.
15	FLT7INPROT	Protect fault 7 input configuration Refer to FLT0INPROT description.
14:11	FLT7INFC[3:0]	Fault 7 input filter control Refer to FLT0INFC[3:0] description.
10	FLT7INSRC[0]	Fault 7 input source Refer to FLT0INSRC description.
9	FLT7INP	Fault 7 input polarity Refer to FLT0INP description.
8	FLT7INEN	Fault 7 input enable Refer to FLT0INEN description.
7	FLT6INPROT	Protect fault 6 input configuration Refer to FLT0INPROT description.
6:3	FLT6INFC[3:0]	Fault 6 input filter control Refer to FLT0INFC[3:0] description.
2	FLT6INSRC[0]	Fault 6 input source Refer to FLT0INSRC description.

1	FLT6INP	Fault 6 input polarity Refer to FLT0INP description.
0	FLT6INEN	Fault 6 input enable Refer to FLT0INEN description.

HRTIMER ADC extended trigger add register (HRTIMER_ADCEXTTRGA)

Address offset: 0x110

Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Resreved										ADC9TRG[5]	ADC8TRG[5]	ADC7TRG[5]	ADC6TRG[5]	ADC5TRG[5]	ADC4TRG[5]
										rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:6	Reserved	Must kept as reserved value.
5	ADC9TRG[5]	ADC trigger 9 selection This bit selects the ADC trigger 9 source. Refer to ADC5TRG[4:0] description.
4	ADC8TRG[5]	ADC trigger 8 selection This bit selects the ADC trigger 8 source. Refer to ADC4TRG[4:0] description.
3	ADC7TRG[5]	ADC trigger 7 selection This bit selects the ADC trigger 7 source. Refer to ADC5TRG[4:0] description.
2	ADC6TRG[5]	ADC trigger 6 selection This bit selects the ADC trigger 6 source. Refer to ADC4TRG[4:0] description.
1	ADC5TRG[5]	ADC trigger 5 selection This bit selects the ADC trigger 5 source. Refer to ADC5TRG[4:0] description.
0	ADC4TRG[5]	ADC trigger 4 selection This bit selects the ADC trigger 4 source. Refer to ADC4TRG[4:0] description.

HRTIMER trigger source 0 to ADC add register (HRTIMER_ADCTRIGS0A)

Address offset: 0x114

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Resreved											TRG0ST7 RST	TRG0ST7 PER	TRG0ST7 C3	TRG0ST7 C2	TRG0ST7 C1
											rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Resreved			TRG0ST6 RST	TRG0ST6 PER	TRG0ST6 C3	TRG0ST6 C2	TRG0ST6 C1	Resreved			TRG0ST5 RST	TRG0ST5 PER	TRG0ST5 C3	TRG0ST5 C2	TRG0ST5 C1
			rw	rw	rw	rw	rw				rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:21	Reserved	Must kept as reserved value.
20	TRG0ST7RST	HRTIMER_ADCTRIG0 on Slave_TIMER7 reset The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG0. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER7 reset. 1: ADC trigger event generated on HRTIMER Slave_TIMER7 reset reset
19	TRG0ST7PER	HRTIMER_ADCTRIG0 on Slave_TIMER7 period event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG0. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER7 period event. 1: ADC trigger event generated on HRTIMER Slave_TIMER7 period event.
18	TRG0ST7C3	HRTIMER_ADCTRG0 on Slave_TIMER7 compare 3 event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRG0. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER7 compare 3 event. 1: ADC trigger event generated on HRTIMER Slave_TIMER7 compare 3 event.
17	TRG0ST7C2	HRTIMER_ADCTRG0 on Slave_TIMER7 compare 2 event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRG0. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER7 compare 2 event. 1: ADC trigger event generated on HRTIMER Slave_TIMER7 compare 2 event.
16	TRG0ST7C1	HRTIMER_ADCTRG0 on Slave_TIMER7 compare 1 event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRG0. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER7 compare 1 event. 1: ADC trigger event generated on HRTIMER Slave_TIMER7 compare 1 event.
15:13	Reserved	Must kept as reserved value.

12	TRG0ST6RST	HRTIMER_ADCTRIG0 on Slave_TIMER6 reset The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG0. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER6 reset. 1: ADC trigger event generated on HRTIMER Slave_TIMER6 reset reset
11	TRG0ST6PER	HRTIMER_ADCTRIG0 on Slave_TIMER6 period event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG0. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER6 period event. 1: ADC trigger event generated on HRTIMER Slave_TIMER6 period event.
10	TRG0ST6C3	HRTIMER_ADCTRG0 on Slave_TIMER6 compare 3 event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRG0. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER6 compare 3 event. 1: ADC trigger event generated on HRTIMER Slave_TIMER6 compare 3 event.
9	TRG0ST6C2	HRTIMER_ADCTRG0 on Slave_TIMER6 compare 2 event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRG0. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER6 compare 2 event. 1: ADC trigger event generated on HRTIMER Slave_TIMER6 compare 2 event.
8	TRG0ST6C1	HRTIMER_ADCTRG0 on Slave_TIMER6 compare 1 event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRG0. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER6 compare 1 event. 1: ADC trigger event generated on HRTIMER Slave_TIMER6 compare 1 event.
7:5	Reserved	Must kept as reserved value.
4	TRG0ST5RST	HRTIMER_ADCTRIG0 on Slave_TIMER5 reset The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG0. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER5 reset. 1: ADC trigger event generated on HRTIMER Slave_TIMER5 reset reset
3	TRG0ST5PER	HRTIMER_ADCTRIG0 on Slave_TIMER5 period event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG0. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER5 period event. 1: ADC trigger event generated on HRTIMER Slave_TIMER5 period event.

2	TRG0ST5C3	HRTIMER_ADCTRG0 on Slave_TIMER5 compare 3 event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRG0. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER5 compare 3 event. 1: ADC trigger event generated on HRTIMER Slave_TIMER5 compare 3 event.
1	TRG0ST5C2	HRTIMER_ADCTRG0 on Slave_TIMER5 compare 2 event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRG0. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER5 compare 2 event. 1: ADC trigger event generated on HRTIMER Slave_TIMER5 compare 2 event.
0	TRG0ST5C1	HRTIMER_ADCTRG0 on Slave_TIMER5 compare 1 event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRG0. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER5 compare 1 event. 1: ADC trigger event generated on HRTIMER Slave_TIMER5 compare 1 event.

HRTIMER trigger source 1 to ADC add register (HRTIMER_ADCTRIGS1A)

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
Resreved											TRG1ST7 RST	TRG1ST7 PER	TRG1ST7 C3	TRG1ST7 C2	TRG1ST7 C1
											rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Resreved			TRG1ST6 RST	TRG1ST6 PER	TRG1ST6 C3	TRG1ST6 C2	TRG1ST6 C1	Resreved			TRG1ST5 RST	TRG1ST5 PER	TRG1ST5 C3	TRG1ST5 C2	TRG1ST5 C1
			rw	rw	rw	rw	rw				rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:21	Reserved	Must kept as reserved value.
20	TRG1ST7RST	HRTIMER_ADCTRIG1 on Slave_TIMER7 reset The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG1. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER7 reset. 1: ADC trigger event generated on HRTIMER Slave_TIMER7 reset reset

19	TRG1ST7PER	HRTIMER_ADCTRIG1 on Slave_TIMER7 period event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG1. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER7 period event. 1: ADC trigger event generated on HRTIMER Slave_TIMER7 period event.
18	TRG1ST7C3	HRTIMER_ADCTRG1 on Slave_TIMER7 compare 3 event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRG1. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER7 compare 3 event. 1: ADC trigger event generated on HRTIMER Slave_TIMER7 compare 3 event.
17	TRG1ST7C2	HRTIMER_ADCTRG1 on Slave_TIMER7 compare 2 event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRG1. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER7 compare 2 event. 1: ADC trigger event generated on HRTIMER Slave_TIMER7 compare 2 event.
16	TRG1ST7C1	HRTIMER_ADCTRG1 on Slave_TIMER7 compare 1 event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRG1. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER7 compare 1 event. 1: ADC trigger event generated on HRTIMER Slave_TIMER7 compare 1 event.
15:13	Reserved	Must kept as reserved value.
12	TRG1ST6RST	HRTIMER_ADCTRIG1 on Slave_TIMER6 reset The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG1. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER6 reset. 1: ADC trigger event generated on HRTIMER Slave_TIMER6 reset reset
11	TRG1ST6PER	HRTIMER_ADCTRIG1 on Slave_TIMER6 period event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG1. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER6 period event. 1: ADC trigger event generated on HRTIMER Slave_TIMER6 period event.
10	TRG1ST6C3	HRTIMER_ADCTRG1 on Slave_TIMER6 compare 3 event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRG1. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER6 compare 3 event.

		1: ADC trigger event generated on HRTIMER Slave_TIMER6 compare 3 event.
9	TRG1ST6C2	HRTIMER_ADCTRG1 on Slave_TIMER6 compare 2 event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRG1. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER6 compare 2 event. 1: ADC trigger event generated on HRTIMER Slave_TIMER6 compare 2 event.
8	TRG1ST6C1	HRTIMER_ADCTRG1 on Slave_TIMER6 compare 1 event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRG1. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER6 compare 1 event. 1: ADC trigger event generated on HRTIMER Slave_TIMER6 compare 1 event.
7:5	Reserved	Must kept as reserved value.
4	TRG1ST5RST	HRTIMER_ADCTRIG1 on Slave_TIMER5 reset The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG1. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER5 reset. 1: ADC trigger event generated on HRTIMER Slave_TIMER5 reset reset
3	TRG1ST5PER	HRTIMER_ADCTRIG1 on Slave_TIMER5 period event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG1. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER5 period event. 1: ADC trigger event generated on HRTIMER Slave_TIMER5 period event.
2	TRG1ST5C3	HRTIMER_ADCTRG1 on Slave_TIMER5 compare 3 event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRG1. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER5 compare 3 event. 1: ADC trigger event generated on HRTIMER Slave_TIMER5 compare 3 event.
1	TRG1ST5C2	HRTIMER_ADCTRG1 on Slave_TIMER5 compare 2 event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRG1. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER5 compare 2 event. 1: ADC trigger event generated on HRTIMER Slave_TIMER5 compare 2 event.
0	TRG1ST5C1	HRTIMER_ADCTRG1 on Slave_TIMER5 compare 1 event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRG1. This bit specifies whether the event can generate the ADC trigger event.

0: No ADC trigger event generated on HRTIMER Slave_TIMER5 compare 1 event.
1: ADC trigger event generated on HRTIMER Slave_TIMER5 compare 1 event.

HRTIMER trigger source 2 to ADC add register (HRTIMER_ADCTRIGS2A)

Address offset: 0x11C
Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:21	Reserved	Must kept as reserved value.
20	TRG2ST7RST	HRTIMER_ADCTRIG2 on Slave_TIMER7 reset The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG2. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER7 reset. 1: ADC trigger event generated on HRTIMER Slave_TIMER7 reset reset
19	TRG2ST7PER	HRTIMER_ADCTRIG2 on Slave_TIMER7 period event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG2. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER7 period event. 1: ADC trigger event generated on HRTIMER Slave_TIMER7 period event.
18	TRG2ST7C3	HRTIMER_ADCTRG2 on Slave_TIMER7 compare 3 event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRG2. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER7 compare 3 event. 1: ADC trigger event generated on HRTIMER Slave_TIMER7 compare 3 event.
17	TRG2ST7C2	HRTIMER_ADCTRG2 on Slave_TIMER7 compare 2 event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRG2. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER7 compare 2 event.

		1: ADC trigger event generated on HRTIMER Slave_TIMER7 compare 2 event.
16	TRG2ST7C1	HRTIMER_ADCTRG2 on Slave_TIMER7 compare 1 event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRG2. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER7 compare 1 event. 1: ADC trigger event generated on HRTIMER Slave_TIMER7 compare 1 event.
15:13	Reserved	Must kept as reserved value.
12	TRG2ST6RST	HRTIMER_ADCTRIG2 on Slave_TIMER6 reset The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG2. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER6 reset. 1: ADC trigger event generated on HRTIMER Slave_TIMER6 reset reset
11	TRG2ST6PER	HRTIMER_ADCTRIG2 on Slave_TIMER6 period event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG2. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER6 period event. 1: ADC trigger event generated on HRTIMER Slave_TIMER6 period event.
10	TRG2ST6C3	HRTIMER_ADCTRG2 on Slave_TIMER6 compare 3 event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRG2. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER6 compare 3 event. 1: ADC trigger event generated on HRTIMER Slave_TIMER6 compare 3 event.
9	TRG2ST6C2	HRTIMER_ADCTRG2 on Slave_TIMER6 compare 2 event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRG2. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER6 compare 2 event. 1: ADC trigger event generated on HRTIMER Slave_TIMER6 compare 2 event.
8	TRG2ST6C1	HRTIMER_ADCTRG2 on Slave_TIMER6 compare 1 event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRG2. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER6 compare 1 event. 1: ADC trigger event generated on HRTIMER Slave_TIMER6 compare 1 event.
7:5	Reserved	Must kept as reserved value.
4	TRG2ST5RST	HRTIMER_ADCTRIG2 on Slave_TIMER5 reset The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG2.

This bit specifies whether the event can generate the ADC trigger event.
0: No ADC trigger event generated on HRTIMER Slave_TIMER5 reset.
1: ADC trigger event generated on HRTIMER Slave_TIMER5 reset reset

3	TRG2ST5PER	HRTIMER_ADCTRIG2 on Slave_TIMER5 period event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG2. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER5 period event. 1: ADC trigger event generated on HRTIMER Slave_TIMER5 period event.
2	TRG2ST5C3	HRTIMER_ADCTRG2 on Slave_TIMER5 compare 3 event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRG2. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER5 compare 3 event. 1: ADC trigger event generated on HRTIMER Slave_TIMER5 compare 3 event.
1	TRG2ST5C2	HRTIMER_ADCTRG2 on Slave_TIMER5 compare 2 event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRG2. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER5 compare 2 event. 1: ADC trigger event generated on HRTIMER Slave_TIMER5 compare 2 event.
0	TRG2ST5C1	HRTIMER_ADCTRG2 on Slave_TIMER5 compare 1 event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRG2. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER5 compare 1 event. 1: ADC trigger event generated on HRTIMER Slave_TIMER5 compare 1 event.

HRTIMER trigger source 3 to ADC add register (HRTIMER_ADCTRIGS3A)

Address offset: 0x120
Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Resreved											TRG3ST7 RST	TRG3ST7 PER	TRG3ST7 C3	TRG3ST7 C2	TRG3ST7 C1
											rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Resreved			TRG3ST6 RST	TRG3ST6 PER	TRG3ST6 C3	TRG3ST6 C2	TRG3ST6 C1	Resreved			TRG3ST5 RST	TRG3ST5 PER	TRG3ST5 C3	TRG3ST5 C2	TRG3ST5 C1
			rw	rw	rw	rw	rw				rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:21	Reserved	Must kept as reserved value.
20	TRG3ST7RST	HRTIMER_ADCTRIG3 on Slave_TIMER7 reset The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG3. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER7 reset. 1: ADC trigger event generated on HRTIMER Slave_TIMER7 reset reset
19	TRG3ST7PER	HRTIMER_ADCTRIG3 on Slave_TIMER7 period event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG3. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER7 period event. 1: ADC trigger event generated on HRTIMER Slave_TIMER7 period event.
18	TRG3ST7C3	HRTIMER_ADCTRG3 on Slave_TIMER7 compare 3 event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRG3. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER7 compare 3 event. 1: ADC trigger event generated on HRTIMER Slave_TIMER7 compare 3 event.
17	TRG3ST7C2	HRTIMER_ADCTRG3 on Slave_TIMER7 compare 2 event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRG3. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER7 compare 2 event. 1: ADC trigger event generated on HRTIMER Slave_TIMER7 compare 2 event.
16	TRG3ST7C1	HRTIMER_ADCTRG3 on Slave_TIMER7 compare 1 event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRG3. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER7 compare 1 event. 1: ADC trigger event generated on HRTIMER Slave_TIMER7 compare 1 event.
15:13	Reserved	Must kept as reserved value.
12	TRG3ST6RST	HRTIMER_ADCTRIG3 on Slave_TIMER6 reset The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG3. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER6 reset. 1: ADC trigger event generated on HRTIMER Slave_TIMER6 reset reset
11	TRG3ST6PER	HRTIMER_ADCTRIG3 on Slave_TIMER6 period event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG3. This bit specifies whether the event can generate the ADC trigger event.

		0: No ADC trigger event generated on HRTIMER Slave_TIMER6 period event. 1: ADC trigger event generated on HRTIMER Slave_TIMER6 period event.
10	TRG3ST6C3	HRTIMER_ADCTRG3 on Slave_TIMER6 compare 3 event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRG3. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER6 compare 3 event. 1: ADC trigger event generated on HRTIMER Slave_TIMER6 compare 3 event.
9	TRG3ST6C2	HRTIMER_ADCTRG3 on Slave_TIMER6 compare 2 event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRG3. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER6 compare 2 event. 1: ADC trigger event generated on HRTIMER Slave_TIMER6 compare 2 event.
8	TRG3ST6C1	HRTIMER_ADCTRG3 on Slave_TIMER6 compare 1 event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRG3. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER6 compare 1 event. 1: ADC trigger event generated on HRTIMER Slave_TIMER6 compare 1 event.
7:5	Reserved	Must kept as reserved value.
4	TRG3ST5RST	HRTIMER_ADCTRIG3 on Slave_TIMER5 reset The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG3. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER5 reset. 1: ADC trigger event generated on HRTIMER Slave_TIMER5 reset reset
3	TRG3ST5PER	HRTIMER_ADCTRIG3 on Slave_TIMER5 period event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRIG3. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER5 period event. 1: ADC trigger event generated on HRTIMER Slave_TIMER5 period event.
2	TRG3ST5C3	HRTIMER_ADCTRG3 on Slave_TIMER5 compare 3 event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRG3. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER5 compare 3 event. 1: ADC trigger event generated on HRTIMER Slave_TIMER5 compare 3 event.
1	TRG3ST5C2	HRTIMER_ADCTRG3 on Slave_TIMER5 compare 2 event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRG3. This

		bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER5 compare 2 event. 1: ADC trigger event generated on HRTIMER Slave_TIMER5 compare 2 event.
0	TRG3ST5C1	HRTIMER_ADCTRG3 on Slave_TIMER5 compare 1 event The HRTIMER can generate an ADC trigger event on HRTIMER_ADCTRG3. This bit specifies whether the event can generate the ADC trigger event. 0: No ADC trigger event generated on HRTIMER Slave_TIMER5 compare 1 event. 1: ADC trigger event generated on HRTIMER Slave_TIMER5 compare 1 event.

HRTIMER fault recovery control register (HRTIMER_FLTRECCTL)

Address offset: 0x17C

Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:2	Reserved	Must kept as reserved value.
1	FLTRECCTL	fault recovery control 0: Fault counter reset on each reset / roll-over event only if no fault event is active. 1: Fault counter reset on each reset / roll-over event only if no fault event happens in one period. **Note:** This configuration is effective only when FLTxRST is set to 1.
0	Reserved	Must kept as reserved value.

26. Infrared interface (IFRP)

26.1. Overview

Infrared interface (IFRP) is used to control infrared light LED, and send out infrared data to implement infrared ray remote control.

There is no register in this module, which is controlled by TIMER15 and TIMER16. The capacity of module's output high current can be improved by set the GPIO pin to Fast Mode.

26.2. Characteristics

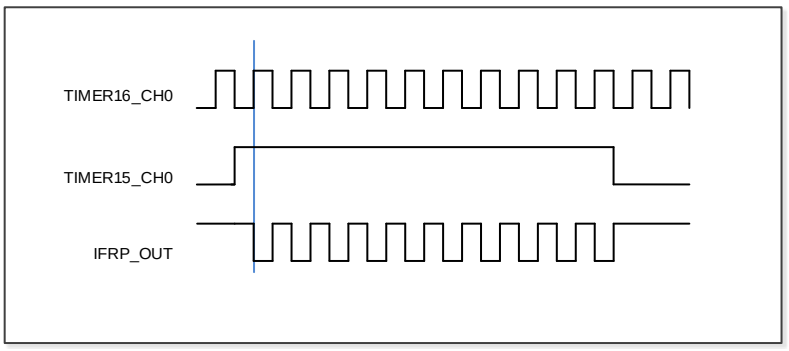
- The IFRP output signal is decided by TIMER15_CH0 and TIMER16_CH0
- To get correct infrared ray signal, TIMER15 should generate low frequence modulation envelope signal, and TIMER16 should generate high frequence carrier signal
- The IFRP output (PB9/PA13) can control LED interface by setting PB9FMPEN in SYSCFG_CFG0

26.3. Function overview

IFRP is a module which is able to integrate the output of TIMER15 and TIMER16 to generate an infrared ray signal.

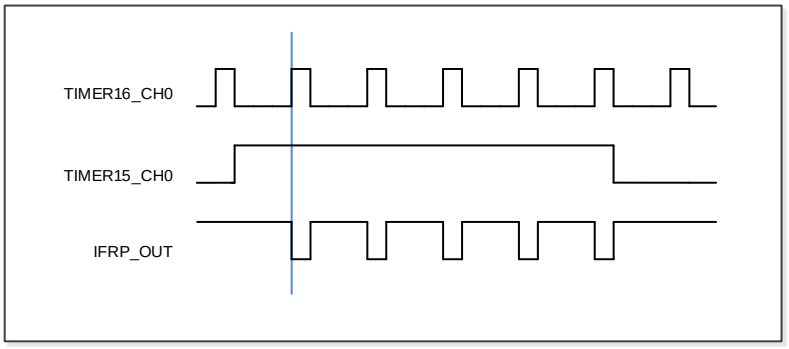
1. The TIMER15's CH0 is programed to generate the low frequence PWM signal which is the modulation envelope signal. The TIMER16's CH0 is programed to generate the high frquence PWM signal which is the carrier signal. And the channel need to be enabled before generating these signals.
2. Program the GPIO remap regisger and enable the PIN.

Figure 26-1. IFRP output timechart 1



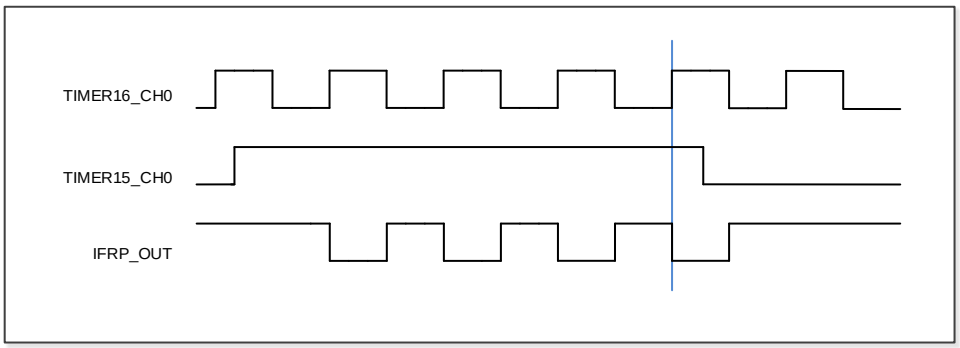
Note: IFRP_OUT has one APB clock delay from TIMER16_CH0.

Figure 26-2. IFRP output timechart 2



Note: Carrier (TIMER15_CH0)'s duty cycle can be changed, and IFRP_OUT has inverted relationship with TIMER16_CH0 when TIMER15_CH0 is high.

Figure 26-3. IFRP output timechart 3



Note: IFRP_OUT will keep the integrity of TIMER16_CH0, even if evelope signal (TIMER15_CH0) is no active.

27. Universal synchronous / asynchronous receiver / transmitter (USART)

27.1. 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_APBx, CK_AHB, CK_LXTAL or CK_IRC8M) to produces a dedicated wide range baudrate clock for the USART transmitter and receiver.

Besides the standard asynchronous receiver and transmitter mode, the USART implements several other types of serial data exchange modes, such as IrDA (infrared data association) SIR mode, smartcard mode, LIN (local interconnection network) mode, half-duplex mode and synchronous mode. It also supports multiprocessor communication mode, and hardware flow control protocol (CTS / RTS). The data frame can be transferred from LSB or MSB bit. The polarity of the TX / RX pins can be configured independently and flexibly.

All USARTs support DMA function for high-speed data communication.

27.2. 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 UCLK reprogramming.
- Programmable baud-rate generator allowing speed up to 27 MBits/s when the clock.frequency is 216 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), Transmit buffer empty (TBE), transfer complete (TC), Transmit FIFO not full(TFNF), Transmit FIFO empty(TFE), Transmit FIFO threshold reached(TFT).
  - Flags for error detection: overrun error (ORERR), noise error (NERR), frame error (FERR) and parity error (PERR).
  - Flag for hardware flow control: CTS changes (CTSF).
  - Flag for LIN mode: LIN break detected (LBDF).
  - Flag for multiprocessor communication: IDLE frame detected (IDLEF).
  - Flag for ModBus communication: address/character match (AMF) and receiver timeout (RTF).
  - Flags for smartcard block mode: end of block (EBF) and receiver timeout (RTF).
  - Wakeup from deep-sleep mode flag.
  - Interrupt occurs at these events when the corresponding interrupt enable bits are set.

While USART0 / USART1 / USART2 is fully implemented, UART3 / UART4 is only partially implemented with the following features not supported.

- Smartcard mode.
- IrDA SIR ENDEC block.
- LIN mode.
- Dual clock domain and wakeup from deep-sleep mode.
- Receiver timeout interrupt.
- ModBus communication.
- Synchronous mode.

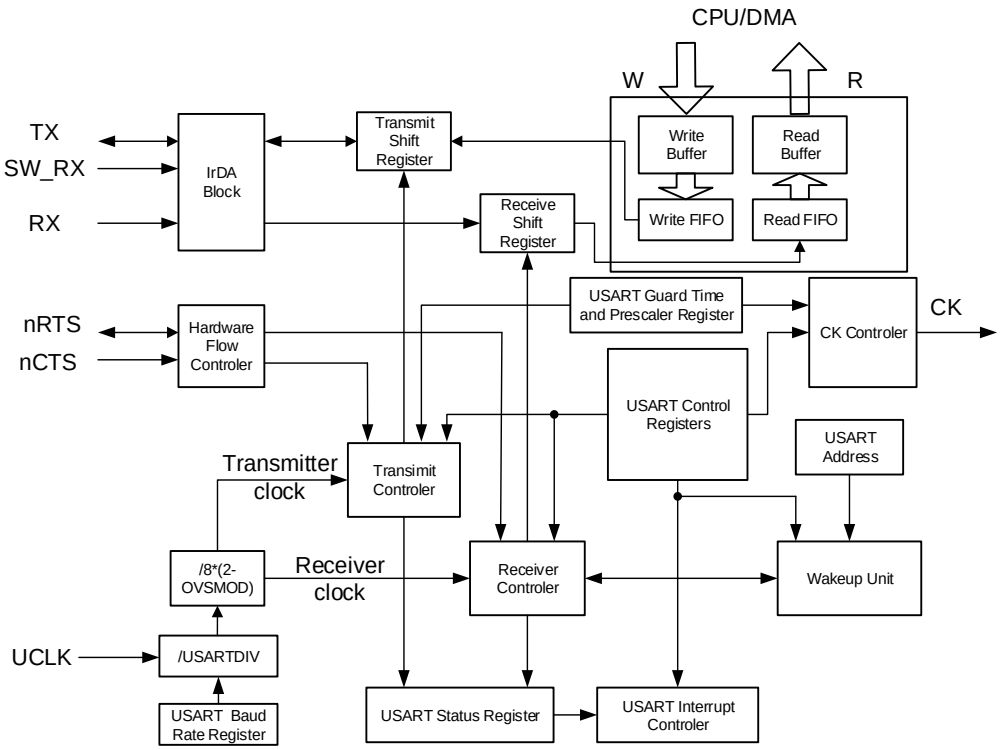
27.3. Function overview

The interface is externally connected to another device by the main pins listed in ***Table 27-1. Description of USART important pins***.

Table 27-1. 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 27-1. USART module block diagram

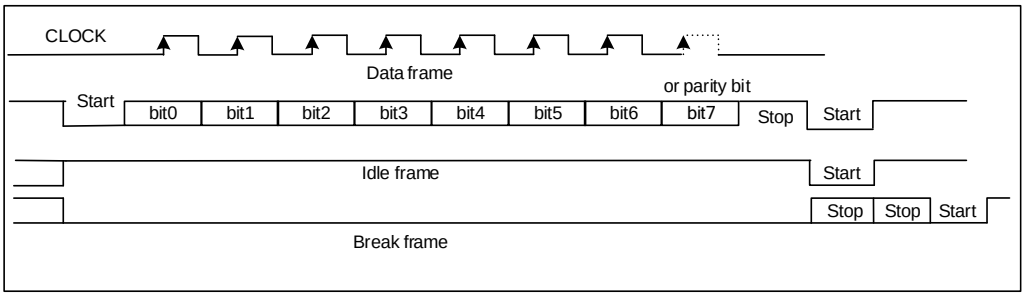


27.3.1. USART frame format

The USART frame starts with a start bit and ends up with a number of stop bits. The length of the data frame is configured by the 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 27-2. 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 27-2. Configuration of stop bits

STB[1:0]	stop bit length (bit)	usage description
00	1	Default value
01	0.5	Smartcard mode for receiving
10	2	Normal USART and single-wire modes
11	1.5	Smartcard mode for transmitting and receiving

In an idle frame, all the frame bits are logic 1. The frame length is equal to the normal USART frame.

The break frame structure is a number of low bits followed by the configured number of stop bits. The transfer speed of a USART frame depends on the frequency of the UCLK, the configuration of the baud rate generator and the oversampling mode.

27.3.2. Baud rate generation

The baud-rate divider is a 16-bit number which consists of a 12-bit integer and a 4-bit fractional part. The number formed by these two values is used by the baud rate generator to determine the bit period. Having a fractional baud-rate divider allows the USART to generate all the standard baud rates.

The baud-rate divider (USARTDIV) has the following relationship with the UCLK:

In case of oversampling by 16, the equation is:

$$\text{USARTDIV} = \frac{\text{UCLK}}{16 \times \text{Baud Rate}} \quad (27\text{-}1)$$

In case of oversampling by 8, the equation is:

$$\text{USARTDIV} = \frac{\text{UCLK}}{8 \times \text{Baud Rate}} \quad (27\text{-}2)$$

For example, when oversampled by 16:

1. Get USARTDIV by caculating the value of USART_BAUD:

If USART_BAUD = 0x21D, then INTDIV = 33 (0x21), FRADIV = 13 (0xD).
USARTDIV=33 + 13 / 16 = 33.81.

2. Get the value of USART_BAUD 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_BAUD = 0x1E6.

   Note: If the roundness of FRADIV is 16 (overflow), the carry must be added to the integer part.

27.3.3. USART transmitter

If the transmit enable bit (TEN) in USART_CTL0 register is set, when the transmit data buffer is not empty, the transmitter shifts out the transmit data frame through the TX pin. The polarity of the TX pin can be configured by the TINV bit in the USART_CTL1 register. Clock pulses can output through the CK pin.

After the TEN bit is set, an idle frame will be sent. The TEN bit should not be cleared while the transmission is ongoing.

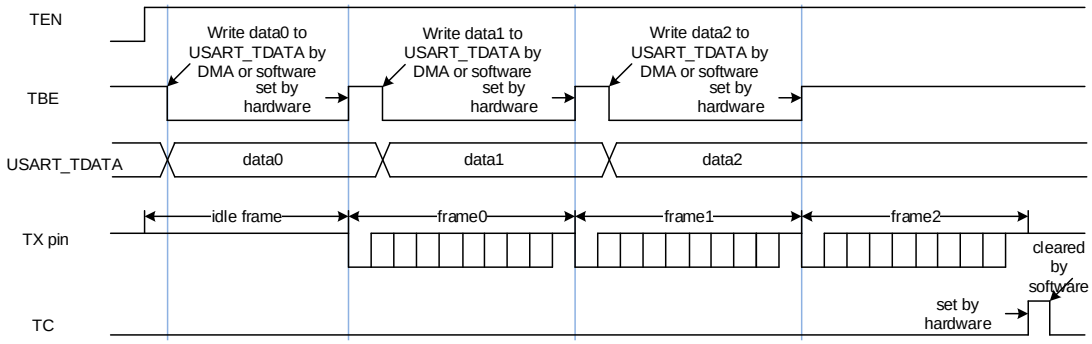
After power on, the TBE bit is high by default. Data can be written to the USART_TDATA when the TBE bit in the USART_STAT register is asserted. The TBE bit is cleared by writing USART_TDATA register and it is set by hardware after the data is put into the transmit shift register. If a data is written to the USART_TDATA register while a transmission is ongoing, it will be firstly stored in the transmit buffer, and transferred to the transmit shift register after the current transmission is done. If a data is written to the USART_TDATA register while no transmission is ongoing, the TBE bit will be cleared and set soon, because the data will be transferred to the transmit shift register immediately.

If a frame is transmitted and the TBE bit is asserted, the TC bit of the USART_STAT register will be set. An interrupt will be generated if the corresponding interrupt enable bit (TCIE) is set in the USART_CTL0 register.

The USART transmit procedure is shown in ***Figure 27-3. USART transmit procedure***. The software operating process is as follows:

1. Write the WL0 bit and WL1 bit in USART_CTL0 to set the data bits length.
2. Set the STB[1:0] bits in USART_CTL1 to configure the number of stop bits.
3. Enable DMA (DENT bit) in USART_CTL2 if multibuffer communication is selected.
4. Set the baud rate in USART_BAUD.
5. Set the UEN bit in USART_CTL0 to enable the USART.
6. Set the TEN bit in USART_CTL0.
7. Wait for the TBE being asserted.
8. Write the data to the USART_TDATA register.
9. Repeat step7-8 for each data, if DMA is not enabled.
10. Wait until TC=1 to finish.

Figure 27-3. 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. The TC bit can be cleared by set the TCC bit in USART_INTC register.

The break frame is sent when the SBKCMD bit is set, and SBKCMD bit is reset after the transmission.

27.3.4. USART receiver

After power on, the USART receiver can be enabled by the following procedure:

1. Write the WL0 bit and WL1 bit in USART_CTL0 to set the data bits length.
2. Set the STB[1:0] bits in USART_CTL1.
3. Enable DMA (DENR bit) in USART_CTL2 if multibuffer communication is selected.
4. Set the baud rate in USART_BAUD.
5. Set the UEN bit in USART_CTL0 to enable the USART.
6. Set the REN bit in USART_CTL0.

After being enabled, the receiver receives a bit stream after a valid start pulse has been detected. Detection on noisy error, parity error, frame error and overrun error is performed during the reception of a frame.

When a frame is received, the RBNE bit in USART_STAT is asserted, an interrupt is generated if the corresponding interrupt enable bit (RBNEIE) is set in the USART_CTL0 register. The status of the reception are stored in the USART_STAT register.

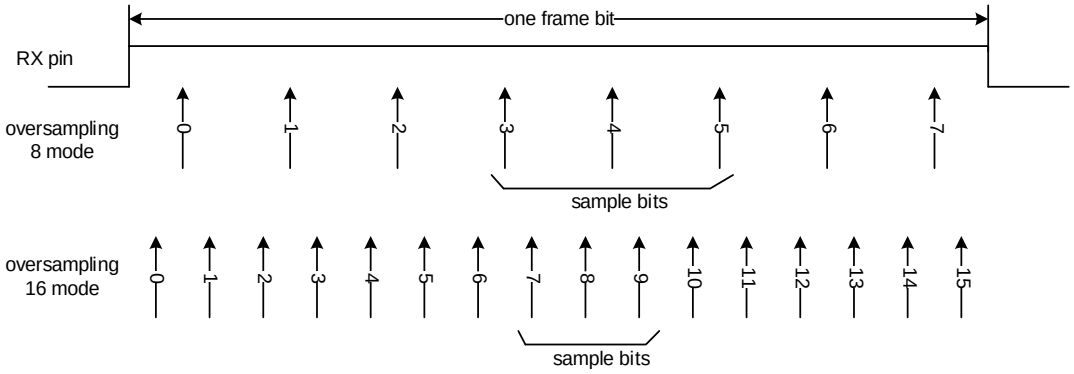
The software can get the received data by reading the USART_RDATA register directly, or through DMA. The RBNE bit is cleared by a read operation on the USART_RDATA register, whatever it is performed by software directly, or through DMA.

The REN bit should not be disabled when reception is ongoing, or the current frame will be lost.

By default, the receiver gets three samples to evaluate the value of a frame bit. If the oversampling 8 mode is enabled, the 3rd, 4th and 5th samples are used, while in the oversampling 16 mode, the 7th, 8th, and 9th samples are used. If two or more samples of a frame bit is 0, the frame bit is confirmed as a 0, else 1. If the value of the three samples of any bit are not the same, whatever it is a start bit, data bit, parity bit or stop bit, a noisy error

(NERR) status will be generated for the frame. An interrupt will be generated, If the ERRIE bit in USART_CTL2 register is set. If the OSB bit in USART_CTL2 register is set, the receiver gets only one sample to evaluate a bit value. In this situation, no noisy error will be detected.

Figure 27-4. Oversampling method of a receive frame bit (OSB = 0)



If the parity check function is enabled by setting the PCEN bit in the USART_CTL0 register, the receiver calculates the expected parity value while receiving a frame. The received parity bit will be compared with this expected value. If they are not the same, the parity error (PERR) bit in USART_STAT register will be set. An interrupt is generated, if the PERRIE bit in USART_CTL0 register is set.

If the RX pin is evaluated as 0 during a stop bit, the frame error (FERR) bit in USART_STAT register will be set. An interrupt is generated, If the ERRIE bit in USART_CTL2 register is set. According to the configuration of the stop bit, there are the following situations:

- 0.5 stop bit: When 0.5 stop bit, stop bit is not sampled.
- 1 stop bit: When 1 stop bit, sampling in the middle of stop bit.
- 1.5 stop bits: When 1.5 stop bits, the 1.5 stop bits are divided into 2 parts: the 0.5 stop bit part is not sampled and sampling in the middle of 1 stop bit.
- 2 stop bits: When 2 stop bits, if a frame error is detected during the first stop bit, the frame error flag is set, the second stop bit is not checked frame error. If no frame error is detected during the first stop bit, then continue to check the second stop bit for frame error.

When a frame is received, if the RBNE bit is not cleared yet, the last frame will not be stored in the receive data buffer. The overrun error (ORERR) bit in USART_STAT register will be set. An interrupt is generated, if the ERRIE bit in USART_CTL2 register is set, or if the RBNEIE is set.

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.

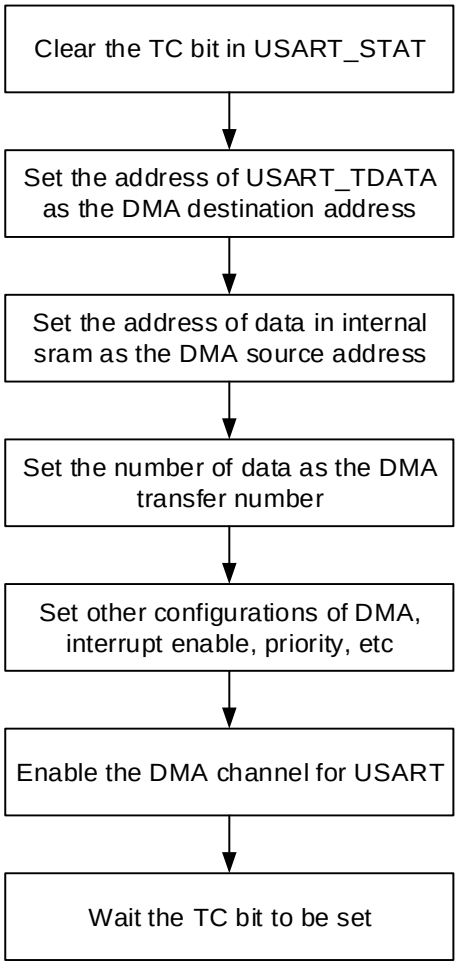
27.3.5. Use DMA for data buffer access

To reduce the burden of the processor, DMA can be used to access the transmitting and

receiving data buffer. The DENT bit in USART_CTL2 is used to enable the DMA transmission, and the DENR bit in USART_CTL2 is used to enable the DMA reception.

When DMA is used for USART transmission, DMA transfers data from internal SRAM to the transmit data buffer of the USART. The configuration step are shown in ***Figure 27-5. Configuration step when using DMA for USART transmission***.

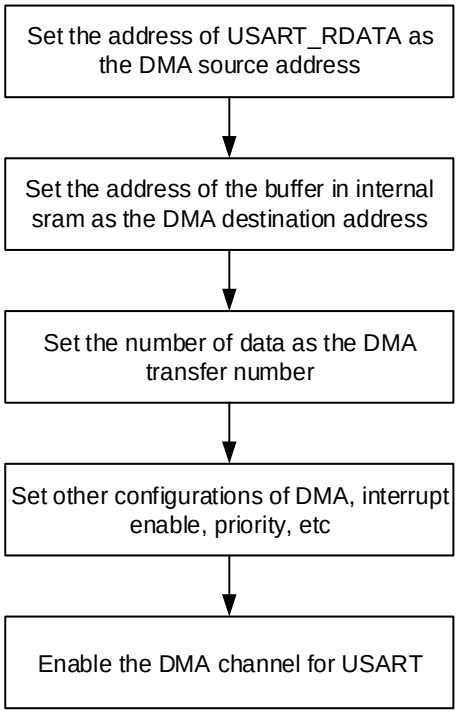
Figure 27-5. Configuration step when using DMA for USART transmission



After all of the data frames are transmitted, the TC bit in USART_STAT is set. An interrupt occurs if the TCIE bit in USART_CTL0 is set.

When DMA is used for USART reception, DMA transfers data from the receive data buffer of the USART to the internal SRAM. The configuration steps are shown in ***Figure 27-6. Configuration step when using DMA for USART reception***. If the ERRIE bit in USART_CTL2 is set, interrupts can be generated by the error status bits (FERR, ORERR and NERR) in USART_STAT.

Figure 27-6. Configuration step when using DMA for USART reception

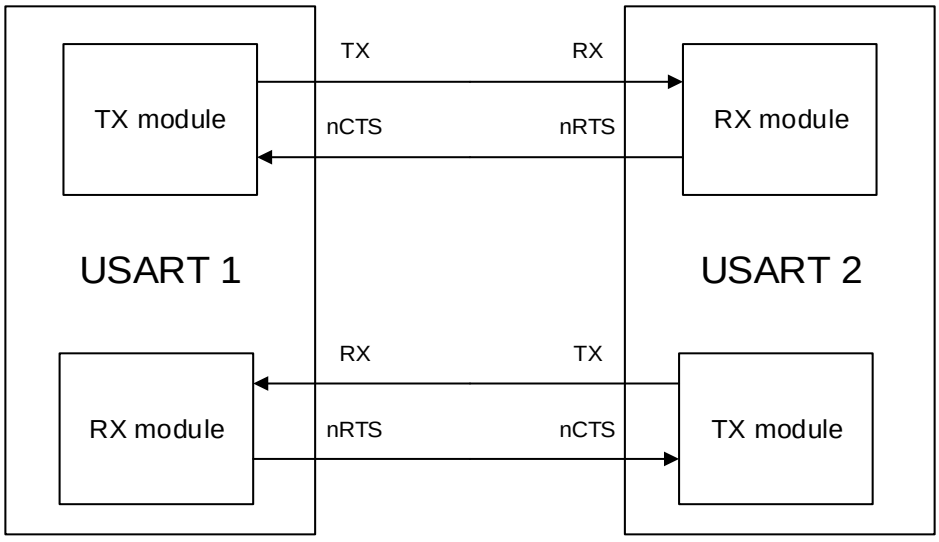


When the number of the data received by USART reaches the DMA transfer number, an end of transfer interrupt can be generated in the DMA module.

27.3.6. Hardware flow control

The hardware flow control function is realized by the nCTS and nRTS pins. The RTS flow control is enabled by writing '1' to the RTSEN bit in USART_CTL2 and the CTS flow control is enabled by writing '1' to the CTSEN bit in USART_CTL2.

Figure 27-7. Hardware flow control between two USARTs



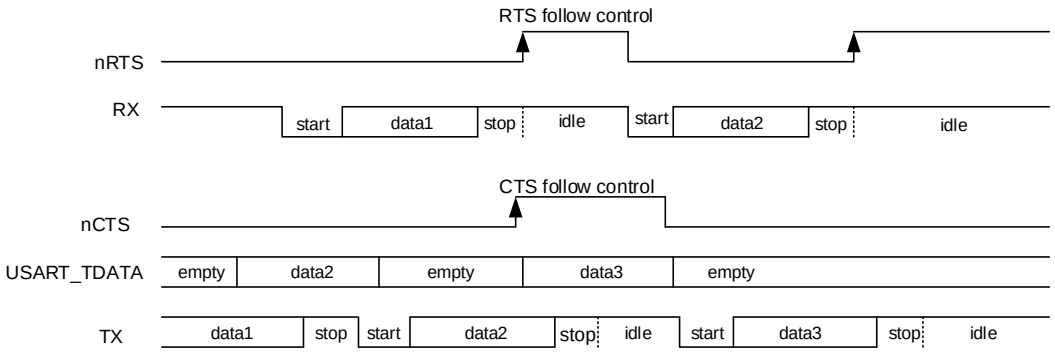
RTS flow control

The USART receiver outputs the nRTS, which reflects the status of the receive buffer. When data frame is received, the nRTS signal goes high to prevent the transmitter from sending next frame. The nRTS signal keeps high when the receive buffer is full.

CTS flow control

The USART transmitter monitors the nCTS input pin to decide whether a data frame can be transmitted. If the TBE bit in USART_STAT is '0' and the nCTS signal is low, the transmitter transmits the data frame. When the nCTS signal goes high during a transmission, the transmitter stops after the current transmission is accomplished.

Figure 27-8. Hardware flow control



RS485 Driver Enable

The driver enable feature, which is enabled by setting bit DEM in the USART_CTL2 control register, allows the user to activate the external transceiver control, through the DE (Driver Enable) signal. The assertion time, which is programmed using the DEA [4:0] bits field in the USART_CTL0 control register, is the time between the activation of the DE signal and the beginning of the START bit. The de-assertion time, which is programmed using the DED [4:0] bits field in the USART_CTL0 control register, is the time between the end of the last stop bit and the de-activation of the DE signal. The polarity of the DE signal can be configured using the DEP bit in the USART_CTL2 control register.

27.3.7. Multi-processor communication

In multiprocessor communication, several USARTs are connected as a network. It will be a big burden for a device to monitor all of the messages on the RX pin. To reduce the burden of a device, software can put an USART module into a mute mode by writing 1 to the MMCMD bit in USART_CMD register.

If a USART is in mute mode, all of the receive status bits cannot be set. The USART can also be wake up by hardware by one of the two methods: idle frame method and address match method.

The idle frame wake up method is selected by default. 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 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 adderss 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.

27.3.8. LIN mode

The local interconnection network mode is enabled by setting the LMEN bit in USART_CTL1. The CKEN, STB[1:0] bit in USART_CTL1 and the SCEN, HDEN, IREN bits in USART_CTL2 should be cleared in LIN mode.

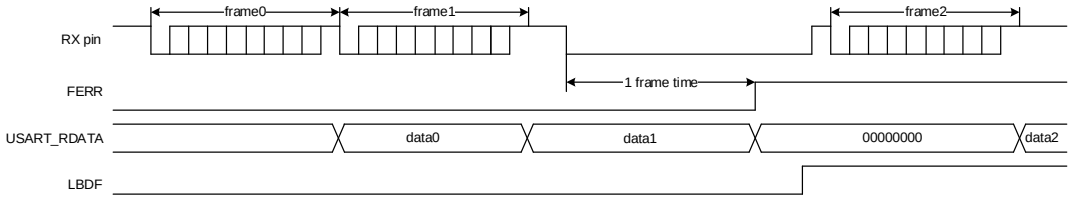
When transmitting a normal data frame, the transmission procedure is the same as the normal USART mode. The data bits length can only be 8. And the break frame is 13-bit ’0’, followed

by 1 stop bit.

The break detection function is totally independent of the normal USART receiver. So a break frame can be detected during the idle state or during a frame. The expected length of a break frame can be selected by configuring LBLEN in USART_CTL1. When the RX pin is detected at low state for a time that is equal to or longer than the expected break frame length (10 bits when LBLEN=0, or 11 bits when LBLEN=1), the LBDF bit in USART_STAT is set. An interrupt occurs if the LBDIE bit in USART_CTL1 is set.

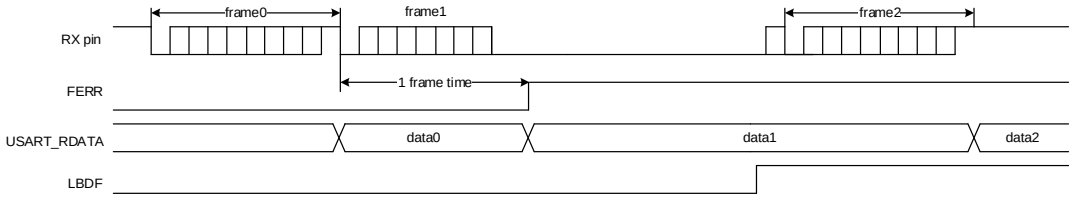
As shown in ***Figure 27-9. Break frame occurs during idle state***, if a break frame occurs during the idle state on the RX pin, the USART receiver will receive an all '0' frame, with an asserted FERR status.

Figure 27-9. Break frame occurs during idle state



As shown in ***Figure 27-10. 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 27-10. Break frame occurs during a frame



27.3.9. Synchronous mode

The USART can be used for full-duplex synchronous serial communications only in master mode, by setting the CKEN bit in USART_CTL1. The LMEN bit in USART_CTL1 and SCEN, HDEN, IREN bits in USART_CTL2 should be cleared in synchronous mode. The CK pin is the clock output of the synchronous USART transmitter, and can be only activated when the TEN bit is enabled. No clock pulse will be sent through the CK pin during the transmission of the start bit and stop bit. The CLEN bit in USART_CTL1 can be used to determine whether the clock is output or not during the last (address flag) bit transmission. The clock output is also not activated during idle and break frame sending. The CPH bit in USART_CTL1 can be used to determine whether data is captured on the first or the second clock edge. The CPL bit in USART_CTL1 can be used to configure the clock polarity in the USART synchronous idle state.

The CPL, CPH and CLEN bits in USART_CTL1 determine the waveform on the CK pin. Software can only change them when the USART is disabled (UEN = 0).

The clock is synchronized with the data transmitted. The receiver in synchronous mode samples the data on the transmitter clock without any oversampling.

Figure 27-11. Example of USART in synchronous mode

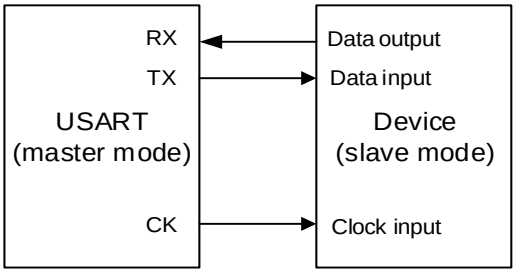
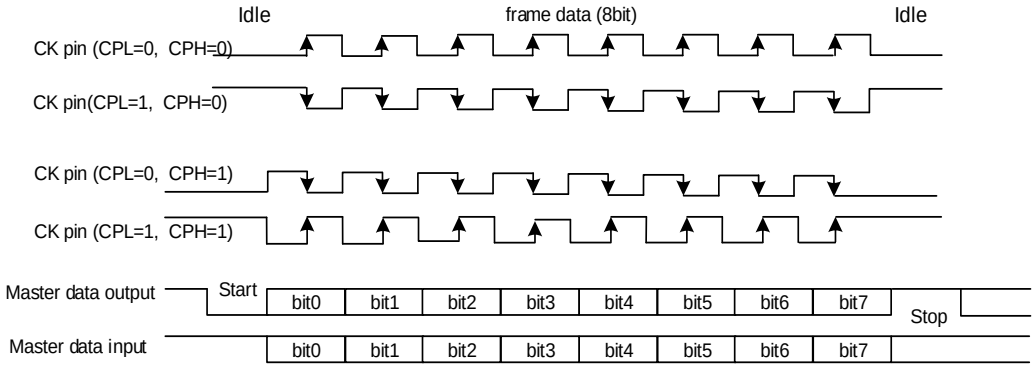


Figure 27-12. 8-bit format USART synchronous waveform (CLEN = 1)

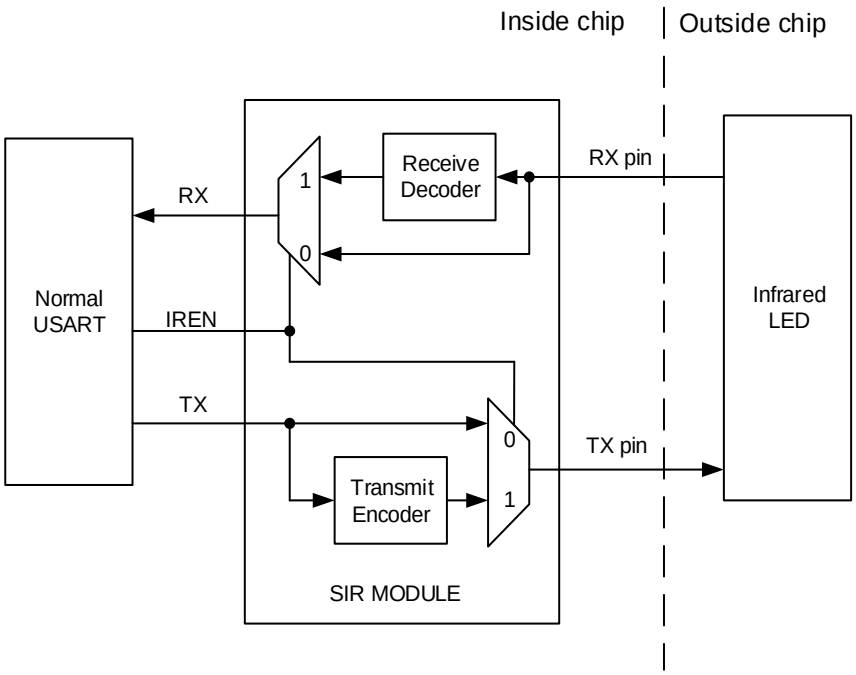


27.3.10. IrDA SIR ENDEC mode

The IrDA mode is enabled by setting the IREN bit in USART_CTL2. The LMEN, STB[1:0], CKEN bits in USART_CTL1 and HDEN, SCEN bits in USART_CTL2 should be cleared in IrDA mode.

In IrDA mode, the USART transmission data frame is modulated in the SIR transmit encoder and transmitted to the infrared LED through the TX pin. The SIR receive decoder receives the modulated signal from the infrared LED through the RX pin, and puts the demodulated data frame to the USART receiver. The baud rate should not be larger than 115200 for the encoder.

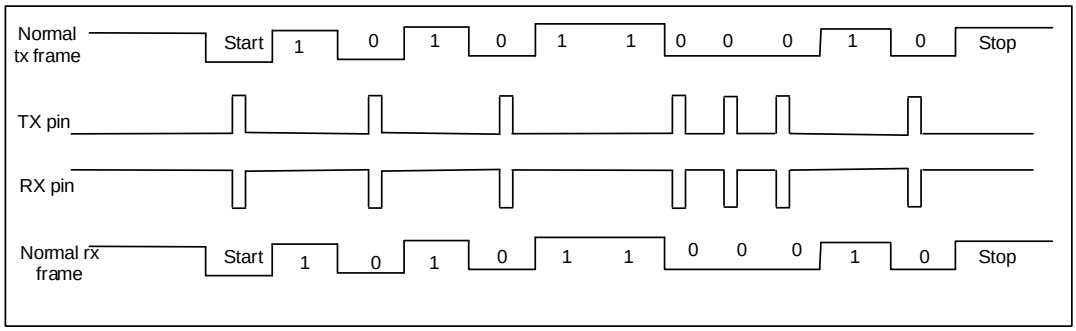
Figure 27-13. IrDA SIR ENDEC module



In IrDA mode, the polarity of the TX and RX pins is different. The TX pin is usually at low state, while the RX pin is usually at high state. The IrDA pins keep stable to represent the logic ‘1’, while an infrared light pulse on the IrDA pins (a Return to Zero signal) represents the logic ‘0’. The pulse width should be 3 / 16 of a bit period. The IrDA could not detect any pulse if the pulse width is less than 1 PSC clock. While it can detect a pulse by chance if the pulse width is greater than 1 but smaller than 2 times of PSC clock.

Because the IrDA is a half-duplex protocol, the transmission and the reception should not be carried out at the same time in the IrDA SIR ENDEC block.

Figure 27-14. IrDA data modulation



The SIR sub module can work in low power mode by setting the IRLP bit in USART_CTL2. The transmit encoder is driven by a low speed clock, which is divided from the PCLK. The division ratio is configured by the PSC[7:0] bits in USART_GP register. The pulse width on the TX pin is 3 cycles of this low speed period. The receiver decoder works in the same manner as the normal IrDA mode.

27.3.11. Half-duplex communication mode

The half-duplex communication mode is enabled by setting the HDEN bit in USART_CTL2. The LMEN, CKEN bits in USART_CTL1 and SCEN, IREN bits in USART_CTL2 should be cleared in half-duplex communication mode.

Only one wire is used in half-duplex mode. The TX and RX pins are connected together internally, and the RX pin is no longer used. The TX pin should be configured as open drain mode and communication conflicts are handled by software

27.3.12. Smartcard (ISO7816-3) mode

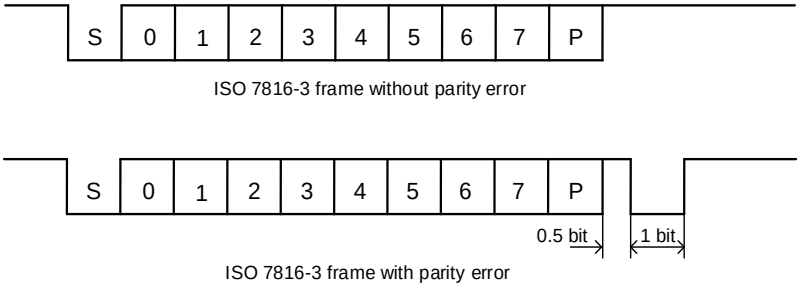
The smartcard mode is an asynchronous mode, which is designed to support the ISO7816-3 protocol. Both the character (T = 0) mode and the block (T = 1) mode are supported. The smartcard mode is enabled by setting the SCEN bit in USART_CTL2. The LMEN bit in USART_CTL1 and HDEN, IREN bits in USART_CTL2 should be reset in smartcard mode.

A clock is provided to the smartcard if the CKEN bit is set. The clock can be divided for other use.

The frame consists of 1 start bit, 9 data bits (1 parity bit included) and 1.5 stop bits.

The smartcard mode is a half-duplex communication protocol. When connected to a smartcard, the TX pin must be configured as open drain mode, and drives a bidirectional line that is also driven by the smartcard.

Figure 27-15. ISO7816-3 frame format



Character (T = 0) mode

Compared to the timing in normal operation, the transmission time from transmit shift register to the TX pin is delayed by half baud clock, and the TC flag assertion time is delayed by a guard time that is configured by the GUAT[7:0] bits in USART_GP. In Smartcard mode, the internal guard time counter starts counting up after the stop bits of the last data frame, and the GUAT[7:0] bits should be configured as the character guard time (CGT) in ISO7816-3 protocol minus 12. The TC status is forced reset while the guard time counter is counting up. When the counter reaches the programmed value TC is asserted high.

During USART transmission, if a parity error event is detected, the smartcard may NACK the current frame by pulling down the TX pin during the last 1 bit time of the stop bits. The USART

can automatically resend data according to the protocol for SCRTNUM times. An interframe gap of 2.5 bits time will be inserted before the start of a resented frame. At the end of the last repeated character the TC bit is set immediately without guard time. The USART will stop transmitting and assert the frame error status if it still receives the NACK signal after the programmed number of retries. The USART will not take the NACK signal as the start bit.

During USART reception, if the parity error is detected in the current frame, the TX pin is pulled low during the last 1 bit time of the stop bits. This signal is the NACK signal to smartcard. Then a frame error occurs in smartcard side. The RBNE / receive DMA request is not activated if the received character is erroneous. According to the protocol, the smartcard can resend the data. The USART stops transmitting the NACK and the error is regarded as a parity error if the received character is still erroneous after the maximum number of retries which is specified in the SCRTNUM bit field. The NACK signal is enabled by setting the NKEN bit in USART_CTL2.

The idle frame and break frame are not supported in the Smartcard mode.

Block (T = 1) mode

In block (T = 1) mode, the NKEN bit in the USART_CTL2 register should be cleared to deactivate the NACK transmission.

When requesting a read from the smartcard, the RT[23:0] bits in USART_RT register should be programmed with the BWT (block wait time) - 11 value and RBNEIE must be set. A timeout interrupt will be generated, if no answer is received from the card before the expiration of this period. If the first character is received before the expiration of the period, it is signaled by the RBNE interrupt. If DMA is used to read from the smartcard in block mode, the DMA must be enabled only after the first character is received.

In order to allow the automatic check of the maximum wait time between two consecutive characters, the USART_R T register must be programmed to the CWT (character wait time) - 11 value, which is expressed in baudtime units, after the reception of the first character (RBNE interrupt). The USART signals to the software through the RT flag and interrupt (when RTIE bit is set), if the smartcard doesn’t send a new character in less than the CWT period after the end of the previous character.

The USART uses a block length counter, which is reset when the USART is transmitting (TBE = 0), to count the number of received characters. The length of the block, which must be programmed in the BL[7:0] bits in the USART_RT register, is received from the smartcard in the third byte of the block (prologue field). This register field must be programmed to the minimum value (0x0), before the start of the block, when using DMA mode. With this value, an interrupt is generated after the 4th received character. The software must read the third byte as block length from the receive buffer.

In interrupt driven receive mode, the length of the block may be checked by software or by programming the BL value. However, before the start of the block, the maximum value of BL (0xFF) may be programmed. The real value will be programmed after the reception of the

third character.

The total block length (including prologue, epilogue and information fields) equals BL+4. The end of the block is signaled to the software through the EBF flag and interrupt (when EBIE bit is set). The RT interrupt may occur in case of an error in the block length.

Direct and inverse convention

The smartcard protocol defines two conventions: direct and inverse.

The direct convention is defined as: LSB first, logical bit value of 1 corresponds to H state of the line and parity is even. In this case, the following control bits must be programmed: MSBF = 0, DINV = 0.

The inverse convention is defined as: MSB first, logical bit value 1 corresponds to an L state on the signal line and parity is even. In this case, the following control bits must be programmed: MSBF = 1, DINV = 1.

27.3.13. ModBus communication

The USART offers basic support for the implementation of ModBus/RTU and ModBus/ASCII protocols by implementing an end of block detection.

In the ModBus/RTU mode, the end of one block is recognized by an idle line for more than 2 characters time. This function is implemented through the programmable timeout function.

To detect the idle line, the RTEN bit in the USART_CTL1 register and the RTIE in the USART_CTL0 register must be set. The USART_RT register must be set to the value corresponding to a timeout of 2 characters time. After the last stop bit is received, when the receive line is idle for this duration, an interrupt will be generated, informing the software that the current block reception is completed.

In the ModBus / ASCII mode, the end of a block is recognized by a specific (CR / LF) character sequence. The USART manages this mechanism using the character match function by programming the LF ASCII code in the ADDR0 or ADDR1 field and activating the address match interrupt (AMIE0 = 1 or AMIE1 = 1). When a LF has been received or can check the CR / LF in the DMA buffer, the software will be informed.

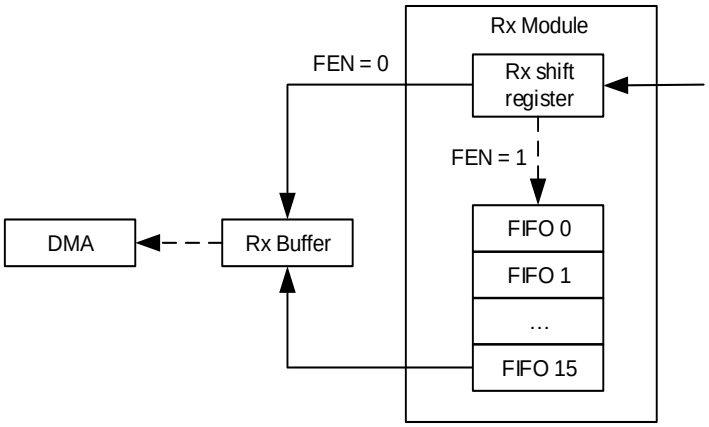
27.3.14. Receive / Transmitter FIFO

Receive FIFO

The receive FIFO can be enabled by setting the FEN bit of the USART_FCS 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 27-16. USART receive FIFO structure

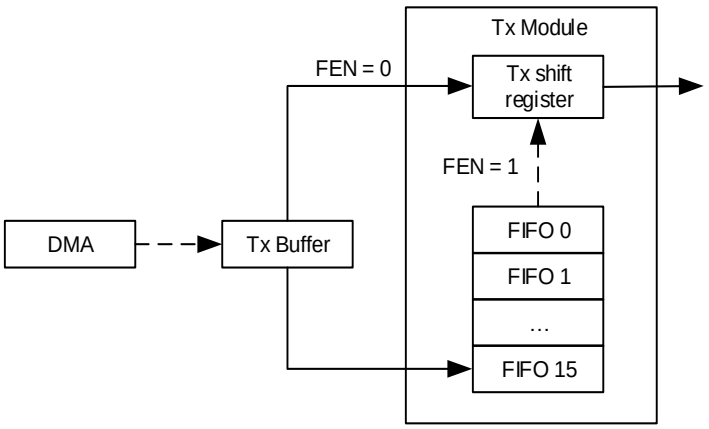


If the software read receive data buffer in the routing of the RBNE interrupt, the RBNEIE bit should be reset at the beginning of the routing and set after all of the receive data is read out. The PERR / NERR / FERR / EBF / ABDE / ABDF flags should be cleared before reading a receive data out.

Transmit FIFO

The Transmitter FIFO can be enabled by setting the FEN bit of the USART_FCS register Up to 16 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 TFT flag will be set when the 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 27-17. USART transmitter FIFO structure



27.3.15. Wakeup from deep-sleep mode

The USART is able to wake up the MCU from deep-sleep mode by the standard RBNE

interrupt or the WUM interrupt.

The UESM bit must be set and the USART clock must be set to CK_IRC8M or CK_LXTAL (refer to the reset and clock unit RCU section).

When using the standard RBNE interrupt, the RBNEIE bit must be set before entering deep-sleep mode.

When using the WUIE interrupt, the source of WUIE interrupt may be selected through the WUM bit fields.

DMA must be disabled before entering deep-sleep mode. Before entering Deep-sleep mode, software must check that the USART is not performing a transfer, by checking the BSY flag in the USART_STAT register. The REA bit must be checked to ensure the USART is actually enabled.

When the wakeup event is detected, the WUF flag is set by hardware and a wakeup interrupt is generated if the WUIE bit is set, independently of whether the MCU is in deep-Sleep or normal mode.

Note: AMEN0 or AMEN1 must be set when using address match method to wakeup MCU from Deep-Sleep mode.

27.3.16. USART interrupts

The USART interrupt events and flags are listed in ***Table 27-3. USART interrupt requests***.

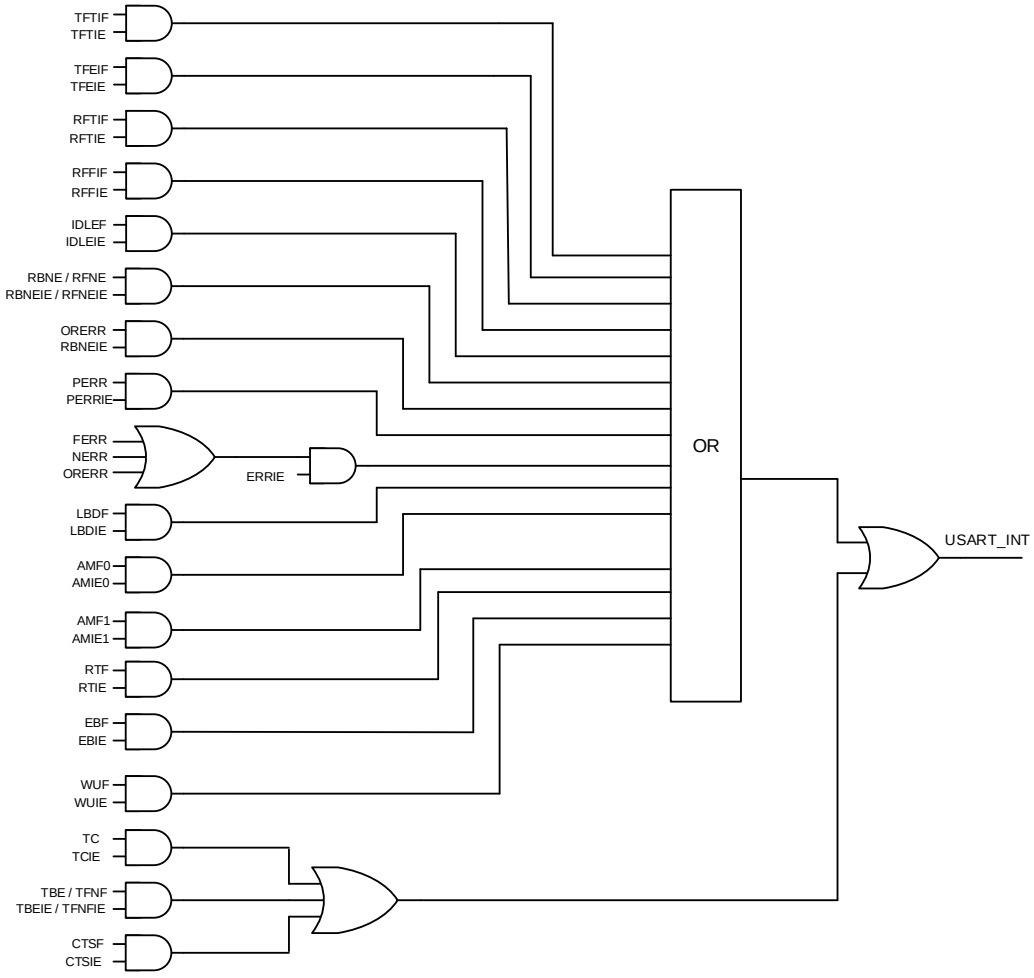
Table 27-3. USART interrupt requests

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

Interrupt event	Event flag	Enable Control bit
Reception errors (noise flag, overrun error, framing error)	NERR or ORERR or FERR	ERRIE
ADDR0 match	AMF0	AMIE0
ADDR1 match	AMF1	AMIE1
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 27-18. USART interrupt mapping diagram



27.4. 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

27.4.1. Control register 0 (USART_CTL0)

Address offset: 0x00
Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
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: Disable ADDR1 character match interrupt 1: Enable ADDR1 character match interrupt
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).
27	EBIE	End of block interrupt enable 0: Disable end of block interrupt 1: Enable end of block interrupt This bit is reserved in UART3 / UART4.
26	RTIE	Receiver timeout interrupt enable

		0: Receiver timeout interrupt is disabled 1: Receiver timeout interrupt is enabled This bit is reserved in UART3 / UART4.
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 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.
This bit is reserved in UART3 / UART4.

0	UEN	USART enable 0: USART prescaler and outputs disabled 1: USART prescaler and outputs enabled

27.4.2. Control register 1 (USART_CTL1)

Address offset: 0x04
Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
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			AMEN0
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 This bit is reserved in UART3 / UART4.
22:20	Reserved	Must be kept at reset value.
19	MSBF	Most significant bit first 0: Data is transmitted / received with the LSB first 1: Data is transmitted / received with the MSB first This bit field cannot be written when the USART is enabled (UEN = 1).
18	DINV	Data bit level inversion

		0: Data bit signal values are not inverted 1: Data bit signal values are inverted This bit field cannot be written when the USART is enabled (UEN = 1).
17	TINV	TX pin level inversion 0: TX pin signal values are not inverted 1: TX pin signal values are inverted This bit field cannot be written when the USART is enabled (UEN = 1).
16	RINV	RX pin level inversion 0: RX pin signal values are not inverted 1: RX pin signal values are inverted This bit field cannot be written when the USART is enabled (UEN = 1).
15	STRP	Swap TX / RX pins 0: The TX and RX pins functions are not swapped 1: The TX and RX pins functions are swapped This bit field cannot be written when the USART is enabled (UEN = 1).
14	LMEN	LIN mode enable 0: LIN mode disabled 1: LIN mode enabled This bit field cannot be written when the USART is enabled (UEN = 1). This bit is reserved in UART3 / UART4.
13:12	STB[1:0]	STOP bits length 00: 1 Stop bit 01: 0.5 Stop bits 10: 2 Stop bits 11: 1.5 Stop bits This bit field cannot be written when the USART is enabled (UEN = 1). **Note:** 0.5 and 1.5 stop bits are not used in UART3 / UART4.
11	CKEN	CK pin enable 0: CK pin disabled 1: CK pin enabled This bit field cannot be written when the USART is enabled (UEN = 1). This bit is reserved in UART3 / UART4.
10	CPL	Clock polarity 0: Steady low value on CK pin outside transmission window in synchronous mode 1: Steady high value on CK pin outside transmission window in synchronous mode This bit field cannot be written when the USART is enabled (UEN = 1).
9	CPH	Clock phase 0: The first clock transition is the first data capture edge in synchronous mode 1: The second clock transition is the first data capture edge in synchronous mode

		This bit field cannot be written when the USART is enabled (UEN = 1).
8	CLEN	CK length 0: The clock pulse of the last data bit (MSB) is not output to the CK pin in synchronous mode 1: The clock pulse of the last data bit (MSB) is output to the CK pin in synchronous mode This bit field cannot be written when the USART is enabled (UEN = 1)
7	Reserved	Must be kept at reset value.
6	LBDIE	LIN break detection interrupt enable 0: LIN break detection interrupt is disabled 1: An interrupt will occur whenever the LBDF bit is set in USART_STAT This bit is reserved in UART3 / UART4.
5	LBLEN	LIN break frame length 0: 10 bit break detection 1: 11 bit break detection This bit field cannot be written when the USART is enabled (UEN = 1). This bit is reserved in UART3 / UART4.
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).
3:1	Reserved	Must be kept at reset value.
0	AMEN0	Address 0 match mode enable 0: Address 0 match mode disable 1: Address 0 match mode enable.

27.4.3. Control register 2 (USART_CTL2)

Address offset: 0x08
Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
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 This bit is reserved in UART3 / UART4.
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). This bit is reserved in UART3 / UART4.
19:17	SCRTNUM[2:0]	Smartcard auto-retry number In smartcard mode, these bits specify the number of retries in transmission and reception. In transmission mode, a transmission error (FERR bit set) will occur after this

		number of automatic retransmission retries. In reception mode, reception error (RBNE and PERR bits set) will occur after this number or erroneous reception trials. When these bits are configured as 0x0, there will be no automatic retransmission in transmit mode. This bit field is only can be cleared to 0 when the USART is enabled (UEN = 1), to stop retransmission. This bit is reserved in UART3 / UART4.
16	AMEN1	Address 1 match mode enable 0: Address 1 match mode disable 1: Address 1 match mode enable
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). This bit is reserved in UART3 / UART4.
4	NKEN	NACK enable in Smartcard mode 0: Disable NACK transmission when parity error 1: Enable NACK transmission when parity error This bit field cannot be written when the USART is enabled (UEN = 1). This bit is reserved in UART3 / UART4.
3	HDEN	Half-duplex enable 0: Half duplex mode is disabled 1: Half duplex mode is enabled This bit field cannot be written when the USART is enabled (UEN = 1).
2	IRLP	IrDA low-power

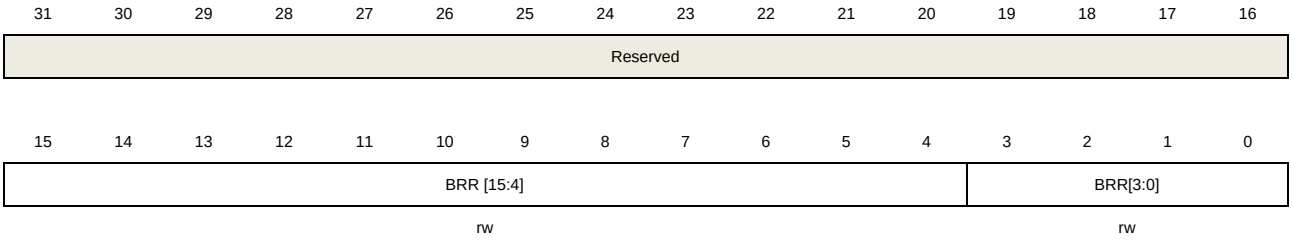
		0: Normal mode 1: Low-power mode This bit field cannot be written when the USART is enabled (UEN = 1).
1	IREN	IrDA mode enable 0: IrDA disabled 1: IrDA enabled This bit field cannot be written when the USART is enabled (UEN = 1). This bit is reserved in UART3 / UART4.
0	ERRIE	Error interrupt enable 0: 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

27.4.4. Baud rate generator register (USART_BAUD)

Address offset: 0x0C
Reset value: 0x0000 0000

This register cannot be written when the USART is enabled (UEN=1).

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



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:4	BRR[15:4]	Integer of baud-rate divider INTDIV = BRR[15:4]
3:0	BRR [3:0]	Fraction of baud-rate divider If OVSMOD = 0, FRADIV = BRR [3:0]; If OVSMOD = 1, FRADIV = BRR [2:0], BRR [3] must be reset.

27.4.5. Prescaler and guard time configuration register (USART_GP)

Address offset: 0x10
Reset value: 0x0000 0000

This register cannot be written when the USART is enabled (UEN = 1).

This register is reserved in UART3 / UART4.

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

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

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

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:8	GUAT[7:0]	Guard time value in smartcard mode This bit field cannot be written when the USART is enabled (UEN=1).
7:0	PSC[7:0]	Prescaler value for dividing the system clock In IrDA Low-power mode, the division factor is the prescaler value. 00000000: Reserved - do not program this value 00000001: divides the source clock by 1 00000010: divides the source clock by 2 ... In IrDA normal mode, 00000001: can be set this value only In smartcard mode, the prescaler value for dividing the system clock is stored in PSC[4:0] bits. And the bits of PSC[7:5] must be kept at reset value. The division factor is twice as the prescaler value. 00000: Reserved - do not program this value 00001: divides the source clock by 2 00010: divides the source clock by 4 00011: divides the source clock by 6 ... This bit field cannot be written when the USART is enabled (UEN=1).

27.4.6. Receiver timeout register (USART_RT)

Address offset: 0x14
Reset value: 0x0000 0000

This bit is reserved in UART3 / UART4.

This register 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 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.

27.4.7. Command register (USART_CMD)

Address offset: 0x18
Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											TXFCMD	RXFCMD	MMCMD	SBKCMD	Reserved
											w	w	w	w	

Bits	Fields	Descriptions
31:5	Reserved	Must be kept at reset value.
4	TXFCMD	Transmit data flush request

		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_FCS. **Note:**When FIFO is enabled, druing 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_FCS.
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.

27.4.8. Status register (USART_STAT)

Address offset: 0x1C
Reset value: 0x0000 00C0

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									REA	TEA	WUF	RWU	SBF	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 flag 0: No wakeup from deep-sleep mode 1: Wakeup from deep-sleep mode. An interrupt is generated if WUFIE = 1 in the USART_CTL2 register and the MCU is in Deep-sleep mode. This bit is set by hardware when a wakeup event, which is defined by the WUM bit field, is detected. Cleared by writing a 1 to the WUC in the USART_INTC register. This bit can also be cleared when UESM is cleared. This bit is reserved in UART3 / UART4.
19	RWU	Receiver wakeup from mute mode This bit is used to indicate if the USART is in mute mode. 0: Receiver in active mode 1: Receiver in mute mode It is cleared/set by hardware when a wakeup/mute sequence (address or IDLEIE) is recognized, which is selected by the WM bit in the USART_CTL0 register. This bit can only be set by writing 1 to the MMCMD bit in the USART_CMD register when wakeup on IDLEIE mode is selected.
18	SBF	Send break flag 0: No break character is transmitted 1: Break character will be transmitted This bit indicates that a send break character was requested. Set by software, by writing 1 to the SBKCMD bit in the USART_CMD register. Cleared by hardware during the stop bit of break transmission.
17	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. This bit is reserved in UART3 / UART4.
11	RTF	Receiver timeout flag 0: Timeout value not reached 1: Timeout value reached without any data reception. An interrupt is generated if RTIE bit in the USART_CTL1 register is set. Set by hardware when the RT value, programmed in the USART_RT register has lapsed without any communication. Cleared by writing 1 to RTC bit in USART_INTC register. The timeout corresponds to the CWT or BWT timings in smartcard mode. This bit is reserved in UART3 / UART4.
10	CTS	CTS level This bit equals to the inverted level of the nCTS input pin. 0: nCTS input pin is in high level 1: nCTS input pin is in low level
9	CTSF	CTS change flag 0: No change occurred on the nCTS status line 1: A change occurred on the nCTS status line. An interrupt will occur if the CTSIE bit is set in USART_CTL2 Set by hardware when the nCTS input toggles. Cleared by writing 1 to CTSC bit in USART_INTC register.
8	LBDF	LIN break detected flag 0: LIN Break is not detected 1: LIN Break is detected. An interrupt will occur if the LBDIE bit is set in USART_CTL1 Set by hardware when the LIN break is detected. Cleared by writing 1 to LBDC bit in USART_INTC register. This bit is reserved in UART3 / UART4.

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_FCS 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_FCS 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.

27.4.9. Interrupt status clear register (USART_INTC)

Address offset: 0x20

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											WUC	Reserved		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. This bit is reserved in UART3 / UART4.
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. This bit is reserved in UART3 / UART4.
11	RTC	Receiver timeout clear Writing 1 to this bit clears the RTF flag in the USART_STAT register. This bit is reserved in UART3 / UART4.
10	Reserved	Must be kept at reset value.
9	CTSC	CTS change clear

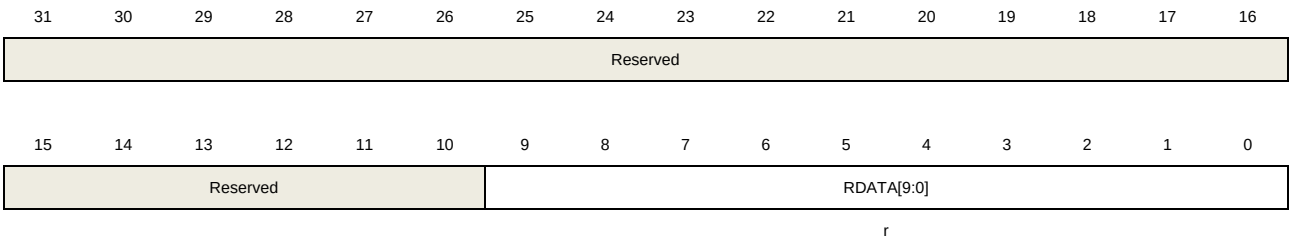
		Writing 1 to this bit clears the CTSF bit in the USART_STAT register.
8	LBDC	LIN break detected clear Writing 1 to this bit clears the LBDF flag in the USART_STAT register. This bit is reserved in UART3 / UART4.
7	Reserved	Must be kept at reset value.
6	TCC	Transmission complete clear Writing 1 to this 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.

27.4.10. Receive data register (USART_RDATA)

Address offset: 0x24

Reset value: 0xXXXX XXXX

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



Bits	Fields	Descriptions
31:10	Reserved	Must be kept at reset value.
9:0	RDATA[9:0]	Receive data value The received data character is contained in these bits. The value read in the MSB (bit6, bit7, bit8 or bit9 depending on the data length) will be the received parity bit, if receiving with the parity is enabled (PCEN bit set to 1 in

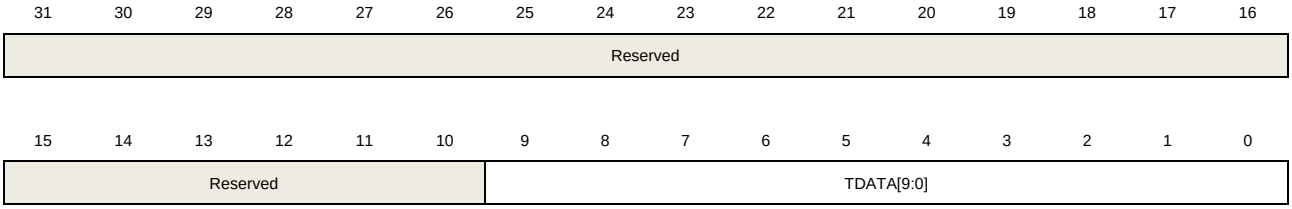
the USART_CTL0 register).

27.4.11. Transmit data register (USART_TDATA)

Address offset: 0x28

Reset value: 0xXXXX XXXX

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



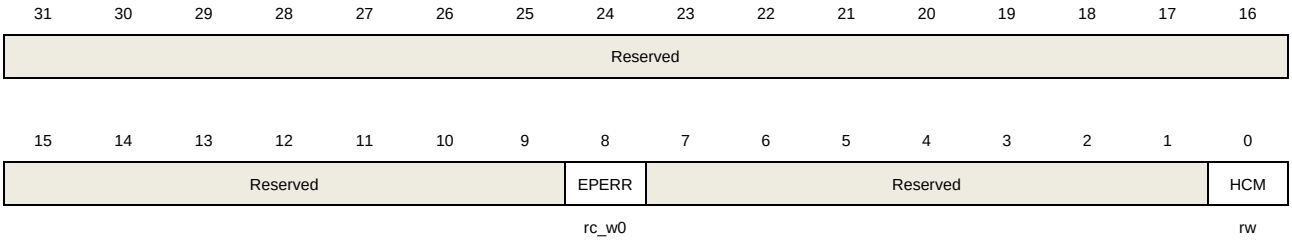
Bits	Fields	Descriptions
31:10	Reserved	Must be kept at reset value
9:0	TDATA[9:0]	Transmit data value The transmit data character is contained in these bits. The value written in the MSB (bit6, bit7, bit8 or bit9 depending on the data length) will be replaced by the parity, when transmitting with the 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.

27.4.12. USART coherence control register (USART_CHC)

Address offset: 0xC0

Reset value: 0x0000 0000

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



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.

27.4.13. USART FIFO control and status register (USART_RFCS)

Address offset: 0xD0

Reset value: 0x0300 0400

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TFEIE	Reserved	TFTIE	Reserved	RFTIE	TFEC	TFTIF	TFEIF	Reserved	RFTIF	TFTCFG[2:0]			RFTCFG[2:0]		
rw		rw		rw	rw	r	r		r	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 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: Transmit FIFO does not reach the programmed threshold 1: Transmit FIFO reached the programmed threshold
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: Transmit FIFO does not reach the programmed threshold 1: Transmit FIFO reached the programmed threshold. An interrupt will occur if the TFTIE bit is set. Set by hardware when the 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 This bit is reserved in UART3 / UART4.

28. Inter-integrated circuit interface (I2C)

28.1. 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.

28.2. 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 addresses, 1 with configurable mask).
- Programmable setup time and hold time.
- Multi-master capability.
- Supports standard mode (up to 100 kHz) and fast mode (up to 400 kHz) and fast mode plus (up to 1MHz, this mode must be enabled in SYSCFG_CFG1).
- 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.
- Wakeup from sleep mode and Deep-sleep mode on I2C address match.
- Independent clock from PCLK.

28.3. Function overview

Figure 28-1. I2C module block diagram below provides details on the internal configuration of the I2C interface.

Figure 28-1. I2C module block diagram

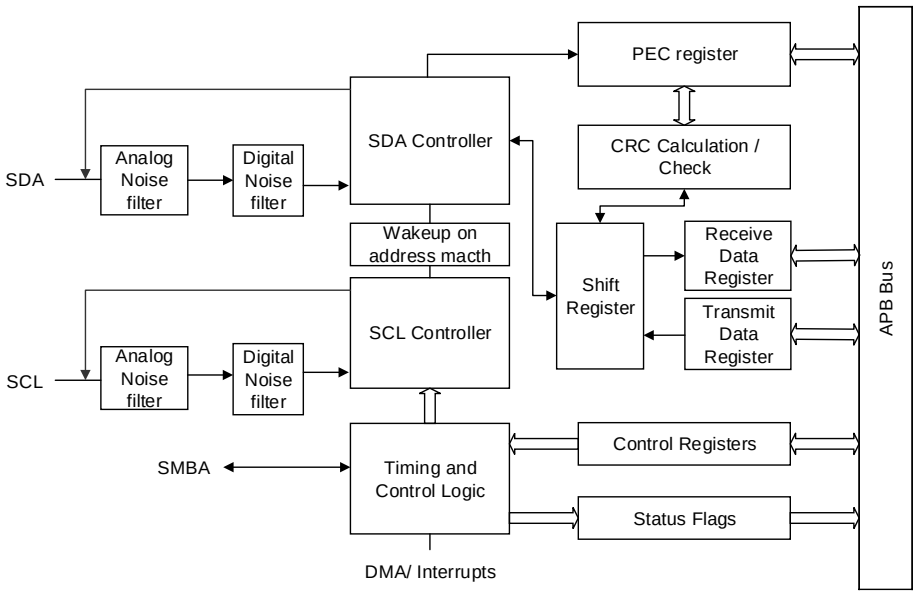


Table 28-1. Definition of I2C-bus terminology (refer to the I2C specification of Philips semiconductors)

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

28.3.1. Clock requirements

The I2C clock is independent of the PCLK frequency, so that the I2C can be operated independently.

This I2C clock (I2CCLK) can be selected from the following three clock sources:

- CK_APB1: APB1 clock (default value)
- CK_SYS: System clock
- CK_IRC8M: Internal 8M RC oscillator

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 130ns. 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}.

28.3.2. I2C communication flow

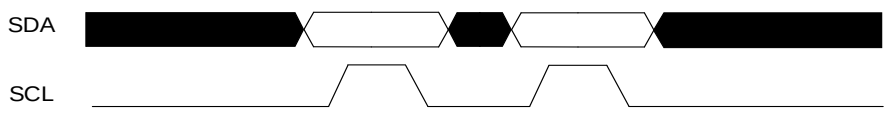
An I2C device is able to transmit or receive data whether it's a master or a slave, thus, there're 4 operation modes for an I2C device:

- 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 28-2. Data validation***). One clock pulse is generated for each data bit transferred.

Figure 28-2. Data validation

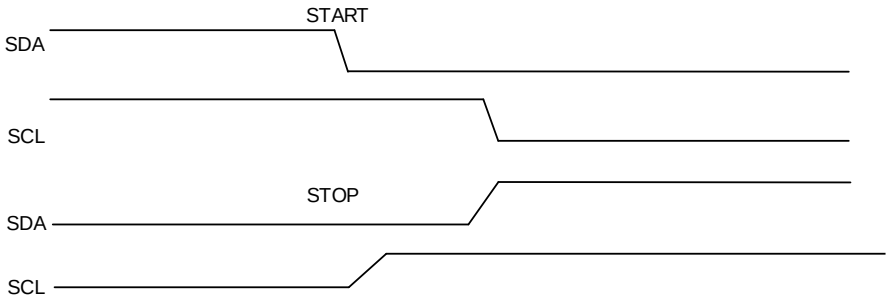


START and STOP signal

All transactions begin with a START and are terminated by a STOP (see ***Figure 28-3. START and STOP*** **condition*condition***). 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 28-3. START and STOP condition



Each I2C device is recognized by a unique address (whether it is a microcontroller, LCD driver, memory or keyboard interface) and can operate as either a transmitter or receiver, depending on the function of the device. It operates in slave mode by default. When it generates a START signal, the interface automatically switches from slave to master. If an arbitration loss or a STOP generation occurs, then the interface switches from master to slave, allowing multimaster capability.

An I2C slave will continue to detect addresses after a START signalon I2C bus and compare the detected address with its slave address which is programmable by software. Once the two addresses match, the I2C slave will send an ACK to the I2C bus and responses to the following command on I2C bus: transmitting or receiving the desired data. Additionally, if General Call is enabled by software, the I2C slave always responses to a General Call Address (0x00). The I2C block support both 7-bit and 10-bit address modes.

Data and addresses are transferred as 8-bit bytes, MSB first. The first byte(s) following the START signalcontain the address (one in 7-bit mode, two in 10-bit mode). The address is always transmitted in master mode.

A 9th clock pulse follows the 8 clock cycles of byte transmission, during which the receiver must send an acknowledge bit to the transmitter. Acknowledge can be enabled or disabled by software.

An I2C master always initiates or end a transfer using START or STOP signaland it’s also responsible for SCL clock generation.

In master mode, if AUTOEND=1, the STOP signalis generated automatically by hardware. If AUTOEND=0, the STOP signalgenerated by software, or the master can generate a RESTART signalto start a new transfer.

Figure 28-4. I2C communication flow with 10-bit address (Master Transmit)

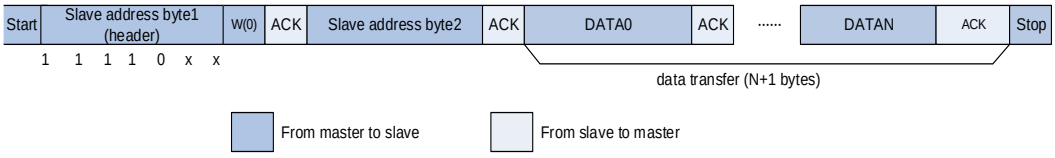


Figure 28-5. I2C communication flow with 7-bit address (Master Transmit)

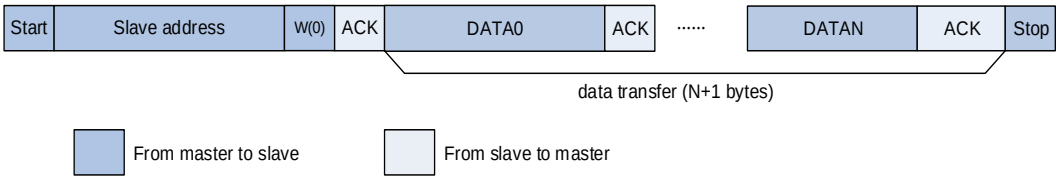
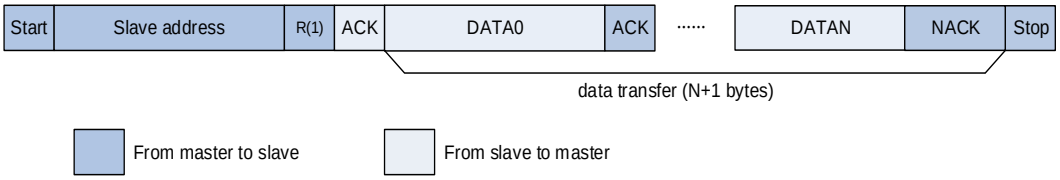


Figure 28-6. I2C communication flow with 7-bit address (Master Receive)



In 10-bit addressing mode, the HEAD10R bit can configured to decide whether the complete address sequence must be executed, or only the header to be sent. When HEAD10R=0, the complete 10-bit address read sequence must be excuted with START + header of 10-bit address in write direction + slave address byte 2 + RESTART + header of 10-bit address in read direction, as is shown in ***Figure 28-7. I2C communication flow with 10-bit address (Master Receive when HEAD10R=0)***.

In 10-bit addressing mode, if the master reception follows a master transmission between the same master and slave, the address read sequence can be RESTART + header of 10-bit address in read direction, as is shown in ***Figure 28-8. I2C communication flow with 10-bit address (Master Receive when HEAD10R=1)***.

Figure 28-7. I2C communication flow with 10-bit address (Master Receive when HEAD10R=0)

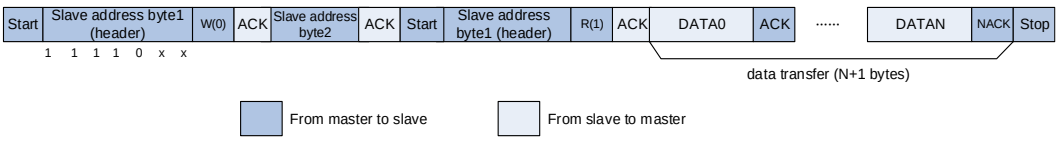
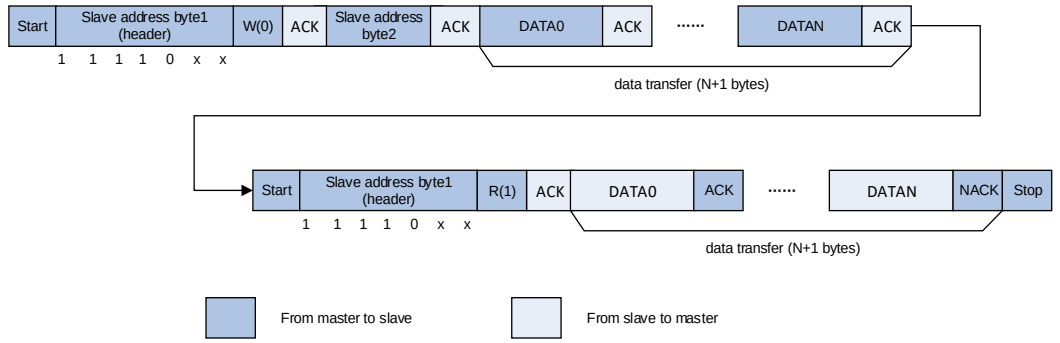


Figure 28-8. I2C communication flow with 10-bit address (Master Receive when HEAD10R=1)



28.3.3. Noise filter

The noise filters must be configured before setting the I2CEN bit in I2C_CTL0 register if it is necessary. The analog noise filter is disabled by setting the ANOFF bit in I2C_CTL0 register and enabled when ANOFF is 0. It can suppress spikes with a pulse width up to 50ns in fast mode and fast mode plus.

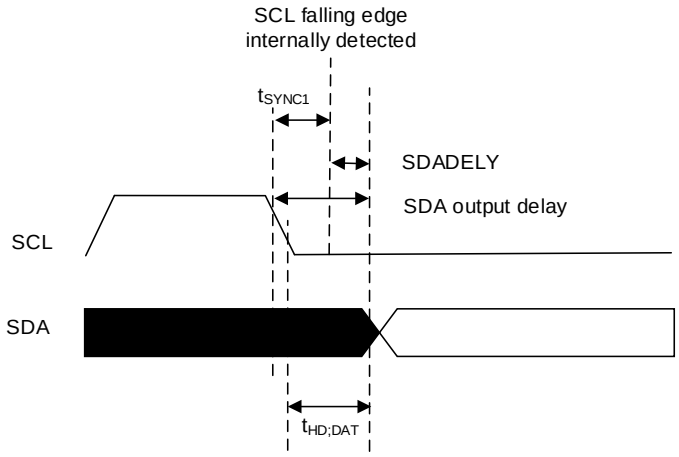
The digital noise filter can be used by configuring the DNF[3:0] bit in I2C_CTL0 register. The the level of the SCL or the SDA will not be changed if the level is stable for no more than DNF[3:0]×t_{I2CCLK}. The length of spikes to be suppressed is configured by DNF[3:0].

28.3.4. 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.

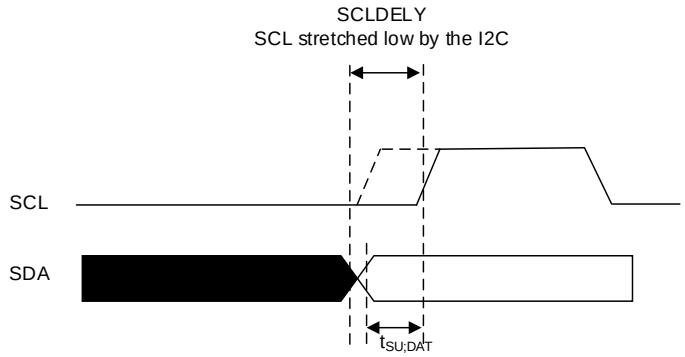
If the data is already available in I2C_TDATA register, the data will be sent on SDA after the SDADELY delay. As is shown in ***Figure 28-9. Data hold time***.

Figure 28-9. Data hold time



The SCLDELY counter starts when the data is sent on SDA output. As is shown in ***Figure 28-10. Data setup time***.

Figure 28-10. Data setup time



When the SCL falling edge is internally detected, a delay is inserted before sending SDA output. This delay is $t_{SDADELY}$ = SDADELY * t_{PSC} + t_{I2CCLK} where t_{PSC} = (PSC+1) * t_{I2CCLK}. $t_{SDADELY}$ effects $t_{HD;DAT}$. The total delay of SDA output is t_{SYNC1} + {[SDADELY * (PSC+1) + 1] * t_{I2CCLK}}. t_{SYNC1} depends on SCL falling slope, the delay of analog filter, the delay of digital filter and delay of SCL synchronization to I2CCLK clock. The delay of SCL synchronization to I2CCLK clock is 2 to 3 t_{I2CCLK}.

SDADELY must match condition as follows:

- SDADELY≥{t_f(max)+$t_{HD;DAT}$(min)-t_{AF}(min)-[(DNF+3)*t_{I2CCLK}]}/[(PSC+1)*t_{I2CCLK}]
- SDADELY≤{$t_{HD;DAT}$(max)-t_{AF}(max)-[(DNF+4)*t_{I2CCLK}]}/[(PSC+1)*t_{I2CCLK}]

Note: t_{AF} is the delay of analog filter. The $t_{HD;DAT}$ should be less than the maximum of $t_{VD;DAT}$.

When SS = 0, after $t_{SDADELY}$ delay, the slave had to stretch the clock before the data writing to I2C_TDATA register, SCL is low during the data setup time. The setup time is $t_{SCLDELY}$=(SCLDELY+1)*t_{PSC}. $t_{SCLDELY}$ effects $t_{SU;DAT}$.

SCLDELY must match condition as follows:

- SCLDELY≥{[t_r(max)+$t_{SU;DAT}$(min)]/[(PSC+1)*t_{I2CCLK}]}-1

In master mode, the SCL clock high and low levels must be configured by programming the PSC[3:0], SCLH[7:0] and SCLL[7:0] bits in the I2C_TIMING register.

When the SCL falling edge is internally detected, a delay is inserted before releasing the SCL output. This delay is t_{SCLL}=(SCLL+1)*t_{PSC} where t_{PSC}=(PSC+1)*t_{I2CCLK}. t_{SCLL} impacts the SCL low time t_{LOW}.

When the SCL rising edge is internally detected, a delay is inserted before forcing the SCL output to low level. This delay is t_{SCLH}=(SCLH+1)*t_{PSC} where t_{PSC}=(PSC+1)*t_{I2CCLK}. t_{SCLH} impacts the SCL high time t_{HIGH}.

Note: When the I2C is enabled, the timing configuration and SS mode must not be changed.

Table 28-2. 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	

28.3.5. 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:

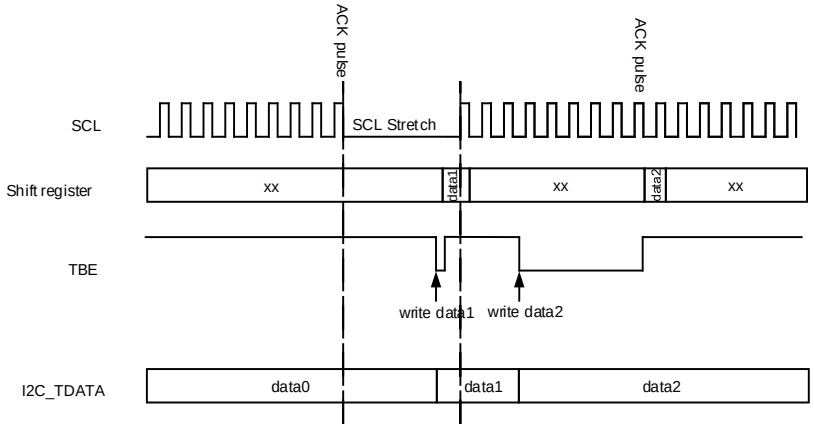
- Write I2CEN = 0
- Check I2CEN = 0
- Write I2CEN = 1

28.3.6. 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.

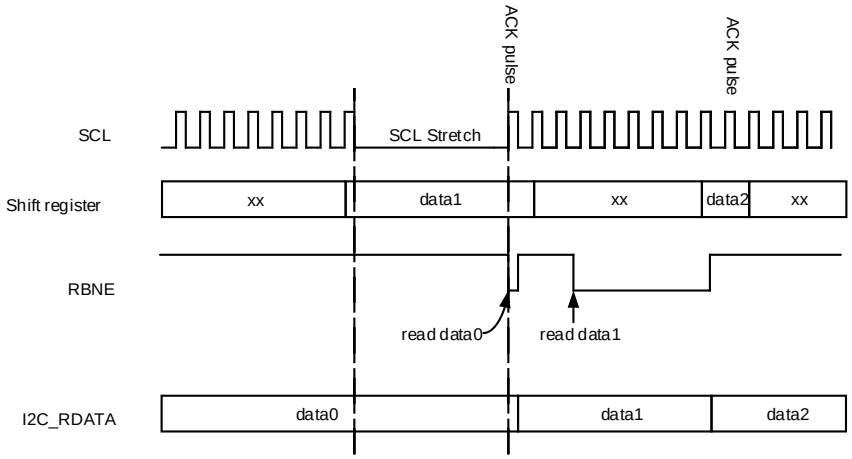
Figure 28-11. Data transmission



Data Reception

When receiving data, the data will be received in the shift register first. If RBNE is 0, the data in the shift register will move into I2C_RDATA register. If RBNE is 1, the SCL line wii be stretched until the previous received data in I2C_RDATA is read. The stretch is inserted before the acknowledge pulse.

Figure 28-12. 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 28-3. Communication modes to be shut down***, the I2C embedded a byte counter in the hardware.

Table 28-3. 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 BYTENUMis 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 both reload mode and automatic end mode are disabled, the I2C communication process needs to be terminated by software. If the number of bytes in BYTENUM[7:0] has been transferred, the software should set STOP 1 to generate a STOP signal and then clear the TC.

28.3.7. 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 should be written in the I2C_TDATA register . Or else the OUERR bit in the I2C_STAT register will be set, if the ERRIE bit is set, an interrupt will be generated. When the STPDET bit is set and the first data transmission starts, OUERR bit in the I2C_STAT register will also be set.
- In slave receiving mode, before the 9th SCL pulse (ACK pulse) occurred by the next data byte, the data must be read out from the I2C_RDATA register. Or else the OUERR bit in the I2C_STAT register will be set, if the ERRIE bit is set, and an interrupt will be generated.

Slave byte control mode

In slave receiving mode, the slave byte control mode can be enabled by setting the SBCTL bit in the I2C_CTL0 register to allow byte ACK control. When SS=1, the slave byte control mode is not allowed.

When using slave byte control mode, the reload mode must be enabled by setting the RELOAD bit in I2C_CTL1 register. In slave byte control mode, BYTENUM[7:0] in I2C_CTL1 register must be configured as 1 in the ADDSEND interrupt service routine and reloaded to 1 after each byte received. The TCR bit in I2C_STAT register will be set when a byte is received, the SCL will be stretched low by slave between the 8th and 9th clock pulses. Then the data can be read from the I2C_RDATA register, and the slave determines to send an ACK or a NACK by configuring the NACKEN bit in the I2C_CTL1 register. When the BYTENUM[7:0] is written a non-zero value, the slave will release the stretch.

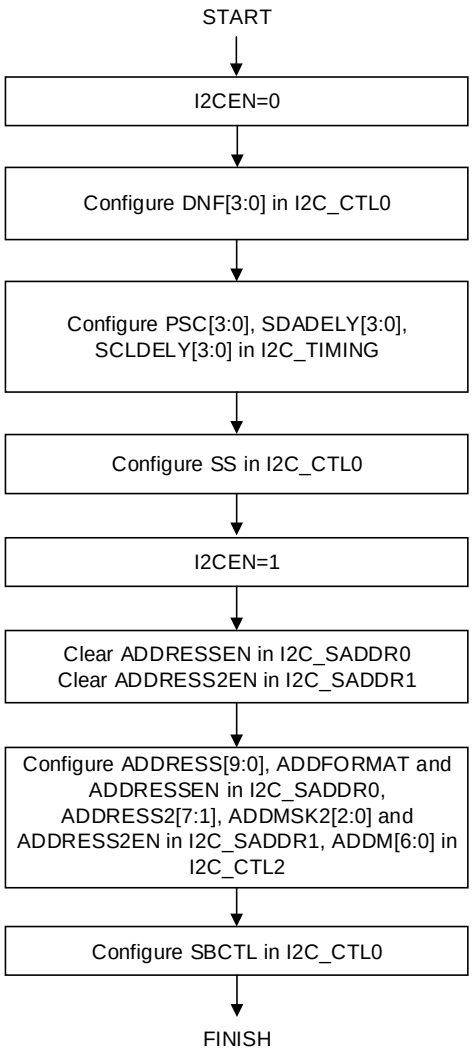
When the BYTENUM[7:0] is greater than 0x1, there is no stretch between the reception of two data bytes.

Note: The SBCTL bit can be configured in following cases:

1. I2CEN=0.
2. The slave has not been addressed.
3. ADDSEND=1.

Only when the ADDSEND=1 or TCR=1, the RELOAD bit can be modified.

Figure 28-13. I2C initialization in slave mode



Programming model in slave transmitting mode

When the I2C_TDATA register is empty, the TI bit in I2C_STAT register will be set. If the TIE bit in I2C_CTL0 register is set, an interrupt will be generated. The NACK bit in I2C_STAT register will be set when a NACK is received. And an interrupt is generated if the NACKIE bit is set in the I2C_CTL0 register. The TI bit in I2C_STAT register will not be set when a NACK is received.

The STPDET bit in I2C_STAT register will be set when a STOP is received. If the STPDETIE in I2C_CTL0 register is set, an interrupt will be generated.

When SBCTL is 0, if ADDSEND=1, and the TBE bit in I2C_STAT register is 0, the data in I2C_TDATA register can be chosed to be transmitted or flushed. The data is flushed by setting the TBE bit.

When SBCTL=1, the slave works in slave byte control mode, the BYTENUM[7:0] must be configured in the ADDSEND interrupt service routine. And the number of TI events is equal to the value of BYTENUM[7:0].

When SS=1, the SCL will not be stretched when ADDSEND bit in I2C_STAT register is set. In this case, the data in I2C_TDATA register can not be flushed in ADDSEND interrupt service routine. So the first byte to be sent must be programmed in the I2C_TDATA register previously.

- This data can be the one which is written in the last TI event of the last transfer.
- Setting the TBE bit can flush the data if it is not the one to be sent, then a new byte can be written in I2C_TDATA register. The STPDET must be 0 when the data transmission begins. Or else the OUERR bit in I2C_STAT register will be set.
- 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 28-14. Programming model for slave transmitting when SS=0

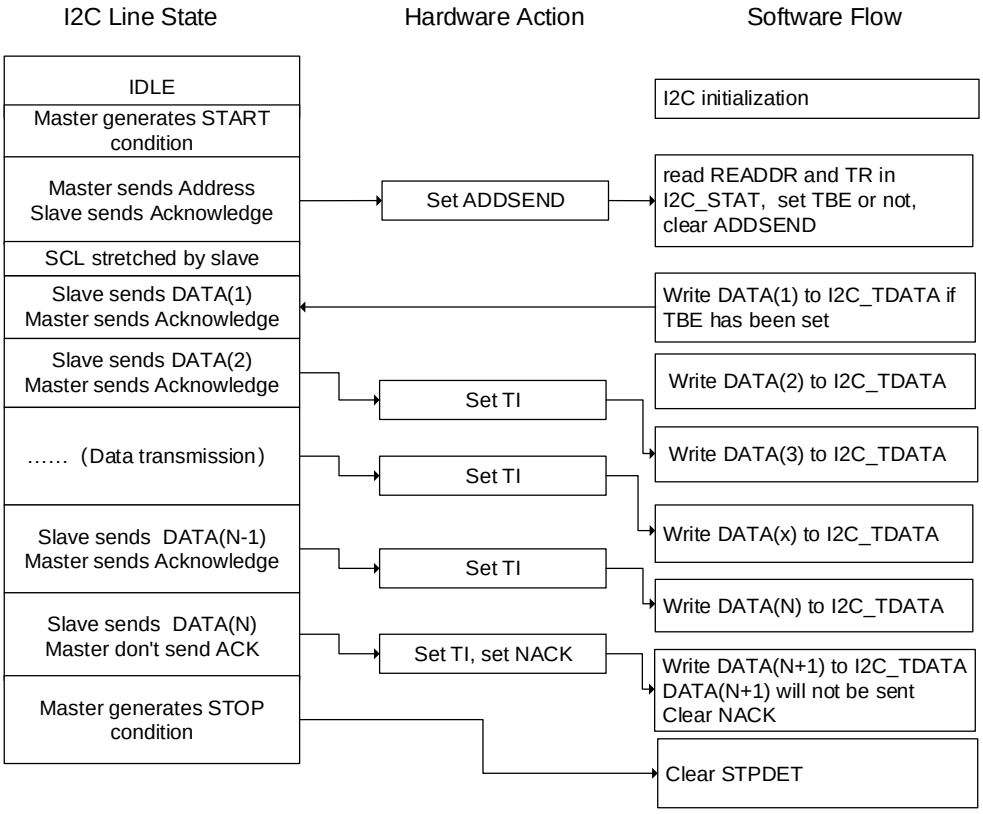
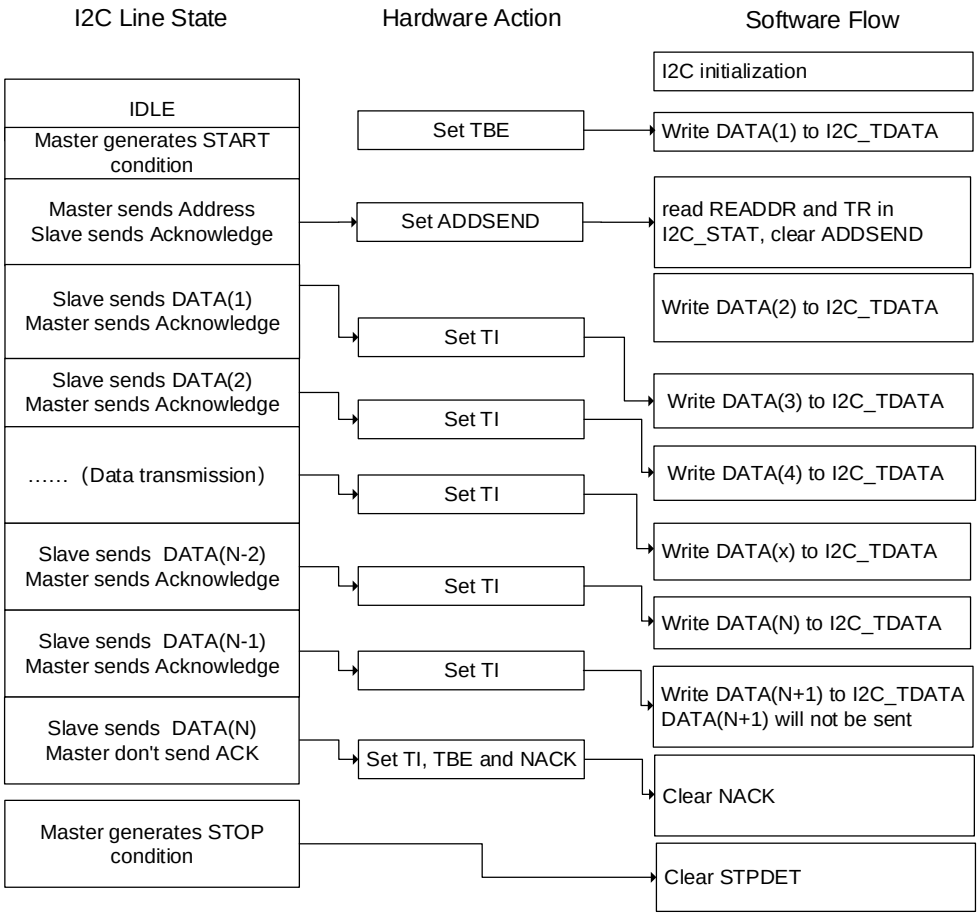


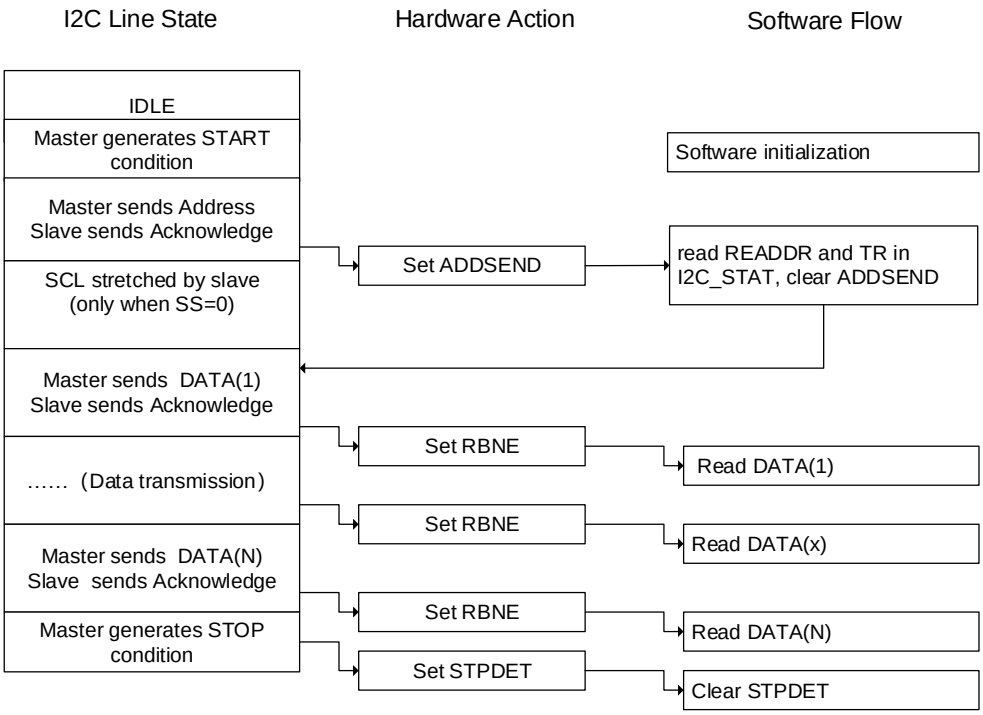
Figure 28-15. Programming model for slave transmitting when SS=1



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 28-16. Programming model for slave receiving



28.3.8. I2C master mode

Initialization

The SCLH[7:0] and SCLL[7:0] in I2C_TIMING register should be configured when I2CEN is 0. In order to support multi-master communication and slave clock stretching, a clock synchronization mechanism is implemented.

The SCLL[7:0] and SCLH[7:0] are used for the low level counting and high level couting respectively. After a t_{SYNC1} delay, when the SCL low level is detected, the SCLL[7:0] starts counting, if the SCLL[7:0] in I2C_TIMING register is reached by SCLL[7:0] counter, the I2C will release the SCL clock. After a t_{SYNC2} delay, when the SCL high level is detected, the SCLH[7:0] starts counting, if the SCLH[7:0] in I2C_TIMING register is reached by SCLH[7:0] counter, the I2C will stretch the SCL clock.

So the master clock period is:

$t_{SCL}=t_{SYNC1}+t_{SYNC2}+\{[(SCLH[7:0]+1)+(SLLL[7:0]+1)]*(PSC+1)*t_{I2CCLK}\}$.

The t_{SYNC1} depends on the SCL falling slope, delay by input analog and digital noise filter and SCL synchronization with I2CCLK clock, which generally 2 to 3 I2CCLK periods. The t_{SYNC2} depends on the SCL rising slope, delay by input analog and digital noise filter and SCL synchronization with I2CCLK clock, which generally 2 to 3 I2CCLK periods. The delay by digital noise filter is DNF[3:0]*t_{I2CCLK}.

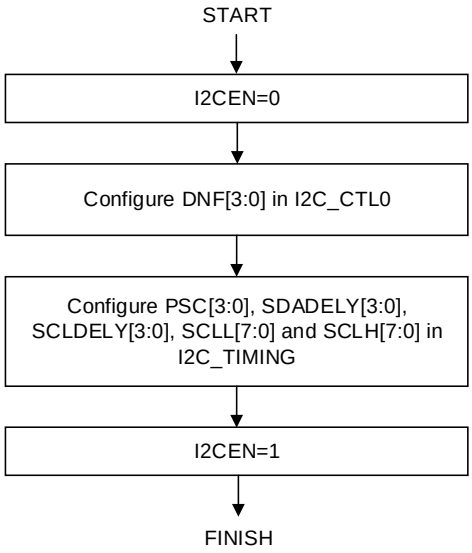
When works in master mode, the ADD10EN bit, SADDRESS[9:0] bits, TRDIR bit should be

configured in I2C_CTL1 register. When the addressing mode is 10-bit in master receiving mode, the HEAD10R bit must be configured to decide whether the complete address sequence must be executed, or only the header to be sent. The number of bytes to be transferred should be configured in BYTENUM[7:0] in I2C_CTL1 register. If the number of bytes to be transferred is equal to or greater than 255, BYTENUM[7:0] should be configured as 0xFF. Then the master sends the START signal. All the bits above should be configured before the START is set. The slave address will be sent after the START signal when the I2CBSY bit I2C_STAT register is detected as 0. When the arbitration is lost, the master changes to slave mode and the START bit will be cleared by hardware. When the slave address has been sent, the START bit will be cleared by hardware.

In 10-bit addressing mode, if the master receives a NACK after the transmission of 10-bit header, the master will resend it until ACK is received. The ADDSENDC bit must be set to stop sending the slave address.

If the START bit is set, meanwhile the ADDSEND is set by addressing as a slave, the master changes to slave mode. The ADDSENDC bit must be set to clear the START bit.

Figure 28-17. I2C initialization in master mode



Programming model in master transmitting mode

In master transmitting mode, the TI bit is set after the ACK is received of each byte transmission. If the TIE bit in I2C_CTL0 register is set, an interrupt will be generated. The bytes to be transferred is programmed in BYTENUM[7:0] in I2C_CTL1 register. If the bytes to be transferred is greater than 255, RELOAD bit in I2C_CTL1 register must be set to enable the reload mode. In reload mode, when data of BYTENUM[7:0] bytes have been transferred, the TCR bit in I2C_STAT register will be set and the SCL stretches unitil BYTENUM[7:0] is modified with a non-zero value.

When a NACK is received, the TI bit will not set.

- If data of BYTENUM[7:0] bytes have been transferred and RELOAD=0, the AUTOEND bit in I2C_CTL1 can be set to generate a STOP 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 signalto 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 28-18. Programming model for master transmitting (N<=255)

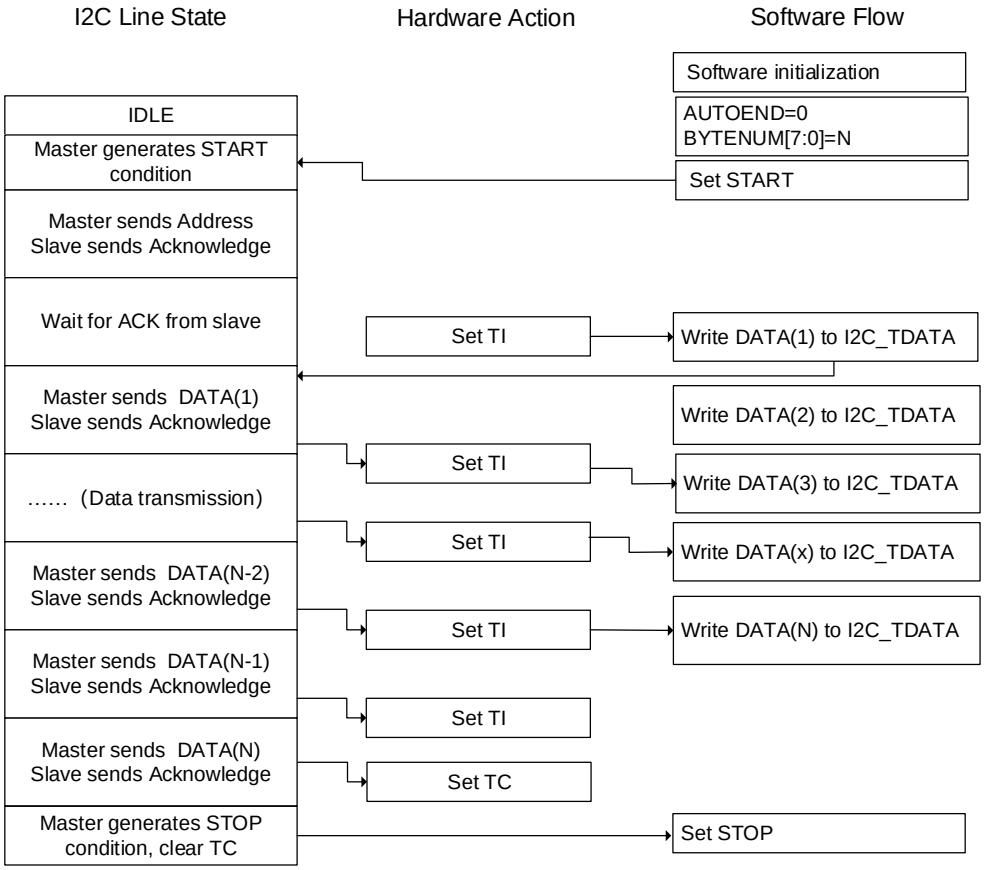
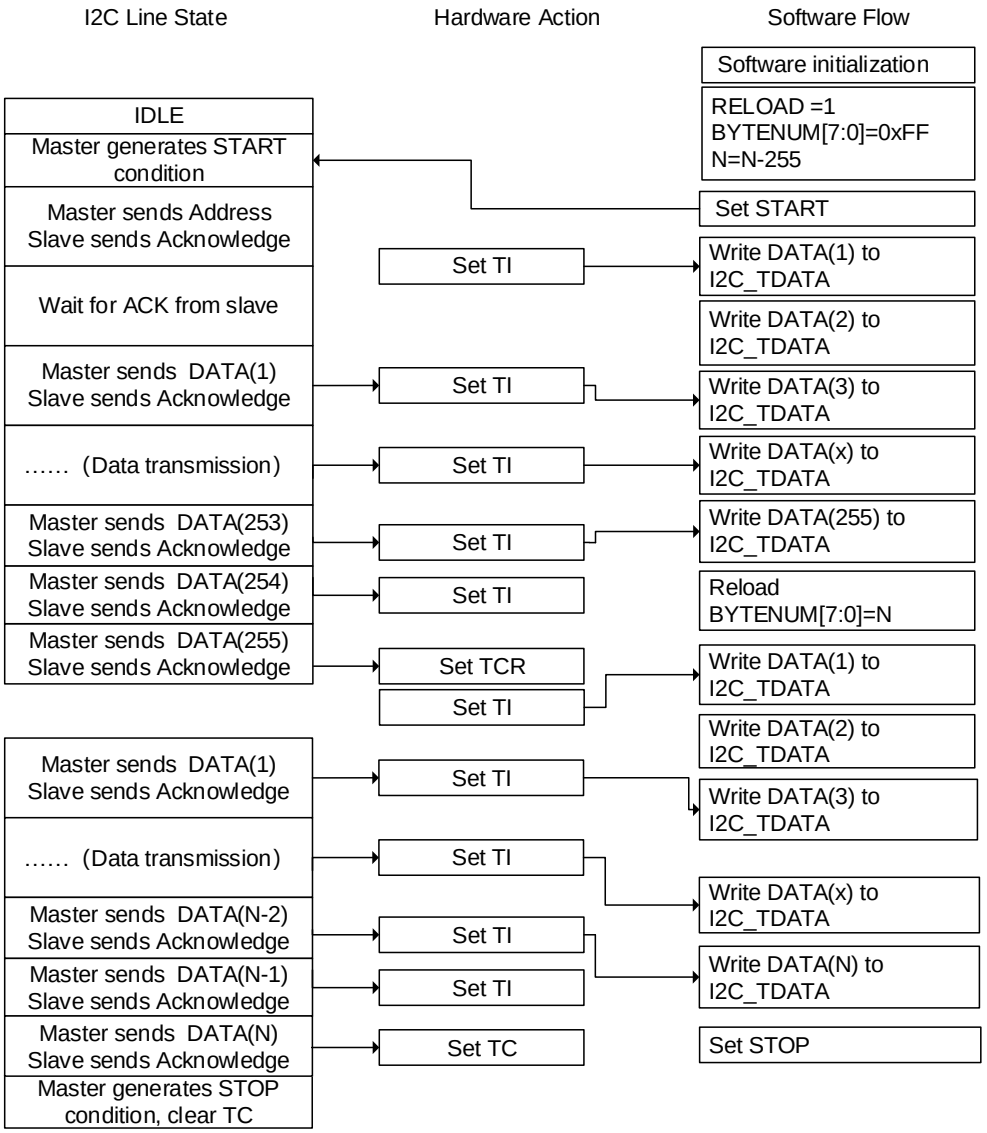


Figure 28-19. Programming model for master transmitting (N>255)



Programming model in master receiving mode

In master receiving mode, the RBNE bit in I2C_STAT register will be set when a byte is received. If the RBNEIE bit is set in I2C_CTL0 register, an interrupt will be generated. If the number of bytes to be received is greater than 255, RELOAD bit in I2C_CTL1 register must be set to enable the reload mode. In reload mode, when data of BYTENUM[7:0] bytes have been transferred, the TCR bit in I2C_STAT register will be set and the SCL stretches unitil BYTENUM[7:0] is modified with a non-zero value.

If data of BYTENUM[7:0] bytes have been transferred and RELOAD=0, the AUTOEND bit in I2C_CTL1 can be set to generate a STOP 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 signalby 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 28-20. Programming model for master receiving (N<=255)

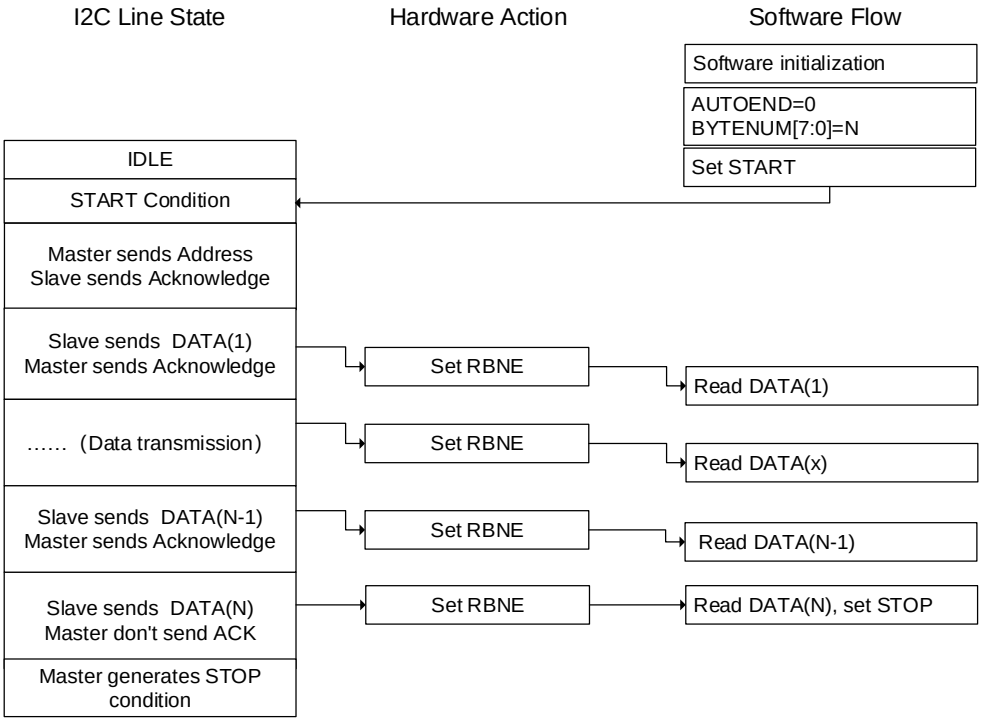
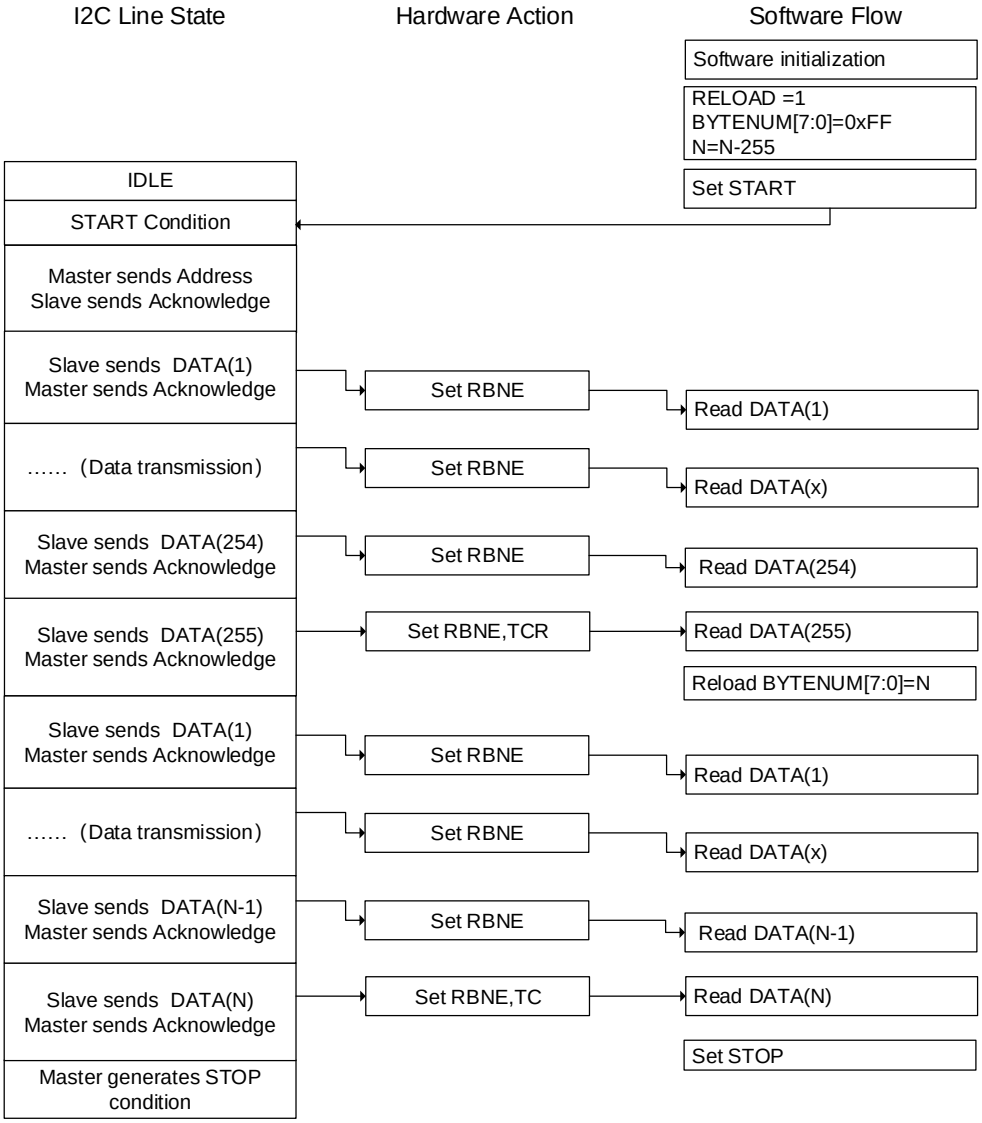


Figure 28-21. Programming model for master receiving (N>255)



28.3.9. 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. Slave byte control mode is 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 range 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.

28.3.10. SMBus mode

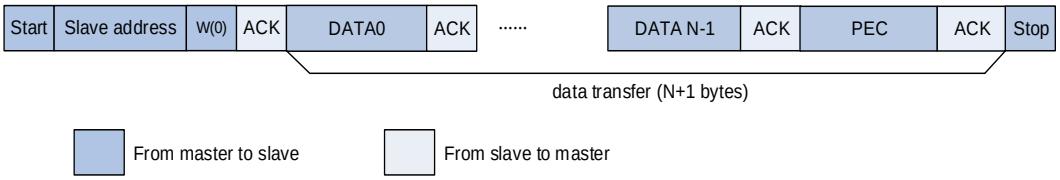
SMBus master transmitter and slave receiver

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, the data in I2C_PEC register will be transmitted automatically. If AUTOEND is 1 the SMBus master will send the STOP signal after the PEC byte automatically. If the AUTOEND is 0, the SMBus master can send a RESTART signal after the PEC. The PEC byte in I2C_PEC register 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 to enable the RELOAD mode. In order to check the PEC byte, it is necessary to clear the RELOAD bit and set PECTRANS bit. After receiving BYTENUM-1 data, the next received byte will be compared with the data in the I2C_PEC register. If the PEC values does not match, the NACK is automatically generated. If the PEC values matches, the ACK is automatically generated, regardless of the NACKEN bit value. When PEC byte is received, it is also copied into the I2C_RDATA register, and RBNE flag will be set. If the ERRIE bit in I2C_CTL0 register is 1, when PEC value does not match, the PECERR flag will be set and the interrupt will be generated. If ACK control is not required, then PECTRANS can be set to 1 and BYTENUM can be programmed according to the number of bytes to be received.

Note: After the RELOAD bit is set, the PECTRANS cannot be changed.

Figure 28-22. SMBus master transmitter and slave receiver communication flow



SMBus master receiver and slave transmitter

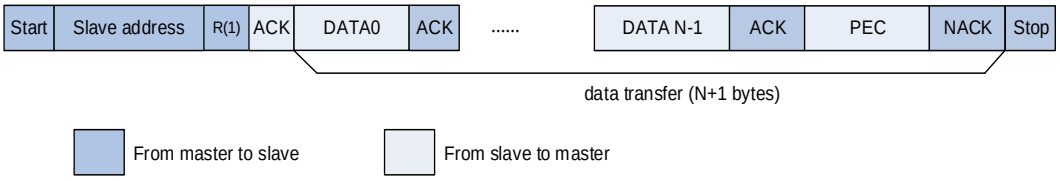
If the SMBus master is required to receive PEC at the end of bytes transfer, automatic end mode can be enabled. Before sending a START condition on the bus, PECTRANS bit must be set and slave addresses must be programmed. After receiving BYTENUM-1 data, the next received byte will be compared with the data in the I2C_PEC register automatically. A NACK is respond to the PEC byte before STOP condition.

If the SMBus master receiver is required to generate a RESTART signal after receiving PEC byte, automatic end mode must be disabled. 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 data in 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 data in the I2C_PEC register will be transmitted automatically.

Note: After the RELOAD bit is set, the PECTRANS cannot be changed.

Figure 28-23. SMBus master receiver and slave transmitter communication flow



28.3.11. Wakeup from power saving mode

When the address of I2C matches correctly, it can wake up from MCU sleep mode and Deep-sleep mode (APB clock is off). In order to wake up from power saving mode, WUEN bit must be set in the I2C_CTL0 register and the IRC8M must be selected as the clock source for I2CCLK. During power saving mode, the IRC8M is switched off. The I2C interface switches the IRC8M on, and stretches SCL low until IRC8M is woken up when a START is detected. Then the 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, IRC8M will be closed again and the MCU will not be wake up.

Only an address match interrupt (ADDMIE = 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 power saving mode is enabled, the digital filter must be disabled and the SS bit in I2C_CTL0 must be cleared. Before entering power saving mode (I2CEN = 0), the I2C peripheral must be disabled if wakeup from power saving mode is disabled (WUEN = 0).

28.3.12. 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.

28.3.13. I2C error and interrupts

The I2C error flags are listed in ***Table 28-4. I2C error flags***.

Table 28-4. 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 28-5. I2C interrupt events***.

Table 28-5. I2C interrupt events

Interrupt event	Event flag	Enable control bit
I2C_RDATA is not empty during receiving	RBNE	RBNEIE
Transmit interrupt	TI	TIE
STOP siganl detected in slave mode	STPDET	STPDETIE
Transfer complete reload	TCR	TCIE
Transfer complete	TC	

Interrupt event	Event flag	Enable control bit
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	

28.3.14. 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.

28.4. Register definition

I2C0 base address: 0x4000 5400

I2C1 base address: 0x4000 5800

I2C2 base address: 0x4000 C000

I2C3 base address: 0x4000 5C00

28.4.1. Control register 0 (I2C_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								PECEN	SMBALTEN	SMBDAEN	SMBHAEN	GCEN	WUEN	SS	SBCTL
								rw	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 power saving mode enable This bit is cleared when mcu wakeup from power saving mode. 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 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: Not acknowledge (NACK) received interrupt is disabled 1: Not acknowledge (NACK) received interrupt is enabled
3	ADDMIE	Address match interrupt enable in slave mode 0: Address match (ADDSEND) interrupt is disabled 1: Address match (ADDSEND) 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

28.4.2. Control register 1 (I2C_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					PECTRANS	AUTOEND	RELOAD	BYTENUM[7:0]							
					rw	rw	rw	rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NACKEN	STOP	START	HEAD10R	ADD10EN	TRDIR	SADDRESS[9:0]									
rw	rw	rw	rw	rw	rw	rw									

Bits	Fields	Descriptions

31:27	Reserved	Must be kept at reset value.
26	PECTRANS	PEC Transfer Set by software. Cleared by hardware in the following cases: When PEC byte is transferred or ADDSEND bit is set or STOP signal is detected or I2CEN=0. 0: Don't transfer PEC value 1: Transfer PEC **Note:** This bit has no effect when RELOAD=1, or SBCTL=0 in slave mode.
25	AUTOEND	Automatic end mode in master mode 0: TC bit is set when the transfer of BYTENUM[7:0] bytes is completed. 1: a STOP signal is sent automatically when the transfer of BYTENUM[7:0] bytes is completed. **Note:** This bit works only when RELOAD=0. This bit is set and cleared by software.
24	RELOAD	Reload mode 0: After the data of BYTENUM[7:0] bytes transfer, the transfer is completed. 1: After data of BYTENUM[7:0] bytes transfer, the transfer is not completed and the new BYTENUM[7:0] will be reloaded. Every time when the BYTENUM[7:0] bytes have been transferred, the TCR bit in I2C_STAT register will be set. This bit is set and cleared by software.
23:16	BYTENUM[7:0]	Number of bytes to be transferred These bits are programmed with the number of bytes to be transferred. When SBCTL=0, these bits have no effect. **Note:** These bits should not be modified when the START bit is set.
15	NACKEN	Generate NACK in slave mode 0: an ACK is sent after receiving a new byte. 1: a NACK is sent after receiving a new byte. **Note:** The bit can be set by software, and cleared by hardware when the NACK is sent, or when a STOP signal is detected or ADDSEND is set, or when I2CEN=0. When PEC is enabled, whether to send an ACK or a NACK is not depend on the NACKEN bit. When SS=1, and the OUERR bit is set, the value of NACKEN is ignored and a NACK will be sent.
14	STOP	Generate a STOP signal on I2C bus This bit is set by software and cleared by hardware when I2CEN=0 or STOP condition is detected. 0: STOP will not be sent 1: STOP will be sent
13	START	Generate a START condition on I2C bus This bit is set by software and cleared by hardware after the address is sent. When the arbitration is lost, or a timeout error occurred, or I2CEN=0, this bit can also be

		cleared by hardware. It can be cleared by software by setting the ADDSENDC bit in I2C_STATC register. 0: START will not be sent 1: START will be sent
12	HEAD10R	10-bit address header executes read direction only in master receive mode 0: The 10-bit master receive address sequence is START + header of 10-bit address (write) + slave address byte 2 + RESTART + header of 10-bit address (read). 1: The 10-bit master receive address sequence is RESTART + header of 10-bit address (read). **Note:** When the START bit is set, this bit can not be changed.
11	ADD10EN	10-bit addressing mode enable in master mode 0: 7-bit addressing in master mode 1: 10-bit addressing in master mode **Note:** When the START bit is set, this bit can not be modified.
10	TRDIR	Transfer direction in master mode 0: Master transmit 1: Master receive **Note:** When the START bit is set, this bit can not be modified.
9:0	SADDRESS[9:0]	Slave address to be sent SADDRESS[9:8]: Slave address bit 9:8 If ADD10EN = 0, these bits have no effect. If ADD10EN = 1, these bits should be written with bits 9:8 of the slave address to be sent. SADDRESS[7:1]: Slave address bit 7:1 If ADD10EN = 0, these bits should be written with the 7-bit slave address to be sent. If ADD10EN = 1, these bits should be written with bits 7:1 of the slave address to be sent. SADDRESS0: Slave address bit 0 If ADD10EN = 0, this bit has no effect. If ADD10EN = 1, this bit should be written with bit 0 of the slave address to be sent **Note:** When the START bit is set, the bit filed can not be modified.

28.4.3. Slave address register 0 (I2C_SADDR0)

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

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:** When ADDRESSEN is set, this bit should not be written.
0	ADDRESS0	Bit 0 of a 10-bit address **Note:** When ADDRESSEN is set, this bit should not be written.

28.4.4. Slave address register 1 (I2C_SADDR1)

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
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.

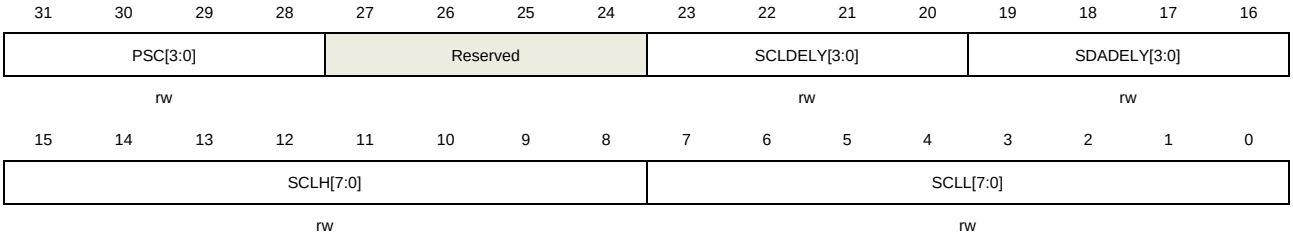
14:11	Reserved	Must be kept at reset value.
10:8	ADDMSK2[2:0]	ADDRESS2[7:1] mask Defines which bits of ADDRESS2[7:1] are compared with an incoming address byte, and which bits are masked (don't care). 000: No mask, all the bits must be compared. N(001~110): ADDRESS2[n:1] is masked. Only ADDRESS2[7:n+1] 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:** When ADDRESS2EN is set, these bits should not be written.
0	Reserved	Must be kept at reset value.

28.4.5. Timing register (I2C_TIMING)

Address offset: 0x10

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:28	PSC[3:0]	Timing prescaler In order to generate the clock period t_{PSC} used for data setup and data hold counters, these bits are used to configure the prescaler for I2CCLK. The t_{PSC} is also used for SCL high and low level counters. $t_{PSC}=(PSC+1)*t_{I2CCLK}$
27:24	Reserved	Must be kept at reset value.
23:20	SCLDELY[3:0]	Data setup time A delay $t_{SCLDELY}$ between SDA edge and SCL rising edge can be generated by configuring these bits. And during $t_{SCLDELY}$, the SCL line is stretched low in master mode and in slave mode when SS = 0. $t_{SCLDELY}=(SCLDELY+1)*t_{PSC}$
19:16	SDADELY[3:0]	Data hold time

		A delay $t_{SDADELY}$ between SCL falling edge and SDA edge can be generated by configuring these bits. And during $t_{SDADELY}$, the SCL line is stretched low in master mode and in slave mode when SS = 0. $t_{SDADELY}$= SDADELY x 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.

28.4.6. Timeout register (I2C_TIMEOUT)

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
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) x 2048 x t_{I2CCLK}. 0: Extended clock timeout detection is disabled. 1: Extended clock timeout detection is enabled.
30:28	Reserved	Must be kept at reset value.
27:16	BUSTOB[11:0]	Bus timeout B Configure the cumulative clock extension timeout. In master mode, the master cumulative clock low extend time $t_{LOW:MEXT}$ is detected. In slave mode, the slave cumulative clock low extend time $t_{LOW:SEXT}$ is detected. $t_{LOW:EXT}$=(BUSTOB+1)*2048*t_{I2CCLK}. **Note:** These bits can be modified only when EXTOEN =0.
15	TOEN	Clock timeout detection enable

		If the SCL stretch time greater than $t_{TIMEOUT}$ when TOIDLE =0 or high for more than t_{IDLE} when TOIDLE =1, a timeout error is detected. 0: SCL timeout detection is disabled 1: SCL timeout detection is enabled
14:13	Reserved	Must be kept at reset value.
12	TOIDLE	Idle clock timeout detection 0: BUSTOA is used to detect SCL low timeout 1: BUSTOA is used to detect both SCL and SDA high timeout when the bus is idle **Note:** This bit can be written only when TOEN =0.
11:0	BUSTOA[11:0]	Bus timeout A When TOIDLE = 0, $t_{TIMEOUT}$=(BUSTOA+1)*2048*t_{I2CCLK}. When TOIDLE = 1, t_{IDLE}=(BUSTOA+1)*4*t_{I2CCLK}. **Note:** These bits can be written only when TOEN =0.

28.4.7. Status register (I2C_STAT)

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								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

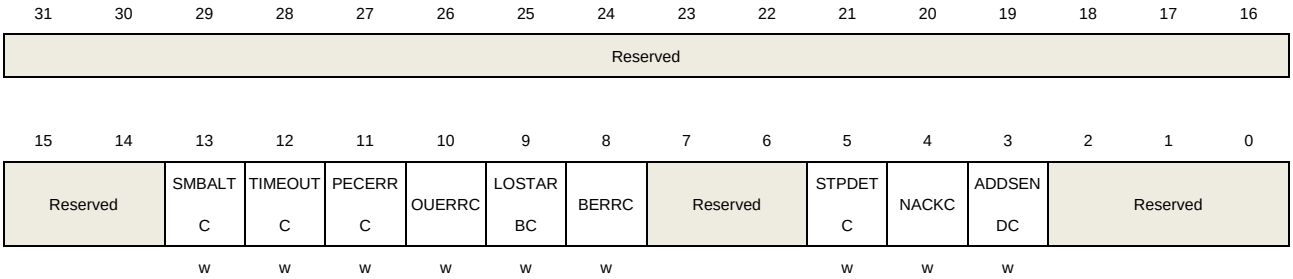
		This bit is set by hardware when RELOAD=1 and data of BYTENUM[7:0] bytes have been transferred. It is cleared by software when BYTENUM[7:0] is written to a non-zero value. 0: When RELOAD=1, transfer of BYTENUM[7:0] bytes is not completed 1: When RELOAD=1, transfer of BYTENUM[7:0] bytes is completed
6	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

28.4.8. Status clear register (I2C_STATC)

Address offset: 0x1C
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	SMBALTC	SMBus alert flag clear. Software can clear the SMBALT bit of I2C_STAT by writing 1 to this bit.
12	TIMEOUTC	TIMEOUT flag clear. Software can clear the TIMEOUT bit of I2C_STAT by writing 1 to this bit.
11	PECERRC	PEC error flag clear. Software can clear the PECERR bit of I2C_STAT by writing 1 to this bit.
10	OUERRC	Overrun/Underrun flag clear. Software can clear the OUERR bit of I2C_STAT by writing 1 to this bit.
9	LOSTARBC	Arbitration Lost flag clear. Software can clear the LOSTARB bit of I2C_STAT by writing 1 to this bit.
8	BERRC	Bus error flag clear. Software can clear the BERR bit of I2C_STAT by writing 1 to this bit.
7:6	Reservced	Must be kept at reset value.
5	STPDETC	STPDET flag clear Software can clear the STPDET bit of I2C_STAT by writing 1 to this bit.
4	NACKC	Not Acknowledge flag clear Software can clear the NACK bit of I2C_STAT by writing 1 to this bit.

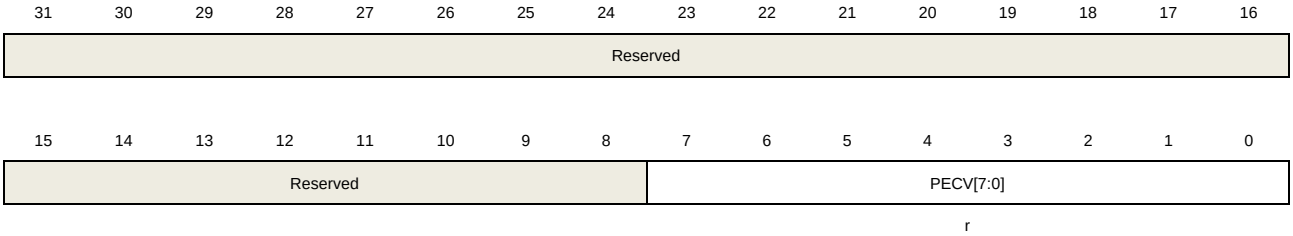
3	ADDSENDC	ADDSEND flag clear Software can clear the ADDSEND bit of I2C_STAT by writing 1 to this bit.
2:0	Reserved	Must be kept at reset value.

28.4.9. PEC register (I2C_PEC)

Address offset: 0x20

Reset value: 0x0000 0000

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



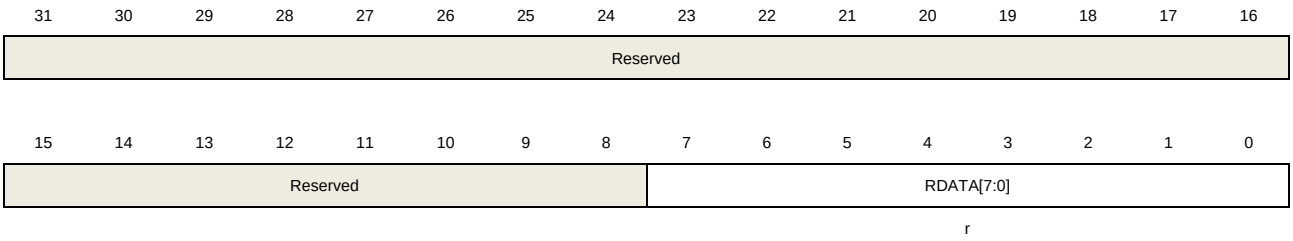
Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	PECV[7:0]	Packet Error Checking Value that calculated by hardware when PEC is enabled. PECV is cleared by hardware when I2CEN = 0.

28.4.10. Receive data register (I2C_RDATA)

Address offset: 0x24

Reset value: 0x0000 0000

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



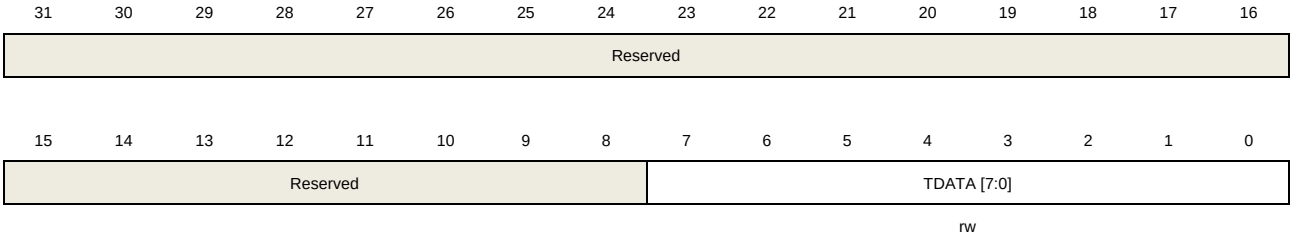
Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	RDATA[7:0]	Receive data value

28.4.11. Transmit data register (I2C_TDATA)

Address offset: 0x28

Reset value: 0x0000 0000

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



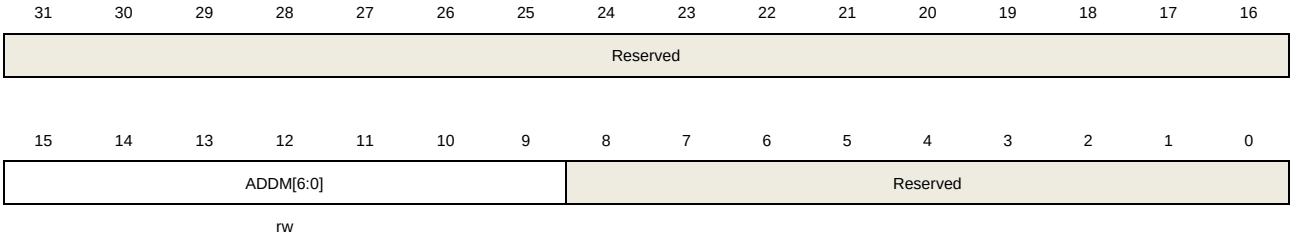
Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	TDATA[7:0]	Transmit data value

28.4.12. Control register 2 (I2C_CTL2)

Address offset: 0x90

Reset value: 0x0000 FE00

This register has to 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.

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 interface supports the CAN 2.0A/B protocol, ISO 11898-1:2015 and BOSCH CAN-FD specification.

The CAN module is a CAN Protocol controller with a very flexible mailbox system for transmitting and receiving CAN frames. The mailbox system consists of a set of mailboxes that store configuration and control data, timestamp, message ID, and data. The space of up to 32 mailboxes can also be configured as Rx FIFO with ID filtering against up to 104 extended IDs or 208 standard IDs or 416 partial 8-bit IDs, and configure receive FIFO/mailbox private filter register for up to 32 ID filter table elements.

29.2. Characteristics

- Supports CAN protocol version 2.0A/B.
- Compliant with the ISO 11898-1:2015 standard.
- Supports CAN FD frame with up to 64 data bytes, baudrate up to 8 Mbit/s.
- Supports CAN classical frame with up to 8 data bytes, baudrate up to 1 Mbit/s.
- Supports time stamp based on 16-bit free running counter.
- Supports transmitter delay compensation for CAN-FD frames at faster data rates.
- Maskable interrupts.
- Supports four communication mode: normal mode, Inactive mode, Loopback and silent mode, and Monitor mode.
- Supports two power saving modes: CAN_Disable mode, and Pretended Networking mode.
- Support two wakeup methods for waking up from Pretended Networking mode: wakeup matching event, and wakup timeout event.
- 32 mailboxes when configures with 8 bytes data length each, configurable as Rx or Tx mailbox.
- Global network time, synchronized by a specific message.

Transmission

- Supports transmission abort.
- Tx mailbox status checkable.
- CRC for transmitted message.
- Supports priority of transmission message: lowest mailbox number, or highest priority.

Reception

- Receive private filter registers per Rx mailbox or Rx FIFO.
- Receive public filter register for Rx mailboxes and receive public filter register for Rx

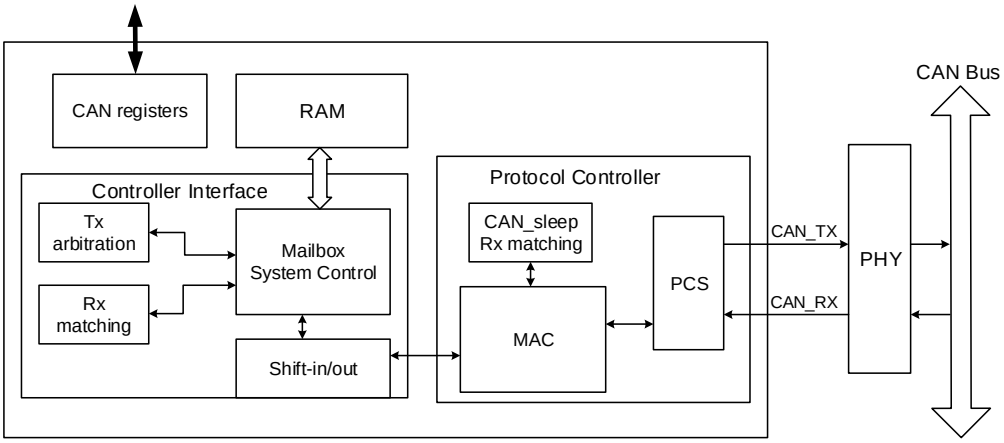
FIFO.

- Supports priority of message reception between mailboxes and Rx FIFO during matching process.
- Rx FIFO identifier filtering, supports identifier matching against either 104 extended, 208 standard, or 416 partial (8 bit) identifiers.
- Rx FIFO up to 6 frames depth, with DMA support.

29.3. Function overview

Figure 29-1. CAN module block diagram shows the CAN block diagram.

Figure 29-1. CAN module block diagram



As shown in *Figure 29-1. CAN module block diagram*, CAN module includes three main parts:

- The Protocol controller

  The Protocol controller manages the communication on the CAN bus, including:

  MAC (Media Access Control):
  - Bit-stuffing/de-stuffing.
  - Stuff bit count for FD Frames.
  - Add CRC.
  - Construction of MAC frame.
  - ACK check/transmission.

  PCS (Physical Coding Sub-layer):
  - Bit timing.
  - Synchronization.
  - TDC (Transmitter delay compensation).

  Pretended Networking Rx matching:
  - Process reception matching in Pretended Networking mode.
- The Controller Interface

The Controller Interface manages RAM space selection for reception and transmission, including:

Tx arbitration:

- Find out the frame with the highest priority.

Rx matching:

- Compare the frame data received in the Rx shift buffer (an internal mailbox descriptor) with the fields in Rx mailbox or Rx FIFO according to the configured matching order.

Mailbox System Controller:

- Manage RAM space selection for reception and transmission, control the mailbox CODE, control the Rx FIFO pointer, and control the access requirement from the APB bus to the RAM space.

  The messages are stored in an embedded RAM dedicated to the CAN module. The dedicated RAM base address is module base address.

Shift in/out:

- Tranmit data between the selected mailbox / Rx FIFO descriptor and the Tx or Rx shift buffer.

■ CAN registers

The CAN registers is responsible for the CAN module communication with the APB bus.

29.3.1. Mailbox descriptor

The mailbox descriptor shown in ***Table 29-1. Mailbox descriptor with 64 byte payload*** can be used for both extended (29-bit identifier) and standard (11-bit identifier) frames. Each mailbox is formed by 16, 24, 40, or 72 bytes, depending on the data bytes allocation for the message payload: 8, 16, 32, or 64 data bytes, respectively. The memory area from offset 0x80 to 0x27F is used by the mailboxes.

Table 29-1. Mailbox descriptor with 64 byte payload

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MDES0	FDF	BRS	ESI	Reserved	CODE[3:0]				Reserved	SRR	IDE	RTR	DLC[3:0]				TIMESTAMP[15:0]															
MDES1	PRIO[2:0]			ID_STD[10:0]													ID_EXD[17:0]															
MDES2	DATA_0[7:0]								DATA_1[7:0]								DATA_2[7:0]								DATA_3[7:0]							
...	...								...								...								...							
MDES17	DATA_60[7:0]								DATA_61[7:0]								DATA_62[7:0]								DATA_63[7:0]							

MDES0: Mailbox descriptor word 0

Address offset: 0x80

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FDF	BRS	ESI	Reserved	CODE[3:0]				Reserved	SRR	IDE	RTR	DLC[3:0]			
rw	rw	rw		rw					rw	rw	rw	rw			

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

Bits	Fields	Descriptions
31	FDF	FD format indicator This bit is used to distinguish between CAN and CAN FD format frames. For reception (Rx mailbox), no need to write this bit, it will be stored with the value received on the CAN bus.
30	BRS	Bit rate switch This bit defines whether the bit rate is switched for a CAN FD frame. For reception (Rx mailbox), no need to write this bit, it will be stored with the value received on the CAN bus.
29	ESI	Error state indicator This bit indicates if the transmitting node is error active or error passive. This bit does not exist in Classical frames. For transmission (Tx mailbox), it is transmitted dominant by error active nodes, and recessive by error passive nodes. For reception (Rx mailbox), no need to write this bit, it will be stored with the value received on the CAN bus.
28	Reserved	Must be kept at rest value.
27:24	CODE[3:0]	Mailbox Code (CODE) This bit field can be accessed by the CPU and by the CAN module, as part of the mailbox matching and arbitration process. The encoding is shown in ***Table 29-3. Mailbox Rx CODE*** and ***Table 29-4. Mailbox Tx CODE***.
23	Reserved	Must be kept at rest value.
22	SRR	Substitute remote request This bit is only used in extended format. For transmission (Tx mailbox), it must be set to ‘1’ (recessive), if the bus transmits this bit as ‘0’ (dominant), then it means an arbitration loss. For reception (Rx mailbox), it will be stored with the value received on the CAN bus. 0: Not valid for transmission in extended format frames. 1: Transmission in extended format frames.
21	IDE	ID extended bit This bit specifies whether the frame is standard or extended format. For reception (Rx mailbox), it will be stored with the value received on the CAN bus. 0: Frame format is standard. 1: Frame format is extended.

20	RTR	Remote transmission request For transmission (Tx mailbox), if this bit is set to '1' (recessive), and the bus transmits this bit as '0' (dominant), then it means an arbitration loss. If this bit is set to '0' (dominant), and the bus transmits this bit as '1' (recessive), it is treated as a bit error. If the value configured matches the value transmitted, it is considered a successful bit transmission. For reception (Rx mailbox), it will be stored with the value received on the CAN bus. 0: In Tx mailbox, the current mailbox has a data frame to be transmitted. In Rx mailbox, it may be considered in matching process. 1: In Tx mailbox, it means the current mailbox has a remote request frame to be transmitted. In Rx mailbox, incoming remote request frames may be stored. **Note:** When configured as CAN FD frames, the RTR bit must be negated. This bit must be considered in classical frames only.
19:16	DLC[3:0]	Data length code in bytes This bit field is the length (in bytes) of the Rx or Tx payload. For reception (Rx mailbox), no need to write this bit field, they are written by the CAN module with the DLC field of the received frame. For transmission (Tx mailbox), this bit field is written by the CPU with value of the frame to be transmitted. When RTR is 1, the frame to be transmitted is a remote request frame and does not include the data field, regardless of the DLC field.
15:0	TIMESTAMP[15:0]	Free-Running counter timestamp This bit field is a copy of the free running counter, captured for Tx and Rx frames at the time when the beginning of the ID field appears on the CAN bus.

Table 29-2. Data bytes for DLC

DLC	Data size in bytes
i (0 ≤ i ≤ 8)	i (0 ≤ i ≤ 8)
9	12
10	16
11	20
12	24
13	32
14	48
15	64

Table 29-3. Mailbox Rx CODE

CODE	Meaning	CODE after reception	Serviced(1)	RRFRMS(2)	Description
0b0000	INACTIVE	-	-	-	Mailbox does not participate in the matching process.
0b0100	EMPTY	FULL	-	-	After a frame is received successfully, the CODE field is automatically switches to FULL.

CODE	Meaning	CODE after reception	Serviced(1)	RRFRMS(2)	Description
0b0010	FULL	FULL	Yes	-	It remains FULL. If a new frame is moved to the mailbox after the mailbox was serviced, the code still remains FULL.
		OVERRUN	No		If the mailbox is FULL and a new frame is moved in before the CPU completes services it, the CODE field is automatically switches to OVERRUN.
0b0110	OVERRUN	FULL	Yes	-	If the CODE is OVERRUN and a new frame is moved in after CPU has serviced the mailbox, the CODE is automatically switches to FULL.
		OVERRUN	No		If the CODE field already indicates OVERRUN, and another new frame must be moved, the mailbox will be overwritten again, and the code will remain OVERRUN.
0b1010	RANSWER(3)	TANSWER (0x1110)	-	0	A Remote Answer was configured to recognize a remote request frame reception. After reception, the mailbox is set to transmit a response frame when RRFRMS bit in CAN_CTL2 register is 0. The code is automatically changed to TANSWER.
		-		1	The CODE is not effect during matching and arbitration process.
CODE[0] = 1	BUSY(4)	FULL	-	-	Indicates that the mailbox is being updated.
		OVERRUN			

(1) Serviced: Mailbox was serviced by CPU read, and was unlocked by reading CAN_TIMER register or other mailbox.

(2) Remote Request Frame Stored bit, refer to ***Control register 2 (CAN_CTL2)***.

(3) A mailbox with CODE 0b1010 should not be aborted. CODE 0b1010 must be used in mailbox which configured as CAN classical format, having the FDF bit reset.

(4) If CODE[0] bit is set, the corresponding mailbox will not participate in the matching process. Notice that for Tx mailboxes, the BUSY bit should be ignored when read, except when MST bit in the CAN_CTL0 register is set.

Table 29-4. Mailbox Tx CODE

CODE	Meaning	CODE after transmission	RTR	Description
0b1000	INACTIVE	-	-	Mailbox does not participate in arbitration process.
0b1001	ABORT	-	-	Mailbox does not participate in arbitration process.
0b1100	DATA	INACTIVE	0	Transmit data frame. After transmission, the mailbox automatically returns to the INACTIVE state.
	REMOTE	EMPTY	1	Transmit remote request frame. After transmission, the mailbox automatically becomes an Rx empty mailbox

				with the same ID.
0b1110	TANSWER	RANSWER	-	This is an intermediate code which is automatically written to the mailbox by the controller interface when a matching remote request frame is received. After transmitting the remote response frame, the mailbox will automatically return to RANSWER state. The CPU can also write this code with the same effect. The remote response frame can be either a data frame or another remote request frame depending on the RTR bit.

MDES1: Mailbox descriptor word 1

Address offset: 0x84

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRIO[2:0]			ID_STD[10:0]											ID_EXD[17:16]	
rw			rw											rw	

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

Bits	Fields	Descriptions
31:29	PRIO[2:0]	Local priority This bit field is only used when LAPRIOEN bit in CAN_CTL0 register is set. This bit filed is only used for Tx mailboxes, while these bits are not transmitted, they are appended to the regular ID to define the transmission priority.
28:18	ID_STD[10:0]	Identifier for standard frame In standard frame format, only these 11 most significant bits (28 to 18) are used for frame identification in both reception and transmission cases. The 18 least significant bits are ignored.
17:0	ID_EXD[17:0]	Identifier for extended frame In extended frame format, ID_STD[10:0] & these bits are used for frame identification in both reception and transmission cases.

MDESx: Mailbox descriptor word x (x = 2..17)

Address offset: 0x80 + 0x04 * x (x = 2..17)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA_i[7:0]								DATA_i+1[7:0]							
rw								rw							

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

Bits	Fields	Descriptions
31:24	DATA_i[7:0]	Data byte i (i = 4*x - 8) Refer to DATA_i+3[7:0] descriptions.
23:16	DATA_i+1[7:0]	Data byte i+1 (i = 4*x - 8) Refer to DATA_i+3[7:0] descriptions.
15:8	DATA_i+2[7:0]	Data byte i+2 (i = 4*x - 8) Refer to DATA_i+3[7:0] descriptions.
7:0	DATA_i+3[7:0]	Data byte i+3 (i = 4*x - 8) Up to 64 bytes can be used for a data frame, depending on the DLC value of the mailbox. For Rx frames, the data received from the CAN bus are stored in this bit field.

Mailbox number

When Rx FIFO is disabled, the dedicated RAM space is occupied by mailboxes only, so the mailbox number is the descriptor number which is incremented by one each time when across the complete mailbox descriptor length (with 8, 16, 32, or 64 data bytes).

When Rx FIFO is enabled (CAN FD mode disabled, data field is 8-byte length), the dedicated RAM space is occupied by both mailboxes and FIFO, so uniformly count the descriptor number by a mailbox descriptor length with 8 data bytes, then the mailbox number is the descriptor number which is occupied by mailbox.

Mailbox size for CAN FD

When CAN FD is enabled, the size of mailboxes that the CAN 512 bytes RAM can be partitioned is configured by MDSZ[1:0] bits in CAN_FDCTL register.

Table 29-5. Mailbox size

MDSZ[1:0]	Payload size in bytes	Mailbox size
0b00	8	32
0b01	16	21
0b10	32	12
0b11	64	7

29.3.2. Rx FIFO descriptor

The Rx FIFO descriptor is shown in ***Table 29-6. Rx FIFO descriptor***.

When RFEN bit in CAN_CTL0 register is 1, the RAM space which normally occupied by mailbox 0–5 with 8 byte payload is used for the Rx FIFO. FDES0 – FDES3 contains the output of the FIFO which is the oldest message that has been received but not yet read by the CPU. The RAM region 0x90-0xDC is reserved for internal use of the FIFO.

When RFEN bit in CAN_CTL0 register is 1, the RAM space which normally occupied by mailbox 6–31 with 8 byte payload is used for the ID filter table (configurable for 8 to up to 104 table elements) for receiving frames matching process into the FIFO. The ID filter table only contains 8 elements from FDES4 to FDES11 by default.

Table 29-6. Rx FIFO descriptor

	31 30 29 28 27 26 25 24 23	22	21	20	19 18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
FDES0	IDFMN[8:0]	SRR	IDE	RTR	DLC[3:0]	TIMESTAMP[15:0]
FDES1	Reserved	ID_STD[10:0]				ID_EXD[17:0]
FDES2	DATA_0[7:0]	DATA_1[7:0]				DATA_2[7:0] DATA_3[7:0]
FDES3	DATA_4[7:0]	DATA_5[7:0]				DATA_6[7:0] DATA_7[7:0]
0x90 - 0xDC	Reserved					
FDES4	ID filter table element 0					
…	…					
FDES107	ID filter table element 103					

FDES0: Rx FIFO descriptor word 0

Address offset: 0x80

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IDFMN[8:0]									SRR	IDE	RTR	DLC[3:0]			
r									r	r	r	r			

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

Bits	Fields	Descriptions
31:23	IDFMN[8:0]	Identifier filter matching number This bit field indicates which ID filter table element matches the received message that is in the output of the Rx FIFO.
22	SRR	Substitute remote request This bit is only used in extended format. It will be stored with the value received on the CAN bus.
21	IDE	ID extended bit This bit specifies whether the frame is standard or extended format. 0: Frame format is standard. 1: Frame format is extended.
20	RTR	Remote transmission request 0: Data frames are accepted

		1: Remote frames are accepted
19:16	DLC[3:0]	Data length code in bytes This bit field is the length (in bytes) of the Rx payload. For reception, this bit field is written by the CAN module with the DLC field of the received frame.
15:0	TIMESTAMP[15:0]	Free-Running counter timestamp This bit field is a copy of the free running counter, captured for Rx frames at the time when the beginning of the ID field appears on the CAN bus.

FDES1: Rx FIFO descriptor word 1

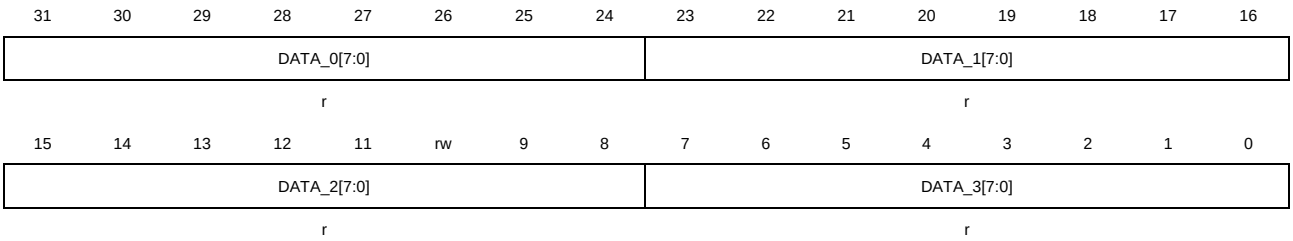
Address offset: 0x84



Bits	Fields	Descriptions
31:29	Reserved	Must be kept at rest value.
28:18	ID_STD[10:0]	Identifier for standard frame In standard frame format, only these 11 most significant bits (28 to 18) are used for frame identification. The 18 least significant bits are ignored.
17:0	ID_EXD[17:0]	Identifier for extended frame In extended frame format, ID_STD[10:0] & these bits are used for frame identification.

FDES2: Rx FIFO descriptor word 2

Address offset: 0x88



Bits	Fields	Descriptions
31:24	DATA_0[7:0]	Data byte 0 Refer to DATA_3[7:0] descriptions.

23:16	DATA_1[7:0]	Data byte 1 Refer to DATA_3[7:0] descriptions.
15:8	DATA_2[7:0]	Data byte 2 Refer to DATA_3[7:0] descriptions.
7:0	DATA_3[7:0]	Data byte 3 Up to 8 bytes can be used for a data frame, depending on the DLC value of the mailbox. FD frames is not supported to receive in Rx FIFO.

FDES3: Rx FIFO descriptor word 3

Address offset: 0x8C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA_4[7:0]								DATA_5[7:0]							
r								r							
15	14	13	12	11	rw	9	8	7	6	5	4	3	2	1	0
DATA_6[7:0]								DATA_7[7:0]							
r								r							

Bits	Fields	Descriptions
31:24	DATA_4[7:0]	Data byte 4 Refer to DATA_7[7:0] descriptions.
23:16	DATA_5[7:0]	Data byte 5 Refer to DATA_7[7:0] descriptions.
15:8	DATA_6[7:0]	Data byte 6 Refer to DATA_7[7:0] descriptions.
7:0	DATA_7[7:0]	Data byte 7 Up to 8 bytes can be used for a data frame, depending on the DLC value of the mailbox. FD frames is not supported to receive in Rx FIFO.

FDESx: Rx FIFO descriptor word x (x = 4..107)

Address offset: 0xE0 + 4 * (x - 4)

This descriptor word shows the three different formats of the ID filter table elements, depending on the configuration of FS[1:0] bits in CAN_CTL0 register.

Note: The format is applied to all ID filter table elements. It is not possible to mix formats within the table.

Format A mode:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RTR_A	IDE_A	Reserved	ID_STD_A[10:0]											Reserved	
			ID_EXD_A[28:16]												

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

Bits	Fields	Descriptions
31	RTR_A	Remote frame for format A This bit specifies whether remote frames can be stored into the FIFO or not when matches. 0: It indicates that remote frames are rejected and data frames can be stored. 1: It indicates that remote frames can be stored and data frames are rejected.
30	IDE_A	ID Extended frame for format A This bit specifies whether extended frames can be stored into the FIFO or not when matches. 0: Extended frames are rejected and standard frames can be stored. 1: Extended frames can be stored and standard frames are rejected.
29	Reserved	Must be kept at rest value.
28:0	ID_STD_A[10:0]/ ID_EXD_A[28:0]	ID in format A This bit field specifies one full standard ID (standard or extended) for Rx FIFO matching process. If IDE_A is 0, the 18 to 28 bits are used for standard ID, and the rest bits are reserved; otherwise, all these bits are used for extended ID.

Format B mode:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RTR_B0	IDE_B0	ID_STD_B_0[10:0]											Reserved		
		ID_EXD_B_0[13:0]													
rw	rw	rw													
15	14	13	12	11	rw	9	8	7	6	5	4	3	2	1	0
RTR_B1	IDE_B1	ID_STD_B_1[10:0]											Reserved		
		ID_EXD_B_1[13:0]													
rw	rw	rw													

Bits	Fields	Descriptions
31	RTR_B0	Remote frame 0 for format B This bit specifies whether remote frames can be stored into the FIFO or not when matches. 0: It indicates that remote frames are rejected and data frames can be stored. 1: It indicates that remote frames can be stored and data frames are rejected.
30	IDE_B0	ID Extended frame 0 for format B

		This bit specifies whether extended frames can be stored into the FIFO or not when matches. 0: Extended frames are rejected and standard frames can be stored. 1: Extended frames can be stored and standard frames are rejected.
29:16	ID_STD_B_0[10:0]/ ID_EXD_B_0[13:0]	ID for frame 0 in format B This bit field specifies a full standard ID or partial 14-bit extended ID for Rx FIFO matching process. If IDE_B0 is 0, the 19 to 29 bits are used for standard ID, and the rest bits are reserved; otherwise, these bits are used for partial 14-bit extended ID, compared with the 14 most significant bits of the received ID.
15	RTR_B1	Remote frame 1 for format B Refer to RTR_B0 descriptions.
14	IDE_B1	ID Extended frame 1 for format B Refer to IDE_B0 descriptions.
13:0	ID_STD_B_1[10:0]/ ID_EXD_B_1[13:0]	ID for frame 1 in format B Refer to ID_STD_B_0[10:0]/ ID_EXD_B_0[13:0] descriptions.

Format C mode:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ID_C_0[7:0]								ID_C_1[7:0]							
rw								rw							

15	14	13	12	11	rw	9	8	7	6	5	4	3	2	1	0
ID_C_2[7:0]								ID_C_3[7:0]							
rw								rw							

Bits	Fields	Descriptions
31:24	ID_C_0[7:0]	ID for frame 0 in format C This bit field specifies a partial 8-bit standard ID or partial 8-bit extended ID for Rx FIFO matching process. In both standard and extended frame formats, the 8 bit field is compared with the 8 most significant bits of the received ID.
23:16	ID_C_1[7:0]	ID for frame 1 in format C Refer to ID_C_0[7:0] descriptions.
15:8	ID_C_2[7:0]	ID for frame 2 in format C Refer to ID_C_0[7:0] descriptions.
7:0	ID_C_3[7:0]	ID for frame 3 in format C Refer to ID_C_0[7:0] descriptions.

29.3.3. Communication modes

The CAN interface has four communication modes:

- Normal mode
- Inactive mode
- Loopback and silent mode
- Monitor mode

Normal mode

In normal mode, the message reception and transmission, and errors are all managed normally, and all CAN protocol functions are enabled.

Inactive mode

To enter Inactive mode, set INAMOD bit in CAN_CTL0 to 1 to enable Inactive mode, then set HALT bit in CAN_CTL0 register to 1 or put the chip into Debug mode.

When Inactive mode is requested, the following steps are performed before INAS bit asserted:

1. Wait for the bus 11 consecutive recessive bits.
2. Wait for the current transmission or reception processes being finished, it means all internal activities such as arbitration, matching, shift-in, and shift-out being finished. A pending shift-in does not prevent entering Inactive mode.
3. The Tx pin is driven as '1' (recessive).
4. Stop the prescaler.
5. Enable write access to the CAN_ERR0 register, which is read-only in other modes.
6. Set NRDY bit and INAS bit in CAN_CTL0 register.

When Inactive mode is entered, INAS bit in CAN_CTL0 register is set to 1 by CAN.

In Inactive mode, neither transmission nor reception is performed, and its prescaler is stopped, all registers are accessible.

To exit from Inactive mode, one of the following methods can meet:

- Clear INAMOD bit in CAN_CTL0.
- Clear HALT bit in CAN_CTL0 register, or the chip is removed from Debug mode.

If exiting from Inactive mode is requested, then INAS bit in CAN_CTL0 register is cleared after the CAN prescaler is running again. When out of Inactive mode, CAN module tries to resynchronize to the CAN bus by waiting for 11 consecutive recessive bits.

Note: When in Inactive mode, the CAN_Disable mode request, or the Pretended Networking mode request will lead to INAS bit in CAN_CTL0 register be cleared and LPS bit in CAN_CTL0 register be set.

Loopback and silent mode

To enter this mode, set the LSCMOD bit in CAN_CTL1 register to 1. In this mode, the messages are internally transmitted back to the receiver input, and the bit sent during the ACK slot in the frame acknowledge field is ignored to ensure reception transmitted by itself. Both transmit and receive interrupts are generated.

Loopback and silent mode is used for self-test. The Rx pin is ignored and the Tx pin holds in recessive state.

Monitor mode

To enter this mode, set MMOD bit in CAN_CTL1 register to 1.

When Monitor mode is entered, ERRSI[1:0] bit field in CAN_ERR1 register is set to 0b01 by CAN to indicate that the module works in an Error Passive state. In this mode, error counters are frozen for transmission and reception.

In Monitor mode, no transmission is performed and reception is performed only when messages are acknowledged by other CAN nodes, detection of a message that has not been acknowledged will lead to a bit dominant error flag (without changing the RECNT[7:0] or REFCNT[7:0] in CAN_ERR0 register).

29.3.4. Power saving modes

The CAN interface has two power saving modes:

- CAN_Disable mode.
- Pretended Networking mode.

In these two power saving modes, the dedicated RAM and the registers in SRAM can not be accessed.

CAN_Disable mode

The CAN module is enabled or disabled by configuring the CANDIS bit in CAN_CTL0 register.

For power saving, if CANDIS bit is set to 1 to disable CAN module, the CAN module will enter CAN_Disable mode after a delay when the LPS bit and NRDY bit in CAN_CTL0 register are changed to 1.

When CAN is disabled, the clocks to the Protocol controller and the Controller Interface are disabled. All registers except the CAN_RMPUBF, CAN_RFIFOPUBF, CAN_RFIFOIFMN, and CAN_RFIFOMPFx (x=0..31) registers are accessible. Also the dedicated RAM can not be accessed.

After CAN is enabled, you need to dalay a time to wait for LPS bit in CAN_CTL0 register to be cleared for Protocol controller recognization. When CAN is enabled, CAN module requests to resume the clocks to the Protocol controller and the Controller Interface.

Pretended Networking mode

Pretended Networking mode is used to receive wakeup messages with low power consumption. This mode can work together with MCU deepsleep mode.

To enter Pretended Networking mode, set PNEN bit and PNMOD bit in CAN_CTL0 register to 1, and optionally put the MCU into deepsleep mode.

When Pretended Networking mode is requested, the following steps are performed:

1. Wait for the bus to be in Idle state, or else wait for the third bit of Intermission, and then checks it to be recessive.
2. Set LPS bit and PNS bit in CAN_CTL0 register.
3. Request to disable the Controller Interface clock, while keeping the Protocol controller clock running.

In Pretended Networking mode, Controller Interface clock is disabled and Protocol controller is kept clocked (if the MCU works in deepsleep mode, the clock of CAN Protocol Controller should be configured as IRC8M in advance, otherwise the the clock of CAN Protocol Controller will be lost), so that the reception process is still active to filter messages. The matching, arbitration, shift-in and shift-out processes are not performed in Pretended Networking mode.

To exit from Pretended Networking mode, the following method can meet:

- When a wakeup event is detected, and a wake up interrupt is occured. Clear LPS bit and PNS bit in CAN_CTL0 register.
- Clear LPS bit and PNS bit in CAN_CTL0 register.

If exiting from Pretended Networking mode is requested, CAN module will wait for the bus to be in Idle state or else wait for the third bit of Intermission to clear LPS bit and PNS bit in CAN_CTL0 register, and resume normal mode, CAN module will be synchronized to the CAN bus.

29.3.5. Data transmission

For transmission, an arbitration mechanism decides whether the Tx mailbox transmission priority is depending on the message ID (the PRIO field can also be configured to participate in arbitration), or on the mailbox number.

The quantity of mailboxes in CAN FD format is determined by MDSZ[1:0] bits in CAN_FDCTL register, refer to ***Table 29-5. Mailbox size***.

Transmit process

To transmit a CAN frame, a Tx mailbox must be prepared for transmission in following steps:

1. Check whether the corresponding mailbox state MSx bit in CAN_STAT register is set and clear it.

2. If the mailbox is active (either Tx or Rx), inactivate the mailbox by method described in ***Tx mailbox inactivation*** or ***Rx mailbox inactivation***, when Tx mailbox inactivation is performed, do the following steps, when Rx mailbox inactivation is performed, go to step 6. While if the mailbox is inactive (either Tx or Rx), go to step 6.
3. Poll the the corresponding MSx bit in CAN_STAT register to be set, or by the interrupt when MIEx bit in CAN_INTEN register is set.
4. Read back the CODE field to get the state of the mailbox (aborted, or transmitted).
5. Clear the corresponding flag MSx in the CAN_STAT register.
6. Write mailbox ID field (plus the mailbox PRIO field if LAPRIOEN bit in CAN_CTL0 register is set to 1) of the MDES1 word.
7. Write payload data bytes in mailbox DATA field of MDESx (x = 2..17) word.
8. Configure the mailbox IDE, RTR, FDF, BRS, ESI, and DLC field to MDES0 word.
9. Activate the mailbox to transmit the frame by setting mailbox CODE field to 0b1100. When the mailbox is activated, it participates in the arbitration process and is eventually transmitted according to its priority. When the mailbox payload size is less than the mailbox DLC value, CAN adds the necessary number of bytes with constant 0xCC to meet the expected DLC.

Upon a successful transmission, the CODE field is automatically updated, and the TIMESTAMP field is automatically updated with the value of the free running counter; the CRC registers (CAN_CRCC and CAN_CRCCFD) are updated, and the corresponding flag MSx in the CAN_STAT register is set, if the interrupt enable bit MIEx in CAN_INTEN register is set, an interrupt will be generated.

Arbitration process

When more than one Tx mailbox is pending, the arbitration process which searching from the lowest number mailbox to the higher ones will give the transmission order. The arbitration algorithm is controlled by the MTO bit in CAN_CTL1 register.

The arbitration process starts when matching one of the following situations:

- The CRC field on CAN bus: number of ASD[4:0] (in CAN_CTL2 register) CAN bits delay after the first bit of the CRC field.
- The Error or Overload Delimiter field on CAN bus.
- CAN bus is recovering from Bus Off state: number of ASD[4:0] (in CAN_CTL2 register) CAN bits delay after the counter TECNT[7:0] counted to 124. Recovering from Bus Off state needs 128 times of 11 continuous recessive bits, which is counted by TECNT[7:0] in CAN_ERR0 register.
- Exit from Inactive mode, or power saving mode (including CAN_Disable mode and Pretended Networking mode).
- Rewrite of MDES0 word of arbitration winner (temporary winner or final winner).
- Rewrite to MDES0 word of the scanned mailbox (arbitration is on-going): if no arbitration winner is found when scan finished, arbitration will restart at soon; otherwise, the arbitration process is finished.
- Write to MDES0 word of a mailbox: when no arbitration is processing, and no arbitration

winner exists, and the CAN bus is not in SOF-DATA field / SOF-Control field of a data / remote frame or Error / Overload flag field of an Error / Overload frame.

- CAN node enters Bus Integration state (refer to ***Bus integration state***): Number of ASD[4:0] (in CAN_CTL2 register) CAN bits delay after the state.

Arbitration process stops when matching one of the following situations:

- All mailboxes are scanned.
- A Tx active mailbox is found when MTO bit in CAN_CTL1 register is set to 1 (lowest-number mailbox first).
- The Error or Overload flag field on CAN bus.
- The SOF field of the next frame on CAN bus.
- When Inactive mode, CAN_Disable mode or Pretended Networking mode is requested.

Lowest-number mailbox first

If MTO bit in CAN_CTL1 register is set to 1, the lowest number Tx mailbox is transmitted first, and LAPRIOEN bit in CAN_CTL0 register has no effect.

Highest-priority mailbox first

If MTO bit in CAN_CTL1 register is set to 0, then the Tx mailbox with the highest priority is transmitted first. The highest priority Tx mailbox is the one that has the lowest arbitration value (refer to ***Table 29-7. Mailbox arbitration value(32 bit) when local priority disabled*** and ***Table 29-8. Mailbox arbitration value(35 bit) when local priority enabled***) among all Tx mailboxes. If more than one mailboxes have equivalent arbitration values, the mailbox with the lowest number is the arbitration winner.

When LAPRIOEN bit in CAN_CTL0 register is set to 1, the local priority is disabled, the bits participate in the internal arbitration process are exactly what will be transmitted to the CAN bus, shown in ***Table 29-7. Mailbox arbitration value(32 bit) when local priority disabled***.

When LAPRIOEN bit in CAN_CTL0 register is set to 1, the local priority is enabled, then the mailbox PRIO field will participate in the internal arbitration process. Shown in ***Table 29-8. Mailbox arbitration value(35 bit) when local priority enabled***, the mailbox PRIO field is the most significant part of the arbitration value, thus mailboxes with low PRIO values have higher priority regardless of the rest of their arbitration values, while the PRIO field will not be transmitted to the CAN bus.

Table 29-7. Mailbox arbitration value(32 bit) when local priority disabled

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

Table 29-8. Mailbox arbitration value(35 bit) when local priority enabled

IDE	34 33 32	31 30 29	28 27 26 25 24	23 22 21	20	19	18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1	0
0	PRIO[2:0]	ID_STD[10:0]			RTR	IDE	Reserved	
1	PRIO[2:0]	ID_EXD[28:18]			SRR	IDE	ID_EXD[17:0]	RTR

Arbitration start delay

Arbitration start delay is configured by ASD[4:0] bits in CAN_CTL2 register to optimize the transmission performance when the arbitration process ends too early to give a chance for the CPU to overwrite the winner Tx mailbox, thus the arbitration is restarted and may not be able to transmit in time.

Shift-out

The shift-out process is the copy operation of the content from a Tx mailbox to the Tx shift buffer (an internal mailbox descriptor) after the arbitration winner is found. The message on the Tx shift buffer is transmitted according to the CAN protocol rules.

When the shift-out process is done, write access to the MDES0 word of the corresponding mailbox is blocked even if MST bit in CAN_CTL0 register is set. The write access to MDES0 word of the corresponding mailbox is recoverd when matching one of the following situations:

- After the mailbox is transmitted and the corresponding flag MSx in the CAN_STAT register is cleared by the CPU.
- CAN node enters Inactive mode or Bus Off state.
- CAN node loses the bus arbitration or there is an error during the transmission.

The shift-out process starts when matching one of the following situations:

- The first bit of Intermission field on CAN bus.
- During Bus Idle state.
- During Wait For Bus Idle state.

During the shift-out process, the CPU has priority to access the corresponding memory in Bus Idle state, and the shift-out operation has the lowest priority to access the corresponding memory.

Abort

To request an abort of the transmission, the recommended operation is setting MST bit in the CAN_CTL0 register to 1, then writing ABORT (0b1001) to the CODE field of the mailbox.

The writing of ABORT (0b1001) to the mailbox MDES0 word is successfully when the mailbox is not the arbitration winner, or when the mailbox is the arbitration winner but the shift-out of the mailbox is not finished yet. In this situation, the corresponding MSx bit in CAN_STAT

register will be set.

The writing of ABORT (0b1001) to the mailbox MDES0 word is blocked when shift-out of the mailbox is already finished, or when the mailbox is being transmitted. In this situation, the abort request is captured and kept pending until the frame is transmitted successfully or failed:

- The frame is transmitted successfully, the mailbox is not aborted: If the frame is transmitted successfully, the pending abort request will be cleared automatically, the corresponding MSx bit in CAN_STAT register will be set, and an interrupt will occur when MIEx bit in CAN_INTEN register is set.
- The frame is transmitted failed, the mailbox is aborted: If the frame failed to be transmitted, the pending abort request is responsed, the write access to the mailbox is recoverd, with ABORT code written to the mailbox MDES0 word, the corresponding MSx bit in CAN_STAT register will be set, and an interrupt will occur when MIEx bit in CAN_INTEN register is set.
  When matching one of the following situations, the frame failed to be transmitted:
  - Lose the bus arbitration.
  - There is an error during the transmission.
  - Enter Bus Off state.
  - There is an overload frame.

Tx mailbox inactivation

The way to inactivate a Tx mailbox:

- Write the CODE field of the Tx mailbox MDES0 with ABORT. This is the recommended way for inactivation, which will not cause the unknown transmission.
  This operation must be done when MST bit in the CAN_CTL0 register is 1.

29.3.6. Data reception

For Classical CAN frames, reception through FIFO and mailbox are both supported.

For CAN FD frames, reception is only supported through mailbox.

Mailbox reception

For mailbox reception, a received frame will be stored into the mailbox only when the frame ID matches the mailbox ID programmed in the ID field (or the mailbox ID group when the filter registers are applied).

To receive a CAN frame into a mailbox, a mailbox must be prepared for reception in following steps:

1. If the mailbox is active (either Tx or Rx), inactivate the mailbox by method described in ***Tx mailbox inactivation*** or ***Rx mailbox inactivation***, when Tx mailbox inactivation is performed, do the following steps, when Rx mailbox inactivation is performed, go to step 4. While if the mailbox is inactive (either Tx or Rx), go to step 4.

2. Poll the the corresponding MSx bit in CAN_STAT register to be set, or by the interrupt when MIEx bit in CAN_INTEN register is set.
3. Read back the CODE field to make sure that the mailbox is aborted, or transmitted.
4. Clear the corresponding flag MSx in the CAN_STAT register.
5. Write the mailbox ID field of MDES1 word, and write IDE, RTR field of MDES0 word if needed.
6. Write the EMPTY (0b0100) to the CODE field of MDES0 word to activate the mailbox.

Upon a successful reception, all bits of the mailbox descriptor (DATA, ID, TIMESTAMP, SRR, IDE, RTR, FDF, BRS, ESI, DLC, CODE) are stored with the received data field or automatically updated, and the corresponding flag MSx in the CAN_STAT register is set, if the interrupt enable bit MIEx in CAN_INTEN register is set, an interrupt will be generated. The TIMESTAMP field is automatically updated with the value of the free running counter at the time of the second bit of frame's ID field.

To service (read) a Rx mailbox, the recommended steps are shown as below:

1. Poll the corresponding flag MSx in the CAN_STAT register to be set, or by the interrupt when MIEx bit in CAN_INTEN register is set.
2. Read the mailbox MDES0 word, and poll until the BUSY bit (in CODE field) is 0. When the BUSY bit is 0, the read operation of the mailbox will lock the mailbox, so that to prevent the mailbox being overwritten.
3. Read the contents of the mailbox.
4. Clear the corresponding flag MSx in the CAN_STAT register.
5. Read CAN_TIMER register to unlock the mailbox.

Rx mailbox locking

A locking mechanism is only applied for Rx mailbox: For CODE field with Rx FULL or Rx OVERRUN, a CPU read to the mailbox MDES0 word will lock the mailbox, thus the locking will prevent a new matching frame overwriting it.

The locking will be released when reading the CAN_TIMER register (global unlocking operation), or when MDES0 word of any other mailbox is read. When unlocked, a shift-in process will start with the pending message (the same in Inactive mode, while when LPS bit in CAN_CTL0 register is 1, the shift-in process will be delayed until LPS bit changes to 0).

If the mailbox is not unlocked in time, while a new matching frame is coming, then the new frame will overwrite the Rx shift buffer without a notification of a lost message, and no error is recorded.

Note: Mailbox inactivation (write CODE with Rx INACTIVE, or Tx ABORT) has higher priority than locking.

Rx mailbox inactivation

The way to inactivate a Rx mailbox:

- Write the CODE field of the Rx mailbox MDES0 with INACTIVE (Tx INACTIVE or Rx

INACTIVE). But this operation may lead to a result that a matched message lost without notice.

Note: The Rx mailbox inactivation will automatically unlocks the mailbox. There is no write protection for Rx FIFO.

Rx FIFO reception

The Rx FIFO is 6-message deep. When RFEN bit in CAN_CTL0 register is 1, Rx FIFO is enabled for reception. Rx FIFO can only be used for reception, and must not be enabled when CAN FD mode is enabled. The Rx FIFO descriptor refers to ***Rx FIFO descriptor***. There is a powerful filter system provided to filter a group of identifiers, reducing the interrupt servicing workload. The number of Rx FIFO filters is configured in RFFN[3:0] bits of CAN_CTL2 register, up to 32 filters are configured in CAN_RFIFOMPFx (x = 0..31) registers (if RPFQEN bit in CAN_CTL0 register is 1), or CAN_RFIFOPUBF and CAN_RFIFOMPFx (x = 0..31) registers (if RPFQEN bit in CAN_CTL0 register is 0).

Rx FIFO has unread messages: Only when MS5_RFNE bit in CAN_STAT register is 1, the FDES0-FDES3 words are valid to be read to get the received message. When MS5_RFNE bit in CAN_STAT register is 1, it means there is at least one frame available to be read from Rx FIFO. If the interrupt enable bit MIEx in CAN_INTEN register is set, an interrupt will be generated; while if DMAEN bit in CAN_CTL0 register is set, the MS5_RFNE flag will generate the DMA request and no Rx FIFO interrupt is generated.

- To service (read) Rx FIFO by CPU, the recommended steps are shown as below:

  1. Poll the flag MS5_RFNE in the CAN_STAT register to be set, or by the interrupt when MIE5 bit in CAN_INTEN register is set.
  2. Read the Rx FIFO FDES0-FDES3 words, and if needed read CAN_RFIFOIFMN register.
  3. Clear the flag MS5_RFNE in the CAN_STAT register. If there are more than one messages in the Rx FIFO, the act of clearing the flag will update the Rx FIFO FDES0-FDES3 words with the next message, and CAN_RFIFOIFMN register will be updated at the same time, the flag MS5_RFNE remains set, and an interrupt occurs again if enabled, repeat step 2 and 3 to get the received messages.

- To service (read) Rx FIFO by DMA controller, the recommended operation are shown as below :

  1. Configure and enable the DMA controller for Rx FIFO reception.
  2. Service (read) Rx FIFO by CPU until the flag MS5_RFNE in the CAN_STAT register is cleared, to avoid an additional DMA request after DMA mode is enabled.
  3. Set DMAEN bit in CAN_CTL0 register to 1 to enable DMA mode.
  4. Wait for the DMA request. When the flag MS5_RFNE in the CAN_STAT register is set, a DMA request is generated.
  5. Upon receiving the DMA request, the DMA will read the Rx FIFO FDES0 to FDES3. FDES3 word must be read to clear the flag MS5_RFNE in the CAN_STAT register,

if there are more than one messages in the Rx FIFO, the act of reading FDES3 word will update the Rx FIFO FDES0-FDES3 words with the next message, and CAN_RFIFOIFMN register (should be read before FDES3) will be updated at the same time, the flag MS5_RFNE remains set, and a DMA request is generated again. Steps 4 and 5 are repeated.

DMA mode

DMA mode is supported for Rx FIFO reception when RFEN bit and DMAEN bit in CAN_CTL0 register are both set. When the DMA mode is enabled, the CPU must not read the Rx FIFO.

When DMA mode is enabled, Rx FIFO FDES0 to FDES3 words will be read by DMA controller when there is unread message in Rx FIFO, to get the received message. In this mode, Rx FIFO warning flag MS6_RFW bit and Rx FIFO overflow flag MS7_RFO bit in CAN_STAT register are reserved.

Before disabling DMA mode by clearing DMAEN bit in CAN_CTL0 register, a clear FIFO operation (when RFEN bit in CAN_CTL0 register is 1, set MS0 in the CAN_STAT register to 1 in Inactive mode) must be performed to clear the Rx FIFO contents. The act of clearing FIFO will clear MS5_RFNE bit in the CAN_STAT register, and cancel the DMA request.

Clear FIFO

when Rx FIFO is enabled (RFEN bit in CAN_CTL0 register is 1), set MS0 bit in the CAN_STAT register to 1 in Inactive mode will clear the Rx FIFO contents, while the Rx FIFO flags will not be cleared (except in DMA mode), thus before clearing FIFO operation, the Rx FIFO must be serviced until the flag MS5_RFNE in the CAN_STAT register is cleared.

Flag

Rx FIFO not empty

When MS5_RFNE bit in CAN_STAT register is 1, it means there is at least one frame available to be read from Rx FIFO.

Rx FIFO warning

When MS6_RFW bit in CAN_STAT register is 1, it means the number of unread messages within the Rx FIFO is increased to five from four due to the reception of a new one, the Rx FIFO is almost full.

Rx FIFO overflow

When MS7_RFO bit in CAN_STAT register is 1, it means there is an incoming message lost because the Rx FIFO is full.

Matching process

The matching process is searching for a Rx mailbox or Rx FIFO (when enabled) with an ID matching with the frame ID on CAN bus, also, the IDE and RTR field will participate in the

matching.

The matching process starts when completes the DLC field reception.

The matching of mailboxes is affected by the RPFQEN bit. If the RPFQEN bit is 0, the first mailbox to be matched is the match winner, regardless of whether it is free-to-receive or non-free-to-receive. If the RPFQEN bit is 1, the first free-to-receive mailbox matched is the winner or the last non-free-to-receive mailbox matched is the winner.

Searching process

- When the Rx FIFO is enabled, the RFO bit in CAN_CTL2 register gives the searching order.
  - If RFO bit is set to 1, matching process starts from Rx mailbox to Rx FIFO. The Rx mailbox is searched from the lowest number mailbox to the higher ones.
    Firstly, searching for a matching mailbox that is available for receiving. If the RPFQEN bit is 0, the first mailbox that matches is the winner, regardless of whether it is free-to-receive or non-free-to-receive. If the RPFQEN bit is 1, the first free-to-receive mailbox matched is the match winner. Rx FIFO is no longer searched in any of the above cases.
    Secondly, If the RPFQEN bit is 1 and no free-to-receive mailbox is matched, but a matching non-free-to-receive mailbox is found, then the Rx FIFO is also searched to determine the winner. when the Rx FIFO is matched and is not full, the Rx FIFO is the matching winner; otherwise, the matching winner is the last non-free-to-receive matched Rx mailbox(leading to mailbox CODE OVERRUN).
    Thirdly, if no matched Rx mailbox is found (means no free-to-receive matched mailbox, nor non-free-to-receive matched mailbox), then matching is processed on Rx FIFO. In this case, if the Rx FIFO is matched but it is full, it will leads to Rx FIFO overflow, while if the Rx FIFO is not matched (no matter it is full or not), the message will not be received.
  - If RFO bit is set to 0, matching process starts from Rx FIFO to Rx mailbox. If the Rx FIFO matches the searching conditions, and is not full, then the Rx FIFO is the matching winner, regardless of searching for mailboxes. If Rx FIFO does not match or it is full, then matching is processed on Rx mailbox. The matching of mailboxes is affected by the RPFQEN bit. If the RPFQEN bit is 0, the first mailbox to be matched is the match winner, regardless of whether it is free-to-receive or non-free-to-receive. If the RPFQEN bit is 1, the first free-to-receive mailbox matched is the winner. If no free-to-receive mailbox is found, the last non-free-to-receive mailbox matched is the winner.
- When the Rx FIFO is disabled, the matching process only searches the Rx mailboxes. Refer to the mailbox matching description above.

A free-to-receive Rx mailbox can be:

- For a data frame reception, or a remote frame reception when RRFRMS bit in CAN_CTL2 register is 1, it can be: A mailbox with CODE field EMPTY; A mailbox with CODE field FULL or OVERRUN, which has already been serviced (read) and unlocked.

- For a remote frame reception when RRFRMS bit in CAN_CTL2 register is 0, it can be a mailbox with CODE field RANSWER.

Searching conditions for matched Rx mailbox

Searching conditions for matched Rx mailbox, refers to ***Table 29-9. Rx mailbox matching***:

- When the frame in Rx shift buffer is a data frame (RTR field is 0), Rx mailbox with CODE EMPTY, FULL, and OVERRUN will be searched:
  - When IDERTR_RMF bit in CAN_CTL2 register is 0, it means the IDE field will be compared and RTR field will not be compared (regardless of bit 30 and bit 31 in related filter register). The ID field will be compared, using bit 0 to bit 28 filter data configurations in related filter register.
  - When IDERTR_RMF bit in CAN_CTL2 register is 1, it means all the IDE, RTR and ID fields will be compared, using bit 0 to bit 28, bit 30, bit 31 filter data configurations in related filter register.
- When the frame in Rx shift buffer is a remote frame (RTR field is 1):
  - If RRFRMS bit in CAN_CTL2 register is 0, it indicates that the Rx mailbox with CODE RANSWER will be searched, and the IDE, ID field will be compared, using bit 0 to bit 28, bit 30 filter data configurations in related filter register.
  - If RRFRMS bit in CAN_CTL2 register is 1, it indicates that the matching process is the same as a data frame, so Rx mailbox with CODE EMPTY, FULL, and OVERRUN will be searched:
    When IDERTR_RMF bit in CAN_CTL2 register is 0, it means the IDE field will be compared and RTR field will not be compared (regardless of bit 30 and bit 31 in related filter register). The ID field will be compared, using bit 0 to bit 28 filter data configurations in related filter register.
    When IDERTR_RMF bit in CAN_CTL2 register is 1, it means all the IDE, RTR and ID fields will be compared, using bit 0 to bit 28, bit 30, bit 31 filter data configurations in related filter register.

Table 29-9. Rx mailbox matching

Received bit	Configuration bit		Field in mailbox descriptor for matching			
RTR	IDERTR_RMF (in CAN_CTL2 register)	RRFRMS (in CAN_CTL2 register)	IDE	RTR	ID	CODE
0	0	-	Compared(1)	Never(2)	Filtered(3)	EMPTY / FULL / OVERRUN
	1		Filtered			EMPTY / FULL / OVERRUN
1	-	0	Compared	Never	Compared	RANSWER
	0	1	Compared	Never	Filtered	EMPTY / FULL / OVERRUN
	1		Filtered			EMPTY / FULL /

				OVERRUN

(1) Compared: This field in Rx mailbox descriptor is always compared with the received bit, regardless of the filter data configurations in related filter register.
(2) Never: This field in Rx mailbox descriptor is not compared with the received bit, regardless of the filter data configurations in related filter register.
(3) Filtered: This field in Rx mailbox descriptor is compared with the received bit, using the filter data configurations in related filter register.

Searching conditions for matched Rx FIFO

Searching conditions for matched Rx FIFO, refers to ***Table 29-10. Rx FIFO matching***:

- If the FS[1:0] bits in CAN_CTL0 register is 0 or 1, it means A or B format of filter table is adopted, then all the IDE, RTR and ID fields will be compared, using bit 0 to bit 31 filter data configurations in related filter register.
- If the FS[1:0] bits in CAN_CTL0 register is 2, it means C format of filter table is adopted, then the IDE, RTR will not be compared (no these fields in FIFO descriptor) and ID fields will be compared, using bit 0 to bit 31 filter data configurations in related filter register.
- If the FS[1:0] bits in CAN_CTL0 register is 3, it means D format of filter table is adopted, then all frames are rejected.

Table 29-10. Rx FIFO matching

Configuration bit	Field in Rx FIFO descriptor for matching		
FS[1:0] (in CAN_CTL0 register)	IDE	RTR	ID
0	Filtered		
1	Filtered		
2	Never		Filtered
3	Not match(1)		

(1) Not match: All frames are rejected.

Shift-in

The shift-in process is the copy operation of the content from a Rx shift buffer (an internal mailbox descriptor) to a Rx mailbox or Rx FIFO that matched it.

When there is a matching descriptor found in the FIFO or in the Rx mailboxes, a shift-in process will be pending. The pending shift-in process starts to transfer when meets all of the following conditions:

- There is a matching winner for the frame in the Rx shift buffer.
- The CAN bus is in:
  - The second bit of Intermission field.
  - The first bit of an Overload frame.
- The target mailbox is not locked.

When the target mailbox of a pending shift-in process is unlocked during Inactive mode, the pending shift-in process starts to transfer. While if the unlocking occurs when LPS bit in CAN_CTL0 register is 1, the pending shift-in process will still be delayed until LPS bit changes to 0.

When the shift-in process is on-going, the BUSY bit (CODE[0]) of the target mailbox is set to indicate that the mailbox is being updated.

The shift-in process can be cancelled for a Rx mailbox, but can not be cancelled for the Rx FIFO. The shift-in process will be cancelled when matching one of the following situations:

- The target mailbox is inactivated after the CAN bus has reached the first bit of Intermission field next to the frame that carried the message and its matching process has finished.
- The Rx shift buffer receives a message transmitted by itself while the self reception is disabled by setting SRDIS bit in CAN_CTL0 register.
- There is a CAN protocol error.

When the shift-in process is done, Rx mailbox descriptor or Rx FIFO descriptor (if Rx FIFO is enabled) will be updated with the received message, and CAN_RFIFOIFMN will be updated if shift-in to the Rx FIFO, CODE field of Rx mailbox descriptor will be updated if shift-in to the Rx mailbox.

Filter data configuration

When Rx FIFO is disabled:

- If RPFQEN bit in CAN_CTL0 register is 0, then CAN_RMPUBF is used for all mailboxes.
- If RPFQEN bit in CAN_CTL0 register is 1, then CAN_RFIFOMPFx (x = 0..31) is used for mailboxes individually.

When Rx FIFO is enabled:

- If RPFQEN bit in CAN_CTL0 register is 0, then CAN_RMPUBF is used for all mailboxes, CAN_RFIFOPUBF and CAN_RFIFOMPFx (x = 0..31) are used for all the Rx FIFO ID filter table elements, and the value of these registers must be all the same.
- If RPFQEN bit in CAN_CTL0 register is 1, then CAN_RFIFOMPFx (x = 0..31) is used for the Rx FIFO ID filter table elements defined by RFFN[3:0] bits in CAN_CTL2 register and the mailboxes individually (because the Rx FIFO descriptor and the Rx mailbox descriptors can not occupy the same RAM space at the same time), CAN_RFIFOPUBF is used for the Rx FIFO ID filter table elements of the rest.

Self reception

When SRDIS bit in CAN_CTL0 register is 1, self reception is disabled, thus the frames transmitted by itself will not be received even if there is a matched Rx mailbox or Rx FIFO is matched, and no flag or interrupt will be generated. When the SRDIS bit is 0, it is allowed to receive a matching frame sent by itself.

29.3.7. Data reception in Pretended Networking mode

When PNEN bit and PNMOD bit in CAN_CTL0 register are configured to 1,the Pretended Networking mode is enabled, then the CAN is able to process received messages in MCU sleep mode. A wakeup event will wake up the CAN module from the Pretended Networking mode.

There are four groups of registers used for matched message storage: CAN_PN_RWMxCS, CAN_PN_RWMxI, CAN_PN_RWMxD0 and CAN_PN_RWMxD1 registers, group x from 0 to 3. Therefore, four messages can be stored at most (when NMM[7:0] bits in CAN_PN_CTL0 register is larger than or equal to 4), and only the latest messages will be stored. The group x indicates the message arrival order. If NMM[7:0] is less than 4, only NMM[7:0] value of messages can be stored, at groups from 0 to NMM[7:0] minus 1.

When the data length of the frame to be stored is less than 8 bytes, the padding values which continued to the received DATA field to be written to the CAN_PN_RWMxD0 and CAN_PN_RWMxD1 registers (x = 0..3) are zeroes. No timestamp is stored for wakeup matched frames.

Note: When in Pretended Networking mode, CAN FD format messages are ignored.

Wakeup interrupt

There are two types of wakeup interrupt events, including wakeup match event, and wakeup timeout event. Each interrupt event has a dedicated flag bit in the CAN_PN_STAT register, and a dedicated enable bit in CAN_PN_CTL0 register. The relationship is described in the ***Table 29-11. Interrupt events***.

An wakeup interrupt can be generated when any type of the wakeup interrupt event occurs and enabled.

Wakeup timeout event

When CAN reaches the timeout value, a wakeup timeout event will occur. The timeout is configured by WTO[15:0] bits in CAN_PN_TO register.

Note: Even if the timeout value is reached, CAN module will not stop the message filtering process until the CPU wakes up.

Wakeup match event

When CAN receives the matched wakeup frame/frames within the timeout, the wakeup match event will occur. MMCNT[7:0] bits in CAN_PN_STAT register reflects the number of matched messages from the time of entering Pretended Networking mode to the time the CPU wakes up.

Note: Even if CAN receives the matched wakeup frame/frames, the timeout counter will not stop counting until the CPU wakes up.

Frame matching

The fields of frame participate in the wakeup matching process are IDE, RTR, ID, DLC, and DATA field.

- When FFT[1:0] bit field in CAN_PN_CTL0 register is configured to 0, it means a wakeup match event occurs when a frame is received with all fields except DATA field, DLC field (that is IDE, RTR, and ID field matched) matched.
- When FFT[1:0] bit field in CAN_PN_CTL0 register is configured to 1, it means a wakeup match event occurs when a frame is received with all fields (that is IDE, RTR, ID, DLC, and DATA field matched) matched.
- When FFT[1:0] bit field in CAN_PN_CTL0 register is configured to 2, it means a wakeup match event occurs when a specified number (configured by NMM[7:0] bits in CAN_PN_CTL0 register) of frames are received with all fields except DATA field, DLC field (that is IDE, RTR, and ID field matched) matched.
- When FFT[1:0] bit field in CAN_PN_CTL0 register is configured to 2, it means a wakeup match event occurs when a specified number (configured by NMM[7:0] bits in CAN_PN_CTL0 register) of frames are received with all fields (that is IDE, RTR, ID, DLC, and DATA field matched) matched.

IDE field matching

The IDE field of a matched message is the same as the configured expected IDE field in CAN_PN_EID0 register, using filter data in CAN_PN_IFEID1 register.

RTR field matching

The RTR field of a matched message is the same as the configured expected RTR field in CAN_PN_EID0 register, using filter data in CAN_PN_IFEID1 register.

ID field matching

- When IDFT[1:0] bit field in CAN_PN_CTL0 register is configured to 0, it means the ID field of a matched message is the same as the configured expected ID field in CAN_PN_EID0 register, using filter data in CAN_PN_IFEID1 register.
- When IDFT[1:0] bit field in CAN_PN_CTL0 register is configured to 1, it means the ID field of a matched message is larger than or equal to the configured expected ID field in CAN_PN_EID0 register. CAN_PN_IFEID1 register is not used.
- When IDFT[1:0] bit field in CAN_PN_CTL0 register is configured to 2, it means the ID field of a matched message is smaller than or equal to the configured expected ID field in CAN_PN_EID0 register. CAN_PN_IFEID1 register is not used.
- When IDFT[1:0] bit field in CAN_PN_CTL0 register is configured to 3, it means the ID field of a matched message is larger than or equal to the configured expected ID field in CAN_PN_EID0 register, and is smaller than or equal to the configured expected ID field in CAN_PN_IFEID1 register.

DLC field matching

- The DLC field of a matched message is larger than or qual to the configured expected DLC low threshold DLCELT[3:0] in CAN_PN_EDLC register, and lower than or qual to the configured expected DLC high threshold DLCEHT[3:0] in CAN_PN_EDLC register.

DATA field matching

- When DATAFT[1:0] bit field in CAN_PN_CTL0 register is configured to 0, it means the DATA field of a matched message is the same as the configured expected DATA field in CAN_PN_EDLx (x = 0,1) registers, using filter data in CAN_PN_DF0EDH0 and CAN_PN_DF1EDH1 registers.
- When DATAFT[1:0] bit field in CAN_PN_CTL0 register is configured to 1, it means the DATA field of a matched message is larger than or equal to the configured expected DATA field in CAN_PN_EDLx (x = 0,1) registers. CAN_PN_DF0EDH0 and CAN_PN_DF1EDH1 registers are reserved.
- When DATAFT[1:0] bit field in CAN_PN_CTL0 register is configured to 2, it means the DATA field of a matched message is smaller than or equal to the configured expected DATA field in CAN_PN_EDLx (x = 0,1) registers. CAN_PN_DF0EDH0 and CAN_PN_DF1EDH1 registers are reserved.
- When DATAFT[1:0] bit field in CAN_PN_CTL0 register is configured to 3, it means the DATA field of a matched message is larger than or equal to the configured expected DATA field in CAN_PN_EDLx (x = 0,1) registers, and is smaller than or equal to the configured expected DATA field in CAN_PN_DF0EDH0 and CAN_PN_DF1EDH1 registers.

  Note: In this case, all the two 8 bytes of the expected data register should be configured, when the DLC of the received message (DLC field matched) is less than 8 bytes, then in DATA field matching, the data that matching with the expected data is the received DATA field plus the padding value zeros.

29.3.8. CAN FD operation

Both ISO CAN FD (ISO11898-1 specification) and non-ISO (Bosch CAN FD Specification V1.0) CAN FD protocols are supported, but they are incompatible with each other, so select the protocol by ISO bit in CAN_CTL2 register. In comparison to the non-ISO CAN FD protocol, a 3-bit counter and a parity bit are introduced in ISO CAN FD protocol. Thus the failure detection capability is improved for ISO CAN FD.

CAN FD mode supports both CAN classical frames and CAN FD frames. The FDF bit (the reserved bit in CAN classical frames) is used to distinguish between CAN classical and CAN FD format frames. When the FDF bit is recessive ‘1’, it is recognized as a CAN FD frame; otherwise, it is a classical frame. Compared with CAN classical frame, CAN FD frame does not support Rx FIFO, Rx FIFO DMA, and Pretended Networking function.

To enable CAN FD mode, set FDEN bit in CAN_CTL0 register to 1.

CAN FD BRS

In CAN FD mode, the data byte length is allowed up to 64 bytes for a CAN FD frame, and the bit time can switch to a higher speed of 8 Mbit/s for the Data Phase (from BRS bit to the first sample point of CRC Delimiter or to the starting of an error frame when an error condition is detected) of a CAN FD frame with BRS bit set (refer to ISO11898-1 or Bosch CAN FD Specification V1.0).

When BRSEN bit in CAN_FDCTL register is set to 1 (takes effect at the next message), and BRS bit in Tx mailbox is written as recessive '1', then higher bit time (called as data bit time) will be used for the Data Phase of CAN FD frame, the nominal bit time will be used for the rest of the bits. The bit time is changed at the sample point of the BRS bit. The data bit time is configured in CAN_FDBT register. The nominal bit time is configured in CAN_BT register.

When BRSEN bit in CAN_FDCTL register is set to 0, or when BRS bit in Tx mailbox is written as 0, then nominal bit time will be used for the entire CAN FD frame.

Note: The length of time quantum should be the same for the entire CAN FD frame, to reduce the possibilities of phase error frames on the CAN bus.

In FD frames, all nodes shall accept an up to two bit long dominant phase of overlapping ACK slot bits as a valid ACK, to compensate for phase shifts between the receivers. (Refer to ISO11898-1 specification)

CAN FD ESI

The transmission of ESI bit (the bit before DLC bits, refer to ISO11898-1 or Bosch CAN FD Specification V1.0) is defined by ESI field in MDES0 word of Tx mailbox and ERRSI[1:0] bits in CAN_ERR1 register. If ESI field in MDES0 is 0, it will transmit the dominant bit by error active nodes and transmit the recessive bit by error passive nodes according to ERRSI[1:0] bits in CAN_ERR1 register. If ESI field in MDES0 is 1, it will transmit ESI field in MDES0 word.

CAN FD CRC

Different CRC polynomials are used for different frame formats, results in a Hamming distance of 6:

- The CRC_15 polynomial is used for frames in CAN classical format: 0xC599
  $x^{15} + x^{14} + x^{10} + x^8 + x^7 + x^4 + x^3 + 1$
- The CRC_17 polynomial is used for frames in CAN FD format with DATA field no more than 16 bytes: 0x3685B
  $x^{17} + x^{16} + x^{14} + x^{13} + x^{11} + x^6 + x^4 + x^3 + x^1 + 1$
- The CRC_21 polynomial is used for frames in CAN FD format with DATA field more than 16 bytes: 0x302899
  $x^{21} + x^{20} + x^{13} + x^{11} + x^7 + x^4 + x^3 + 1$

For transmission, these three types of CRC will all be calculated at the start of the frame, and the final CRC to be transmitted is determined by the FDF field and DLC field of the frame.

After a successful transmission, when corresponding MSx bit of CAN_STAT register is set, the CAN_CRCCFD register is updated at the same time, with the calculated CRC for both CAN FD and non-FD messages. The CAN_CRCC register only stores the calculated CRC for CAN classical format frames.

For reception, the CRC polynomial used for CRC check is determined by the received FDF and DLC field.

Note: In Classical frames, the CRC delimiter is one single recessive bit. In FD frames, the CRC delimiter may consist of one or two recessive bits. A transmitter shall send only one recessive bit as CRC delimiter, but it shall accept two recessive bits before the edge from recessive to dominant that starts the acknowledge slot. A receiver will send its acknowledge bit after the first CRC delimiter bit. Refer to ISO11898-1 specification.

Bit stuff

The bit stuffing in CAN FD format frames is different from that in CAN classical format frames.

For transmission of CAN FD format frames, a fixed stuff bit is inserted before the first bit of the CRC field (regardless of the bit stuff conditions), and other fixed stuff bits are inserted after each 4 bits of the CRC field (fixed stuff bits are not included). The value of these fixed stuff bits are the inverse value of their preceding bit. Refer to ISO11898-1 specification.

For reception of CAN FD format frames, the fixed stuff bits will be descarded. When the value of the fixed stuff bit is the same as the value of its preceding bit, a stuff error is detected.

Note: For CAN FD format frames, fixed stuff bits are included in CRC calculation. For CAN classical format frames, stuff bits are not included.

Resynchronization

Resynchronization and hard synchronization occur in CAN FD frames in the same way as in CAN classical frames. Resynchronization is not performed in transmitting the CAN FD data phase.

Transmitter delay compensation

The transmitter delay compensation is used for the data phase of CAN FD frames with BRS set, for the reason that in CAN FD frames with BRS bit set, the length of the CAN bit time in the data phase is shorter than the transmitter delay, thus the bit error check is influenced, the transmitter cannot receive its own transmitted bit latest at the sample point of that bit. The transmitter delay is measured from the falling edge of FDF bit of transmitted frame to the falling edge of FDF bit of received frame, shown in ***Figure 29-2. Transmitter delay***.

The transmitter delay compensation mechanism defines a secondary sample point SSP. When it is used, the transmitter shall ignore bit errors detected at the sample point. When TDCEN bit in CAN_FDCTL register is 1, this feature is enabled, then the bit check will be done between the actually received bit and the delayed transmitted bit (the delay is calculated

based on the measured transmitter delay).

The transmitter delay compensation value is calculated in the equation follows:

$$t_{compensation} = t_{measure} + t_{offset} \quad (29\text{-}1)$$

with

$$t_{offset} = TDCO[4:0] \times t_{CANCLK} \quad (29\text{-}2)$$

$$t'_{offset} = t_{PBS1_FD} + t_{PTS_FD} + t_{SYNC_SEG} \quad (29\text{-}3)$$

$$t_{PBS1_FD} = (DPBS1[2:0] + 1) \times t_{q_FD} \quad (29\text{-}4)$$

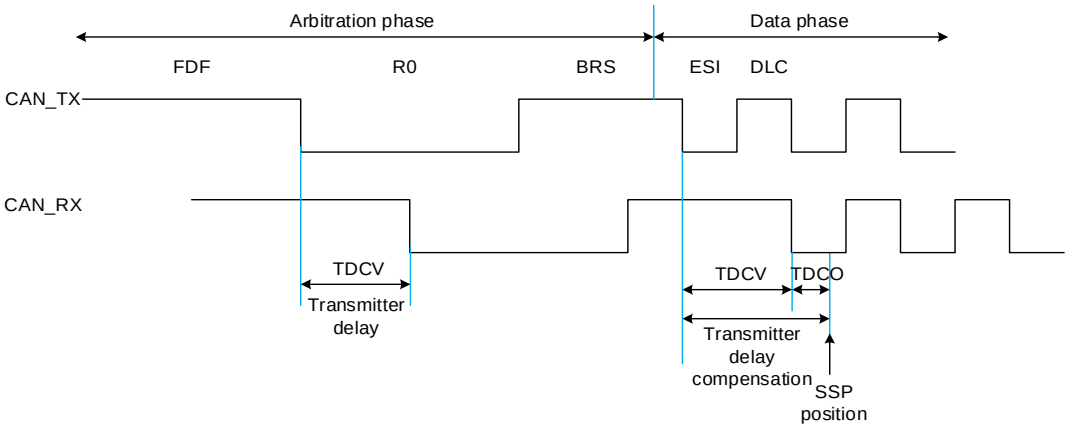
$$t_{PTS_FD} = DPTS[4:0] \times t_{q_FD} \quad (29\text{-}5)$$

$$t_{q_FD} = (DBAUDPSC[9:0] + 1) \times t_{CANCLK} \quad (29\text{-}6)$$

where the $t_{measure}$ is the measured transmitter delay, t_{offset} is the transmitter delay compensation offset which is saved in the TDCO[4:0] bits of CAN_FDCTL register in unit of t_{CANCLK}, t_{offset} should not be larger than the CAN data bit time. $t_{compensation}$ is the transmitter delay compensation value saved in TDCV[5:0] bits of CAN_FDCTL register in unit of t_{CANCLK}.

In the equations, the DPBS1[2:0] bits, DPTS[4:0] bits, DBAUDPSC[9:0] bits are all configured in CAN_FDBT register.

Figure 29-2. Transmitter delay



The maximum $t_{compensation}$ is $(3 \times \text{data bit time} - 2 \times t_{q_FD})$. When exceed this value, it is unable to compensate the transmitter delay, then TDCS bit in CAN_FDCTL register will be set. The implementation shall be able to compensate transmitter delays of at least two data bit times.

29.3.9. Errors and states

Transmit Error Counter (TECNT[7:0] bits in CAN_ERR0 register) and Receive Error Counter (RECNT[7:0] bits in the CAN_ERR0 register) take into account all errors in both CAN FD and non-FD messages, which get incremented or decremented according to the error condition. For detailed information about TECNT[7:0] and RECNT[7:0] management, please refer to the

CAN standard.

For CAN FD format frames, a Receive Error Counter for data phase of CAN FD messages (REFCNT[7:0] bits in the CAN_ERR0 register) and a Transmit Error Counter for data phase of CAN FD messages (TEFCNT[7:0] bits in the CAN_ERR0 register) are used additionally only when the BRS field of the frame is set. They stop counting and keep their values when in Bus off state, and they restart counting after returned to error active state by Bus off recovery.

Note: When in Pretended Networking mode, receive error counters keep counting and error flags are saved, transmit error counters stop counting and save their values. When returns to normal mode, the CAN_ERR0 and CAN_ERR1 registers will be updated with the counter value and saved error flags.

States

Error Passive State

If the value of TECNT[7:0] or RECNT[7:0] in CAN_ERR0 register increments to greater than 127, ERRSI[1:0] in CAN_ERR1 register is updated to 1 (error passive state).

Error Active state

If the node is in Error Passive state, and the value of either TECNT[7:0] or RECNT[7:0] in CAN_ERR0 register decrements to less than or equal to 127 when the other already satisfies this condition, ERRSI[1:0] in CAN_ERR1 register is updated to 0 (error active state).

Bus off state

If the value of TECNT[7:0] in CAN_ERR0 register becomes greater than 255, ERRSI[1:0] in CAN_ERR1 register is updated to 0b1x (Bus off state), and BOF bit in CAN_ERR1 register will be set, when BOIE bit in CAN_CTL1 register is set, an interrupt will be generated. The value of TECNT[7:0] will be reset to 0.

Bus off recovery:

To exit from Bus off state, the CAN has to wait for the recovery sequence specified in the CAN standard (128 occurrences of 11 consecutive recessive bits monitored on CAN RX). When TECNT[7:0] in CAN_ERR0 register reaches 128, ERRSI[1:0] in CAN_ERR1 register is updated to 0 (error active state) and both TECNT[7:0] and RECNT[7:0] in CAN_ERR0 register are reset to 0.

Depending on ABORDIS bit in CAN_CTL1 register, CAN will recover from Bus off automatically or remain in Bus off state.

When ABORDIS is 0, enable automatic bus off recovery, CAN wil recover from Bus off automatically after the recovery sequence. If the ABORDIS is changed to 0 after the recovery sequence, then CAN will resynchronize to the bus by detecting 11 consecutive recessive bits.

When ABORDIS is 1, not enable automatic bus off recovery. If the ABORDIS is changed to 1 after the CAN entered Bus off state, automatic bus off recovery will be disabled at the next time the CAN entered Bus off state.

Bus integration state

If the node starts the protocol operation during Bus off recovery, or detects the protocol exception event (the event occurs when FDEN bit in CAN_CTL0 register is set to 0, and a FDF bit of a FD frame is received), the node enters the bus integration state. In this state, the synchronicity to the CAN bus is lost. CAN node can leave the bus integration state when the bus idle condition (the sequence of 11 consecutive recessive bits) is detected. Refer to the CAN Protocol standard (ISO 11898-1).

The protocol exception detection is controlled by PREEN bit in CAN_CTL2 register.

The edge filtering can be configured by EFDIS bit in CAN_CTL2 register, which is used during the bus integration state. When the edge filtering is enabled, two consecutive nominal time quanta with dominant bus state are required to detect an edge that causes synchronization. When synchronization occurs, the counting for bus idle condition (the sequence of 11 consecutive recessive bits) is restarted. When edge filtering is performed, dominant bus-states shorter than a nominal bit time (the bits in data phase of a FD frame) will be ignored, stopped from being mistaken for an idle condition. Refer to the CAN Protocol standard (ISO 11898-1).

Note: Recommend to always enable edge filtering by reset EFDIS bit, to avoid mistakenly detection of bus idle condition.

Errors

If at least one of the error flags (ACKERR, BRERR, BDERR, CRCERR, FMERR, and STFERR bit in CAN_ERR1 register) is set, ERRSF bit in CAN_ERR1 register will be set. If ERRSIE bit in CAN_CTL1 register is set, an error interrupt will be generated.

If at least one of the error flags (BRFERR, BDFERR, CRCFERR, FMFERR, and STFFERR bit in CAN_ERR1 register) is set, ERRFSF bit in CAN_ERR1 register will be set. If ERRFSIE bit in CAN_CTL2 register is set, an error interrupt will be generated for errors detected in CAN FD frame data phase with BRS bit set.

Acknowledge error

If there is only one node operating, then it will lead to TECNT[7:0] in CAN_ERR0 register incrementing (to 128 at most by acknowledge error) in each message transmission, and an acknowledge error will occur, which is indicated by ACKERR bit in the CAN_ERR1 register.

Bit recessive error

When at least one bit sent as recessive ‘1’ is received as dominant ‘0’, a bit recessive error occurs. Refers to BRFERR and BRERR bit in CAN_ERR1 register.

Bit dominant error

When at least one bit sent as dominant '0' is received as recessive '1', a bit dominant error occurs. Refers to BDFERR and BDERR bit in CAN_ERR1 register.

CRC error

When the calculated CRC is different from the received CRC field of the frame, a CRC error occurs. Refers to CRCFERR and CRCERR bit in CAN_ERR1 register.

Form error

When a fixed-form bit field contains at least one illegal bit, a form error occurs. Refers to FMFERR and FMERR bit in CAN_ERR1 register.

Stuff error

Refers to STFFERR and STFERR bit in CAN_ERR1 register.

29.3.10. Communication parameters

Bit time

The CAN bit time from the CAN protocol has three segments as follows:

Synchronization segment (SYNC_SEG): A bit change is expected to occur within this time segment. It has a fixed length of one time quantum ($1 \times t_q$).

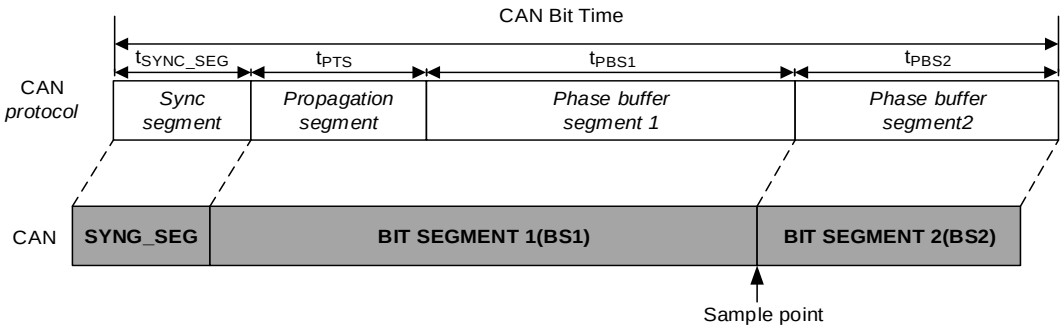
Bit segment 1 (BS1): It defines the location of the sample point. It includes the Propagation segment and Phase buffer segment 1 in the CAN standard. It may be automatically lengthened to compensate for positive phase drifts due to different frequency of the various nodes of the network.

Bit segment 2 (BS2): It defines the location of the sample point. It may also be automatically shortened to compensate for negative phase drifts. Its duration should programmed no less than 2 time quanta.

Note: The bit time configuration ranges must be in compliance with the CAN Protocol standard (ISO 11898-1).

CAN bit time is shown as in the ***Figure 29-3. CAN bit time***.

Figure 29-3. CAN bit time



Synchronization Jump Width (SJW): It can be lengthened or shortened to compensate for the Synchronization error of the CAN network node. It is configured by SJW[4:0] bits in CAN_BT register for nomainal bit time, or configured by DSJW[2:0] bits in CAN_FDBT register for data bit time.

A valid edge is defined as the first toggle in a bit time from dominant to recessive bus level before the controller sends a recessive bit.

If a valid edge is detected in BS1, not in SYNC_SEG, BS1 is added with up to SJW maximumly, so that the sample point is delayed.

Conversely, if a valid edge is detected in BS2, not in SYNC_SEG, BS2 is cut down with up to SJW maximumly, so that the transmit point is moved earlier.

Bit sampling

BSPMOD in CAN_CTL1 register defines the sampling mode of CAN bits at the Rx input pin.

When BSPMOD is 0, only one sample (the sample point) is used.

When BSPMOD is 1, three samples are used to determine the received bit value, that is the one on the sample point, and the two preceding samples.

Note: This bit cannot be set when CAN FD is enabled.

Baudrate

CAN module has two clock domains:

- The clock of Control Interface and CAN registers derives from the APB2 clock.
- The clock of Protocol controller (CANCLK) can be configured by CANxSEL[1:0] bit in RCU_CFG2 register, to derive from oscillator clock, or APB2 clock, or APB2 clock devided by 2, or IRC64MDIV internal clock.

The CAN calculates its baudrate as follows:

$$\text{BaudRate} = \frac{1}{\text{CAN Bit Time}} \quad (29\text{-}7)$$

$$\text{CAN Bit Time} = t_{SYNC_SEG} + t_{PTS} + t_{PBS1} + t_{PBS2} \quad (29\text{-}8)$$

with

$$t_{SYNC_SEG} = 1 \times t_q \quad (29\text{-}9)$$

$$t_{PTS} = (N_{PTS} + 1) \times t_q \text{ or } t_{PTS} = N_{DPTS} \times t_q \quad (29\text{-}10)$$

$$t_{PBS1} = (N_{PBS1} + 1) \times t_q \quad (29\text{-}11)$$

$$t_{PBS2} = (N_{PBS2} + 1) \times t_q \quad (29\text{-}12)$$

$$t_q = (N_{BAUDPSC} + 1) \times t_{CANCLK} \quad (29\text{-}13)$$

In the equations, for nominal bit rate:

N_{PTS}, N_{PBS1}, N_{PBS2}, and $N_{BAUDPSC}$ are configured by the PTS[5:0] bits, PBS1[4:0] bits, PBS2[4:0] bits, and BAUDPSC[9:0] bits respectively in CAN_BT register.

For data bit rate:

N_{DPTS}, N_{PBS1}, N_{PBS2}, and $N_{BAUDPSC}$ are configured by the DPTS[4:0] bits, DPBS1[2:0] bits, DPBS2[2:0] bits, and DBAUDPSC[9:0] bits respectively in CAN_FDBT register.

Timestamp

A 16-bit internal counter of the CAN hardware in CAN_TIMER register is used to generate the timestamp value. The value of the internal counter is sampled at SOF field on the CAN bus, and is written into the TIMESTAMP field of MDES0 or FDES0 word after a successful reception or transmission of a message.

The counter does not count in Inactive mode, or when LPS bit in CAN_CTL0 register is 1.

Counter clock source

When ITSRC bit in CAN_CTL2 register is 1, the TRIGSEL output CANx_EX_TIME_TICK is selected as the increasing condition of internal counter. The CANx_EX_TIME_TICK and APB2 bus clock are in the same clock domain, to ensure the internal counter be increased effectively, the pulse width of CANx_EX_TIME_TICK signal must be greater than or equal to the APB2 bus clock period.

When ITSRC bit in CAN_CTL2 register is 0, the CAN baudrate is selected as the increasing condition of internal counter, internal counter increments by one for each bit that is received or transmitted. When there is no message on the bus, it is counted with the baudrate programmed previously.

Time synchronization

If TSYNC bit in CAN_CTL1 register is 1, a SYNC message reception in the first mailbox descriptor will reset the internal counter for network time synchronization.

29.3.11. Interrupts

The CAN interrupt events and flags are list in ***Table 29-11. Interrupt events***.

Table 29-11. Interrupt events

Interrupt event		Flag			Enable control			
		Bit		Register	Enable bit	Control bit	Enable register	Control register
Bus off		BOF		CAN_ERR1	BOIE		CAN_CTL1	
Bus off recovery		BORF			BORIE		CAN_CTL2	
Error summary	Bit recessive error	ERRSF	BRERR		ERRSIE		CAN_CTL1	
	Bit dominant error		BDERR					
	ACK error		ACKERR					
	CRC error		CRCERR					
	Form error		FMERR					
	Stuff error		STFERR					
Error summary for FD frames with data bit time	Bit recessive error	ERRFSF	BRFERR		ERRFSIE		CAN_CTL2	
	Bit dominant error		BDFERR					
	CRC error		CRCFERR					
	Form error		FMFERR					
	Stuff error		STFFERR					
Tx error warning		TWERRIF			TWERRIE	WERREN	CAN_CTL1	CAN_CTL0
Rx error warning		RWERRIF			RWERRIE			
Wakeup match		WMS		CAN_PN_STAT	WMIE		CAN_PN_CTL0	
Wakeup timeout		WTOS			WTOIE			
Mailbox successful transmission or reception		All bits		CAN_STAT	All bits	RFEN = 0	CAN_INTEN	CAN_CTL0
		MSx			MIEx	RFEN = 1		
Rx FIFO not empty		MS5_RFNE			MIE5	RFEN = 1 & DMAEN = 0		
Rx FIFO warning		MS6_RFW			MIE6			
Rx FIFO overflow		MS7_RFO			MIE7			

29.4. Example for a typical configuration flow of CAN

After power-on reset or system reset, the following operation flow is a typical process for application to configure and run CAN:

- Configure CAN module clock source CANCLK, and enable CAN clock
  - Configure CANxSEL[1:0] bits in CAN_CFG2 register to select the CAN module clock source. Program the RCU_APB2EN register to enable the CAN module clock.
- Setup the communication interface
  - Configure GPIO and AFIO module to select PADs to alternate functions.
- Enter Inactive mode
  - Because INAMOD bit, HALT bit, NRDY bit and INAS bit are default set after power-

on reset or system reset, so CAN will automatically enters Inactive mode for configuration of CAN registers.

- Service the flags in CAN_STAT register
  - Read the Rx mailbox or Rx FIFO descriptor contents, clear the corresponding asserted flag bit in CAN_STAT register, then read the CAN_TIMER register at last for a complete flag bit service. If Rx FIFO is enabled, do a clearing FIFO operation by setting MS0 bit in CAN_STAT register to 1. Also clear the asserted flags by Tx mailboxes.
- Initialize the physical memory space for mailbox and Rx FIFO descriptors
  - Configure memory space for mailbox and Rx FIFO descriptors totally by MSZ[4:0] bits in CAN_CTL0 register.
- Configure the communication parameters
  - Configure the CAN nominal bit rate by PTS[5:0] bits, PBS1[4:0] bits, PBS2[4:0] bits, SJW[4:0] bits and BAUDPSC[9:0] bits in CAN_BT register.
  - Configure bit sampling mode by BSPMOD bit in CAN_CTL1 register if needed.
  - Configure PREEN bit and EFDIS bit for bus integration state if needed.
- Configure the control parameters for transmission
  - Configure arbitration priority by MTO bit in CAN_CTL1 register and LAPRIOEN bit in CAN_CTL0 register.
  - Configure arbitration start delay by ASD[4:0] bits of CAN_CTL2 register if needed.
  - Enable transmission abort function for Tx mailbox descriptor configuration by MST bit in CAN_CTL0 register.
- Configure the control parameters for reception
  - Choose whether use Rx FIFO and Rx FIFO DMA for reception or not by RFEN bit and DMAEN bit in CAN_CTL0 register.
  - Configure Rx private filter & Rx mailbox queue feature by RPFQEN bit in CAN_CTL0 register.
  - Configure receive filter related parameters by RFO bit, RRFRMS bit and IDERTR_RMF bit of CAN_CTL2 register.
  - Configure filter data of the Rx mailbox and Rx FIFO by CAN_RMPUBF, CAN_RFIFOPUBF and CAN_RFIFOMPFx (x = 0..31) registers. If Rx FIFO is enabled, configure Rx FIFO ID filter table element format by FS[1:0] bits of CAN_CTL0 register, configure Rx FIFO ID filter table element number by RFFN[3:0] bits of CAN_CTL2 register.
- If CAN FD operation is needed
  - Select CAN FD operation protocol by ISO bit in CAN_CTL2 regitser.
  - Enable CAN FD operation by FDEN bit in CAN_CTL0 register.
  - Initialize the mailbox data size by MDSZ[1:0] bits of CAN_FDCTL register.
  - Configure CAN FD related transmitter delay compensation feature by TDCEN bit and TDCO[4:0] bits of CAN_FDCTL register if needed.
  - Configure the CAN data bit rate by DPTS[4:0] bits, DPBS1[2:0] bits, DPBS2[2:0] bits DSJW[2:0] bits and DBAUDPSC[9:0] bits in CAN_FDBT register.
- Configure interrupts
  - Enable the needed interrupts in CAN_CTL0, CAN_CTL1, CAN_CTL2 and

CAN_INTEN registers.

- Initialize the Tx / Rx mailbox descriptors
  - If message transmission is needed, initialize the Tx mailbox descriptors.
  - If message reception is needed, initialize the Rx mailbox descriptors, if Rx FIFO is enabled, also initialize the Rx FIFO descriptors including the ID filter table elements.
- If Pretended Networking mode is required, set PNEN bit and PNMOD bit in CAN_CTL0 register and configure the necessary registers for wakeup.
- Exit Inactive mode
  - Clear HALT bit in CAN_CTL0 register to exit Inactive mode, and CAN starts to synchronize to the CAN bus.

29.5. CAN registers

CAN0 base address: 0x4001 A000

CAN1 base address: 0x4001 B000

CAN2 base address: 0x4001 C000

29.5.1. Control register 0 (CAN_CTL0)

Address offset: 0x00
Reset value: 0x5900 000F

All bits except bit 30, 28, 25, 19 of this register should be configured in Inactive mode only, because they are blocked by hardware in other modes.

All bits except bit 31, 27, 24, 20 of this register will be reset by software reset bit SWRST in CAN_CTL0 register.

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CANDIS	INAMOD	RFEN	HALT	NRDY	Reserved	SWRST	INAS	Reserved		WERREN	LPS	PNEN	PNS	SRDIS	RPFQEN
rw	rw	rw	rw	r		rw	r			rw	r	rw	r	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMAEN	PNMOD	LAPRIOEN	MST	FDEN	Reserved	FS[1:0]		Reserved			MSZ[4:0]				
rw	rw	rw	rw	rw		rw					rw				

Bits	Fields	Descriptions
31	CANDIS	CAN disable This bit is not affected by software reset bit SWRST in CAN_CTL0 register. 0: Enable CAN module 1: Disable CAN module
30	INAMOD	Inactive mode enable 0: Disable Inactive mode 1: Enable Inactive mode
29	RFEN	Rx FIFO enable 0: Disable Rx FIFO 1: Enable Rx FIFO
28	HALT	Halt CAN 0: No enter Inactive mode request 1: Enter Inactive mode if the INAMOD bit in CAN_CTL0 register is set
27	NRDY	Not ready

		This bit indicates the state of whether the Protocol controller clock is disabled or not. When in Inactive mode, or in CAN_Disable mode, the Protocol controller clock is disabled, and CAN is not ready. 0: CAN is ready 1: CAN is not ready
26	Reserved	Must be kept at reset value.
25	SWRST	Software reset When this bit is set, CAN internal state machines and CAN registers will be reset. This bit is automitically cleared by hardware when software reset is completed. Software reset has no effect when LPS bit in CAN_CTL0 register is set. 0: No effect 1: Software reset request
24	INAS	Inactive mode state 0: Not in Inactive mode 1: In Inactive mode
23:22	Reserved	Must be kept at reset value.
21	WERREN	Error warning enable When this bit is set, the warning interrupt flag TWERRIF and RWERRIF bit in CAN_ERR1 register will be enabled to reflect the state change of TWERRF and RWERRF bit in CAN_ERR1 register respectively. 0: Disable Tx and Rx error warining 1: Enable Tx and Rx error warining
20	LPS	Low power state 0: Not in low power state 1: In low power state
19	PNEN	Pretended Networking mode enable 0: Disable Pretended Networking mode 1: Enable Pretended Networking mode
18	PNS	Pretended Networking state 0: Not in Pretended Networking state 1: In Pretended Networking state
17	SRDIS	Self reception disable 0: Enable self reception 1: Disable self reception
16	RPFQEN	Rx private filters enable & Rx mailbox queue enable 0: Disable Rx private filters & disable Rx mailbox queue 1: Enable Rx private filters & enable Rx mailbox queue
15	DMAEN	DMA enable

		0: DMA feature for RX FIFO disabled. 1: DMA feature for RX FIFO enabled.
14	PNMOD	Pretended Networking mode selection 0: Not select Pretended Networking mode 1: Select Pretended Networking mode
13	LAPRIOEN	Local arbitration priority enable 0: Disable local arbitration priority 1: Enable local arbitration priority
12	MST	Mailbox stop transmission 0: Disable transmission abort 1: Enable transmission abort
11	FDEN	CAN FD operation enable 0: Disable CAN FD operation 1: Enable CAN FD operation
10	Reserved	Must be kept at reset value.
9:8	FS[1:0]	Format selection This bit field defines the format of the Rx FIFO ID filter table elements. 00: Format A: One full ID (standard and extended) per ID filter table element 01: Format B: Two full standard IDs or two partial 14-bit extended IDs per ID filter table element 10: Format C: Four partial 8-bit IDs (standard and extended) per ID filter table element 11: Format D: All frames rejected
7:5	Reserved	Must be kept at reset value.
4:0	MSZ[4:0]	Memory size This bit field defines the maximum size of memory for message transmission and reception. The size is counted in uint of 4 words (equals to the size of a mailbox descriptor with 8-byte data), including mailbox and Rx FIFO. Before configuring this bit field, the flags in CAN_STAT register must be serviced. 00000: 1 unit 00001: 2 units … 11111: 32 units

29.5.2. Control register 1 (CAN_CTL1)

Address offset: 0x04

Reset value: 0x0000 0000

The bits 12, 7, 5, 4, 3 of this register should be configured in Inactive mode only, because

they are blocked by hardware in other modes.

All bits of this register are not affected by software reset bit SWRST in CAN_CTL0 register.

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BOIE	ERRSIE	Reserved	LSCMOD	TWERRIE	RWERRIE	Reserved	BSSSEL	BSPMOD	ABORDIS	TSYNC	MTO	MMOD	Reserved		
rw	rw		rw	rw	rw		rw	rw	rw	rw	rw	rw			

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	BOIE	Bus off interrupt enable 0: Disable Bus off interrupt 1: Enable Bus off interrupt
14	ERRSIE	Error summary interrupt enable 0: Disable error summary interrupt 1: Enable error summary interrupt
13	Reserved	Must be kept at reset value.
12	LSCMOD	Loopback and silent communication mode 0: Disable loopback and silent communication mode 1: Enable loopback and silent communication mode **Note:** In this mode, SRDIS bit in CAN_CTL0 register, and TDCEN in CAN_FDCTL register cannot be set.
11	TWERRIE	Tx error warning interrupt enable This bit can be written only when WERREN in CAN_CTL0 register is 1. This bit is read as zero when WERREN in CAN_CTL0 register is 0. 0: Disable Tx error warning interrupt 1: Enable Tx error warning interrupt
10	RWERRIE	Rx error warning interrupt enable This bit can be written only when WERREN in CAN_CTL0 register is 1. This bit is read as zero when WERREN in CAN_CTL0 register is 0. 0: Disable Rx error warning interrupt 1: Enable Rx error warning interrupt
9	Reserved	Must be kept at reset value.
8	BSSSEL	Bit sampling synchronization select 0: two satges synchronization for CAN bus sampling 1: one satge synchronization for CAN bus sampling

7	BSPMOD	Bit sampling mode 0: One sample for the received bit 1: Three samples for the received bit
6	ABORDIS	Automatic Bus off recovery not enable 0: Enable automatic Bus off recovery 1: Not enable automatic Bus off recovery
5	TSYNC	Time synchronization enable 0: Disable time synchronization 1: Enable time synchronization
4	MTO	Mailbox transmission order 0: Highest priority mailbox is transmitted first 1: Lowest number mailbox is transmitted first
3	MMOD	Monitor mode 0: Disable Monitor mode 1: Enable Monitor mode
2:0	Reserved	Must be kept at reset value.

29.5.3. Timer register (CAN_TIMER)

Address offset: 0x08
Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CNT[15:0]	Counter value This bit field contains the internal counter value used for timestamp generation.

29.5.4. Receive mailbox public filter register (CAN_RMPUBF)

Address offset: 0x10
Reset value: 0xXXXX XXXX

This register is located in RAM.

All bits of this register should be configured in Inactive mode only, because they are blocked by hardware in other modes.

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MFD31	MFD30	MFD29	MFD28	MFD27	MFD26	MFD25	MFD24	MFD23	MFD22	MFD21	MFD20	MFD19	MFD18	MFD17	MFD16
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MFD15	MFD14	MFD13	MFD12	MFD11	MFD10	MFD9	MFD8	MFD7	MFD6	MFD5	MFD4	MFD3	MFD2	MFD1	MFD0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:0	MFDx	Mailbox filter data MFD31 bit is used to filter the mailbox descriptor RTR field. MFD30 bit is used to filter the mailbox descriptor IDE field. MFDx (x = 0..28) bits are used to filter the mailbox descriptor ID field. 0: The bit is "don't care" 1: The bit is checked **Note:** For standard frame, MDF18~MFD28 bits are use to filter the mailbox descriptor ID field.

29.5.5. Error register 0 (CAN_ERR0)

Address offset: 0x1C
Reset value: 0x0000 0000

All bits of this register are read-only except in Inactive mode.

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REFCNT[7:0]								TEFCNT[7:0]							
rw0								rw0							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RECNT[7:0]								TECNT[7:0]							
rw								rw							

Bits	Fields	Descriptions
31:24	REFCNT[7:0]	Receive error counter for data phase of FD frames with BRS bit set This bit field can only be written as zero in Inactive mode.
23:16	TEFCNT[7:0]	Transmit error count for the data phase of FD frames with BRS bit set This bit field can only be written as zero in Inactive mode.
15:8	RECNT[7:0]	Receive error count defined by the CAN standard

7:0	TECNT[7:0]	Transmit error count defined by the CAN standard

29.5.6. Error register 1 (CAN_ERR1)

Address offset: 0x20

Reset value: 0x0004 0009

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BRFERR	BDFERR	Reserved	CRCFERR	FMFERR	STFFERR	Reserved				ERROVR	ERRFSF	BORF	SYN	TWERRIF	RWERRIF
rc	rc		rc	rc	rc					rc_w1	rc_w1	rc_w1	r	rc_w1	rc_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BRERR	BDERR	ACKERR	CRCERR	FMERR	STFERR	TWERRF	RWERRF	IDLEF	TS	ERRSI[1:0]		RS	BOF	ERRSF	Reserved
rc	rc	rc	rc	rc	rc	r	r	r	r	r		r	rc_w1	rc_w1	

Bits	Fields	Descriptions
31	BRFERR	Bit recessive error in data phase of FD frames with the BRS bit set 0: No error occurrence 1: At least one bit sent as recessive is received as dominant
30	BDFERR	Bit dominant error in data phase of FD frames with the BRS bit set 0: No error occurrence 1: At least one bit sent as dominant is received as recessive
29	Reserved	Must be kept at reset value.
28	CRCFERR	CRC error in data phase of FD frames with the BRS bit set 0: No error occurrence 1: A CRC error occurred
27	FMFERR	Form error in data phase of FD frames with the BRS bit set 0: No error occurrence 1: A form error occurred
26	STFFERR	Stuff error in data phase of FD frames with the BRS bit set 0: No error occurrence 1: A stuff error occurred
25:22	Reserved	Must be kept at reset value.
21	ERROVR	Error overrun This bit indicates that an error condition occurred when any error flag is already set. 0: Error overrun not occurred 1: Error overrun occurred
20	ERRFSF	Error summary flag for data phase of FD frames with BRS bit set This bit is logical ORed by the following bits:

		CAN_ERR1[31]: Bit recessive error CAN_ERR1[30]: Bit dominant error CAN_ERR1[28]: CRC error CAN_ERR1[27]: Form error CAN_ERR1[26]: Stuff error
19	BORF	Bus off recovery flag This bit is set when the the recovery sequence specified in the CAN standard on the CAN bus is detected and CAN is ready to recovery from Bus off. 0: No event occurrence 1: Bus off recovery sequence event occurs
18	SYN	Synchronization flag 0: Not synchronized to the CAN bus 1: Synchronized to the CAN bus
17	TWERRIF	Tx error warning interrupt flag This bit is not used during Bus off state. 0: No event occurrence 1: TWERRF bit in CAN_ERR1 register changes from 0 to 1
16	RWERRIF	Rx error warning interrupt flag This bit is updated when exiting from Pretended Networking mode. 0: No event occurrence 1: RWERRF bit in CAN_ERR1 register changes from 0 to 1
15	BRERR	Bit recessive error for all format frames This bit is updated when exiting from Pretended Networking mode. 0: No error occurrence 1: At least one bit sent as recessive is received as dominant
14	BDERR	Bit dominant error for all format frames This bit is updated when exiting from Pretended Networking mode. 0: No error occurrence 1: At least one bit sent as dominant is received as recessive
13	ACKERR	ACK error This bit is updated when exiting from Pretended Networking mode. 0: No error occurrence 1: An ACK error occurred
12	CRCERR	CRC error This bit is updated when exiting from Pretended Networking mode. 0: No error occurrence 1: A CRC error occurred
11	FMERR	Form error This bit is updated when exiting from Pretended Networking mode.

		0: No error occurrence 1: A form error occurred
10	STFERR	Stuff error This bit is updated when exiting from Pretended Networking mode. 0: No error occurrence 1: A stuffing error occurred
9	TWERRF	Tx error warning flag 0: No event occurrence 1: TECNT[7:0] in CAN_ERR0 register is greater than or equal to 96
8	RWERRF	Rx error warning flag This bit is updated when exiting from Pretended Networking mode. 0: No event occurrence. 1: RECNT[7:0] in CAN_ERR0 register is greater than or equal to 96
7	IDLEF	IDLE flag 0: No event occurrence 1: In Bus idle state
6	TS	Transmitting state 0: CAN is not working in transmitting state 1: CAN is working in transmitting state
5:4	ERRSI[1:0]	Error state indicator When MMOD bit in CAN_CTL1 register and SWRST bit in CAN_CTL0 register are both set to 1, this bit will be reset for one CAN bit time, and then changes to 0b01 to reflect Monitor mode state. 00: Error active 01: Error passive 1x: Bus off
3	RS	Receiving state 0: CAN is not working in receiving state 1: CAN is working in receiving state
2	BOF	Bus off flag 0: No event occurrence 1: In Bus off state
1	ERRSF	Error summary flag This bit is logical ORed by the following bits: CAN_ERR1[15]: Bit recessive error CAN_ERR1[14]: Bit dominant error CAN_ERR1[13]: ACK error CAN_ERR1[12]: CRC error CAN_ERR1[11]: Form error

		CAN_ERR1[10]: Stuff error
0	Reserved	Must be kept at reset value.

29.5.7. Interrupt enable register (CAN_INTEN)

Address offset: 0x28

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MIE31	MIE30	MIE29	MIE28	MIE27	MIE26	MIE25	MIE24	MIE23	MIE22	MIE21	MIE20	MIE19	MIE18	MIE17	MIE16
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MIE15	MIE14	MIE13	MIE12	MIE11	MIE10	MIE9	MIE8	MIE7	MIE6	MIE5	MIE4	MIE3	MIE2	MIE1	MIE0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:0	MIEx	Message transmission and reception interrupt enable When Rx FIFO is disabled, these bits are used for mailbox number x (refers to ***Mailbox number***) interrupt configuration. When Rx FIFO is enabled, MIE5 to MIE7 are used for Rx FIFO interrupt configuration, and mailbox interruption configuration bits are the bits x that are the same with the mailbox number x (refers to ***Mailbox number***). 0: Disable the corresponding interrupt 1: Enable the corresponding interrupt

29.5.8. Status register (CAN_STAT)

Address offset: 0x30

Reset value: 0x0000 0000

The bits 1 to 7 of this register will be cleared by configuration change of RFEN bit in CAN_CTL0 register.

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MS31	MS30	MS29	MS28	MS27	MS26	MS25	MS24	MS23	MS22	MS21	MS20	MS19	MS18	MS17	MS16
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MS15	MS14	MS13	MS12	MS11	MS10	MS9	MS8	MS7_RFO	MS6_RFW	MS5_RFNE	MS4_RES	MS3_RES	MS2_RES	MS1_RES	MS0_RFC
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

Bits	Fields	Descriptions
31:8	MSx	Mailbox x state

		x is the mailbox number, refers to ***Mailbox number***. 0: No successful transmission or reception has occurred in the mailbox descriptor 1: A successful transmission or reception has occurred in the mailbox descriptor
7	MS7_RFO	Mailbox 7 state / Rx FIFO overflow 0: No successful transmission or reception has occurred in the mailbox descriptor 7 when Rx FIFO is disabled. / Rx FIFO is not overflow when Rx FIFO is enabled. 1: A successful transmission or reception has occurred in the mailbox descriptor 7 when Rx FIFO is disabled. / Rx FIFO is overflow when Rx FIFO is enabled.
6	MS6_RFW	Mailbox 6 state / Rx FIFO warning 0: No successful transmission or reception has occurred in the mailbox descriptor 6 when Rx FIFO is disabled. / Rx FIFO has no warning when Rx FIFO is enabled. 1: A successful transmission or reception has occurred in the mailbox descriptor 6 when Rx FIFO is disabled. / Rx FIFO almost full warning when Rx FIFO is enabled.
5	MS5_RFNE	Mailbox 5 state / Rx FIFO not empty 0: No successful transmission or reception has occurred in the mailbox descriptor 5 when Rx FIFO is disabled. / Rx FIFO is empty when Rx FIFO is enabled. 1: A successful transmission or reception has occurred in the mailbox descriptor 5 when Rx FIFO is disabled. / Rx FIFO is not empty when Rx FIFO is enabled.
4	MS4_RES	Mailbox 4 state / Reserved Similar to MS1_RES description.
3	MS3_RES	Mailbox 3 state / Reserved Similar to MS1_RES description.
2	MS2_RES	Mailbox 2 state / Reserved Similar to MS1_RES description.
1	MS1_RES	Mailbox 1 state / Reserved 0: No successful transmission or reception has occurred in the mailbox descriptor 1 when Rx FIFO is disabled. / Reserved when Rx FIFO is enabled. 1: A successful transmission or reception has occurred in the mailbox descriptor 1 when Rx FIFO is disabled. / Reserved when Rx FIFO is enabled.
0	MS0_RFC	Mailbox 0 state / Clear Rx FIFO bit 0: No successful transmission or reception has occurred in the mailbox descriptor 0 when Rx FIFO is disabled. / No effect when Rx FIFO is enabled. 1: A successful transmission or reception has occurred in the mailbox descriptor 0 when Rx FIFO is disabled. / Clear Rx FIFO when Rx FIFO is enabled, only allowed to written in Inactive mode, refers to ***Clear FIFO***.

29.5.9. Control register 2 (CAN_CTL2)

Address offset: 0x34

Reset value: 0x00A0 0000

All bits except bit 31, 30 of this register should be configured in Inactive mode only, because they are blocked by hardware in other modes.

All bits of this register are not reset by software reset bit SWRST in CAN_CTL0 register.

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ERRFSIE	BORIE	Reserved		RFFN[3:0]				ASD[4:0]					RFO	RRFRMS	IDERTR_RMF
rw	rw			rw				rw					rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ITSRC	PREEN	Reserved	ISO	EFDIS	Reserved										
rw	rw		rw	rw											

Bits	Fields	Descriptions
31	ERRFSIE	Error summary interrupt enable bit for data phase of FD frames with BRS bit set 0: Disable error summary interrupt for data phase of FD frames with BRS bit set 1: Enable error summary interrupt for data phase of FD frames with BRS bit set
30	BORIE	Bus off recovery interrupt enable 0: Disable Bus off recovery interrupt 1: Enable Bus off recovery interrupt
29:28	Reserved	Must be kept at reset value.
27:24	RFFN[3:0]	Rx FIFO filter number

Table 29-12. Rx FIFO filter element number

RFFN[3:0]	Rx FIFO filter element number	Rx FIFO occupied space	Available mailboxes
0000	8	Mailbox descriptor 0 - 7	Mailbox 8 - 31
0001	16	Mailbox descriptor 0 - 9	Mailbox 10 - 31
0002	24	Mailbox descriptor 0 - 11	Mailbox 12 - 31
0003	32	Mailbox descriptor 0 - 13	Mailbox 14 - 31
0004	40	Mailbox descriptor 0 - 15	Mailbox 16 - 31
0005	48	Mailbox descriptor 0 - 17	Mailbox 18 - 31
0006	56	Mailbox descriptor 0 – 19	Mailbox 20 - 31
0007	64	Mailbox descriptor 0 – 21	Mailbox 22 - 31
0008	72	Mailbox descriptor 0 – 23	Mailbox 24 - 31
0009	80	Mailbox descriptor 0 – 25	Mailbox 26 - 31
000A	88	Mailbox descriptor 0 – 27	Mailbox 28 - 31

000B	96	Mailbox descriptor 0 – 29	Mailbox 30 - 31
000C	104	Mailbox descriptor 0 – 31	none
others	104	Mailbox descriptor 0 - 31	none

This bit field must not be programmed with values that cause memory occupied by Rx FIFO to exceed the available memory size which is defined by MSZ[4:0] bits in CAN_CTL0 register, otherwise the exceeding ones will not be functional.

23:19 ASD[4:0] Arbitration start delay
This bit field defines how many CAN bits the Tx arbitration process start point can be delayed.

18 RFO Receive filter order
0: Rx FIFO is filtered first
1: Mailboxes are filtered first

17 RRFRMS Remote request frame is stored
0: Remote response frame is generated when a mailbox with CODE RANSWER is found with the same ID
1: Remote request frame is stored as a data frame without automatic remote response frame transmitted

16 IDERTR_RMF IDE and RTR field filter type for Rx mailbox reception
This bit defines the matching of IDE and RTR field in Rx mailbox descriptor with the received bit.
0: IDE field is always compared, and RTR is never compared. Regardless of the filter data configurations in related filter register.
1: Filtering of IDE and RTR fields are enabled, by filter data configurations in related filter register.

15 ITSRC Internal counter source
0: CAN baudrate
1: External trigger CANx_EX_TIME_TICK from TRIGSEL output

14 PREEN Protocol exception detection enable by CAN standard
0: Disable protocol exception detection
1: Enable protocol exception detection

13 Reserved Must be kept at reset value.

12 ISO ISO CAN FD
0: Non-ISO CAN FD protocol operation is applied
1: ISO CAN FD protocol operation is applied

11 EFDIS Edge filtering disable
0: Enable edge filtering
1: Disable edge filtering

10:0	Reserved	Must be kept at reset value.

29.5.10. CRC for classical frame register (CAN_CRCC)

Address offset: 0x44
Reset value: 0x0000 0000

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

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

Bits	Fields	Descriptions
31:21	Reserved	Must be kept at reset value.
20:16	ANTM[4:0]	Associated number of mailbox for transmitting the CRCTC[14:0] value This bit field contains the number of the mailbox which transmits the CRCTC[14:0] value.
15	Reserved	Must be kept at reset value.
14:0	CRCTC[14:0]	Transmitted CRC value for classical frames This bit field contains the CRC value of the last successfully transmitted message in classical format.

29.5.11. Receive FIFO public filter register (CAN_RFIFOPUBF)

Address offset: 0x48
Reset value: 0xXXXX XXXX

This register is located in RAM.

All bits of this register should be configured in Inactive mode only, because they are blocked by hardware in other modes.

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

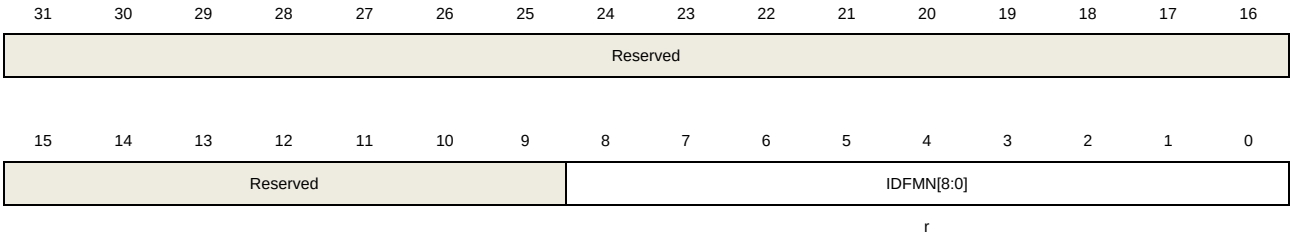
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FFD31	FFD30	FFD29	FFD27	FFD27	FFD26	FFD25	FFD24	FFD23	FFD22	FFD21	FFD20	FFD19	FFD18	FFD17	FFD16
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FFD15	FFD14	FFD13	FFD12	FFD11	FFD10	FFD9	FFD8	FFD7	FFD6	FFD5	FFD4	FFD3	FFD2	FFD1	FFD0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:0	FFDx	Rx FIFO filter data Each bit is used for filtering the corresponding ID filter table element bit, except the reserved bit in ID filter table element. 0: The bit is "don't care" 1: The bit is checked

29.5.12. Receive FIFO identifier filter matching number register (CAN_RFIFOIFMN)

Address offset: 0x4C
Reset value: 0xXXXX XXXX

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



Bits	Fields	Descriptions
31:9	Reserved	Must be kept at reset value.
8:0	IDFMN[8:0]	Identifier filter matching number This field is valid only when MS5_RFNE bit in CAN_STAT register is 1. This bit field indicates which ID filter table element matches the received message that is in the output of the Rx FIFO. If more than one element is matched, the ID filter table element with the lowest number is stored.

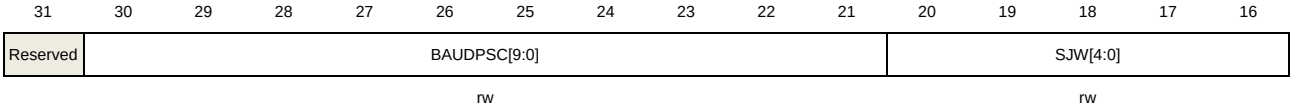
29.5.13. Bit timing register (CAN_BT)

Address offset: 0x50
Reset value: 0x0100 0000

All bits of this register should be configured in Inactive mode only, because they are blocked by hardware in other modes.

This register is not affected by software reset bit SWRST in CAN_CTL0 register.

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



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PTS[5:0]						PBS1[4:0]					PBS2[4:0]				
rw						rwr					rw				

Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30:21	BAUDPSC[9:0]	Baud rate prescaler The CAN baud rate prescaler = BAUDPSC[9:0] + 1.
20:16	SJW[4:0]	Resynchronization jump width Resynchronization jump width time quantum = SJW[4:0] + 1
15:10	PTS[5:0]	Propagation time segment Propagation time segment time quantum = PTS[5:0] + 1
9:5	PBS1[4:0]	Phase buffer segment 1 Phase buffer segment 1 time quantum = PBS1[4:0] + 1
4:0	PBS2[4:0]	Phase buffer segment 2 Phase buffer segment 2 time quantum = PBS2[4:0] + 1

29.5.14. Receive FIFO/mailbox private filter x register (CAN_RFIFOMPFx)(x=0..31)

Address offset: 0x880 + 4 × x
Reset value: 0xXXXX XXXX

These register is located in RAM.

All bits of these registers should be configured in Inactive mode only, because they are blocked by hardware in other modes.

These registers are not affected by software reset bit SWRST in CAN_CTL0 register.

These registers have to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FMFD31	FMFD30	FMFD29	FMFD27	FMFD27	FMFD26	FMFD25	FMFD24	FMFD23	FMFD22	FMFD21	FMFD20	FMFD19	FMFD18	FMFD17	FMFD16
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FMFD15	FMFD14	FMFD13	FMFD12	FMFD11	FMFD10	FMFD9	FMFD8	FMFD7	FMFD6	FMFD5	FMFD4	FMFD3	FMFD2	FMFD1	FMFD0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:0	FMFDx	FIFO/mailbox filter data If used as mailbox filters, refer to the MFDx bits in CAN_RMPUBF register. If used as Rx FIFO filters, refer to the FFDx bits in CAN_RFIFOPUBF register. 0: The bit is "don't care"

1: The bit is checked

29.5.15. Pretended Networking mode control register 0 (CAN_PN_CTL0)

Address offset: 0xB00
Reset value: 0x0000 0100

All bits except bit 17, 16 of this register should be configured in Inactive mode only, because they are blocked by hardware in other modes.

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

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

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

Bits	Fields	Descriptions
31:18	Reserved	Must be kept at reset value.
17	WTOIE	Wakeup timeout interrupt enable 0: Disable wakeup timeout interrupt 1: Enable wakeup timeout interrupt
16	WMIE	Wakeup match interrupt enable 0: Disable wakeup match interrupt 1: Enable wakeup match interrupt
15:8	NMM[7:0]	Number of messages matching times An event counter is used in the wakeup message filter, in which a transistion on the output of event after N input matching events. 00000001: N = 1 00000010: N = 2 …… 11111111: N = 255
7:6	Reserved	Must be kept at reset value.
5:4	DATAFT[1:0]	DATA field filtering type in Pretended Networking mode 00: Only messages with DATA field equal to the expected data field through data filter are matched 01: Messages with DATA field greater than or equal to the expected data low threshold are matched 10: Messages with DATA field smaller than or equal to the expected data high threshold are matched 11: Messages with DATA field greater than or equal to the expected data low

		threshold, and smaller than or equal to the expected data high threshold are matched
3:2	IDFT[1:0]	ID field filtering type in Pretended Networking mode 00: Only messages with ID field equal to the expected identifier through identifier filter are matched 01: Messages with ID field greater than or equal to the expected identifier low threshold are matched 10: Messages with ID field smaller than or equal to the expected identifier high threshold are matched 11: Messages with ID field greater than or equal to the expected identifier low threshold, and smaller than or equal to the expected identifier high threshold are matched
1:0	FFT[1:0]	Frame filtering type in Pretended Networking mode 00: All fields except DATA field, DLC field are filtered 01: All fields are filtered 10: All fields except DATA field, DLC field are filtered with NMM[7:0] matching times 11: All fields are filtered with NMM[7:0] matching times

29.5.16. Pretended Networking mode timeout register (CAN_PN_TO)

Address offset: 0xB04
Reset value: 0x0000 0000

All bits of this register should be configured in Inactive mode only, because they are blocked by hardware in other modes.

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

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

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

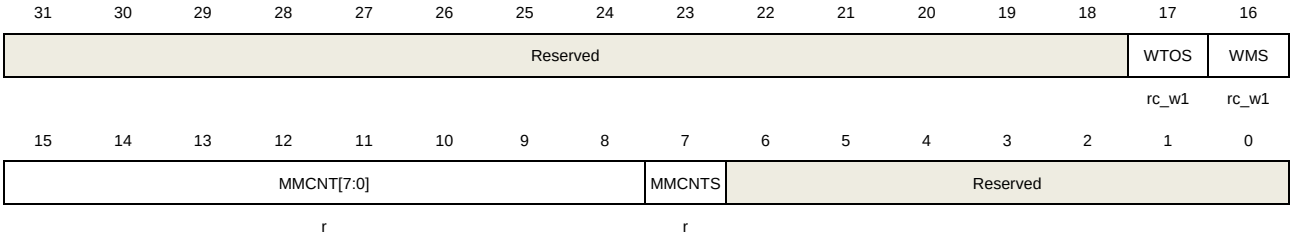
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	WTO[15:0]	Wakeup timeout The timeout is counted by step of 64 times the CAN Bit Time. Wakeup timeout is default disabled.

29.5.17. Pretended Networking mode status register (CAN_PN_STAT)

Address offset: 0xB08

Reset value: 0x0000 0080

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



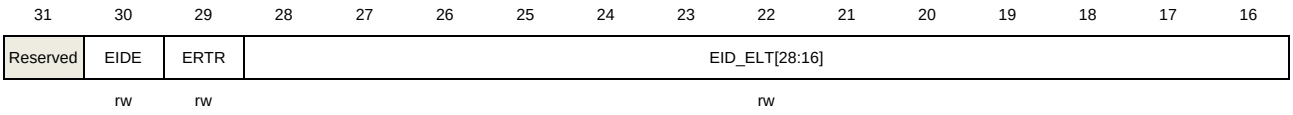
Bits	Fields	Descriptions
31:18	Reserved	Must be kept at reset value.
17	WTOS	Wakeup timeout flag status 0: No wakeup timeout event occured 1: Wakeup timeout event occured
16	WMS	Wakeup match flag status 0: No wakeup match event occured 1: Wakeup match event occured
15:8	MMCNT[7:0]	Matching message counter in Pretended Networking mode This bit field indicates the matching message number during Pretended Networking mode. These bits are cleared when node enters Pretended Networking mode, they are not affected by software reset.
7	MMCNTS	Matching message counter state This bit is set to 1 to show the value of MMCNT[7:0] is valid. 0: Matching message counter MMCNT[7:0] is updating 1: Matching message counter MMCNT[7:0] is valid
6:0	Reserved	Must be kept at reset value.

29.5.18. Pretended Networking mode expected identifier 0 register (CAN_PN_EID0)

Address offset: 0xB0C
Reset value: 0x0000 0000

All bits of this register should be configured in Inactive mode only, because they are blocked by hardware in other modes.

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



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

Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30	EIDE	Expected IDE in Pretended Networking mode 0: Standard frame format 1: Extended frame format
29	ERTR	Expected RTR in Pretended Networking mode 0: Data frame 1: Remote frame
28:0	EIDF_ELT[28:0]	Expected ID field / expected ID low threshold in Pretended Networking mode This bit field is used as expected ID field when IDFT[1:0] bit field in CAN_PN_CTL0 register is 0 / 1 / 2, or is used as expected ID low threshold when IDFT[1:0] bit field is 3. For extended frame format, all 29 bits are used. For standard frame format, bits 18 to 28 are used.

29.5.19. Pretended Networking mode expected DLC register (CAN_PN_EDLC)

Address offset: 0xB10
Reset value: 0x0000 0008

All bits of this register should be configured in Inactive mode only, because they are blocked by hardware in other modes.

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												DLCELT[3:0]			
												rw			

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

Bits	Fields	Descriptions
31:20	Reserved	Must be kept at reset value.
19:16	DLCELT[3:0]	DLC expected low threshold in Pretended Networking mode
15:4	Reserved	Must be kept at reset value.
3:0	DLCEHT[3:0]	DLC expected high threshold in Pretended Networking mode

29.5.20. Pretended Networking mode expected data low 0 register (CAN_PN_EDL0)

Address offset: 0xB14
Reset value: 0x0000 0000

All bits of this register should be configured in Inactive mode only, because they are blocked by hardware in other modes.

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DB0ELT[7:0]								DB1ELT[7:0]							
rw								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DB2ELT[7:0]								DB3ELT[7:0]							
rw								rw							

Bits	Fields	Descriptions
31:24	DB0ELT[7:0]	Data byte 0 expected low threshold in Pretended Networking mode Refer to DB3ELT[7:0] descriptions.
23:16	DB1ELT[7:0]	Data byte 1 expected low threshold in Pretended Networking mode Refer to DB3ELT[7:0] descriptions.
15:8	DB2ELT[7:0]	Data byte 2 expected low threshold in Pretended Networking mode Refer to DB3ELT[7:0] descriptions.
7:0	DB3ELT[7:0]	Data byte 3 expected low threshold in Pretended Networking mode This bit field is used as expected DATA field when DATAFT[1:0] bit field in CAN_PN_CTL0 register is 0 / 1 / 2, or is used as expected DATA low threshold when DATAFT[1:0] bit field is 3.

29.5.21. Pretended Networking mode expected data low 1 register (CAN_PN_EDL1)

Address offset: 0xB18
Reset value: 0x0000 0000

All bits of this register should be configured in Inactive mode only, because they are blocked by hardware in other modes.

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DB4ELT[7:0]								DB5ELT[7:0]							
rw								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DB6ELT[7:0]								DB7ELT[7:0]							
rw								rw							

Bits	Fields	Descriptions
31:24	DB4ELT[7:0]	Data byte 4 expected low threshold in Pretended Networking mode Refer to DB3ELT[7:0] descriptions.
23:16	DB5ELT[7:0]	Data byte 5 expected low threshold in Pretended Networking mode Refer to DB3ELT[7:0] descriptions.
15:8	DB6ELT[7:0]	Data byte 6 expected low threshold in Pretended Networking mode Refer to DB3ELT[7:0] descriptions.
7:0	DB7ELT[7:0]	Data byte 7 expected low threshold in Pretended Networking mode Refer to DB3ELT[7:0] descriptions.

29.5.22. Pretended Networking mode identifier filter / expected identifier 1 register (CAN_PN_IFEID1)

Address offset: 0x B1C
Reset value: 0x0000 0000

All bits of this register should be configured in Inactive mode only, because they are blocked by hardware in other modes.

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	IDEFD	RTRFD	IDFD_EHT[28:16]												
	rw	rw	rw												

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

Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30	IDEFD	IDE filter data in Pretended Networking mode 0: The bit is "don't care" 1: The bit is checked
29	RTRFD	RTR filter data in Pretended Networking mode 0: The bit is "don't care" 1: The bit is checked
28:0	IDFD_EHT[28:0]	ID filter data / ID expected high threshold in Pretended Networking mode

ID filter data (when IDFT[1:0] bit field in CAN_PN_CTL0 register is 0):

0: The bit is "don't care"

1: The bit is checked

ID expected high threshold (when IDFT[1:0] bit field is 3).

Bits reserved (when IDFT[1:0] bit field is 1 or 2).

For extended frame format, all 29 bits are used.

For standard frame format, bits 18 to 28 are used.

29.5.23. Pretended Networking mode data 0 filter / expected data high 0 register (CAN_PN_DF0EDH0)

Address offset: 0xB20

Reset value: 0x0000 0000

All bits of this register should be configured in Inactive mode only, because they are blocked by hardware in other modes.

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DB0FD_EHT[7:0]								DB1FD_EHT[7:0]							
rw								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DB2FD_EHT[7:0]								DB3FD_EHT[7:0]							
rw								rw							

Bits	Fields	Descriptions
31:24	DB0FD_EHT[7:0]	Data byte 0 filter data / Data byte 0 expected high threshold in Pretended Networking mode Refer to DB3FD_EHT[7:0] descriptions.
23:16	DB1FD_EHT[7:0]	Data byte 1 filter data / Data byte 1 expected high threshold in Pretended Networking mode Refer to DB3FD_EHT[7:0] descriptions.
15:8	DB2FD_EHT[7:0]	Data byte 2 filter data / Data byte 2 expected high threshold in Pretended Networking mode Refer to DB3FD_EHT[7:0] descriptions.
7:0	DB3FD_EHT[7:0]	Data byte 3 filter data / Data byte 2 expected high threshold in Pretended Networking mode **Data byte 3 filter data** (when DATAFT[1:0] bit field in CAN_PN_CTL0 register is 0): 0: The bit is "don't care" 1: The bit is checked **Data byte 3 expected high threshold** (when DATAFT[1:0] bit field is 3).

Bits reserved (when DATAFT[1:0] bit field is 1 or 2).

29.5.24. Pretended Networking mode data 1 filter / expected data high 1 register (CAN_PN_DF1EDH1)

Address offset: 0xB24
Reset value: 0x0000 0000

All bits of this register should be configured in Inactive mode only, because they are blocked by hardware in other modes.

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DB4FD_HTF[7:0]								DB5FD_HTF[7:0]							
rw								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DB6FD_HTF[7:0]								DB7FD_HTF[7:0]							
rw								rw							

Bits	Fields	Descriptions
31:24	DB4FD_HTF[7:0]	Data byte 4 filter data / Data byte 4 expected high threshold in Pretended Networking mode Refer to DB3FD_EHT[7:0] descriptions.
23:16	DB5FD_HTF[7:0]	Data byte 5 filter data / Data byte 5 expected high threshold in Pretended Networking mode Refer to DB3FD_EHT[7:0] descriptions.
15:8	DB6FD_HTF[7:0]	Data byte 6 filter data / Data byte 6 expected high threshold in Pretended Networking mode Refer to DB3FD_EHT[7:0] descriptions.
7:0	DB7FD_HTF[7:0]	Data byte 7 filter data / Data byte 7 expected high threshold in Pretended Networking mode Refer to DB3FD_EHT[7:0] descriptions.

29.5.25. Pretended Networking mode received wakeup mailbox x control status information register (CAN_PN_RWMxCS)(x=0..3)

Address offset: 0xB40 + 16 * x
Reset value: 0x0000 0000

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

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

Reserved									RSRR	RIDE	RRTR	RDLC[3:0]			
									r	r	r	r			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															

Bits	Fields	Descriptions
31:23	Reserved	Must be kept at reset value.
22	RSRR	Received SRR bit
21	RIDE	Received IDE bit 0: Frame format is standard 1: Frame format is extended
20	RRTR	Received RTR bit 0: Frame is data frame 1: Frame is remote frame
19:16	RDLC[3:0]	Received DLC bits The bit field indicates the valid data byte length.
15:0	Reserved	Must be kept at reset value.

29.5.26. Pretended Networking mode received wakeup mailbox x identifier register (CAN_PN_RWMxI)(x=0..3)

Address offset: 0xB44 + 16 * x

Reset value: 0x0000 0000

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

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

Bits	Fields	Descriptions
31:29	Reserved	Must be kept at reset value.
28:0	RID[28:16]	Received ID bits For extended frame format, all 29 bits are used for ID storage. For standard frame format, bits 18 to 28 are used for ID storage.

29.5.27. Pretended Networking mode received wakeup mailbox x data 0 register (CAN_PN_RWMxD0)(x=0..3)

Address offset: 0xB48 + 16 * x

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RDB0[7:0]								RDB1[7:0]							
r								r							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDB2[7:0]								RDB3[7:0]							
r								r							

Bits	Fields	Descriptions
31:24	RDB0[7:0]	Received data byte 0
23:16	RDB1[7:0]	Received data byte 1
15:8	RDB2[7:0]	Received data byte 2
7:0	RDB3[7:0]	Received data byte 3

29.5.28. Pretended Networking mode received wakeup mailbox x data 1 register (CAN_PN_RWMxD1)(x=0..3)

Address offset: 0xB4C + 16 * x

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RDB4[7:0]								RDB5[7:0]							
r								r							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDB6[7:0]								RDB7[7:0]							
r								r							

Bits	Fields	Descriptions
31:24	RDB4[7:0]	Received data byte 4
23:16	RDB5[7:0]	Received data byte 5
15:8	RDB6[7:0]	Received data byte 6
7:0	RDB7[7:0]	Received data byte 7

29.5.29. FD control register (CAN_FDCTL)

Address offset: 0xC00

Reset value: 0x8000 0101

Bits 17:16, 15, 12:8 of this register should be configured in Inactive mode only, because they are blocked by hardware in other modes.

This register is not affected by software reset bit SWRST in CAN_CTL0 register.

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BRSEN	Reserved													MDSZ[1:0]	
rw														rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDCEN	TDCS	Reserved	TDCO[4:0]					Reserved		TDCV[5:0]					
rw	rc_w1		rw							r					

Bits	Fields	Descriptions
31	BRSEN	Bit rate of data switch enable 0: Bit rate not switch 1: The bit rate shall be switched from the nominal bit rate to the preconfigured data bit rate during the data phase when BRS bit in Tx mailbox is recessive ‘1’
30:18	Reserved	Must be kept at reset value.
17:16	MDSZ[1:0]	Mailbox data size 00: 8 bytes per mailbox 01: 16 bytes per mailbox 10: 32 bytes per mailbox 11: 64 bytes per mailbox
15	TDCEN	Transmitter delay compensation enable **Note:** Transmitter delay compensation must be disabled when loopback and silent mode is enabled. 0: Transmitter delay compensation is disabled 1: Transmitter delay compensation is enabled
14	TDCS	Transmitter delay compensation status When this bit is set, the transmitter delay is out of compensation range, it is unable to compensate the transmitter delay for bit check. 0: Transmitter delay is in compensation range 1: Transmitter delay is out of compensation range
13	Reserved	Must be kept at reset value.
12:8	TDCO[4:0]	Transmitter delay compensation offset These bits are set to the transmitter delay compensation offset value which defines

		the distance between the measured delay from CANTX to CANRX and the second sample point for CAN FD frames with BRS bit set.
7:6	Reserved	Must be kept at reset value.
5:0	TDCV[5:0]	Transmitter delay compensation value These bits are set by hardware to display the summary of the measured transmitter delay value and the transmitter delay compensation offset.

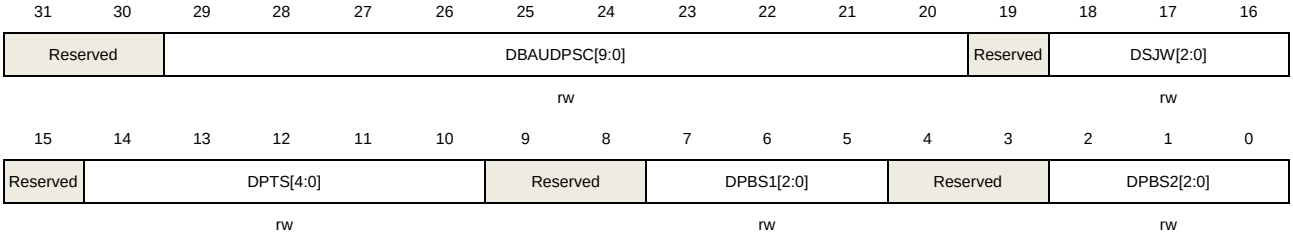
29.5.30. FD bit timing register (CAN_FDBT)

Address offset: 0xC04
Reset value: 0x0000 0000

All bits of this register should be configured in Inactive mode only, because they are blocked by hardware in other modes.

This register is not affected by software reset bit SWRST in CAN_CTL0 register.

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



Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29:20	DBAUDPSC[9:0]	Baud rate prescaler for data bit time The CAN data bit time baud rate prescaler = BAUDPSC[9:0] + 1.
19	Reserved	Must be kept at reset value.
18:16	DSJW[2:0]	Resynchronization jump width for data bit time Resynchronization jump width time quantum = DSJW[2:0] + 1
15	Reserved	Must be kept at reset value.
14:10	DPTS[4:0]	Propagation time segment for data bit time Propagation time segment time quantum = DPTS[4:0]
9:8	Reserved	Must be kept at reset value.
7:5	DPBS1[2:0]	Phase buffer segment 1 for data bit time Phase buffer segment 1 time quantum = DPBS1[2:0] + 1
4:3	Reserved	Must be kept at reset value.

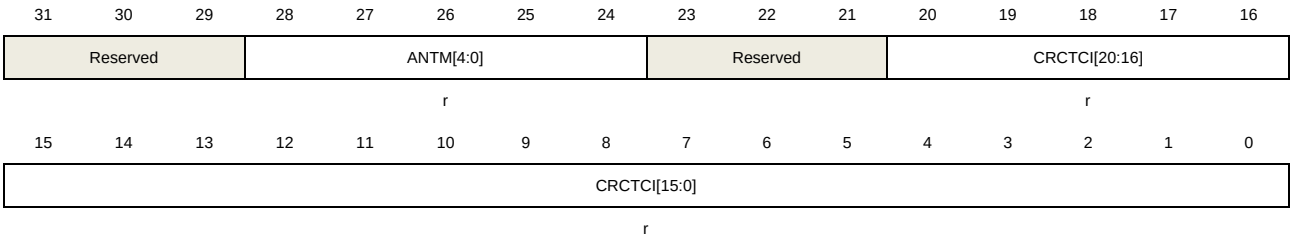
2:0	DPBS2[2:0]	Phase buffer segment 2 for data bit time Phase buffer segment 2 time quantum = DPBS2[2:0] + 1

29.5.31. CRC for classical and FD frame register (CAN_CRCCFD)

Address offset: 0xC08

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:29	Reserved	Must be kept at reset value.
28:24	ANTM[4:0]	Associated number of mailbox for transmitting the CRCTCI[20:0] value This bit field contains the number of the mailbox which transmits the CRCTCI[20:0] value for both classical and FD frames.
23:21	Reserved	Must be kept at reset value.
20:0	CRCTCI[20:0]	Transmitted CRC value for classical and ISO / non-ISO FD frames For CRC_15, bits 0 to 14 are used, the other bits are zeros, and the value is the same as the value of CRCTC[14:0] in CAN_CRCC register. For CRC_17, bits 0 to 16 are used, the other bits are zeros. For CRC_21, all 21 bits are used.

30. Serial peripheral interface (SPI)

30.1. Overview

The SPI module can communicate with external devices using the SPI 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.

30.2. Characteristics

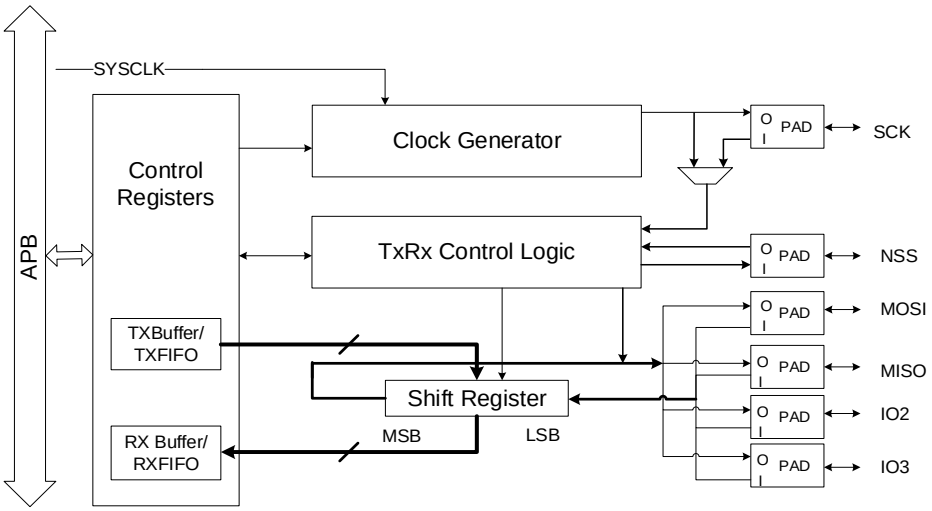
30.2.1. SPI characteristics

- Master or slave operation with full-duplex or simplex mode.
- Separate transmission and reception 32-bit FIFO.
- Data frame size can be 4 to 16 bits.
- Bit order can be LSB or MSB.
- Software and hardware NSS management.
- Hardware CRC calculation, transmission and checking.
- Transmission and reception using DMA.
- SPI TI mode supported.
- SPI NSS pulse mode supported.
- Quad-SPI configuration available in master mode (only in SPI0).

30.3. SPI function overview

30.3.1. SPI block diagram

Figure 30-1. Block diagram of SPI



30.3.2. SPI signal description

Normal configuration (Not Quad-SPI Mode)

Table 30-1. SPI signal description

Pin name	Direction	Description
SCK	I/O	Master: SPI clock output Slave: SPI clock input
MISO	I/O	Master: Data reception line Slave: Data transmission line Master with bidirectional mode: Not used Slave with bidirectional mode: Data transmission and reception line.
MOSI	I/O	Master: Data transmission line Slave: Data reception line Master with bidirectional mode: Data transmission and reception line. Slave with bidirectional mode: Not used
NSS	I/O	Software NSS mode: not used Master in hardware NSS mode: when NSSDRV=1, it is NSS output, suitable for single master application; when NSSDRV=0, it is NSS input, suitable for multi-master

Pin name	Direction	Description
		application. Slave in hardware NSS mode: NSS input, as a chip select signal for slave.

Quad-SPI configuration

SPI is in single wire mode by default and enters into Quad-SPI mode after QMOD bit in SPI_QCTL register is set (only available in SPI0). Quad-SPI mode can only work in master mode.

The IO2 and IO3 pins can be driven high in normal Non-Quad-SPI mode by configuring IO23_DRV bit in SPI_QCTL register.

The SPI is connected to external devices through 6 pins in Quad-SPI mode:

Table 30-2. Quad-SPI signal description

Pin name	Direction	Description
SCK	O	SPI clock output
MOSI	I/O	Transmission/Reception data 0
MISO	I/O	Transmission/Reception data 1
IO2	I/O	Transmission/Reception data 2
IO3	I/O	Transmission/Reception data 3
NSS	O	NSS output

30.3.3. 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.

In SPI0 normal mode, the length of data is configured by the DZ bits in the SPI_CTL1 register. It can be set from 4-bit up to 16-bit length and the setting applies for both transmission and reception, and the read access to the FIFO must be aligned with the BYTEN bit setting in the SPI_CTL1 register. 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.

When the SPI_DATA register is accessed, data frames are always right-aligned into either a byte (if the data fits into a byte) or a half-word. During communication, only bits within the data frame are clocked and transferred.

Figure 30-2. SPI0 timing diagram in normal mode

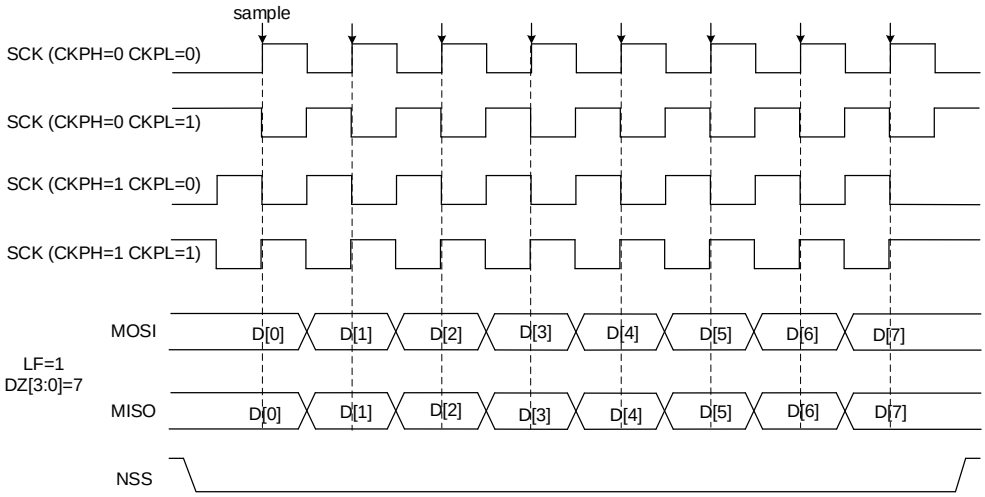
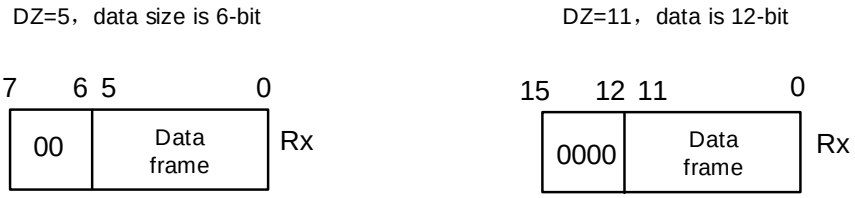
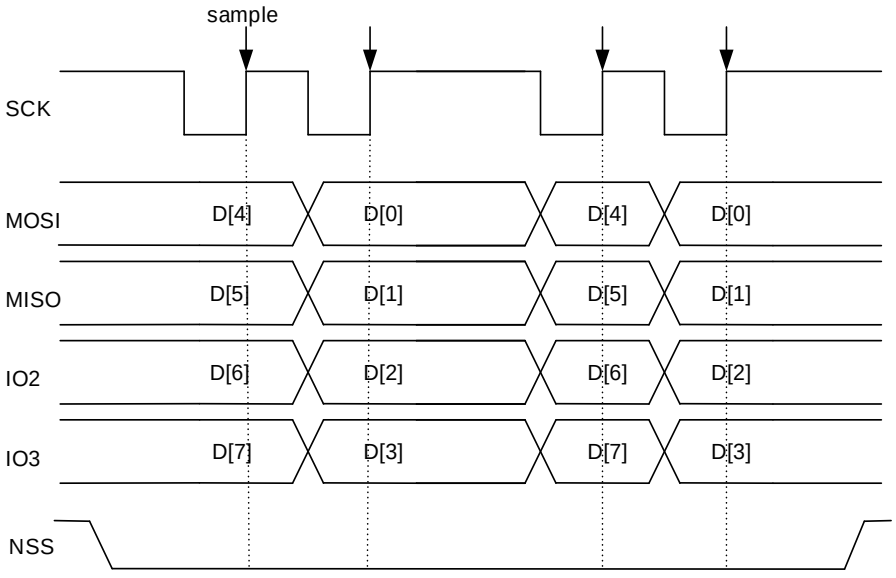


Figure 30-3. SPI0 data frame right-aligned diagram



Data order is configured by LF bit in SPI_CTL0 register, and SPI1 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.

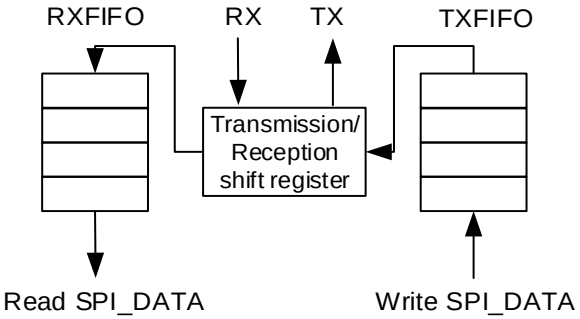
Figure 30-4. SPI timing diagram in Quad-SPI mode (CKPL=1, CKPH=1, LF=0)



30.3.4. Separate transmission and reception FIFO

The separate 32-bit reception FIFO (RXFIFO) and transmission FIFO (TXFIFO) are used in different directions for SPI data transactions, and they can enable the SPI to work in a continuous flow.

Figure 30-5. Transmission and reception FIFO



When the current TXFIFO level is less than or equal to half of its capacity, the TXFIFO is considered empty(1) and TBE is set to 1 by hardware at this time. When the TBE bit is set, writing data to the SPI_DATA register will store the data at the end of the TXFIFO. Hardware sets the RBNE bit when the RXFIFO is considered non-empty(2). When the RBNE bit is set, reading data from the SPI_DATA register will get the oldest data from the RXFIFO.

Note: (1) The TXFIFO empty means that the TXFIFO level is less than or equal to half of its capacity. The meaning of TXFIFO full is the opposite. Therefore, when the data frame format is not greater than 8 bits, the TXFIFO can store up to three data frames. If the TXFIFO empty or full appears below and there is no special explanation, the meaning is the same as that described here.

(2) The meaning of RXFIFO empty is divided into the following two conditions: If BYTEN bit in SPI_CTL1 is set, the RXFIFO empty means the RXFIFO level is less than quarter of its capacity. At this time, when the data frame format is not more than 8 bits, the RXFIFO can store up to 4 data frames. If BYTEN is cleared, the RXFIFO empty means the RXFIFO level is less than half of its capacity. The meaning of RXFIFO full is the opposite. If the RXFIFO empty or full appears below and there is no special explanation, the meaning is the same as that described here.

Data merging

When DZ[3:0] in the SPI_CTL1 register configures the transmission data bit width to be 8 bits or less than 8 bits, by configuring the BYTEN bit in the SPI_CTL1 register to 0, the data merge transmission mode function is enabled. When DZ[3:0] in the configuration SPI_CTL1 register configures the transmission data bit width to be less than or equal to 8 bits, this function can realize that when 16-bit write access is performed to the SPI_DATA register, two data frames are sent in parallel instead of serial line method.

Similarly, at the receiving end, the receiver obtains these two data frames through a 16-bit read access to SPI_DATA, and only one RBNE event will be generated when the two frames of data are received.

Note: when an odd number of data bytes will be transferred, on the transmitter side, writing the last data frame of any odd sequence with an 8-bit access to SPI_DATA is enough. The receiver has to change BYTEN for the last data frame received in the odd sequence of frames in order to generate the RBNE event.

30.3.5. 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 30-3. NSS function in slave mode

Mode	Register configuration	Description
Slave hardware NSS mode	MSTMOD = 0 SWNSSEN = 0	SPI slave gets NSS level from NSS pin.
Slave software NSS mode	MSTMOD = 0 SWNSSEN = 1	SPI slave NSS level is determined by the SWNSS bit. SWNSS = 0: NSS level is low SWNSS = 1: NSS level is high

Master mode

In master mode (MSTMOD=1) if the application uses multi-master connection, NSS can be configured to hardware input mode (SWNSSEN=0, NSSDRV=0) or software mode (SWNSSEN=1). Then, once the NSS pin (in hardware NSS mode) or the SWNSS bit (in software NSS mode) goes low, the SPI automatically enters slave mode and triggers a master fault flag CONFERR.

If the application wants to use NSS line to control the SPI slave, NSS should be configured to hardware output mode (SWNSSEN=0, NSSDRV=1). NSS stays high after SPI is enabled and goes low when transmission or reception process begins. When SPI is disabled, the NSS goes high.

The application may also use a general purpose IO as NSS pin to realize more flexible NSS.

Table 30-4. NSS function in master mode

Mode	Register configuration	Description
Master hardware NSS output mode	MSTMOD = 1 SWNSSEN = 0 NSSDRV=1	Applicable to single-master mode. The master uses the NSS pin to control the SPI slave device. At this time, the NSS is configured as the hardware output mode. NSS goes low after enabling SPI.
Master hardware NSS input mode	MSTMOD = 1 SWNSSEN = 0 NSSDRV=0	Applicable to multi-master mode. At this time, NSS is configured as hardware input mode. Once the NSS pin is pulled low, SPI will automatically enter slave mode, and a master configuration error will occur and the CONFERR bit will be set to 1.
Master software NSS mode	MSTMOD = 1 SWNSSEN = 1 SWNSS = 0 NSSDRV: Don't care	Applicable to multi-master mode. Once SWNSS = 0, SPI will automatically enter slave mode, and a master configuration error will occur and the CONFERR bit will be 1.
	MSTMOD = 1 SWNSSEN = 1 SWNSS = 1 NSSDRV: Don't care	The slave can use hardware or software NSS mode.

30.3.6. SPI operation modes

Table 30-5. 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	MOSI: Not used MISO: Reception

Mode	Description	Register configuration	Data pin usage
		BDOEN: Don't care	
MTB	Master transmission with bidirectional connection	MSTMOD = 1 RO = 0 BDEN = 1 BDOEN = 1	MOSI: Transmission MISO: Not used
MRB	Master reception with bidirectional connection	MSTMOD = 1 RO = 0 BDEN = 1 BDOEN = 0	MOSI: Reception MISO: Not used
SFD	Slave full-duplex	MSTMOD = 0 RO = 0 BDEN = 0 BDOEN: Don't care	MOSI: Reception MISO: Transmission
STU	Slave transmission with unidirectional connection	MSTMOD = 0 RO = 0 BDEN = 0 BDOEN: Don't care	MOSI: Not used MISO: Transmission
SRU	Slave reception with unidirectional connection	MSTMOD = 0 RO = 1 BDEN = 0 BDOEN: Don't care	MOSI: Reception MISO: Not used
STB	Slave transmission with bidirectional connection	MSTMOD = 0 RO = 0 BDEN = 1 BDOEN = 1	MOSI: Not used MISO: Transmission
SRB	Slave reception with bidirectional connection	MSTMOD = 0 RO = 0 BDEN = 1 BDOEN = 0	MOSI: Not used MISO: Reception

Figure 30-6. A typical full-duplex connection

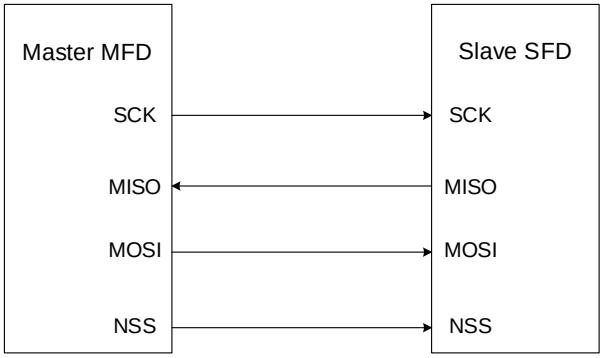


Figure 30-7. A typical simplex connection (Master: Receive, Slave: Transmit)

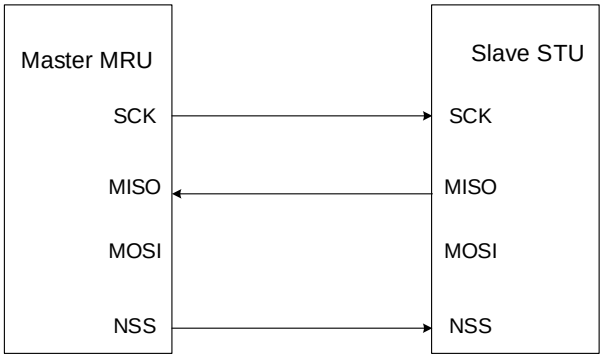


Figure 30-8. A typical simplex connection (Master: Transmit only, Slave: Receive)

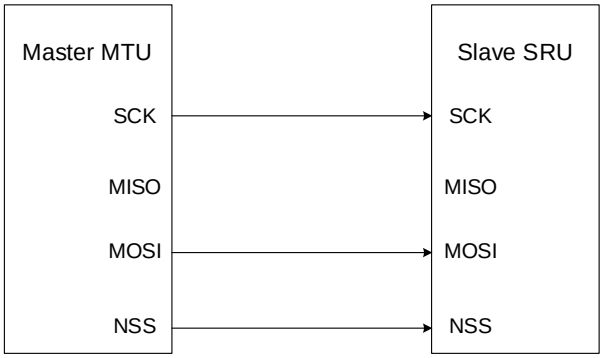
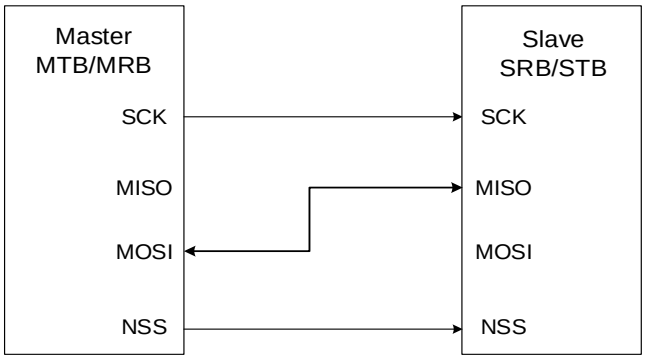


Figure 30-9. A typical bidirectional connection



Initialization sequence

Before transmitting 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 the clock timing register (CKPL and CKPH bits in the SPI_CTL0 register).
3. Program the frame format (LF bit in the SPI_CTL0 register).
4. Program data format (DZ bits in the SPI_CTL1 register) and the access size for the SPI_DATA register (BYTEN bit in the SPI_CTL1 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 in ***SPI operation modes*** section.
9. Initialize TXDMA_ODD/RXDMA_ODD bits if they are needed when DMA is used in packed mode.
10. If Quad-SPI mode is used, set the QMOD bit in SPI_QCTL register. Ignore this step if Quad-SPI mode is not used.
11. Enable the SPI (set the SPIEN bit).

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/TXFIFO. 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/TXFIFO 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/TXFIFO to the shift register and then begins to transmit the loaded data frame. After TBE flag is set, which means the transmit buffer/TXFIFO is empty, the application should write SPI_DATA register again if it has more data to transmit.

In master mode, software should write the next data into SPI_DATA register before the transmission of current data frame is completed if it desires to generate continuous transmission.

Reception sequence

After the last valid sample clock, the incoming data will be moved from shift register to the receive buffer/RXFIFO and RBNE will be set. The application should read SPI_DATA register to get the received data and this will clear the RBNE flag automatically when receive buffer/RXFIFO is empty. In MRU and MRB modes, hardware continuously sends clock signal to receive the next data frame, while in full-duplex master mode (MFD), hardware only receives the next data frame when the transmit buffer/TXFIFO is not empty.

SPI operation sequence in different modes (Not Quad-SPI, TI mode or NSSP mode)

In full-duplex mode, either MFD or SFD, the RBNE and TBE flags should be monitored and then follow the sequences described above.

The transmission mode (MTU, MTB, STU or STB) is similar to the transmission sequence of full-duplex mode regardless of the RBNE and OVRE bits.

The master reception mode (MRU or MRB) is different from the reception sequence of full-duplex mode. In MRU or MRB mode, after SPI is enabled, the SPI continuously generates SCK until the SPI is disabled. So the application should ignore the TBE flag and read out reception buffer/RXFIFO in time after the RBNE flag is set, otherwise a data overrun fault will occur.

The slave reception mode (SRU or SRB) is similar to the reception sequence of full-duplex mode regardless of the TBE flag.

SPI TI mode

SPI TI mode takes NSS as a special frame header flag signal and its operation sequence is similar to normal mode described above. The modes described above (MFD, MTU, MRU, MTB, MRB, SFD, STU, SRU, STB and SRB) are still supported in TI mode. While, in TI mode the CKPL and CKPH bits in SPI_CTL0 registers take no effect and the SCK sample edge is falling edge.

Figure 30-10. Timing diagram of TI master mode with discontinuous transfer

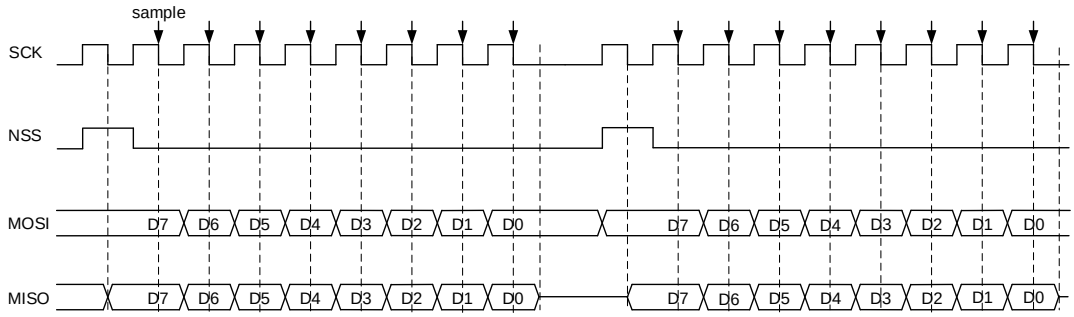
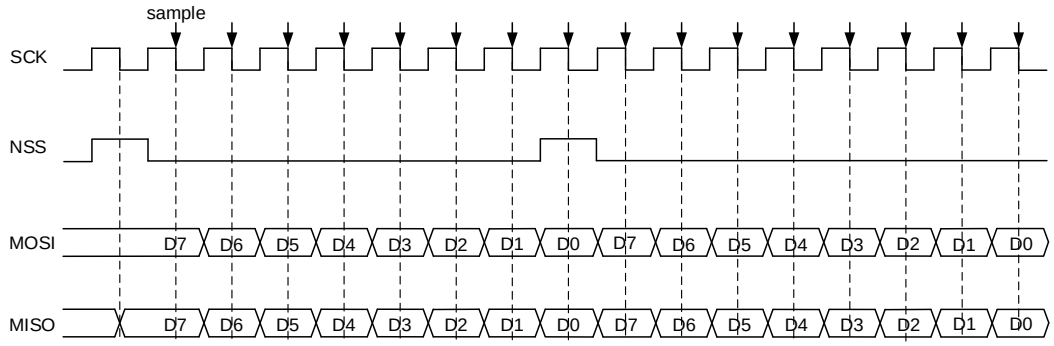
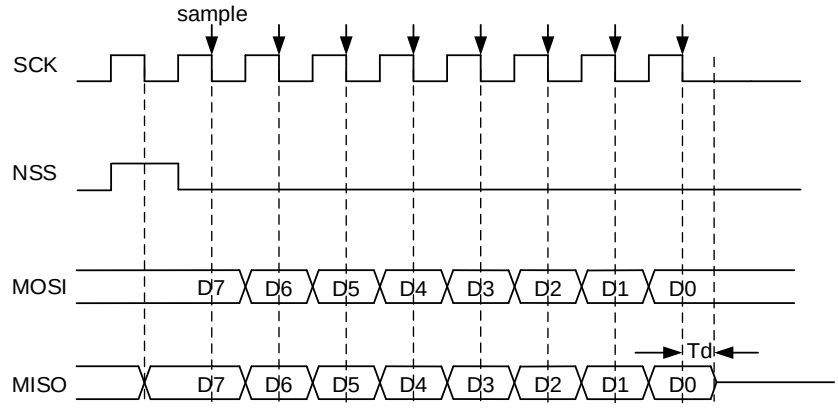


Figure 30-11. Timing diagram of TI master mode with continuous transfer



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 30-12. 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.

$$T_d = \frac{T_{bit}}{2} + 5 * T_{pclk} \quad (30\text{-}1)$$

For example, if PSC [2:0] = 010, T_d is $9*T_{pclk}$.

In slave mode, the slave also monitors the NSS signal and sets an error flag FERR if it detects an incorrect NSS behavior, for example: toggles at the middle bit of a byte.

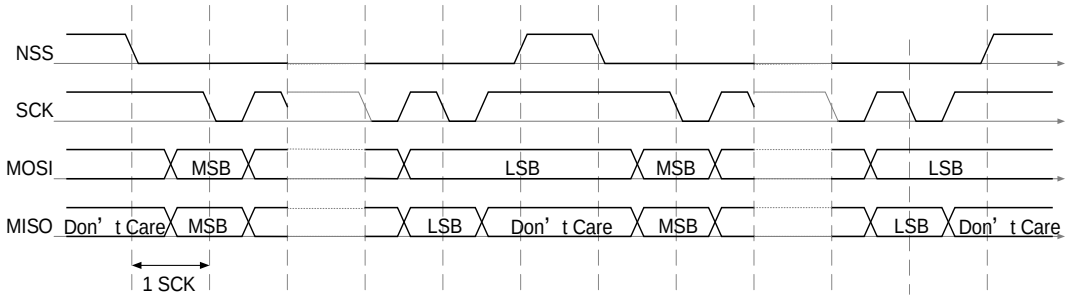
NSS pulse mode operation sequence

This function is controlled by NSSP bit in SPI_CTL1 register. In order to implement this function, several additional conditions must be met: configure the device to master mode, frame format should follow the normal SPI protocol, select the first clock transition as the data capture edge.

In summary, MSTMOD = 1, NSSP = 1, CKPH = 0.

When NSS pulse mode is enabled, a pulse duration of at least 1 SCK clock period is inserted between two successive data frames depending on the status of internal data transmit buffer/TXFIFO. Multiple SCK clock cycle intervals are possible if the transfer buffer/TXFIFO 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 30-13. 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, CRCL, RO and LF in SPI_CTL0 register should be kept cleared and DZ[3:0] should be set to ensure that SPI data size is 8-bit, MSTMOD should be set to ensure that SPI is in master mode. SPIEN, PSC, CKPL and CKPH should be configured as desired.

There are two operation modes in Quad-SPI mode: quad write and quad read, decided by QRD bit in SPI_QCTL register.

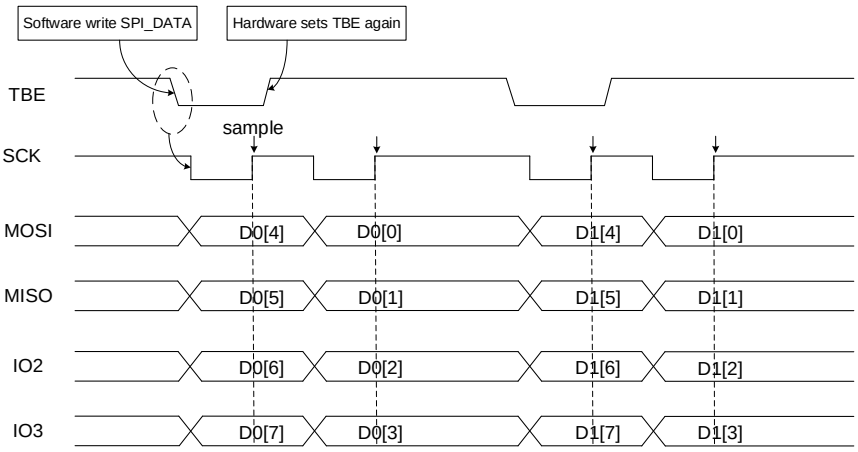
Quad write operation

SPI works in quad write mode when QMOD is set and QRD is cleared in SPI_QCTL register. In this mode, MOSI, MISO, IO2 and IO3 are all used as output pins. SPI begins to generate clock on SCK line and transmit data on MOSI, MISO, IO2 and IO3 as soon as data is written into SPI_DATA (TBE is cleared) and SPIEN is set. Once SPI starts transmission, it always checks TBE status at the end of a frame and stops when condition is not met.

The operation flow for transmitting in quad mode:

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 a 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 30-14. Timing diagram of quad write operation in Quad-SPI mode



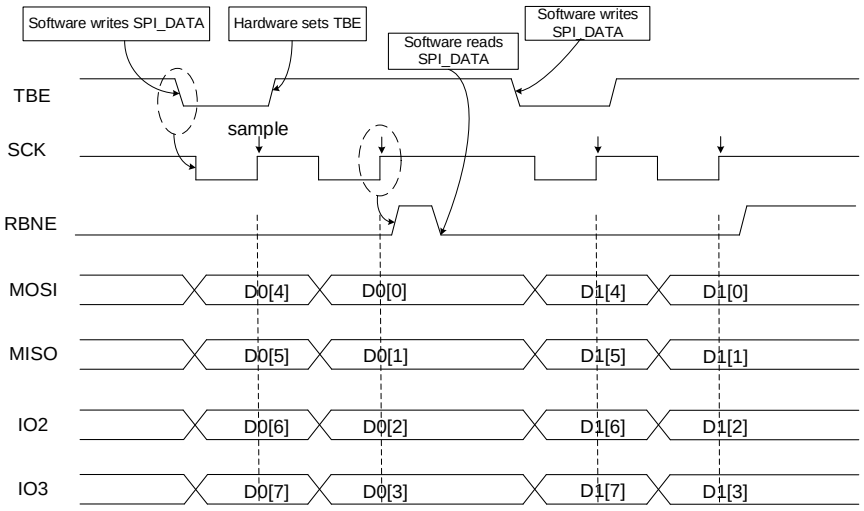
Quad read operation

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, dummy data should always be written into SPI_DATA to generate SCK.

The operation flow for receiving in quad mode is shown below:

1. Configure clock prescaler, clock polarity, phase, etc. 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 30-15. 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

For SPI, wait until TXLVL[1:0]=00 and confirm TRANS=0. Then disable the SPI by clearing SPIEN bit. At last, read data until RXTVL[1:0]=00.

MTU MTB STU STB

For SPI, wait until TXLVL[1:0]=00 and confirm TRANS=0. Then disable the SPI by clearing SPIEN bit.

MRU MRB

For SPI, application can disable the SPI when it doesn't want to receive data, and then confirm the TRANS=0 and read data until RXLVL[1:0]=00.

SRU SRB

For SPI, application can disable the SPI when it doesn't want to receive data, and then confirm the TRANS=0 and read data until RXLVL[1:0]=00.

TI mode

The disabling sequence of TI mode is the same as the sequences described above.

NSS pulse mode

The disabling sequence of NSSP mode is the same as the sequences described above.

Quad-SPI mode

Before leaving quad wire mode or disabling SPI, software should first check that TBE bit is set and TRANS bit is cleared, then the QMOD bit in SPI_QCTL register and SPIEN bit in SPI_CTL0 register are cleared.

30.3.7. DMA function

The DMA frees the application from data writing and reading process during transfer, to improve the system efficiency.

DMA function in SPI is enabled by setting DMATEN and DMAREN bits in SPI_CTL1 register. To use DMA function, application should first correctly configure DMA modules, then configure SPI module according to the initialization sequence, at last enable SPI.

After being enabled, If DMATEN is set, SPI will generate a DMA request each time when TBE=1, then DMA will acknowledge to this request and write data into the SPI_DATA register automatically. If DMAREN is set, SPI will generate a DMA request each time when RBNE=1, then DMA will acknowledge to this request and read data from the SPI_DATA register automatically.

Data merging with DMA

In the case of using DMA for data transmission, when BYTEN is set to 0 and the data length configured by DZ[3:0] is less than or equal to 8 bits and the data merging mode is enabled, the DMA will access the SPI_DATA register in 16-bit mode, automatically Complete the data transmission.

In the case that the data packetization mode is enabled and the frame number of the data frame is not an even multiple, in order to avoid the problem of one more frame of data in the last DMA transmission, the TXDMA_ODD/RXDMA_ODD bit in the SPI_CTL1 register needs to be set to 1

30.3.8. CRC function

There are two CRC calculators in SPI: one for transmission and the other for reception. The CRC calculation uses the polynomial defined in SPI_CRCPOLY register.

Application can enable the CRC function by setting CRCEN bit in SPI_CTL0 register. The CRC calculators continuously calculate CRC for each bit transmitted and received on lines, and the calculated CRC values can be read from SPI_TCRC and SPI_RCRC registers.

To transmit the calculated CRC value, application should set the CRCNT bit in SPI_CTL0 register after the last data is written to the transmit buffer/TXFIFO. In full-duplex mode (MFD or SFD), when the SPI transmits a CRC and prepares to check the received CRC value, the SPI treats the incoming data as a CRC value. In reception mode (MRB, MRU, SRU and SRB), the application should set the CRCNT bit after the second last data frame is received. When

CRC checking fails, the CRCERR flag will be set.

For SPI, a CRC-format transaction usually takes one more data frame to communicate at the end of data sequence. However, when setting an 8-bit data frame checked by 16-bit CRC, two more frames are necessary to send the complete CRC. If DMA function is enabled, the counter for the SPI transmission DMA channel has to be set to the number of data frames to transmit excluding the CRC frame. On the receiver side, the DMA counter should be configured as follows:

1. Full duplex mode: Suppose the amount of data received by SPI is L, when CRCL = 0 and DZ = 8, then the count of the DMA receive channel is L + 1, otherwise the count of the DMA receive channel is L + 2.

2. Receive only mode: DMA receive channel count is only equal to the amount of data received. After receiving data, the CRC value is obtained by reading SPI_RCRC register by software.

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.

30.3.9. SPI interrupts

Status flags

- **Transmit buffer/TXFIFO empty flag (TBE)**

This bit is set when the transmit buffer is empty or the TXFIFO level is lower or equal to 1/2 of FIFO depth, the software can write the next data to the transmit buffer/TXFIFO by writing the SPI_DATA register.

- **Receive buffer/RXFIFO not empty flag (RBNE)**

For SPI, this bit is set depending on the BYTEN bit in the SPI_CTL1: If BYTEN = 0, the RBNE is set when the RXFIFO level is greater or equal to 1/2(16-bit). If BYTEN = 1, the RBNE is set when the RXFIFO level is greater or equal to 1/4(8-bit).

- **SPI transmitting ongoing flag (TRANS)**

TRANS is a status flag to indicate whether the transfer is ongoing or not. It is set and cleared by hardware and not controlled by software. This flag doesn't generate any interrupt.

Error conditions

- **Configuration fault error (CONFERR)**

CONFERR is an error flag in master mode. In NSS hardware mode and the NSSDRV is not enabled, the CONFERR is set when the NSS pin is pulled low. In NSS software mode, the

CONFERR is set when the SWNSS bit is 0. When the CONFERR is set, the SPIEN bit and the MSTMOD bit are cleared by hardware, the SPI is disabled and the device is forced into slave mode.

The SPIEN and MSTMOD bit are write protection until the CONFERR is cleared. The CONFERR bit of the slave cannot be set. In a multi-master configuration, the device can be in slave mode with CONFERR bit set, which means there might have been a multi-master conflict for system control.

- **Rx overrun error (RXORERR)**

The RXORERR bit is set if a data is received when the RBNE is set. For SPI1, that means the last data has not been read out and the newly incoming data is received. For SPI, that means the RXFIFO has not enough space to store this received data. The receive buffer/RXFIFO contents won't be covered with the newly incoming data, so the newly incoming data is lost.

- **Format error (FERR)**

In slave TI mode, the slave also monitors the NSS signal and set an error flag if it detects an incorrect NSS behavior, for example: toggles at the middle bit of a byte.

- **CRC error (CRCERR)**

When the CRCEN bit is set, the CRC calculation result of the received data in the SPI_RCRC register is compared with the received CRC value after the last data, the CRCERR is set when they are different.

Table 30-6. SPI interrupt requests

Flag	Description	Clear method	Interrupt enable bit
TBE	Transmit buffer/TXFIFO empty	Write SPI_DATA register.	TBEIE
RBNE	Receive buffer/RXFIFO 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	

30.4. Register definition

SPI0 base address: 0x4001 3000
SPI1 base address: 0x4000 3800
SPI2 base address: 0x4000 3C00

30.4.1. Control register 0 (SPI_CTL0)

Address offset: 0x00
Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	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	CRCL	RO	SWNSSEN	SWNSS	LF	SPIEN	PSC [2:0]			MSTMOD	CKPL	CKPH
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw			rw	rw	rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	BDEN	Bidirectional enable 0: 2 line unidirectional transmit mode 1: 1 line bidirectional transmit mode. The information transfers between the MOSI pin in master and the MISO pin in slave.
14	BDOEN	Bidirectional transmit output enable When BDEN is set, this bit determines the direction of transfer. 0: Work in receive-only mode 1: Work in transmit-only mode
13	CRCEN	CRC calculation enable 0: CRC calculation is disabled. 1: CRC calculation is enabled.
12	CRCNT	CRC next transfer 0: Next transfer is data 1: Next transfer is CRC value (TCRC) When the transfer is managed by DMA, CRC value is transferred by hardware. This bit should be cleared. In full-duplex or transmit-only mode, set this bit after the last data is written to SPI_DATA register. In receive only mode, set this bit after the second last data is received.

11	CRCL	CRC length (only for SPI0) 0: 8-bit crc length. 1: 16-bit crc length.
10	RO	Receive only When BDEN is cleared, this bit determines the direction of transfer. 0: Full-duplex mode 1: Receive-only mode
9	SWNSSEN	NSS software mode selection 0: NSS hardware mode. The NSS level depends on NSS pin. 1: NSS software mode. The NSS level depends on SWNSS bit. 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/2 001: PCLK/4 010: PCLK/8 011: PCLK/16 100: PCLK/32 101: PCLK/64 110: PCLK/128 111: PCLK/256 PCLK means PCLK2 when using SPI0. PCLK means 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.

30.4.2. Control register 1 (SPI_CTL1)

Address offset: 0x04

Reset value: 0x0000 0700 for SPI

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	TXDMA_ ODD	RXDMA_ ODD	BYTEN	DZ[3:0]				TBEIE	RBNEIE	ERRIE	TMOD	NSSP	NSSDRV	DMATEN	DMAREN
	rw	rw	rw	rw				rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:15	Reserved	Must be kept at reset value.
14	TXDMA_ODD	Odd bytes in TX DMA channel In data merging mode, this bit is set if the total number of data to transmit by DMA is odd. It has effect only when DMATEN is set and data merging mode enable (data size is less than or equal to 8-bit and write access to SPI_DATA is 16-bit wide). This field can be written only when SPI is disabled. 0: The total number of data to transmit by DMA is even. 1: The total number of data to transmit by DMA is odd.
13	RXDMA_ODD	Odd bytes in RX DMA channel In data merging mode, this bit is set if the total number of data to receive by DMA is odd. It has effect only when DMAREN is set and data merging mode enable (data size is less than or equal to 8-bit and write access to SPI_DATA is 16-bit wide). This field can be written only when SPI is disabled. 0: The total number of data to receive by DMA is even. 1: The total number of data to receive by DMA is odd.
12	BYTEN	Byte access enable This bit is used to indicate the access size to FIFO, and set the threshold of the RXFIFO that generate RBNE. 0: Half-word access, and RBNE is generated when RXLVL >= 2. 1: Byte access, and RBNE is generated when RXLVL >= 1.
11:8	DZ[3:0]	Date size

		This field indicates the data size for transfer. 0000: Force to “0111” 0001: Force to “0111” 0010: Force to “0111” 0011: 4-bit 0100: 5-bit …… 1111: 16-bit
7	TBEIE	Transmit buffer / TXFIFO 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 / RXFIFO 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 / TXFIFO DMA enable 0: Transmit buffer / TXFIFO DMA is disabled. 1: Transmit buffer / TXFIFO DMA is enabled, when the TBE bit in SPI_STAT is set, it will be a DMA request on corresponding DMA channel.
0	DMAREN	Receive buffer / RXFIFO DMA enable 0: Receive buffer / RXFIFO DMA is disabled. 1: Receive buffer / RXFIFO DMA is enabled, when the RBNE bit in SPI_STAT is set, it will be a DMA request on corresponding DMA channel.

30.4.3. Status register (SPI_STAT)

Address offset: 0x08

Reset value: 0x0000 0002

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			TXLVL[1:0]		RXLVL[1:0]		FERR	TRANS	RXORERR	CONFERR	CRCERR	Reserved		TBE	RBNE
			r		r		rc_w0	r	r	r	rc_w0			r	r

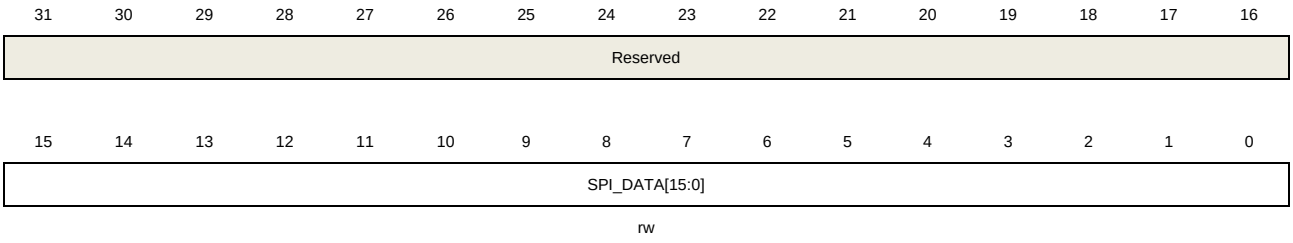
Bits	Fields	Descriptions
31:13	Reserved	Must be kept at reset value.
12:11	TXLVL[1:0]	TXFIFO level 00: Empty 01: 1/4 full 10: 1/2 full 11: Full **Note:** The FIFO level here refers to the current actual storage of the FIFO. Here, the FIFO is considered full when the FIFO level is greater than 1/2.
10:9	RXLVL[1:0]	RXFIFO level 00: Empty 01: 1/4 full 10: 1/2 full 11: Full This field has no meaning when SPI is in receive-only mode with CRC function enabled. **Note:** The FIFO level here refers to the current actual storage of the FIFO. Here, the FIFO is considered full when the FIFO level is greater than 1/2.
8	FERR	Format error SPI TI mode: 0: No TI mode format error 1: TI mode format error occurs. This bit is set by hardware and is able to be cleared by writing 0.
7	TRANS	Transmitting ongoing bit 0: SPI is idle. 1: SPI is currently transmitting and/or receiving a frame. This bit is set and cleared by hardware.

6	RXORERR	Reception overrun error bit 0: No reception overrun error occurs. 1: Reception overrun error occurs. This bit is set by hardware and cleared by a read operation on the SPI_DATA register followed by a read access to the SPI_STAT register.
5	CONFERR	SPI configuration error 0: No configuration fault occurs. 1: Configuration fault occurred. (In master mode, the NSS pin is pulled low in NSS hardware mode or SWNSS bit is low in NSS software mode.) This bit is set by hardware and cleared by a read or write operation on the SPI_STAT register followed by a write access to the SPI_CTL0 register.
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.
3:2	Reserved	Must be kept at reset value.
1	TBE	Transmit buffer / TXFIFO empty 0: Transmit buffer / TXFIFO is not empty. 1: Transmit buffer / TXFIFO is empty.
0	RBNE	Receive buffer / RXFIFO not empty 0: Receive buffer / RXFIFO is empty. 1: Receive buffer / RXFIFO is not empty.

30.4.4. Data register (SPI_DATA)

Address offset: 0x0C
Reset value: 0x0000 0000

For SPI, this register can be accessed by byte (8-bit) or half-word (16-bit).



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	SPI_DATA[15:0]	Data transfer register. For SPI, the hardware has two FIFOs, including TXFIFO and RXFIFO. The SPI_DATA register serves as an interface between the Rx and Tx FIFOs. Write data

to SPI_DATA will save the data to TXFIFO and read data from SPI_DATA will get the data from RXFIFO.

Note: In fact, SPI hardware determines the size of each access to SPI_DATA only based on the BYTEN bit in SPI_CTL1, regardless of the size of the software's current operation.

30.4.5. CRC polynomial register (SPI_CRCPOLY)

Address offset: 0x10

Reset value: 0x0000 0007

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

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

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

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.

30.4.6. Receive CRC register (SPI_RCRC)

Address offset: 0x14

Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	RCRC[15:0]	RX CRC value When the CRCEN bit of SPI_CTL0 is set, the hardware computes the CRC value of the received bytes and saves them in RCRC register. For SPI, CRC function is

valid only when the data length is 8 bits or 16 bits. And if the CRC length is set to 8-bit and the data size is equal to 8-bit, the CRC calculation is based on CRC8 standard, and saves the value in RCRC [7:0]. In addition to this, the calculation is based on CRC16 standard, and saves the value in RCRC [15:0].

The hardware computes the CRC value after each received bit, when the TRANS is set, a read to this register could return an intermediate value.

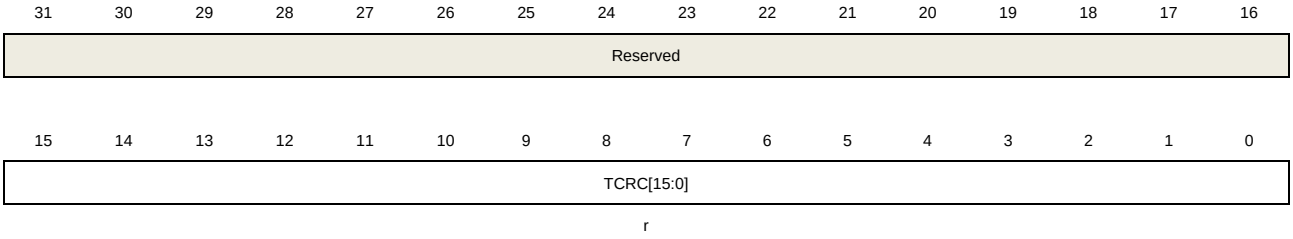
This register is reset when the CRCEN bit in SPI_CTL0 register or the SPIxRST bit in RCU reset register is set.

30.4.7. Transmit CRC register (SPI_TCRC)

Address offset: 0x18

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	TCRC[15:0]	TX CRC value When the CRCEN bit of SPI_CTL0 is set, the hardware computes the CRC value of the transmitted bytes and saves them in TCRC register. For SPI, CRC function is valid only when the data length is 8 bits or 16 bits. And if the CRC length is set to 8-bit and the data size is equal to 8-bit, the CRC calculation is based on CRC8 standard, and saves the value in TCRC[7:0]. In addition to this, the calculation is based on CRC16 standard, and saves the value in TCRC[15:0]. The hardware computes the CRC value after each transmitted bit, when the TRANS is set, a read to this register could return an intermediate value. The different frame formats (LF bit of the SPI_CTL0) will get different CRC values. This register is reset when the CRCEN bit in SPI_CTL0 register or the SPIxRST bit in RCU reset register is set.

30.4.8. Quad-SPI mode control register (SPI_QCTL) of SPI0

Address offset: 0x80

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	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													IO23_DRV	QRD	QMOD
													rw	rw	rw

Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value.
2	IO23_DRV	Drive IO2 and IO3 enable 0: IO2 and IO3 are not driven in single wire mode. 1: IO2 and IO3 are driven to high in single wire mode. This bit is only available in SPI0.
1	QRD	Quad-SPI mode read select. 0: SPI is in quad wire write mode. 1: SPI is in quad wire read mode. This bit should be only be configured when SPI is not busy (TRANS bit cleared) This bit is only available in SPI0.
0	QMOD	Quad-SPI mode enable. 0: SPI is in single wire mode. 1: SPI is in Quad-SPI mode. This bit should only be configured when SPI is not busy (TRANS bit cleared). This bit is only available in SPI0.

31. Quad-SPI interface (QSPI)

31.1. Overview

The QSPI is a specialized interface that can communicate with flash memories. This interface supports single, dual or quad SPI Flash. It can operate in normal mode, read polling mode and memory map mode.

31.2. Characteristics

- Three functional modes: normal mode (address extend), read polling mode and memory map mode.
- Fully programmable command format for both normal mode and memory map mode.
- Integrated FIFO for transmission/reception.
- Support SDR and DDR mode.
- Support DQS signal.
- 8, 16, or 32-bit data accesses.
- DMA channel for normal mode.

31.3. Function overview

31.3.1. QSPI block diagram

7 signals are used to interface with an external flash memory. The description is shown in *Table 31-1. QSPI signal description*.

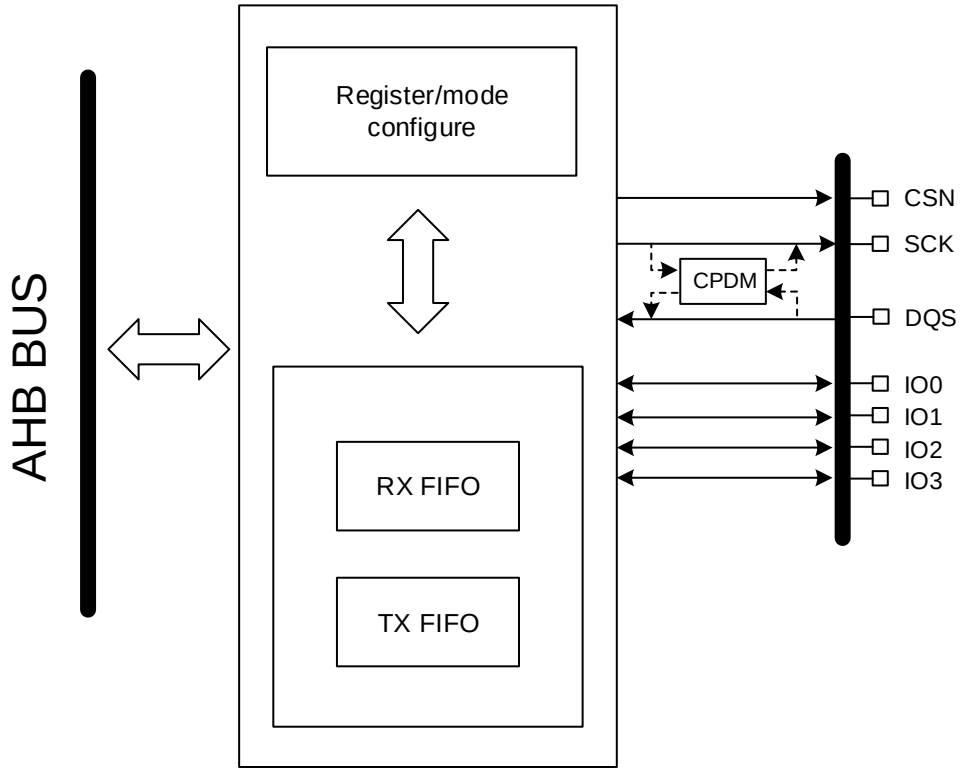
Table 31-1. QSPI signal description

Pin name	Direction	Description
CSN	O	chip select output (active low)
SCK	O	clock output
IO0/SO	I/O	single mode:data output dual mode: data intput or output quad mode: data intput or output
IO1/SI	I/O	single mode:data input dual mode: data intput or output quad mode: data intput or output
IO2	I/O	single mode: connect WP pin of flash, control “write protect” function dual mode: connect WP pin of flash, control “write protect” function quad mode: data intput or output

Pin name	Direction	Description
IO3	I/O	single mode: connect HOLD pin of flash, control “hold” function dual mode: connect HOLD pin of flash, control “hold” function quad mode: data intput or output
DQS	I	data strobe signal

*Figure 31-1 QSPI diagram*shows the block diagram of the QSPI unit.

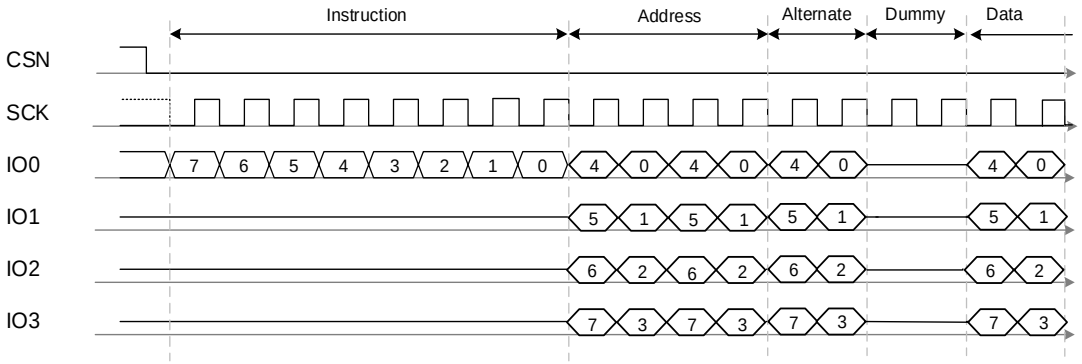
Figure 31-1 QSPI diagram



31.3.2. QSPI command format

The QSPI communicates with the flash memory using commands in various formats. There are totally 5 phases which can be included or not: instruction, address, alternate byte, dummy and data. Any phase can be skipped, but it must contain at least one of the instruction phase, address phase, alternate byte phase, and data phase. This is guaranteed by the software, and the hardware design has no protection method. The sequence diagram of the QSPI command format is shown in *Figure 31-2 QSPI command format*.

Figure 31-2 QSPI command format



The commands and the corresponding configuration are shown in ***Table 31-2. QSPI command description***.

Table 31-2. QSPI command description

Command	Send information	Configuration	Note
instruction	8-bit instruction	QSPI_TCFG register defines the instruction and signal line mode	-
address	1-4 bytes of address	QSPI_ADDR register defines the information of address. QSPI_TCFG register defines the number of address and signal line mode	-
Alternate byte	1-4 alternate-bytes	QSPI_ALTE register defines the information of alternate bytes, QSPI_TCFG register defines the number of alternate-bytes and signal line mode	-
dummy	0-31 cycles	QSPI_TCFG register defines the cycles and DATAMOD field (QSPI_TCFG register) defines the dummy signal line mode	given without any data being transferred for external flash, in order to wait flash prepare data
data	any number of bytes	In normal mode, QSPI_DTLEN register defines the number of bytes. DATAMOD filed (QSPI_TCFG register) defines the data signal line mode, and the configuration of DATAMOD = 00 must only be used in normal write mode	In memory map mode, the number of bytes to be transmitted is specified as single AHB bus access operation, these could be 8, 16 or 32 read/write access, corresponding to 1, 2, or 4 bytes.

Note: Signal line mode can be no command, single line mode, dual line mode, or quad line mode.

31.3.3. QSPI signal line modes

Each of the instruction, address, alternate-byte, or data phase can be configured separately into signal line modes by setting IMOD / ADDRMOD / ALTEMOD / DATAMOD.

Table 31-3. QSPI singnal line modes

Signal line modes		Single line mode	Dual line mode	Quad line mode
Configure filed	IMOD	01 or 00	10 or 00	11 or 00
	ADDRMOD			
	ALTEMOD			
	DATAMOD			
Pins	IO0 (SO)	Output	Input: data read (high impedance) output: all other phases	Input: data read (high impedance) Output: all other phases.
	IO1 (SI)	Input (high inpedance)		
	IO2	Output 0 (deactivate “write protect function”)		
	IO3	Output 1 (deactivate “hold” function)		
Description		In dummy phase when DATAMOD = 01, IO0 output, IO1 input (high inpedance)	In dummy phase when DATAMOD = 10, IO0 / IO1 are always high-impedance.	In dummy phase when DATAMOD = 11, IO0 / IO1 / IO2 / IO3 are always high-impedance.

IO2 / IO3 are used only in quad mode, and if none of the 5 phases is configured in quad mode, IO2 / IO3 will be released and can be used for other functions even when QSPI is enabled.

QSPI pins

QSPI supports a maximum clock of 200M, but the highest communication clock frequency supported by different groups of pins varies. Please refer to ***Table 31-4. Correspondence between QSPI pins used and the highest supported communication clock***.

Table 31-4. Correspondence between QSPI pins used and the highest supported communication clock

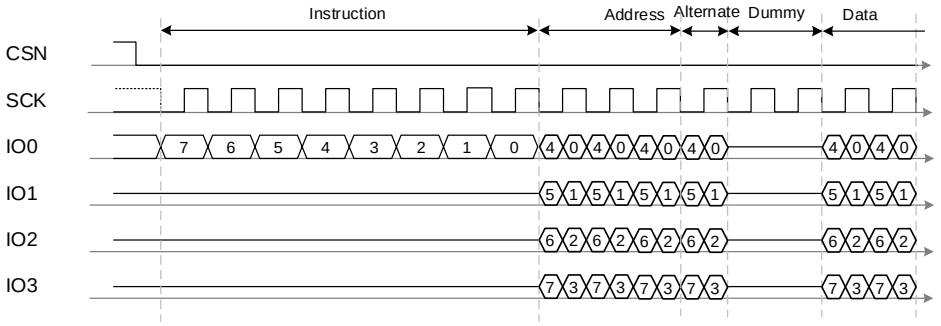
Pin function	Max clock 120M	Max clock 120M	Max clock 200M	Max clock 120M
QSPI_CSN	PA2	PB11\PE11	PD3	PA2
QSPI_SCK	PA3	PB10\PE10	PD2	PF10
QSPI_IO0	PB1	PE12	PD4	PC1\PF8
QSPI_IO1	PB0\PB2	PE13	PD5	PC2\PF9
QSPI_IO2	PA7	PE14	PD6	PC3\PF7
QSPI_IO3	PA6\PC4	PE15	PD7	PF6
QSPI_DQS	PB7	PB7	PD1	PB7

Note: It is strongly recommended to use pins in groups, otherwise, the maximum communication clock frequency supported by QSPI cannot be guaranteed.

31.3.4. QSPI DDR mode

By default, QSPI operates in SDR (single data rate) mode, when the DDREN bit in QSPI_TCFG register is set to 1, the QSPI operates in DDR (double data rate) mode. In DDR mode, during the address / alternate-byte / data phases, IO0, IO1, IO2 and IO3 signals all transmit data along the two clock edges (rising and falling edges) of SCK signal. During the instruction phase, SDR mode is still used for transmission, and IO0, IO1, IO2 and IO3 signals are sampled along the falling edge of SCK.

Figure 31-3 QSPI DDR mode

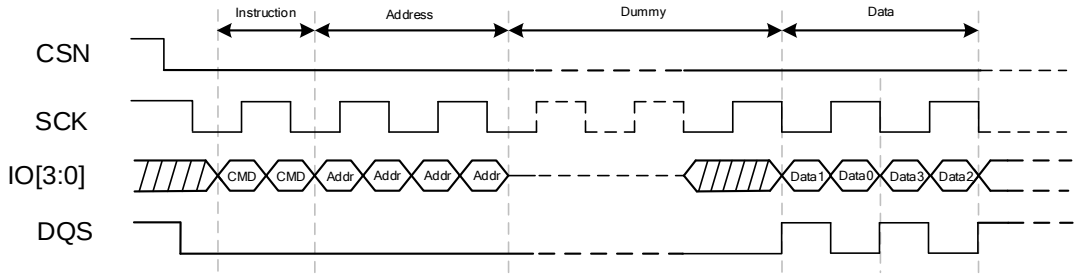


In DDR mode, according to actual usage requirements, user need to use CPDM to fine-tune the QSPI receive clock phase. For CPDM usage, please refer to ***Clock phase delay module (CPDM)***.

31.3.5. 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 DDR read operations, the data should be latched on both rising edge and falling edge of the DQS signal.

Figure 31-4 DDR read timing using DQS



31.3.6. CSN and SCK

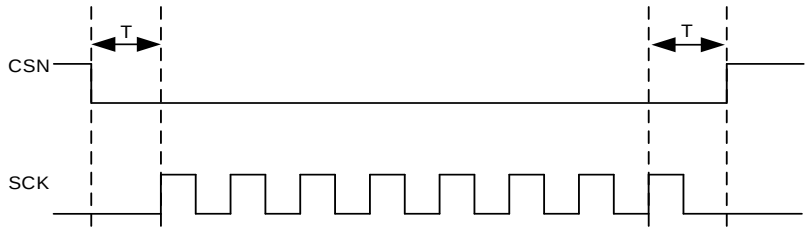
The default value of CSN is high, and it falls before a command begins and rises as soon as

it finishes.

SCK output signal is a gate signal from internal sck, where the internal sck is present all the time.

To accommodate some high-speed devices, the QSPI supports selecting the CSN falls one or two SCK cycle before the first valid rising SCK edge, and rises on one or two SCK cycle after the final valid rising SCK edge by configuring the CSN SCKSEL bit in the QSPI DCFG register.

Figure 31-5 Behavior of CSN and SCK (CSNCKM = 0)



When the FIFO stays empty in a write command phase, or full in a read command phase, SCK will be stalled and keep low until the FIFO can work again. At this moment if the CSN is high, SCK will rises back up one half of a SCK cycle after the rising edge of the CSN.

31.4. Operating modes

The QSPI can operate in normal mode, read polling mode and memory map mode. In normal mode, all operations are performed depends on QSPI registers. In read polling mode, the values of status registers in external flash memory are periodically read and checked. In memory map mode, the external flash memory is mapped to the microcontroller address space (range from 0x9000 0000 to 0x9FFF FFFF) and can be accessed as an internal memory.

31.4.1. Normal mode

The write operation in normal mode is selected by configuring the FMOD[1:0] in QSPI_TCFG register to “00”. The data to be transmitted is written into QSPI_DATA. The read operation in normal mode is selected by setting the FMOD[1:0] in QSPI_TCFG register to “01”, and the data to be received is read from QSPI_DATA.

The DTLEN[31:0] in QSPI_DTLEN register defines the number of bytes to be transferred. If DTLEN is 0xFFFF FFFF, the number of bytes to be transferred is considered undefined, and the transmission continues until the memory size boundary is reached as specified by FMSZ[4:0] in QSPI_DCFG register. If DTLEN is 0xFFFF FFFF and FMSZ[4:0] is configured as 0x1F, the flash memory capacity is 4GB, the transmission will continue indefinitely until a request of abort is occurred or the QSPI is disabled.

The transfer complete flag TC will be set when the number of bytes programmed in DTLEN has been transferred. If the DTLEN is 0xFFFF FFFF, TC will be set when the transmitted/received byte number equals to the external memory size defined by FMSZ[4:0]. An interrupt is generated if TCIE and TC are both set. The TC bit is cleared by writing 1 to the TCC bit in QSPI_STATC register.

Initialize a command sequence

The command sequence starts immediately after the last information is provided by software according to communication requirement.

When neither address nor data are required, the sequence starts immediately after QSPI_TCFG has been accessed.

When address is required and no data is required, the sequence starts after QSPI_ADDR has been accessed.

When both address and data are required in normal write mode, the command sequence starts after QSPI_DATA has been accessed.

FIFO

A FIFO 16 bytes is implemented to transfer data. In normal write mode, the relationship between the AHB write access mode and the number of bytes add to FIFO is shown in ***Table 31-5. AHB write access mode and number of bytes add to FIFO***.

Table 31-5. AHB write access mode and number of bytes add to FIFO

AHB write access mode	Number of bytes add to FIFO
32-bit	4 bytes
16-bit	2 bytes
8-bit	1 byte

Note: When the AHB write access mode is 8-bit or 16-bit, the least significant bytes are valid in QSPI_DATA register.

FIFO threshold is defined by FTL[3:0] in QSPI_CTL register, in normal read mode, when the amount of bytes in the FIFO is equal or above the defined threshold, FIFO threshold flag FT in QSPI_STAT register will be set. FT is also set after data phase is completed if FIFO is not empty. In normal write mode, when the amount of the empty bytes in the FIFO is above the threshold, FT will be 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 normal read mode, when the FIFO becomes full, the QSPI temporarily stop SCK clock to avoid overrun. The reading sequence is not resumed until more than 4 bytes are available in FIFO.

31.4.2. Read polling mode

The read polling mode can be selected by configuring the FMOD[1:0] to “10”. In read polling mode, the QSPI periodically starts a read command with up to 4-bytes data. The received data can be bit-wise masked and compared with a defined data content. If a match happens, an interrupt will be generated if RPMFIE is set.

The read polling access sequence is started the same as it of normal read mode. The BUSY bit keeps 1 between periodic intervals.

Polling match mode configured by RPMM controls the comparison match mode.If RPMM = 0, the AND mode is selected. In this mode, status match flag RPMF will be set only when there is a match on all the unmasked bits. While if RPMM = 1, the OR mode is selected. In this mode, RPMF will be set if there is a match on any of the unmasked bit.

In read polling mode, if RPMS is set, read polling sequence will stop when a match is detected, and the BUSY flag is cleared at the end of data phase. Otherwise, the periodic sequence always continues until ABORT bit is set or the QSPI is disabled.

In read polling mode, FIFO is bypassed and the status bytes read are stored in QSPI_DATA, and the status bytes stored are not affected by the MASK control field. Contents in QSPI_DATA is renewed at the beginning of data phase if there is any.

FT is set at the end of data phase, where the external flash memory status bytes are considered read, and it is cleared when QSPI_DATA is read.

31.4.3. Memory map mode

The memory map mode can be selected by configuring the FMOD[1:0] to “11”. In memory map 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 can not access the address no more than 256MB but outside the range which is specified by FMSZ. Or else, an error will be generated. If the AHB master is the CPU, a hard fault interrupt will be generated. If the AHB master is the DMA, a transfer error will be generated, and the corresponding DMA channel is disabled by hardware.

In this mode, byte, half-word, and word single or burst access are supported.

The memory map mode supports prefetch function on sequential address accessing. The QSPI load the data at the following address before accessing it, if the next access is indeed at the next address, the access with be faster since the data is already prefetched. Otherwise, the read sequence is restarted, pulling CSN low before the read sequence starts.

When the FIFO is full, the output of SCK will be stopped, and the CSN is low during this period. If the TMOUTEN bit in QSPI_CTL register is set, CSN will be pulled high when the duration of the low period reaches the number of SCK clock cycles specified in QSPI_TMOUT register.

At the beginning of a transfer, BUSY goes high before CSN falls, and is cleared when a

timeout occurred or abort / disable is issued.

31.5. QSPI configuration

31.5.1. Flash configuration

The configuration in QSPI_DCFG register can be used to specify the characteristics of the external flash memory, so that the QSPI interface can work consistently.

The size of the external memory is defined by FMSZ[4:0] in QSPI_DCFG register. FMSZ + 1 is the number of address bits in the flash memory. The maximum of the flash capacity can be up to 4GB in normal mode.

CSHC[2:0] defines the chip select high time, it specifies the minimum number of SCK cycles that CSN must stay high between two command sequences.

31.5.2. IP configuration

The configuration in QSPI_CTL register can be used to specify the characteristics of the QSPI IP.

PSC[7:0] in QSPI_CTL register indicates the clock prescaler division factor.

SSAMPLE is used to define which SCK edge is used to sample data. By default, the QSPI samples data one half of a SCK cycle after the external flash drives. However, it may be beneficial to sample data later because of the external signal delays. The sample edge can be shifted half one of SCK cycle using SSAMPLE bit. In addition, with the RXSFT[2:0] bit in QSPI_DCFG register, the sampling edge can be modified for more than half a cycle

The DMA request is enabled by setting the DMAEN bit in QSPI_STAT register. The FIFO threshold level is configured in FTL[3:0] bits in QSPI_CTL register.

31.6. Send instruction only once

Sending instruction only once mode is enabled by setting the SIOO bit in QSPI_TCFG register, this function is valid for all functional modes. If SIOO bit is set, the instruction is sent only once after QSPI_TCFG has been accessed. The subsequent command sequence will skip the instruction phase before the next configuration of QSPI_TCFG register.

31.7. Error and interrupts

TERR and AHB error will be generated if one of the conditions occurs in ***Table 31-6. TERR and AHB error conditions***.

Table 31-6. TERR and AHB error conditions

Error name	Condition
TERR	1. In normal mode or read polling mode, a wrong address has been programed in QSPI_ADDR register according to the address defined by FMSZ. 2. In normal mode, the address (ADDR) plus data length (DTLEN) is greater than external memory size.
AHB error	1. In memory map mode, out of range access is done by AHB master, or when the QSPI is disabled. 2. AHB master is accessing the memory mapped space while the memory map mode is not enabled. 3. The read polling mode don't match until the timeout counter reaches 0.

The QSPI interrupt event and flags are listed in ***Table 31-7. QSPI interrupt events***.

Table 31-7. QSPI interrupt events

Interrupt event	Event Flag	Interrupt enable bit	Clear method
FIFO threshold interrupt	FT	FTIE	By hardware
Transfer complete interrupt	TC	TCIE	Set TCC bit in QSPI_STATC register
Transfer error interrupt	TERR	TERRIE	Set TERRC bit in QSPI_STATC register
Timeout interrupt	TMOUT	TMOUTIE	Set TMOUTC bit in QSPI_STATC register
Status match interrupt	RPMF	RPMFIE	Set RPMFC bit in QSPI_STATC register

31.8. QSPI register definition

QSPI base address: 0xA000 1000

31.8.1. Control register (QSPI_CTL)

Address offset: 0x00

Reset value: 0x0000 0010

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PSC[7:0]								RPMM	RPMS	Reserved	TMOUTIE	RPMFIE	FTIE	TCIE	TERRIE
rw								rw	rw		rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OCKDV[3:0]				FTL[3:0]				Reserved		OCKDEN	SSAMPLE	TMOUTEN	DMAEN	ABORT	QSPIEN
rw				rw						rw	rw	rw	rw	w1s	rw

Bits	Fields	Descriptions
31:24	PSC[7:0]	The clock prescaler of AHB clock for generating SCK. The relationship between the frequency of SCK and the AHB is $f_{SCK}=f_{AHB}/(PSC+1)$. 00000000: $f_{SCK}=f_{AHB}$ 00000001: $f_{SCK}=f_{AHB}/2$ 00000010: $f_{SCK}=f_{AHB}/3$ ... 11111111: $f_{SCK}=f_{AHB}/256$ When the clock division factor is odd, the duty cycle of SCK is not 50%. And the low level is one cycle longer than the high level. These bits can be modified only when BUSY = 0.
23	RPMM	Read polling match mode 0: AND match mode. If all the unmasked bits received from the flash memory match with the corresponding bits in QSPI_STATMATCH register, RPMF will be set. 1: OR match mode. If any one of the unmasked bits received from the flash memory matches its corresponding bit in the QSPI_STATMATCH register, RPMF will be set. This bit can be modified only when BUSY = 0.
22	RPMS	Read polling mode stop This bit determines if read polling is stopped after a match. 0: Read polling mode stop when ABORT bit is set or the QSPI is disabled. 1: Read polling mode stop when a match happens. This bit can be modified only when BUSY = 0.
21	Reserved	Must be kept at reset value.

20	TMOUTIE	Timeout interrupt enable 0: Disable timeout interrupt 1: Enable timeout interrupt
19	RPMFIE	Read polling mode match interrupt enable 0: Disable read polling mode match interrupt 1: Enable read polling mode 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
15:12	OCKDV[3:0]	Output clock delay value. These bits only useful when OCKDEN is enabled. The output clock delay function is only effective when the QSPI clock is not divided.
11:8	FTL[3:0]	FIFO threshold level These bits are useful in normal mode, the threshold number of bytes in the FIFO that will cause the FIFO threshold flag to be set. In normal write mode (FMOD = 00): 0000: FT is set if there are 1 or more free bytes available to be written to in the FIFO 0001: FT is set if there are 2 or more free bytes available to be written to in the FIFO ... 1111: FT is set if there are 16 free bytes available to be written to in the FIFO In normal read mode (FMOD = 01): 0000: FT is set if there are 1 or more valid bytes that can be read from the FIFO 0001: FT is set if there are 2 or more valid bytes that can be read from the FIFO ... 1111: FT is set if there are 16 valid bytes that can be read from the FIFO If DMAEN = 1, then the DMA controller for the corresponding channel must be disabled before changing the FTL value.
7:6	Reserved	Must be kept at reset value.
5	OCKDEN	Output clock delay enable when write data to flash. 0: Disable 1: Enable
4	SSAMPLE	Sample delay QSPI samples data 1/2 SCK clock cycle after flash memory drives data by default.

		Taking the delay of external signals into account, thes bits can be configured to allow the data to be sampled later. 0: No delay 1: 1/2 cycle delay This bit can be modified only when BUSY = 0.
3	TMOUTEN	Timeout counter enable In memory map mode (FMOD = 11). When setting this bit, the chip select output (CSN) will be high if there is no access after a certain amount of time which specified by TMOUTCYC[15:0]. 0: Disable the timeout counter, the chip select (CSN) will remain low after an access in memory map mode. 1: Enable the timeout counter, the chip select (CSN) will be high if there is no access to flash memory after TMOUTCYC[15:0] cycles in memory map mode. This bit can be modified only when BUSY = 0.
2	DMAEN	DMA enable DMA can be used to transfer data in normal mode. And when FT is set, DMA transfers initiates. 0: Disable DMA 1: Enable DMA
1	ABORT	Abort request This bit is used to stop the current command. It is automatically cleared once the abort is completed. In read polling mode or memory map mode, when setting this bit, the RPMS bit or the DMAEN bit will be cleared. 0: No abort request 1: Abort request
0	QSPIEN	QSPI enable bit 0: Disable QSPI 1: Enable QSPI

31.8.2. Device configuration register (QSPI_DCFG)

Address offset: 0x04

Reset value: 0x001F 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	CSNCKM	Reserved									FMSZ[4:0]				
	rw												rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					CSHC[2:0]			Reserved		DLYSCEN	RCKSEL	RXSFT[2:0]			CKMOD

rw rw rw rw rw

Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30	CSNCKM	Select whether the CSN falls and rises one or two SCK clock cycles before the first SCK effective rising edge and after the last SCK effective rising edge. 0: 1 cycle 1: 2 cycles
29:21	Reserved	Must be kept at reset value.
20:16	FMSZ[4:0]	Flash memory size These bits defines the size of external memory. And the number of bytes is $2^{[FMSZ+1]}$. FMSZ+1 is the number of address bits in the flash memory. In normal mode, the flash memory capacity can be up to 4GB, while it is limited to 256MB in memory map mode. These bits can be modified only when BUSY = 0.
15:11	Reserved	Must be kept at reset value.
10:8	CSHC[2:0]	Chip select high cycle The chip select (CSN) must stay high for at least CSHC+1 SCK cycles between two command sequences. 000: CSN must stay high for at least 1 SCK cycle between two flash memory command sequences. 001: CSN must stay high for at least 2 SCK cycles between two flash memory command sequences. ... 111 CSN must stay high for at least 8 SCK cycles between two flash memory command sequences. These bits can be modified only when BUSY = 0.
7:6	Reserved	Must be kept at reset value.
5	DLYSCEN	Delay scan enable 0: Disable 1: Enable **Note**: This bit only takes effect when RCKSEL=0. This bit enables the SCK phase fine-tuning function, which is implemented through the CPDM module. Implementation. After the CPDM clock adjustment is completed, DLYSCEN needs to be set to 0 again.
4	RCKSEL	Receive clock select 0: SCK 1: DQS Select the receiving clock source is the SCK generated inside the QSPI or the DQS

		provided by an external device. When the DQS clock is selected, CPDM can also be used for clock adjustment, and there is no need to set DLYSCEN bit.
3:1	RXSFT[2:0]	Shift receive step if rececive data delay more than 0.5 cycle. 000: No shift 001: 1 cycle 010: 2 cycles 011: 3 cycles 100: 4 cycles 101: 5 cycles 110: 6 cycles 111: 7 cycles This bit field can be used to adjust the receiving sampling point together with the SSAMPLE bit.
0	CKMOD	This bit indicates the SCK level when QSPI is free. 0: SCK must stay low when CSN is high. 1: SCK must stay high when CSN is high. This bit can be modified only when BUSY = 0.

31.8.3. Status register (QSPI_STAT)

Address offset: 0x08

Reset value: 0x0000 0004

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

31	30	29	28	27	26	25	24	23	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			FL[4:0]					Reserved		BUSY	TMOUT	RPMF	FT	TC	TERR
			rw							r	r	r	r	r	r

Bits	Fields	Descriptions
31:13	Reserved	Must be kept at reset value.
12:8	FL[4:0]	FIFO level These bits is used to configure the number of valid bytes which are being stored in the FIFO in normal mode. In memory map mode and in read polling mode, FL is 0.
7:6	Reserved	Must be kept at reset value.
5	BUSY	Busy flag This bit is set when a command is transferring. In normal mode, this bit will be cleared once the command phase is completed and if in read opreation of normal mode, FIFO also needs be empty.

4	TMOUT	Timeout flag When the TMOUTEN is set and there is no access to flash memory after TMOUTCYC[15:0] cycles. This bit will be set.
3	RPMF	Read polling match flag This bit is set in read polling mode when the unmasked received data matches the expected value in QSPI_STATMATCH register.
2	FT	FIFO threshold flag In normal mode, when the FIFO threshold has been reachedor if the FIFO is not empty after the last read operation from the flash memory, this bit will be set. And it is cleared by hardware as soon as the threshold condition is not true. In read polling mode, this bit will be set. When the status register is read by the external flash, and it is cleared when the QSPI_DATA register is read.
1	TC	Transfer complete flag This bit will be set when the number of data programmed in QSPI_DTLEN register has been transmitted in normal mode or when abort operation is completed.
0	TERR	Transfer error flag In normal mode, when an invalid address has been accessed. This bit will be set

31.8.4. Status clear register (QSPI_STATC)

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											TMOUTC	RPMFC	Reserved	TCC	TERRC
											w	w		w	w

Bits	Fields	Descriptions
31:5	Reserved	Must be kept at reset value.
4	TMOUTC	Clear timeout flag Writing 1 to this bit clears the TMOUT bit in the QSPI_STAT register.
3	RPMFC	Clear read polling mode match flag Writing 1 to this bit clears the RPMF bit in the QSPI_STAT register.
2	Reserved	Must be kept at reset value.
1	TCC	Clear transfer complete flag

		Writing 1 to this bit clears the TC bit in the QSPI_STAT register.
0	TERRC	Clear transfer error flag Writing 1 to this bit clears the TERR bit in the QSPI_STAT register.

31.8.5. Data length register (QSPI_DTLEN)

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
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 These bits specify the number of data to be retrieved (value+1) in normal and read polling modes. When in read polling mode, the value of these bits should no greater than 3 (indicating 4 bytes). When these bits are configured as 0xFFFF FFFF in normal mode, it means undefined length, the QSPI will continue transferring data until the end of memoryaccording to FMSZ[4:0] in QSPI_DCFG register. 0x0000 0000: the data length to be transferred is 1 byte 0x0000 0001: the data length to be transferred is 2 bytes 0x0000 0002: the data length to be transferred is 3 bytes 0x0000 0003: the data length to be transferred is 4 bytes ... 0xFFFF FFFD: the data length to be transferred is 4,294,967,294 (4G-2) bytes 0xFFFF FFFE: the data length to be transferred is 4,294,967,295 (4G-1) bytes 0xFFFF FFFF: undefined length, all bytes defined by FMSZ[4:0] in QSPI_DCFG register until the end of flash memorywill be transferred. If FMSZ[4:0] is 0x1F, it will continue reading indefinitely. When in memory map mode, these bits have no effect. These bits can be modified only when BUSY = 0.

31.8.6. Transfer configuration register (QSPI_TCFG)

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
DDREN	DDRHEN	Reserved	SIOO	FMOD[1:0]		DATAMOD[1:0]		Reserved	DUMYC[4:0]					ALTESZ[1:0]	
rw	rw		rw	rw		rw			rw					rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ALTEMOD[1:0]		ADDRSZ[1:0]		ADDRMOD[1:0]		IMOD[1:0]		INSTRUCTION[7:0]							
rw		rw		rw		rw		rw							

Bits	Fields	Descriptions
31	DDREN	Double data rate mode enable 0: Disable 1: Enable
30	DDRHEN	DDR output hold enable This bit delays the data output by 1/4 of the QSPI output clock cycle in DDR mode. 0: Disable 1: Enable This function can be used when the QSPI clock is divided.
29	Reserved	Must be kept at reset value.
28	SIOO	Send instruction only once mode When IMOD is 00, this bit has no effect. 0: Send instruction on every command sequence 1: Send instruction only for the first command sequence This bit can be modified only when BUSY = 0.
27:26	FMOD[1:0]	Functional mode 00: Write operation in normal mode 01: Read operation in normal mode 10: Read polling mode 11: Memory map mode These bits can be modified when the DMA controller for the corresponding channel is disabled if DMAEN is 1. These bits can be modified only when BUSY = 0.
25:24	DATAMOD[1:0]	Data mode These bits define the data phase's mode of operation. It is also specify the dummy phase mode of operation. 00: No data 01: Data on a single line 10: Data on two lines 11: Data on four lines These bits can be modified only when BUSY = 0.
23	Reserved	Must be kept at reset value.
22:18	DUMYC[4:0]	Number of dummy cycles

		These bits define the duration of the dummy phase. These bits can be modified only when BUSY = 0.
17:16	ALTESZ[1:0]	Alternate bytes size 00: 8-bit alternate byte 01: 16-bit alternate bytes 10: 24-bit alternate bytes 11: 32-bit alternate bytes These bits can be modified only when BUSY = 0.
15:14	ALTEMOD[1:0]	Alternate bytes mode 00: No alternate bytes 01: Alternate bytes on a single line 10: Alternate bytes on two lines 11: Alternate bytes on four lines These bits can be modified only when BUSY = 0.
13:12	ADDRSZ[1:0]	Address size 00: 8-bit address 01: 16-bit address 10: 24-bit address 11: 32-bit address These bits can be modified only when BUSY = 0.
11:10	ADDRMOD[1:0]	Address mode 00: No address 01: Address on a single line 10: Address on two lines 11: Address on four lines These bits can be modified only when BUSY = 0.
9:8	IMOD[1:0]	Instruction mode 00: No instruction 01: Instruction on a single line 10: Instruction on two lines 11: Instruction on four lines These bits can be modified only when BUSY = 0.
7:0	INSTRUCTION[7:0]	Instruction Command information to be send to the flash memory. These bits can be modified only when BUSY = 0.

31.8.7. Address register (QSPI_ADDR)

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
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 to be send to the external Flash memory When BUSY=0 or in memory mapped mode, writing values to these bits will be ignored.

31.8.8. Alternate bytes register (QSPI_ALTE)

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
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 Optional data to be send to the flash memory follow by address. These bits can be modified only when BUSY = 0

31.8.9. Data register (QSPI_DATA)

Address offset: 0x20

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
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]	Interactive data between QSPI and flash memory. In write operation of normal mode, the data written to this register will be stored in the FIFO before sending to the flash memory. If the FIFO is full, a write operation will be stopped and wait until the FIFO has enough space. In read operation of normal mode, the data received from the flash memory can be read in this register. If the bytes in the FIFO are less than the requested bytes by the read command, and BUSY=1, the read operation will be stopped and wait until enough data is stored in the FIFO or until the transfer is completed. In read polling mode, this register contains the last data read from the flash memory.

31.8.10. Status mask register (QSPI_STATMK)

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
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 in read polling mode Mask of the status bytes received from the flash memory in read polling mode. For bit n of MASK[31:0]: 0: Bit n of the data received can not match 1: Bit n of the data received must match These bits can be modified only when BUSY = 0.

31.8.11. Status match register (QSPI_STATMATCH)

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

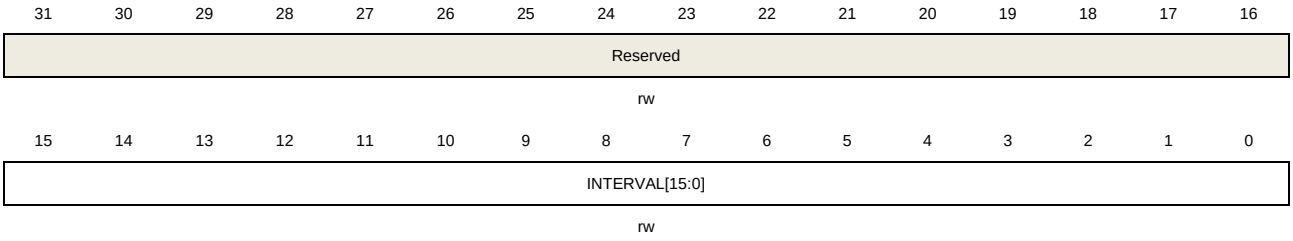
Bits	Fields	Descriptions
31:0	MATCH[31:0]	Status match in read polling mode Expected value to be compared with the value in the QSPI_STATMK register to get a match. These bits can be modified only when BUSY = 0.

31.8.12. Interval register (QSPI_INTERVAL)

Address offset: 0x2C

Reset value: 0x0000 0000

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



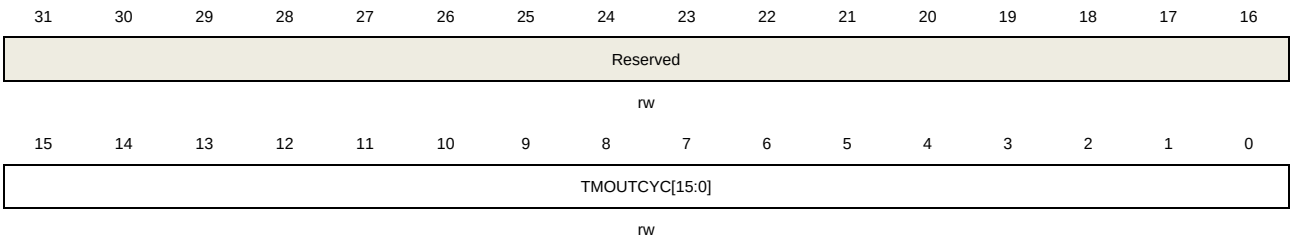
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	INTERVAL[15:0]	Interval cycle Number of SCK cycles between two read commands in read polling mode. These bits can be modified only when BUSY = 0.

31.8.13. Timeout register (QSPI_TMOUT)

Address offset: 0x30

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	TMOUTCYC[15:0]	Timeout cycle When the FIFO is full in memory map mode, these bits indicate how many SCK

cycles the QSPI waits for next access. In this duration, CSN keeps low.

These bits can be modified only when BUSY = 0.

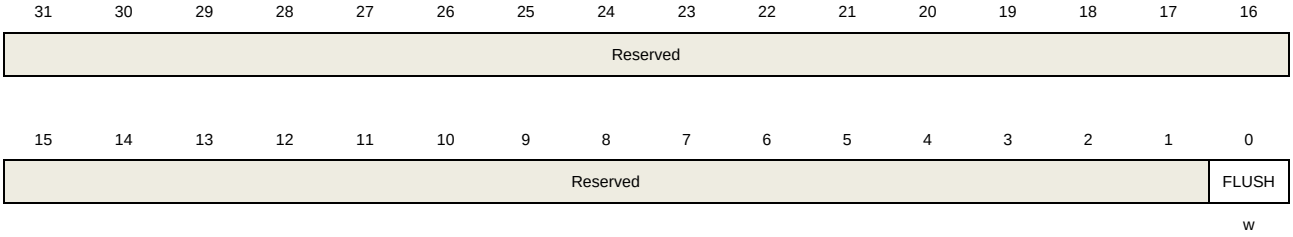
Note: This bit field must not be set to 0 if the timeout function is enabled.

31.8.14. FIFO flush register (QSPI_FLUSH)

Address offset: 0x34

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	FLUSH	Used to flush all qspi interal fifo.

32. Clock phase delay module (CPDM)

32.1. Overview

The Clock Phase Delay Module (CPDM) is used to delay the phase of the input clock and then output the clock. When used, the application needs to first program the phase of the output clock, and then use the output clock in other peripherals to receive data.

Phase delay is related to voltage and temperature and may require reconfiguration of the application and redetermination of the phase relationship between the output clock and the received data as parameters change.

32.2. Characteristics

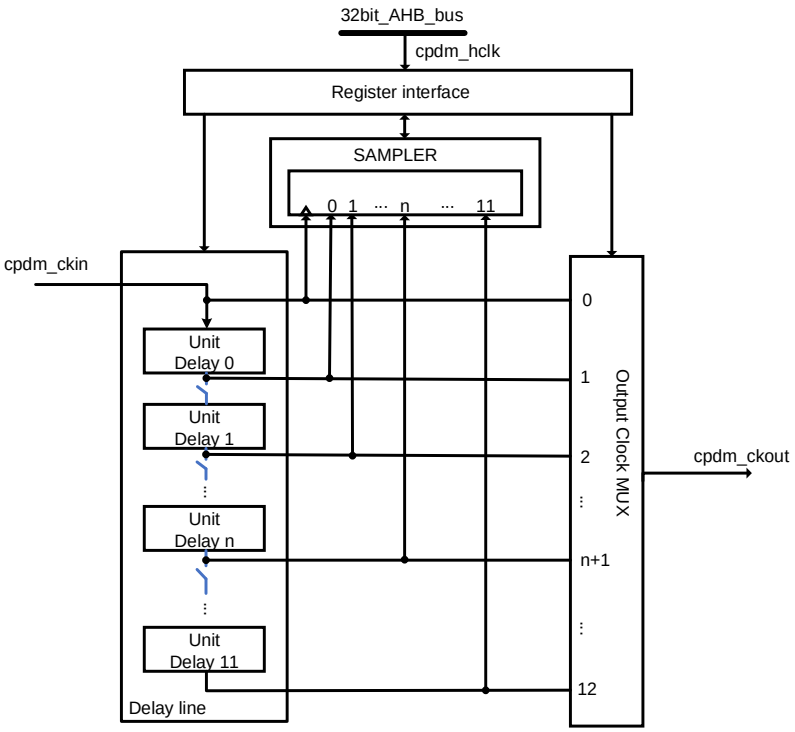
- Supports the input clock frequency ranges: 25 MHz ~ 208MHz.
- Supports up to 12 output clock phase selections.

32.3. Function overview

The Clock Phase Delay Module (CPDM) consists of register interface module, delay line module, delay line length sampling module and output clock selection multiplexer. The delay line module is used to generate unit delay.

Note:
CPDM is only used in QSPI, please refer to ***Figure 31-1 QSPI diagram***.

Figure 32-1. Schematic diagram of CPDM



Note:

cpdm_hclk: register interface clock of CPDM.
cpdm_ckin: the input clock of CPDM.
cpdm_ckout: the output clock of CPDM.

32.3.1. Overview

The Clock Phase Delay Module (CPDM) can be enabled by configuring the CPDMEN bit in the CPDM_CTL register. The delay line length sampling module can be enabled by configuring the DLSEN bit in the CPDM_CTL register.

Before shifting the phase of the input clock, the delay line length should be configured as one input clock period. When the CPDM is enabled and the delay line length sampling module is enabled (DLSEN = 1), define the number of delay steps required for a unit delay unit by configuring the DLSTCNT[11:0] bit field in the CPDM_CFG register. When the delay line length sampling module is enabled (DLSEN = 1), the length sampler can access the delay line length (DLLEN) and the length valid flag (DLLENF) in the CPDM_CFG register. Therefore, the delay line length can be configured by enabling the length sampler. When the delay line length sampling module is disabled (DLSEN = 0), the CPSEL bit in the CPDM_CFG register is used to select the output clock phase.

When the delay line length configuration is complete and the delay line length sampling module is disabled (DLSEN = 0), the clock that passes through the phase shift can be finally output through the clock phase selector.

- When CPDMEN = 0 and DLSEN = 0, CPDM output clock is enabled, and the output clock is equal to the input clock.
- When DLSEN = 1, the delay sampling module is enabled, and the CPDM output clock is disabled.
- When CPDMEN = 1 and DLSEN = 0, the CPDM output clock is enabled, and the phase of the output clock is determined by the configuration of CPSEL[3:0] bits in CPDM_CFG.

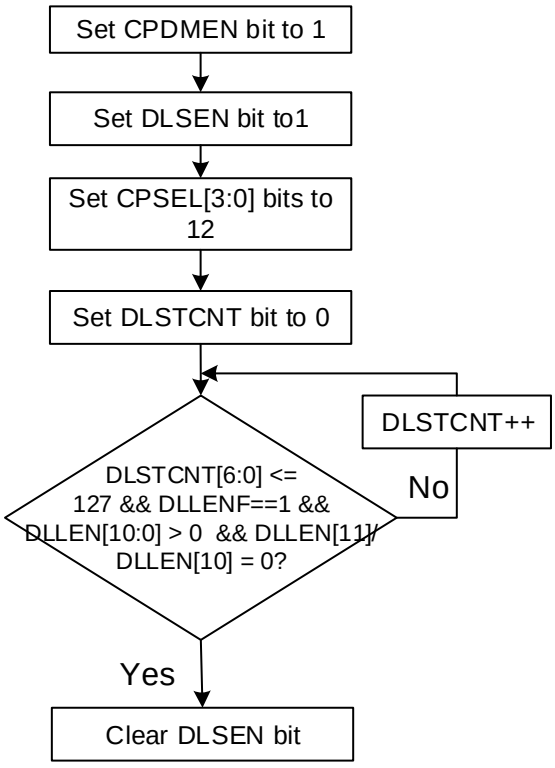
Note: The unit delay and output clock phase can only be changed when DLSEN is 1.

32.3.2. Operation process

Step one

The delay line length needs to be configured before configuring the output clock phase. The configured delay line length should cover a complete input clock cycle. ***Figure 32-2. CPDM delay line length configuration flowchart*** shows the configuration process.

Figure 32-2. CPDM delay line length configuration flowchart



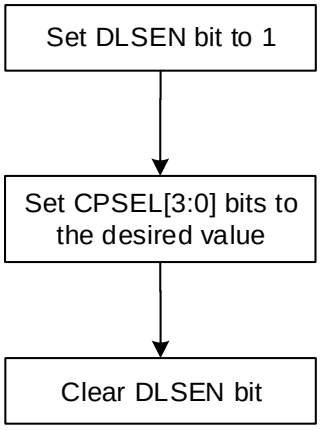
Note:

1. If DLLEN[10:0] > 0 and DLLEN[11] / DLLEN[10] = 0, configure the delay line length to an input clock cycle.
2. For N = 10 to 0, if DLLEN[N] = 1, the number of unit delays covering an input clock cycle is N.

Step two

The user can configure any output clock phase between unit delay N when the delay line length has been configured for one full input clock cycle. The specific method is as follows *Figure 32-3. CPDM output clock phase configuration flowchart*.

Figure 32-3. CPDM output clock phase configuration flowchart



32.4. Register definition

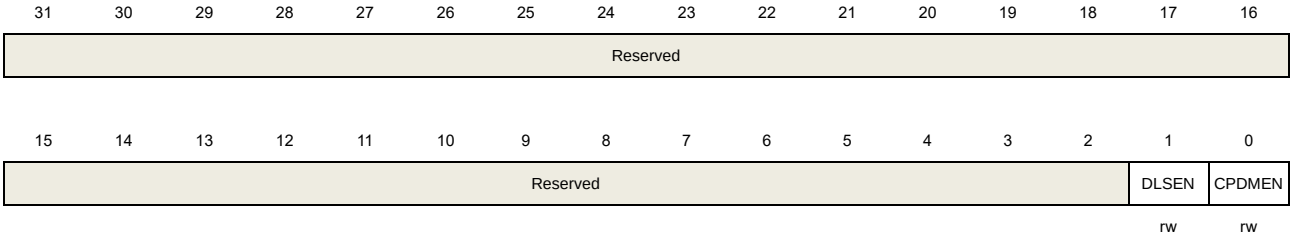
CPDM base address: 0x4802 2800

32.4.1. Control register (CPDM_CTL)

Address offset: 0x00

Reset value: 0x0000 0000

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



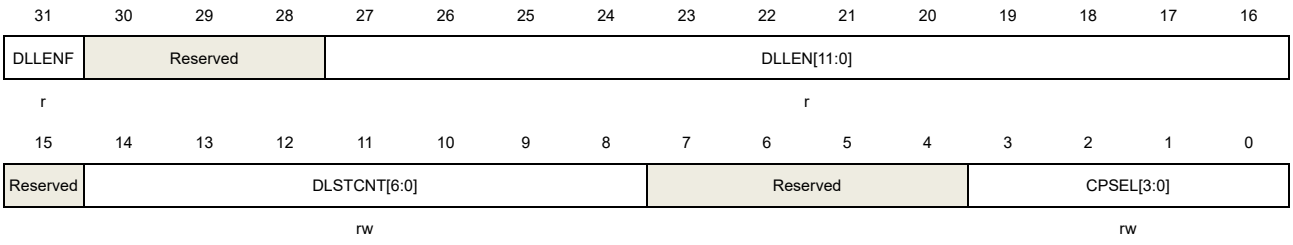
Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	DLSEN	CPDM delay line sample module enable bit 0: disable CPDM delay line sample module 1: enable CPDM delay line sample module
0	CPDMEN	CPDM enable bit 0: disable CPDM 1: enable CPDM

32.4.2. Configuration register (CPDM_CFG)

Address offset: 0x04

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31	DLLENF	Valid mark of delay line length 0: The length value in DLLEN[11:0] is invalid.

		1: The length value in DLLEN[11:0] is valid.
30:28	Reserved	Must be kept at reset value.
27:16	DLLEN[11:0]	Delay line length 12 unit delay values sampled on the rising edge of the input clock Only valid if DLLENF = 1.
15	Reserved	Must be kept at reset value.
14:8	DLSTCNT[6:0]	Defines a delay step count for a unit delay UNIT This bits can be written only when DLSEN = 1. 0000000: UNIT delay = Initial delay 0000001: UNIT delay = Initial delay + 1 * Delay step … 1111111: UNIT delay = Initial delay + 127 * Delay step
7:4	Reserved	Must be kept at reset value.
3:0	CPSEL[3:0]	Output clock phase selection These bits can be written only when DLSEN = 1. 0000: Output clock phase = input clock 0001: Output clock phase = input clock + 1 * UNIT delay .. 1100: Output clock phase = input clock + 12 * UNIT delay 1101 ~ 1111: Reserved

33. External memory controller (EXMC)

33.1. 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 SRAM, ROM, NOR Flash, PSRAM. Users can also adjust the timing parameters in the configuration registers to improve memory access efficiency. EXMC access space is divided into multiple banks; each bank is assigned to access a specific memory type with flexible parameter configuration as defined in the control registers.

33.2. Characteristics

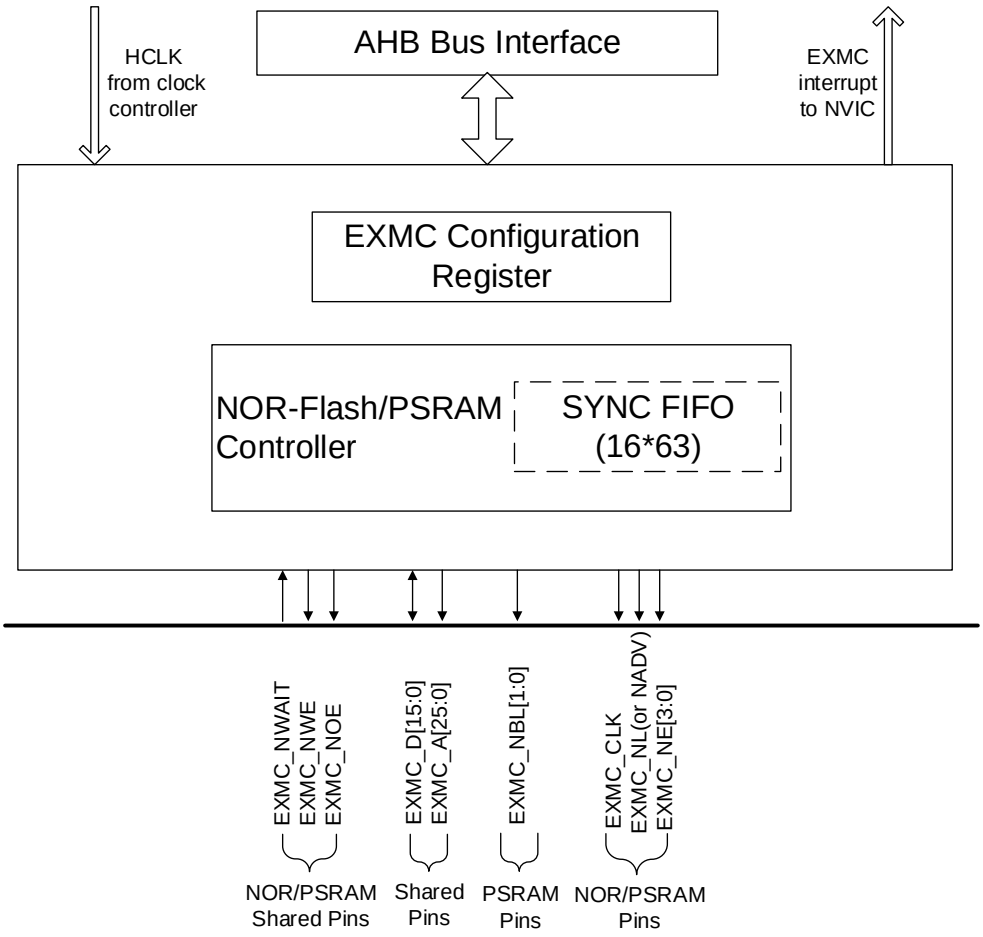
- Supported external memory:
  - SRAM
  - PSRAM
  - ROM
  - NOR Flash
- Protocol translation between the AMBA and the multitude of external memory protocol.
- Offering a variety of programmable timing parameters to meet user's specific needs.
- Each bank has its own chip-select signal which can be configured independently.
- Independent read/write timing configuration to a sub-set memory type.
- 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.
- Write FIFO with max 16 words data storage.

33.3. Function overview

33.3.1. Block diagram

EXMC is the combination of five modules: The AHB bus interface, EXMC configuration registers, NOR/PSRAM controller and external device interface. AHB clock (HCLK) is the reference clock.

Figure 33-1. The EXMC block diagram



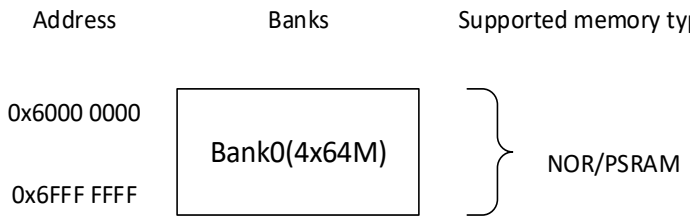
33.3.2. Basic regulation of EXMC access

EXMC is the conversion interface between AHB bus and external device protocol. 32-bit of AHB read/write accesses can be split into several consecutive 8-bit or 16-bit read/write operations respectively. In the process of data transfer, AHB access data width and memory data width may not be the same. In order to ensure consistency of data transmission, EXMC's read/write accesses follows the following basic regulation.

- When the width of AHB bus equals to the memory bus width, no conversion is applied.
- When the width of AHB bus is greater than memory bus width, the AHB accesses will automatically split into several continuous memory accesses.
- When the width of AHB bus is smaller than memory bus width, if the external memory devices have the byte selection function, such as 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.

33.3.3. External device address mapping

Figure 33-2. EXMC memory banks



EXMC access space is divided into multiple banks. Each bank is 256 Mbytes and Bank0 is valid. The first bank (Bank0) is further divided into four regions, and each region is 64 Mbytes.

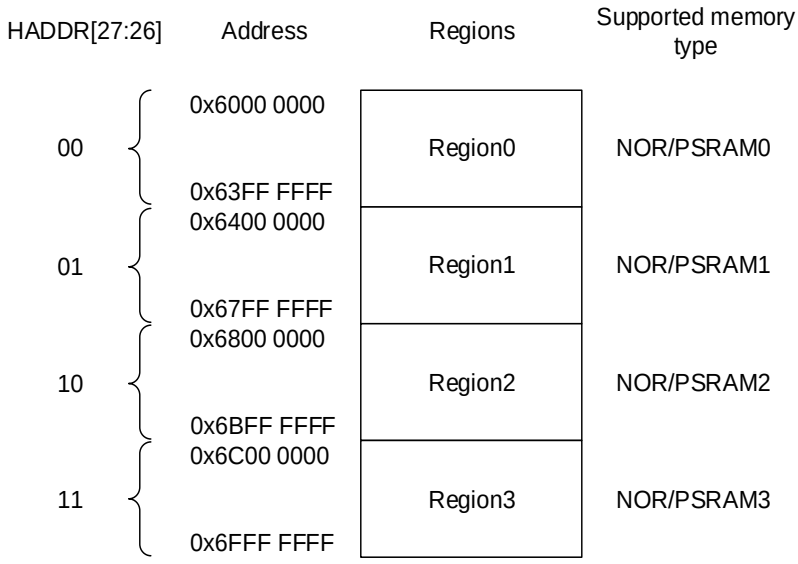
Each bank or region has a separate chip-select control signal, which can be configured independently.

Bank0 is used for NOR and PSRAM device access.

NOR/PSRAM address mapping

Figure 33-3. Four regions of bank0 address mapping reflects the address mapping of the four regions of bank0. Internal AHB address lines HADDR[27:26] bit are used to select the four regions.

Figure 33-3. 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.

- 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.
- 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.

33.3.4. NOR/PSRAM controller

NOR/PSRAM memory controller controls bank0, which is designed to support NOR Flash, PSRAM, SRAM, ROM and honeycomb RAM external memory. EXMC has 4 independent chip-select signals for each of the 4 sub-banks within bank0, named NE[x] (x = 0, 1, 2, 3). 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 33-1. NOR Flash interface signals description

EXMC Pin	Direction	Mode	Functional description
EXMC_CLK	Output	Sync	Clock signal for sync
Non-muxed EXMC_A[25:0]	Output	Async/Sync	Address bus signal
Muxed EXMC_A[25:16]			
EXMC_D[15:0]	Input/output	Async/Sync (muxed)	Address/Data bus
	Input/output	Async/Sync (non-muxed)	Data bus
EXMC_NE[x]	Output	Async/Sync	Chip selection, x=0/1/2/3
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 33-2. PSRAM non-muxed signal description

EXMC Pin	Direction	Mode	Functional description
EXMC_CLK	Output	Sync	Clock signal for sync
EXMC_A[25:0]	Output	Async/Sync	Address Bus
EXMC_D[15:0]	Input/output	Async/Sync	Data Bus
EXMC_NE[x]	Output	Async/Sync	Chip selection, x=0/1/2/3
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 8 or 16-bit for NOR, PSRAM and SRAM.

Table 33-3. Supported 16-bit transactions

Memory	Access Mode	R/W	AMBA Transaction Size	Memory Transaction Size	Comments
NOR Flash	Async	R	8	16	
	Async	W	8	16	Not allowed
	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	8	16	Not allowed
	Sync	R	16	16	
	Sync	R	32	16	Split into 2 EXMC accesses
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

Memory	Access Mode	R/W	AMBA Transaction Size	Memory Transaction Size	Comments
					accesses
	Sync	R	8	16	Not allowed
	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	
SRAM and ROM	Async	R	8	16	
	Async	R	16	16	
	Async	R	32	16	Split into 2 EXMC accesses
	Async	W	8	16	Use of byte lanes EXMC_NBL[1:0]
	Async	W	16	16	
	Async	W	32	16	

Table 33-4. Supported 8-bit transactions

Memory	Access Mode	R/W	AMBA Transaction Size	Memory Transaction Size	Comments
NOR Flash	Async	R	8	8	
	Async	W	8	8	
	Async	R	16	8	Split into 2 EXMC accesses
	Async	W	16	8	Split into 2 EXMC accesses
	Async	R	32	8	Split into 4 EXMC accesses
	Async	W	32	8	Split into 4 EXMC accesses
	Sync	R	8	8	
	Sync	R	16	8	Split into 2 EXMC accesses
	Sync	R	32	8	Split into 4 EXMC accesses
PSRAM	Async	R	8	8	
	Async	W	8	8	Use of byte lanes EXMC_NBL[1:0]
	Async	R	16	8	Split into 2 EXMC

Memory	Access Mode	R/W	AMBA Transaction Size	Memory Transaction Size	Comments
					accesses
	Async	W	16	8	Split into 2 EXMC accesses
	Async	R	32	8	Split into 4 EXMC accesses
	Async	W	32	8	Split into 4 EXMC accesses
	Sync	R	8	8	
	Sync	R	16	8	Split into 2 EXMC accesses
	Sync	R	32	8	Split into 4 EXMC accesses
	Sync	W	8	8	Use of byte lanes EXMC_NBL[1:0]
	Sync	W	16	8	Split into 2 EXMC accesses
	Sync	W	32	8	Split into 4 EXMC accesses
SRAM and ROM	Async	R	8	8	
	Async	R	16	8	Split into 2 EXMC accesses
	Async	R	32	8	Split into 4 EXMC accesses
	Async	W	8	8	Use of byte lanes EXMC_NBL[1:0]
	Async	W	16	8	Split into 2 EXMC accesses
	Async	W	32	8	Split into 4 EXMC accesses

Note: EXMC may not be able to reach the highest frequency when splitting multiple EXMC accesses in asynchronous mode.

NOR Flash/PSRAM controller timing

EXMC provides various programmable timing parameters and timing models for SRAM, ROM, PSRAM, NOR Flash and other external static memory.

Table 33-5. NOR / PSRAM controller timing parameters

Parameter	Function	Access mode	Unit	Min	Max
CKDIV	Sync Clock divide ratio	Sync	HCLK	2	16

Parameter	Function	Access mode	Unit	Min	Max
DLAT	Data latency	Sync	EXMC_CLK	2	17
BUSLAT	Bus latency	Async/Sync read	HCLK	0	15
DSET	Data setup time	Async	HCLK	1	255
AHLD	Address hold time	Async(muxed)	HCLK	1	15
ASET	Address setup time	Async	HCLK	0	15
BLSET	byte lanes setup time	Async	HCLK	0	3

Table 33-6. EXMC_timing models

Timing model		Extend mode	Mode description	Write timing parameter	Read timing parameter
Async	Mode 1	0	SRAM/PSRAM/CRAM	DSET ASET BLSET	DSET ASET BLSET
	Mode 2	0	NOR Flash	DSET ASET	DSET ASET
	Mode A	1	SRAM/PSRAM/CRAM with EXMC_NOE toggling on data phase	WDSET WASET BLSET	DSET ASET BLSET
	Mode B	1	NOR Flash	WDSET WASET	DSET ASET
	Mode C	1	NOR Flash with EXMC_NOE toggling on data phase	WDSET WASET	DSET ASET
	Mode D	1	With address hold capability	WDSET WAHLD WASET BLSET	DSET AHLD ASET BLSET
	Mode AM	0	NOR Flash address/data mux	DSET AHLD ASET BUSLAT BLSET	DSET AHLD ASET BUSLAT BLSET
Sync	Mode E	0	NOR/PSRAM/CRAM synchronous read PSRAM/CRAM synchronous write	DLAT CKDIV	DLAT CKDIV
	Mode SM	0	NOR Flash/ PSRAM/CRAM address/data mux	DLAT CKDIV	DLAT CKDIV

As shown in ***Table 33-6. EXMC_timing models***, EXMC NOR Flash / PSRAM controller provides a variety of timing model, users can modify those parameters listed in ***Table 33-5. NOR / PSRAM controller timing parameters*** to satisfy different external memory type and user's requirements. When extended mode is enabled via the EXMODEN bit in

EXMC_SNCTLx register, different timing patterns for read and write access could be generated independently according to EXMC_SNTCFGx and EXMC_SNWTCFGx register's configuration.

EXMC_CLK can be configured through the consecutive clock (CCK) bit. If CCK is set to 0, when NOR flash synchronous access is performed, EXMC_CLK will be generated. If CCK is set to 1, EXMC_CLK will be generated unconditionally whether the NOR flash is accessed in synchronous or asynchronous mode.

Asynchronous access timing diagram

Mode 1 - SRAM/CRAM

Figure 33-4. Mode 1 read access

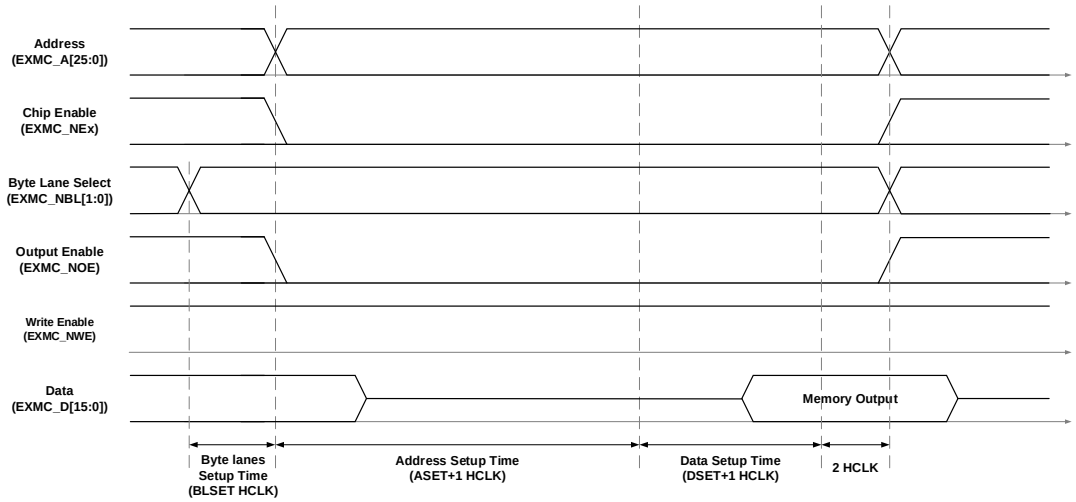


Figure 33-5. Mode 1 write access

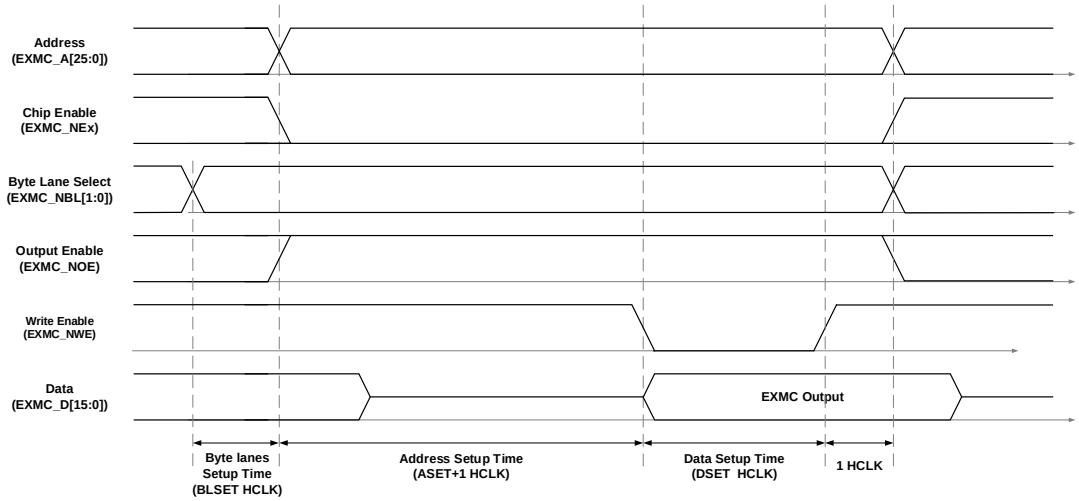


Table 33-7. Mode 1 related registers configuration

Bit Position	Bit Name	Reference Setting Value
EXMC_SNCTLx		
31-24	Reserved	0x00
23-22	BLSET	Depends on user
21	WFIFODIS	Depends on user
20	CCK	Depends on user
19	SYNCWR	0x0
18-16	CPS	0x0
15	ASYNCWAIT	Depends on memory
14	EXMODEN	0x0
13	NRWTEN	0x0
12	WEN	Depends on user
11	NRWTCFG	No effect
10	WRAPEN	0x0
9	NRWTPOL	Meaningful only when the bit 15 is set to 1
8	SBRSTEN	0x0
7	Reserved	0x1
6	NREN	No effect
5-4	NRW	Depends on memory
3-2	NRTP	Depends on memory, except 2(Nor Flash)
1	NRMUX	0x0
0	NRBKEN	0x1
EXMC_SNTCFGx		
31-30	Reserved	0x0000
29-28	ASYNCMOD	No effect
27-24	DLAT	No effect
23-20	CKDIV	No effect
19-16	BUSLAT	Time between EXMC_NE[x] rising edge to EXMC_NE[x] falling edge
15-8	DSET	Depends on memory and user (DSET+1 HCLK for write, DSET+3 HCLK for read)
7-4	AHLD	No effect
3-0	ASET	Depends on memory and user
EXMC_SNLATDECx		
31-3	Reserved	0x0
2-0	LATDEC	No effect

Mode A - SRAM/PSRAM(CRAM) OE toggling

Figure 33-6. Mode A read access

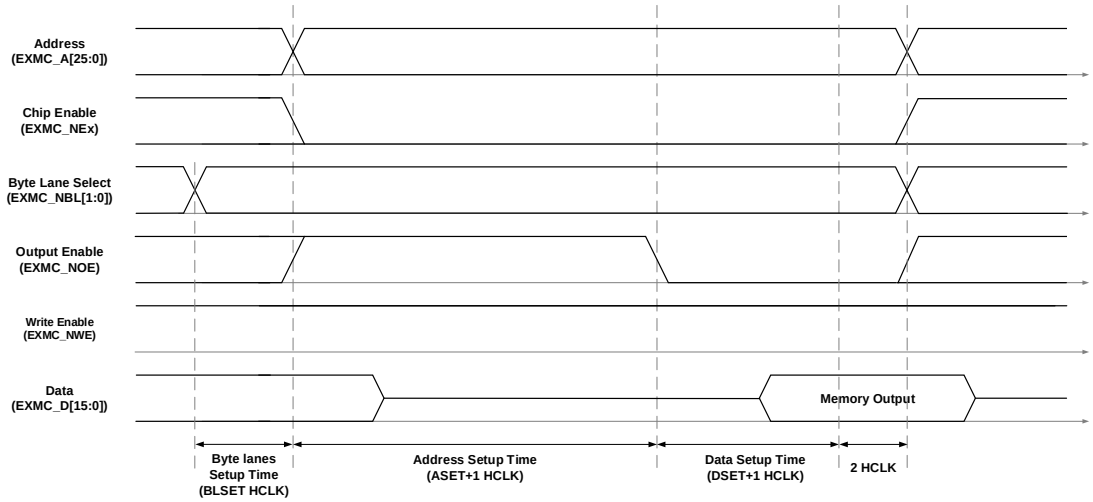
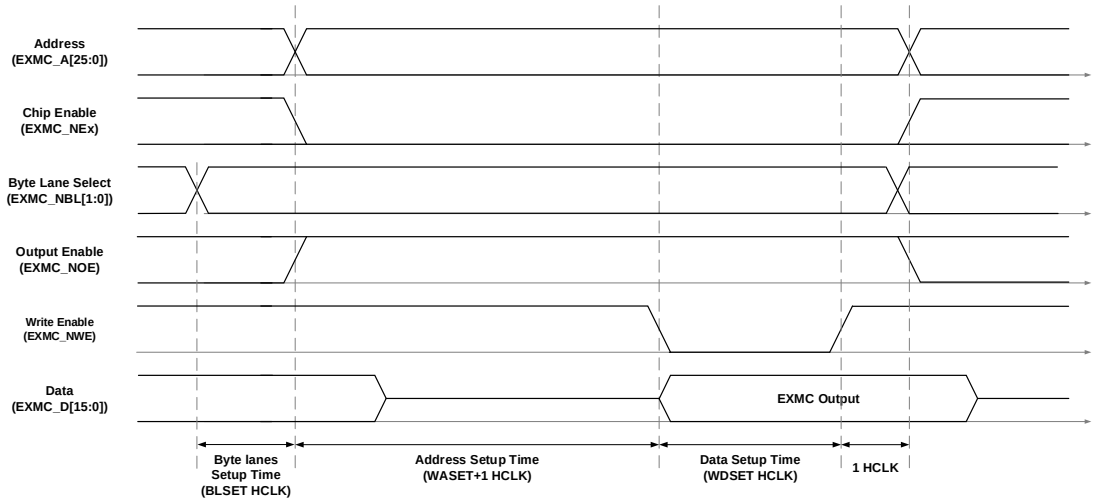


Figure 33-7. Mode A write access



The difference between mode A and mode 1 write timing is that read/write timing is specified by the same set of timing configuration, while mode A write timing configuration is independent of its read configuration.

Table 33-8. Mode A related registers configuration

Bit Position	Bit Name	Reference Setting Value
EXMC_SNCTLx		
31-24	Reserved	0x00
23-22	BLSET	Depends on user
21	WFIFODIS	Depends on user
20	CCK	Depends on user
19	SYNCWR	0x0
18-16	CPS	0x0
15	ASYNCWAIT	Depends on memory

Bit Position	Bit Name	Reference Setting Value
14	EXMODEN	0x1
13	NRWTEN	0x0
12	WEN	Depends on user
11	NRWTCFG	No effect
10	WRAPEN	0x0
9	NRWTPOL	Meaningful only when the bit 15 is set to 1
8	SBRSTEN	0x0
7	Reserved	0x1
6	NREN	No effect
5-4	NRW	Depends on memory
3-2	NRTP	Depends on memory, except 2(Nor Flash)
1	NRMUX	0x0
0	NRBKEN	0x1
EXMC_SNTCFGx(Read)		
31-30	Reserved	0x0
29-28	ASYNCMOD	0x0
27-24	DLAT	No effect
23-20	CKDIV	No effect
19-16	BUSLAT	Time between EXMC_NE[x] rising edge to EXMC_NE[x] falling edge
15-8	DSET	Depends on memory and user (DSET+3 HCLK for read)
7-4	AHLD	No effect
3-0	ASET	Depends on memory and user
EXMC_SNWTCFGx(Write)		
31-30	Reserved	0x0
29-28	WASYNCMOD	0x0
27-20	Reserved	0x00
19-16	WBUSLAT	Time between EXMC_NE[x] rising edge to EXMC_NE[x] falling edge
15-8	WDSET	Depends on memory and user (WDSET+1 HCLK for write)
7-4	WAHLD	0x0
3-0	WASET	Depends on memory and user
EXMC_SNLATDECx		
31-3	Reserved	0x00
2-0	LATDEC	No effect

Mode 2/B - NOR Flash

Figure 33-8. Mode 2/B read access

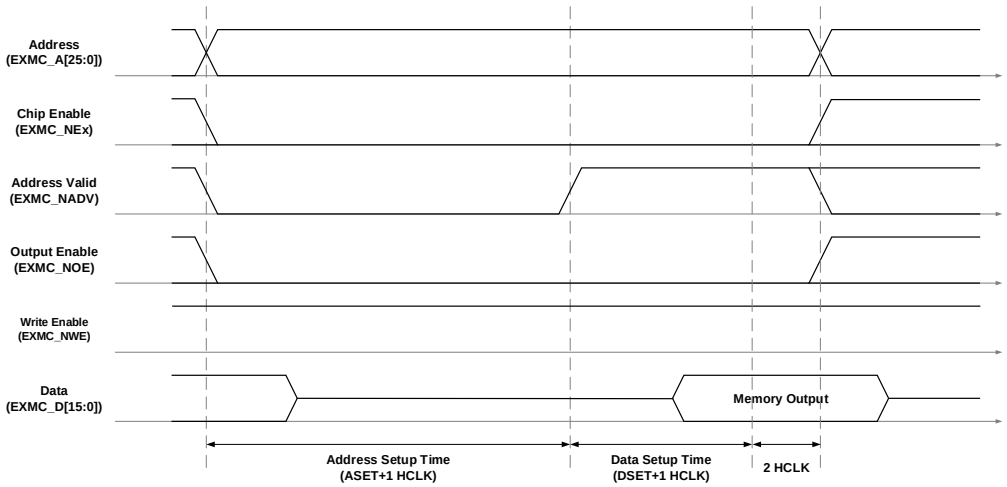


Figure 33-9. Mode 2 write access

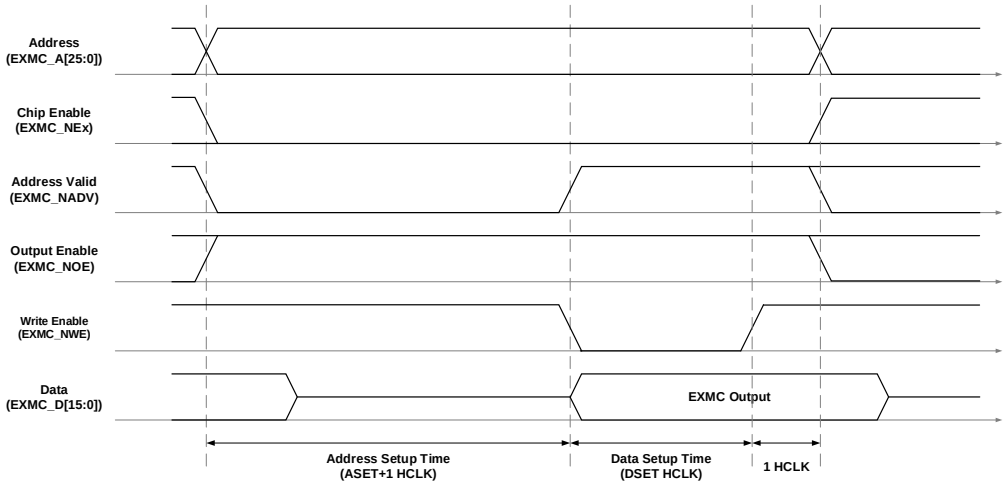


Figure 33-10. Mode B write access

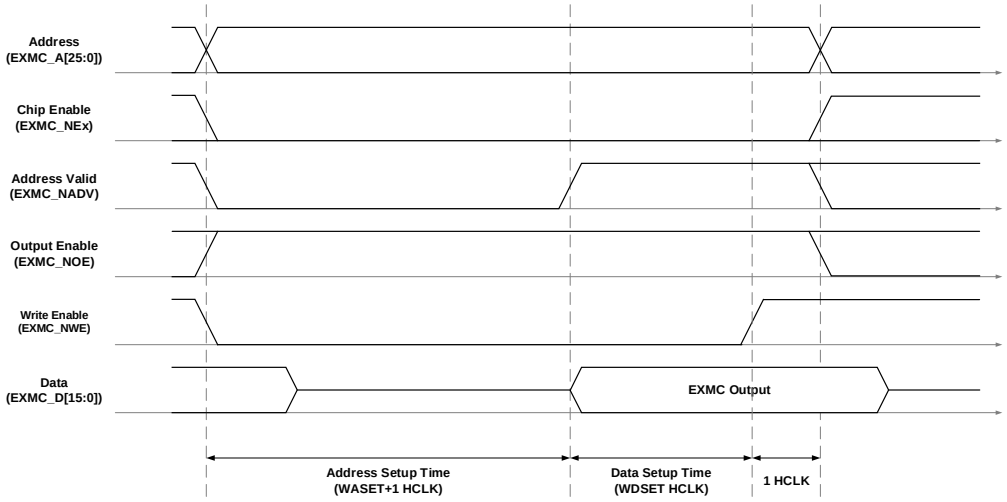


Table 33-9. Mode 2/B related registers configuration

Bit Position	Bit Name	Reference Setting Value
EXMC_SNCTLx(Mode 2, Mode B)		
31-24	Reserved	0x00
23-22	BLSET	No effect
21	WFIFODIS	Depends on user
20	CCK	Depends on user
19	SYNCWR	0x0
18-16	CPS	0x0
15	ASYNCWAIT	Depends on memory
14	EXMODEN	Mode 2:0x0, Mode B:0x1
13	NRWTEN	0x0
12	WEN	Depends on user
11	NRWTCFG	No effect
10	WRAPEN	0x0
9	NRWTPOL	Meaningful only when the bit 15 is set to 1
8	SBRSTEN	0x0
7	Reserved	0x1
6	NREN	0x1
5-4	NRW	Depends on memory
3-2	NRTP	0x2, NOR Flash
1	NRMUX	0x0
0	NRBKEN	0x1
EXMC_SNTCFGx(Read and write in mode 2,read in mode B)		
31-30	Reserved	0x0000
29-28	ASYNCMOD	Mode B:0x1
27-24	DLAT	No effect
23-20	CKDIV	No effect
19-16	BUSLAT	Time between EXMC_NE[x] rising edge to EXMC_NE[x] falling edge
15-8	DSET	Depends on memory and user (DSET+3 HCLK for read)
7-4	AHLD	0x0
3-0	ASET	Depends on memory and user
EXMC_SNWTCFGx(Write in mode B)		
31-30	Reserved	0x0000
29-28	WASYNCMOD	Mode B:0x1
27-20	Reserved	0x00
19-16	WBUSLAT	Time between EXMC_NE[x] rising edge to EXMC_NE[x] falling edge
15-8	WDSET	Depends on memory and user (WDSET+1 HCLK for write)

Bit Position	Bit Name	Reference Setting Value
7-4	WAHLD	0x0
3-0	WASET	Depends on memory and user
EXMC_SNLATDECx		
31-3	Reserved	0x00
2-0	LATDEC	No effect

Mode C - NOR Flash OE toggling

Figure 33-11. Mode C read access

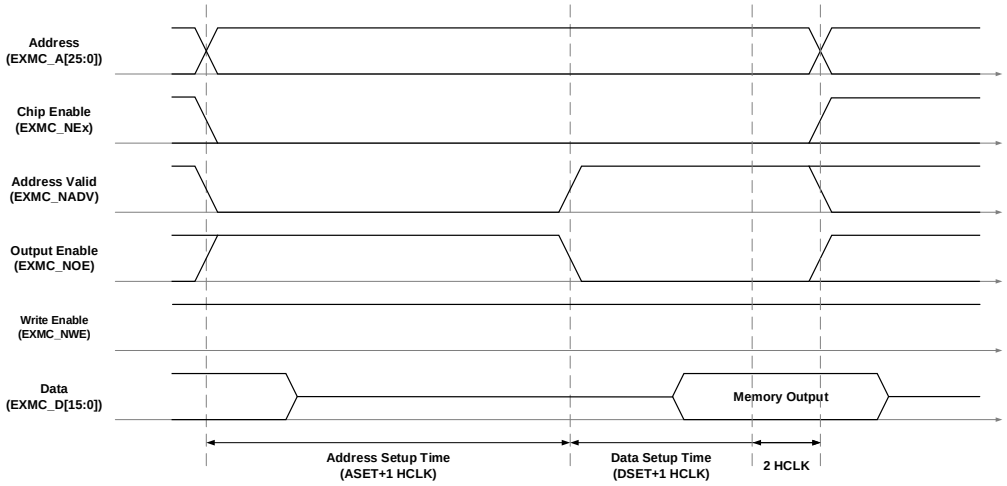
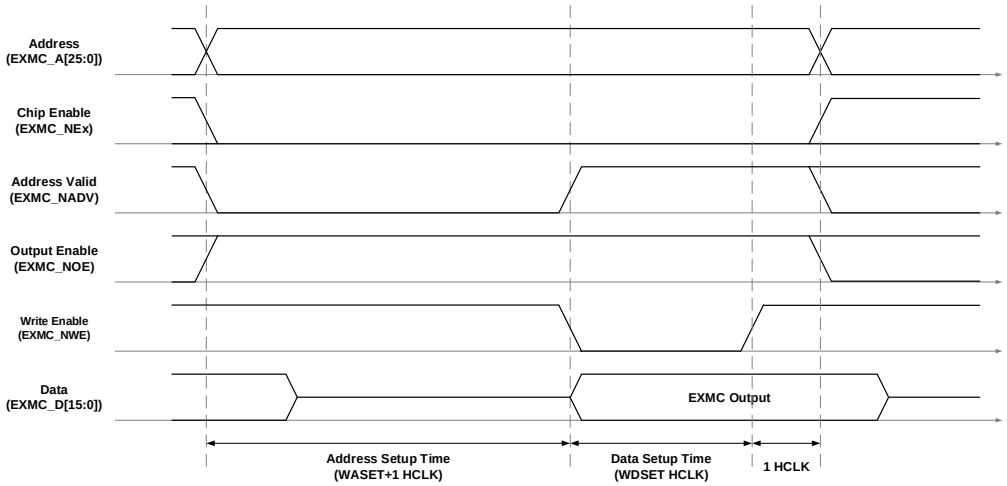


Figure 33-12. Mode C write access



The differences between mode C and mode 1 write timing are that read/write timing is specified by the same set of timing configuration, while mode C write timing configuration is independent of its read configuration, and the toggle of NOE and NADV are different.

Table 33-10. Mode C related registers configuration

Bit Position	Bit Name	Reference Setting Value
EXMC_SNCTLx		
31-24	Reserved	0x00

Bit Position	Bit Name	Reference Setting Value
23-22	BLSET	No effect
21	WFIFODIS	Depends on user
20	CCK	Depends on user
19	SYNCWR	0x0
18-16	CPS	0x0
15	ASYNCWAIT	Depends on memory
14	EXMODEN	0x1
13	NRWTEN	0x0
12	WEN	Depends on user
11	NRWTCFG	No effect
10	WRAPEN	0x0
9	NRWTPOL	Meaningful only when the bit 15 is set to 1
8	SBRSTEN	0x0
7	Reserved	0x1
6	NREN	0x1
5-4	NRW	Depends on memory
3-2	NRTP	0x2, NOR Flash
1	NRMUX	0x0
0	NRBKEN	0x1
EXMC_SNTCFGx(Read)		
31-30	Reserved	0x0000
29-28	ASYNCMOD	Mode C:0x2
27-24	DLAT	No effect
23-20	CKDIV	No effect
19-16	BUSLAT	Time between EXMC_NE[x] rising edge to EXMC_NE[x] falling edge
15-8	DSET	Depends on memory and user (DSET+3 HCLK for read)
7-4	AHLD	0x0
3-0	ASET	Depends on memory and user
EXMC_SNWTCFGx(Write)		
31-30	Reserved	0x0
29-28	WASYNCMOD	Mode C:0x2
27-20	Reserved	0x00
19-16	WBUSLAT	Time between EXMC_NE[x] rising edge to EXMC_NE[x] falling edge
15-8	WDSET	Depends on memory and user (WDSET+1 HCLK for write)
7-4	WAHLD	0x0
3-0	WASET	Depends on memory and user
EXMC_SNLATDECx		

Bit Position	Bit Name	Reference Setting Value
31-3	Reserved	0x00
2-0	LATDEC	No effect

Mode D - Asynchronous access with extended address

Figure 33-13. Mode D read access

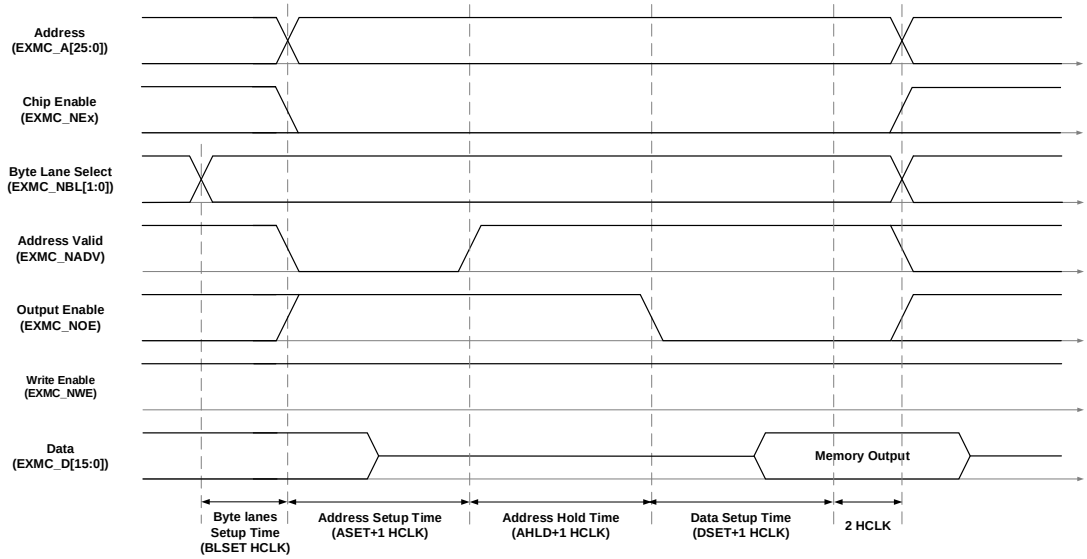


Figure 33-14. Mode D write access

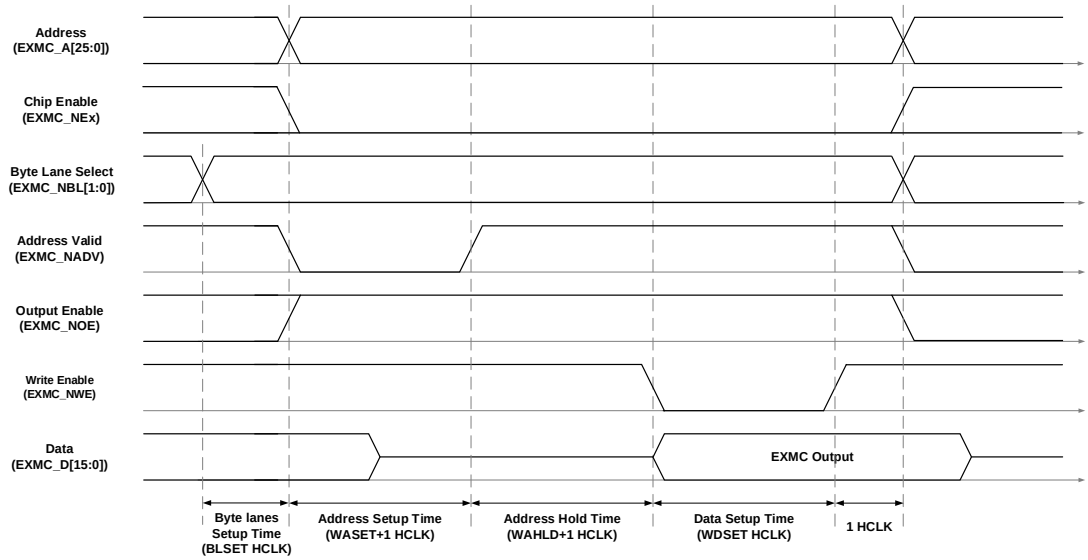


Table 33-11. Mode D related registers configuration

Bit Position	Bit Name	Reference Setting Value
EXMC_SNCTLx		
31-24	Reserved	0x00
23-22	BLSET	Depends on user

Bit Position	Bit Name	Reference Setting Value
21	WFIFODIS	Depends on user
20	CCK	Depends on user
19	SYNCWR	0x0
18-16	CPS	0x0
15	ASYNCWAIT	Depends on memory
14	EXMODEN	0x1
13	NRWTEN	0x0
12	WEN	Depends on user
11	NRWTCFG	No effect
10	WRAPEN	0x0
9	NRWTPOL	Meaningful only when the bit 15 is set to 1
8	SBRSTEN	0x0
7	Reserved	0x1
6	NREN	Depends on memory
5-4	NRW	Depends on memory
3-2	NRTP	Depends on memory
1	NRMUX	0x0
0	NRBKEN	0x1
EXMC_SNTCFGx(Read)		
31-30	Reserved	0x0
29-28	ASYNCMOD	Mode D:0x3
27-24	DLAT	Don’t care
23-20	CKDIV	No effect
19-16	BUSLAT	Time between EXMC_NE[x] rising edge to EXMC_NE[x] falling edge
15-8	DSET	Depends on memory and user (DSET+3 HCLK for read)
7-4	AHLD	Depends on memory and user
3-0	ASET	Depends on memory and user
EXMC_SNWTCFGx(Write)		
31-30	Reserved	0x0
29-28	WASYNCMOD	Mode D:0x3
27-20	Reserved	0x00
19-16	WBUSLAT	Time between EXMC_NE[x] rising edge to EXMC_NE[x] falling edge
15-8	WDSET	Depends on memory and user (WDSET+1HCLK for write)
7-4	WAHLD	Depends on memory and user
3-0	WASET	Depends on memory and user
EXMC_SNLATDECx		
31-3	Reserved	0x00

Bit Position	Bit Name	Reference Setting Value
2-0	LATDEC	No effect

Mode AM - NOR Flash address / data bus multiplexing

Figure 33-15. Multiplex mode read access

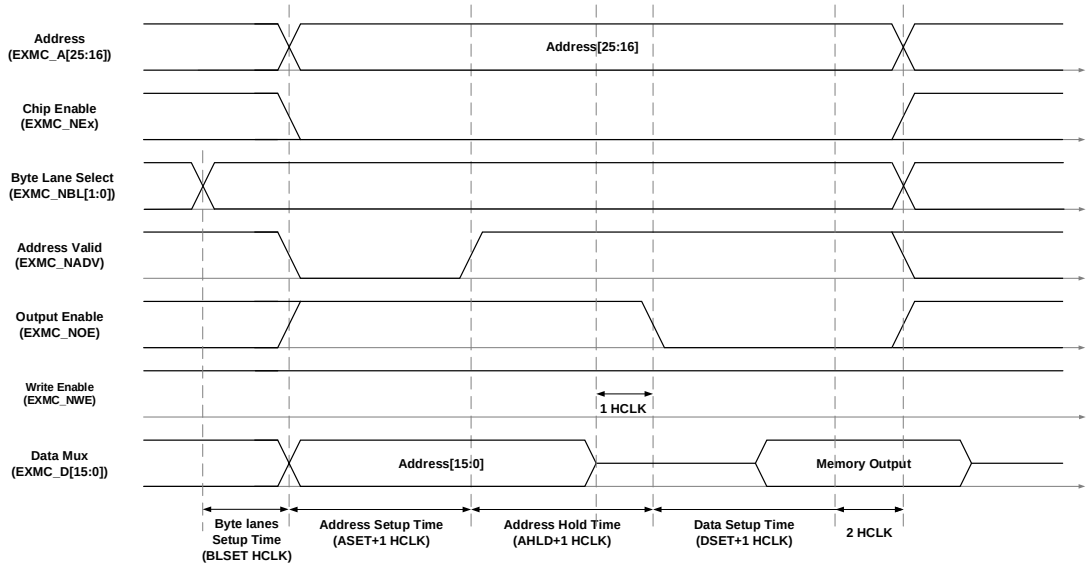


Figure 33-16. Multiplex mode write access

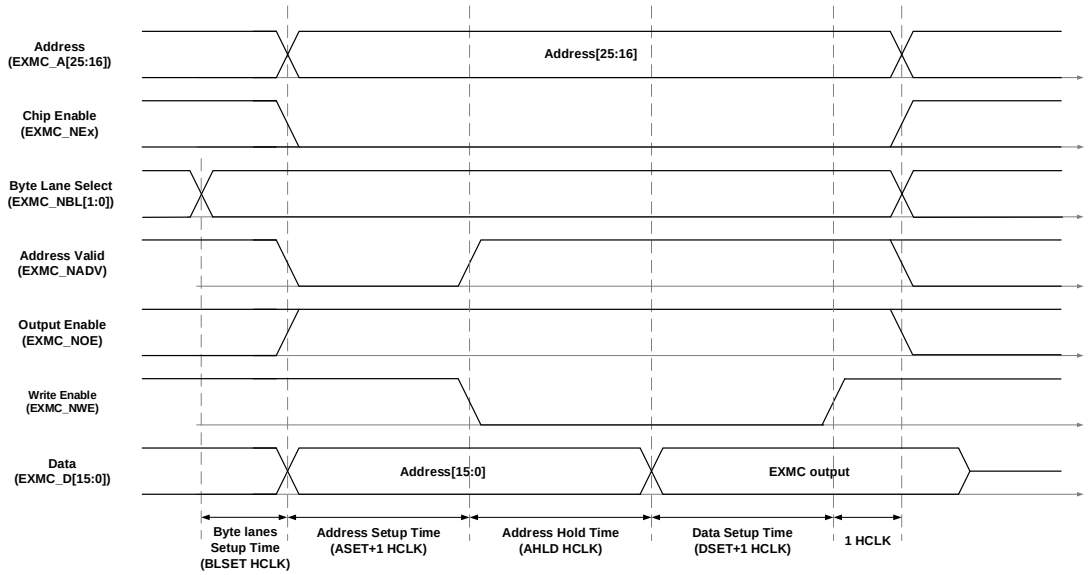


Table 33-12. Multiplex mode related registers configuration

Bit Position	Bit Name	Reference Setting Value
EXMC_SNCTLx		
31-24	Reserved	0x00
23-22	BLSET	Depends on user

Bit Position	Bit Name	Reference Setting Value
21	WFIFODIS	Depends on user
20	CCK	Depends on user
19	SYNCWR	0x0
18-16	CPS	0x0
15	ASYNCWAIT	Depends on memory
14	EXMODEN	0x0
13	NRWTEN	0x0
12	WEN	Depends on memory
11	NRWTCFG	No effect
10	WRAPEN	0x0
9	NRWTPOL	Meaningful only when the bit 15 is set to 1
8	SBRSTEN	0x0
7	Reserved	0x1
6	NREN	0x1
5-4	NRW	Depends on memory
3-2	NRTP	0x2:NOR Flash
1	NRMUX	0x1
0	NRBKEN	0x1
EXMC_SNTCFGx		
31-30	Reserved	0x0
29-28	ASYNCMOD	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
EXMC_SNLATDECx		
31-3	Reserved	0x00
2-0	LATDEC	No effect

Wait timing of asynchronous communication

Wait feature is controlled by the bit ASYNCWAIT in register EXMC_SNCTLx. During extern memory access, data setup phase will be automatically extended by the active EXMC_NWAIT signal if ASYNCWAIT bit is set. The extend time is calculated as follows:

If memory wait signal is aligned to EXMC_NOE/ EXMC_NWE:

$$T_{DATA_SETUP} \geq \max T_{WAIT_ASSERTION} + 4HCLK \quad (33\text{-}1)$$

If memory wait signal is aligned to EXMC_NE:

If

$$maxT_{WAIT_ASSERTION} \geq T_{ADDRES_PHASE} + T_{HOLD_PHASE} \quad (33\text{-}2)$$

be

$$T_{DATA_SETUP} \geq (maxT_{WAIT_ASSERTION} - T_{ADDRES_PHASE} - T_{HOLD_PHASE}) + 4HCLK \quad (33\text{-}3)$$

Otherwise

$$T_{DATA_SETUP} \geq 4HCLK \quad (33\text{-}4)$$

Figure 33-17. Read access timing diagram under async-wait signal assertion

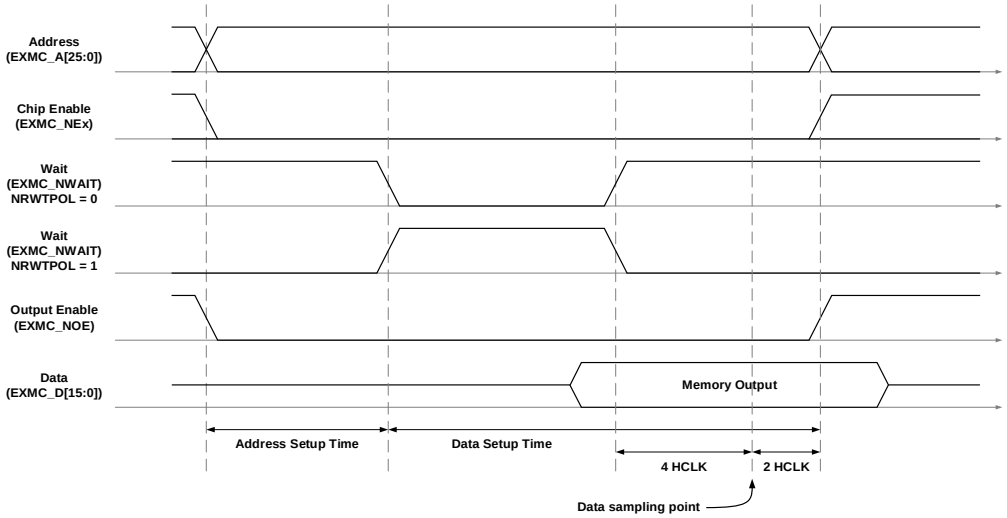
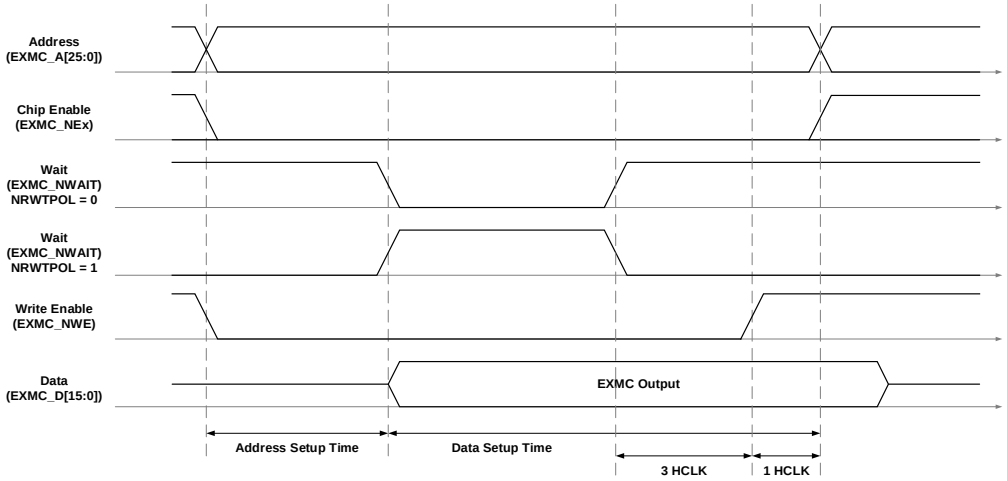


Figure 33-18. Write access timing diagram under async-wait signal assertion



Synchronous access timing diagram

The relations between memory clock (EXMC_CLK) and system clock (HCLK) clock are as follows:

$$EXMC_CLK = \frac{HCLK}{CKDIV+1} \quad (33\text{-}5)$$

CKDIV is the synchronous clock divider ratio, it is configured through the CKDIV control field in the EXMC_SNTCFGx 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 (33-6)

For NOR Flash's specification including the EXMC_NADV cycle, their relationship should be:

NOR Flash latency=DLAT+3 (33-7)

Note:

During read access, the data latency is determined by both DLAT in the EXMC_SNTCFGx register and LATDEC in the EXMC_SNLATDECx registers. For details, see ***SRAM/NOR flash data latency decrease registers (EXMC_SNLATDECx) (x=0, 1, 2, 3)***.

2. Data wait

Users should guarantee that EXMC_NWAIT signal matches that of the external device. This signal is configured through the EXMC_SNCTLx 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_SNCTLx register is set, EXMC_NWAIT signal will be detected after a period of data latency. If EXMC_NWAIT signal detected is valid, wait cycles will be inserted until EXMC_NWAIT becomes invalid.

- The valid polarity of EXMC_NWAIT:

NRWTPOL= 1: valid level of EXMC_NWAIT signal is high.
NRWTPOL= 0: valid level of EXMC_NWAIT signal is low.

- In synchronous burst mode, EXMC_NWAIT signal has two kinds of configurations:

NRWTCFG = 1: When EXMC_NWAIT signal is active, current cycle data is not valid.
NRWTCFG = 0: When EXMC_NWAIT signal is active, the next cycle data is not valid. It is the default state after reset.

During wait-state inserted via the EXMC_NWAIT signal, the controller continues to send clock pulses to the memory, keep the chip select and output signals availably, and ignore the invalid data signal.

3. Automatic burst split at CRAM page boundary

Crossing page boundary burst access is prohibited in CRAM 1.5, an automatic burst split

functionality is implemented by the EXMC. To guarantee correct burst split operation, users should specify CRAM page size by configuring the CPS bit in EXMC_SNCTLx register to inform the EXMC when this functionality should be performed.

4. Mode SM - Single burst transmission

For synchronous burst transmission, if the needed data of AHB is 16-bit, EXMC will perform a burst transmission whose length is 1. If the needed data of AHB is 32-bit, EXMC will make the transmission divided into two 16-bit transmissions, that is, EXMC performs a burst transmission whose length is 2.

For other configurations please refers to ***Supported memory access mode***.

Synchronous mux burst read timing - NOR, PSRAM (CRAM)

Figure 33-19. Read timing of synchronous multiplexed burst mode

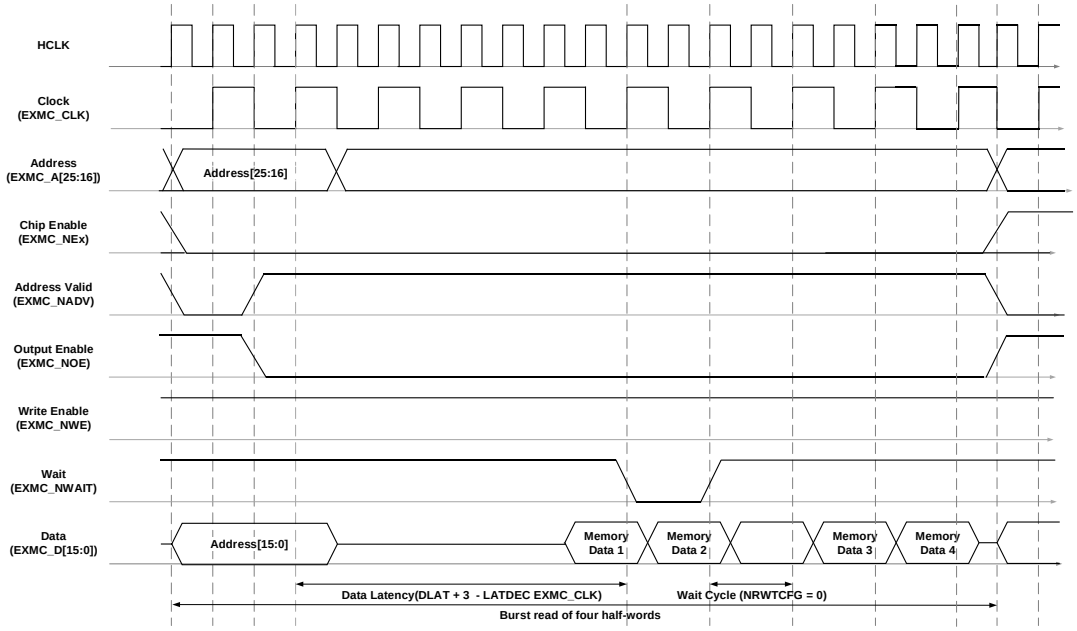


Table 33-13. Timing configurations of synchronous multiplexed read mode

Bit Position	Bit Name	Reference Setting Value
EXMC_SNCTLx		
31-24	Reserved	0x00
23-22	BLSET	No effect
21	WFIFODIS	No effect
20	CCK	Depends on user
19	SYNCWR	No effect
18-16	CPS	0x0
15	ASYNCWAIT	0x0
14	EXMODEN	0x0
13	NRWTEN	Depends on memory
12	WEN	No effect
11	NRWTCFG	Depends on memory

Bit Position	Bit Name	Reference Setting Value
EXMC_SNCTLx		
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_SNTCFGx(Read)		
31-30	Reserved	0x0
29-28	ASYNCMOD	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
EXMC_SNLATDECx		
31-3	Reserved	0x0
2-0	LATDEC	Depends on memory and users

Mode SM –Synchronous mux burst write timing – NOR, PSRAM (CRAM)

Figure 33-20. Write timing of synchronous multiplexed burst mode

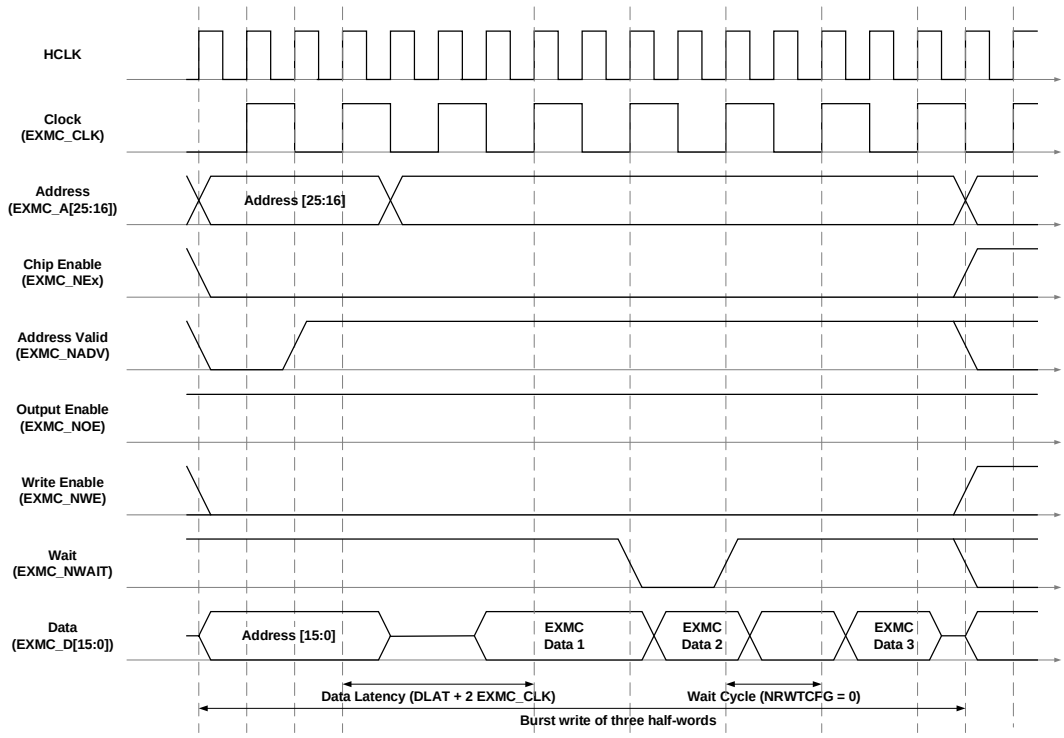


Table 33-14. Timing configurations of synchronous multiplexed write mode

Bit Position	Bit Name	Reference Setting Value
EXMC_SNCTLx		
31-24	Reserved	0x00
23-22	BLSET	No effect
21	WFIFODIS	No effect
20	CCK	Depends on user
19	SYNCWR	0x1, synchronous write enable
18-16	CPS	0x0
15	AYSNCWAIT	0x0
14	EXMODEN	0x0
13	NRWTEN	Depends on memory
12	WEN	0x1
11	NRWTCFG	0x0(Here must be zero)
10	WRAPEN	0x0
9	NTWTPOL	Depends on memory
8	SBRSTEN	No effect
7	Reserved	0x1
6	NREN	Depends on memory
5-4	NRW	0x1
3-2	NRTP	0x1
1	NRMUX	0x1, Depends on users
0	NRBKEN	0x1

Bit Position	Bit Name	Reference Setting Value
EXMC_SNCTLx		
EXMC_SNTCFGx(Write)		
31-30	Reserved	0x0
29-28	ASYNCMOD	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
EXMC_SNLATDECx		
31-3	Reserved	0x0
2-0	LATDEC	No effect

Write FIFO Support

The write data FIFO is used to accelerate AHB write access to external memory, refer to the WFIFODIS bit in EXMC_SNCTL register for details.

While max depth of FIFO is 16, it is suggested to use burst size not larger than 16. When in FIFO mode (default mode), value of all control registers can not be changed when FIFO is not empty.

33.4. Registers definition

EXMC base address: 0xA000 0000

33.4.1. NOR/PSRAM controller registers

SRAM/NOR Flash control registers (EXMC_SNCTLx) (x=0, 1, 2, 3)

Address offset: 0x00 + 8 * x, (x = 0, 1, 2, and 3)
Reset value: 0x0000 30DA (Region 0)
Reset value: 0x0000 30D2 (Region 1)
Reset value: 0x0000 30D2 (Region 2)
Reset value: 0x0000 30D2 (Region 3)

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								BLSET[1:0]		WFIFODIS	CCK	SYNCWR	CPS[2:0]		
								rw		rw	rw	rw		rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ASYNCWAIT	EXMODEN	NRWTEN	WEN	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
31:24	Reserved	Must be kept at reset value.
23:22	BLSET[1:0]	NBL (Byte lane) setup time 00: NBL setup time from NBL low to chip select NE low is 0 AHB clock cycle 01: NBL setup time from NBL low to chip select NE low is 1 AHB clock cycle 10: NBL setup time from NBL low to chip select NE low is 2 AHB clock cycle 11: NBL setup time from NBL low to chip select NE low is 3 AHB clock cycle
21	WFIFODIS	Write FIFO disabled This bit controlles the Write FIFO. 0: Enable Write FIFO (default after reset) 1: Disable Write FIFO **Note:** The WFIFODIS bit of the EXMC_SNCTL1..3 registers is don't care. It can be enabled only through the EXMC_SNCTL0 register.
20	CCK	Consecutive Clock 0: EXMC_CLK is generated only during synchronous access. 1: EXMC_CLK is generated unconditionally. **Note:** The CCK bit of the EXMC_SNCTL1..3 registers is don't care. It can be enabled only through the EXMC_SNCTL0 register.

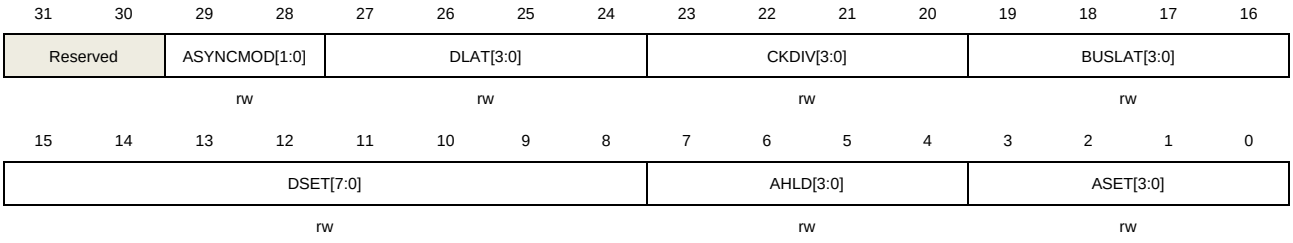
		When this bit set, only EXMC_SNCTL0 CKDIV[3:0] divide ratio can effect EXMC_CLK output.
19	SYNCWR	Synchronous write 0: Asynchronous write 1: Synchronous write
18:16	CPS[2:0]	CRAM page size 000: No burst split on page boundary crossing 001: 128 bytes 010: 256 bytes 011: 512 bytes 100: 1024 bytes Others: Reserved
15	ASYNCWAIT	Asynchronous wait 0: Disable the asynchronous wait function 1: Enable the asynchronous wait function
14	EXMODEN	Extended mode enable 0: Disable extended mode 1: Enable extended mode
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	WEN	Write enable 0: Disabled write in the bank by the EXMC, otherwise an AHB error is reported 1: Enabled write in the bank by the EXMC (default after reset)
11	NRWTCFG	NWAIT signal configuration, only work in synchronous mode 0: NWAIT signal is active one data cycle before wait state 1: NWAIT signal is active during wait state
10	WRAPEN	Wrapped burst mode enable 0: Disable wrap burst mode support 1: Enable wrap burst mode support
9	NRWTPOL	NWAIT signal polarity 0: Low level is active of NWAIT 1: High level is active of NWAIT
8	SBRSTEN	Synchronous burst enable 0: Disable burst access mode 1: Enable burst access mode

7	Reserved	Must be kept at reset value.
6	NREN	NOR Flash access enable 0: Disable NOR Flash access 1: Enable NOR Flash access
5:4	NRW[1:0]	NOR region memory data bus width 00: 8 bits 01: 16 bits(default after reset) 10/11: Reserved
3:2	NRTP[1:0]	NOR region memory type 00: SRAM(default after reset for region1-region3) 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 Accessing a disabled bank region causes an ERROR on AHB bus. 0: Disable the corresponding memory bank 1: Enable the corresponding memory bank

SRAM/NOR Flash timing configuration registers (EXMC_SNTCFGx) (x=0, 1, 2, 3)

Address offset: 0x04 + 8 * x, (x = 0, 1, 2, and 3)
Reset value: 0x0FFF FFFF

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



Bits	Fields	Descriptions
31: 30	Reserved	Must be kept at reset value.
29:28	ASYNCMOD[1:0]	Asynchronous access mode The bits are valid only when the EXMODEN bit in the EXMC_SNCTLx register is 1.

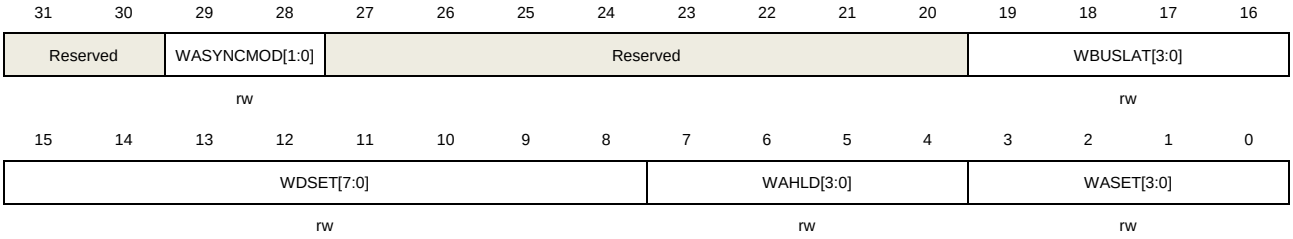
		00: Mode A access 01: Mode B access 10: Mode C access 11: Mode D access
27:24	DLAT[3:0]	Data latency for NOR Flash. Only valid in synchronous access 0x0: Data latency of first burst access is 2 EXMC_CLK 0x1: Data latency of first burst access is 3 EXMC_CLK …… 0xF: Data latency of first burst access is 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 = 0 * HCLK period 0x1: Bus latency = 1 * HCLK period …… 0xF: Bus latency = 15 * HCLK period
15:8	DSET[7:0]	Data setup time This field is meaningful only in asynchronous access. 0x00: Reserved 0x01: Data setup time = 1 * HCLK period …… 0xFF: Data setup time = 255 * 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 = 1 * HCLK …… 0xF: Address hold time = 15 * 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 SRAM,ROM,NOR Flash 0x0: Address setup time = 0 * HCLK …… 0xF: Address setup time = 15 * HCLK

SRAM/NOR Flash write timing configuration registers (EXMC_SNWTCFGx) (x=0, 1, 2, 3)

Address offset: 0x104 + 8 * x, (X = 0, 1, 2, and 3)
Reset value: 0x0FFF FFFF

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

This register is meaningful only when the EXMODEN bit in EXMC_SNCTLx is set to 1.



Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29:28	WASYNCMOD[1:0]	Asynchronous access mode The bits are valid only when the EXMODEN bit in the EXMC_SNCTLx register is 1. 00: Mode A access 01: Mode B access 10: Mode C access 11: Mode D access
27:20	Reserved	Must be kept at reset value.
19:16	WBUSLAT[3:0]	Bus latency Bus latency added at the end of each write transaction to match with the minimum time between consecutive transactions. 0x0: Bus latency = 0 * HCLK period 0x1: Bus latency = 1 * HCLK period …… 0xF: Bus latency = 15 * HCLK period
15:8	WDSET[7:0]	Data setup time This field is meaningful only in asynchronous access. 0x00: Reserved 0x01: Data setup time = 1 * HCLK period …… 0xFF: Data setup time = 255 * HCLK period
7:4	WAHLD[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 = 1 * HCLK …… 0xF: Address hold time = 15 * HCLK
3:0	WASET[3:0]	Address setup time This field is used to set the time of address setup phase. Note: Meaningful only in asynchronous access of SRAM,ROM,NOR Flash 0x0: Address setup time = 0 * HCLK 0x1: Address setup time = 1 * HCLK …… 0xF: Address setup time = 15 * HCLK

SRAM/NOR flash status register (EXMC_SNSTAT)

Address offset: 0x84
Reset value: 0x0000 0040

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

31	30	29	28	27	26	25	24	23	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									FIFOEPT	Reserved					
									r						

Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value.
6	FIFOEPT	FIFO empty flag 0: FIFO is not empty. 1: FIFO is empty.
5:0	Reserved	Must be kept at reset value.

SRAM/NOR flash data latency decrease registers (EXMC_SNLATDECx) (x=0, 1, 2, 3)

Address offset: 0x300+ 4 * x, (x= 0, 1, 2, and 3)
Reset value: 0x0000 0000

This register is meaningful only in synchronous access.

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

31	30	29	28	27	26	25	24	23	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													LATDEC[1:0]		
													rw		

Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value.
2:0	LATDEC[2:0]	Data latency decrease for NOR Flash. Only valid in synchronous read access. This field is used to adjust read access time along with DLAT. Burst read: 000: Data latency of first burst access is (DLAT + 3) CLK 001: Data latency of first burst access is (DLAT + 2) CLK 010: Data latency of first burst access is (DLAT + 1) CLK 011: Data latency of first burst access is (DLAT + 0) CLK 100: Data latency of first burst access is (DLAT - 1) CLK 101: Data latency of first burst access is (DLAT - 2) CLK 110: Data latency of first burst access is (DLAT - 3) CLK 111: Data latency of first burst access is (DLAT - 4) CLK **Note:** For example, if the data latency in read mode needs to be configured with 3 CLK, the DLAT[3:0] should be 0b'0000 and LATDEC[2:0] should be 0b'010.

34. High-Performance Digital Filter (HPDF)

34.1. Overview

A high performance digital filter module (HPDF) for external sigma delta (Σ-Δ) modulator is integrated in GD32G553. HPDF supports SPI interface and Manchester-coded single-wire interface. The external sigma delta modulator can be connected with MCU by the serial interface, and the serial data stream output by sigma delta modulator can be filtered. In addition, HPDF also supports the parallel data stream input, which can be selected from internal ADC peripherals or from MCU memory.

34.2. Characteristics

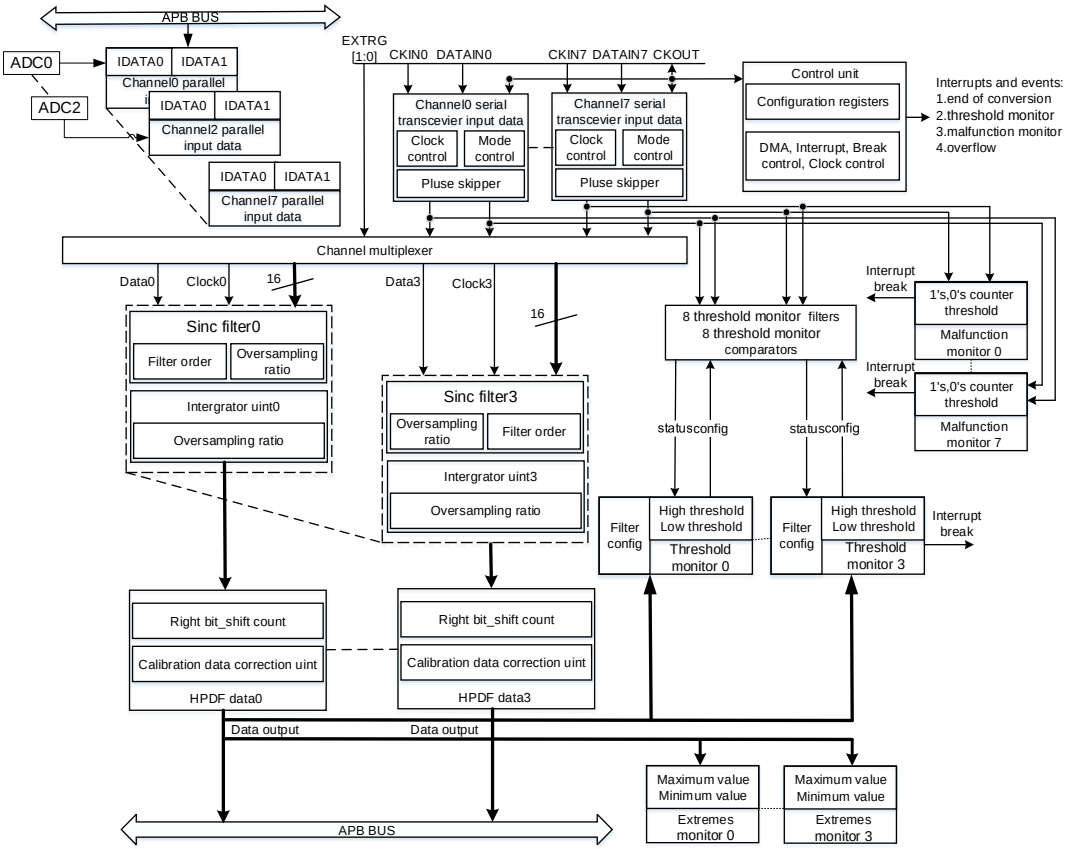
- 8 multiplex digital serial input channels
  - configurable SPI and Manchester interfaces
- 8 internal digital parallel input channels
  - input with up to 16-bit resolution
  - internal source: ADC data or memory (CPU / DMA write) data stream
- Configurable Sinc filter and integrator
  - the order and oversampling rate (decimation rate) of Sinc filter can be configured;
  - sampling rate of configurable integrator
- Threshold monitor function
  - independent Sinc filter, configurable order and oversampling rate (decimation rate)
  - configurable data input source: serial channel input data or HPDF output data
- Malfunction monitor function
  - A counter with 8 bits is used to monitor the continuous 0 or 1 in the serial channel input data stream
- Extreme monitor function
  - store minimum and maximum values of output data values of HPDF
- Up to 24-bit output data resolution
- Clock signal can be provided to external sigma delta modulator
  - provide configurable clock signal by the CKOUT pin
- Flexible conversion configuration function
  - the conversion channel is divided into regular group and inserted group
  - support multiple conversion modes and startup modes
- HPDF output data is in signed format

34.3. Function overview

34.3.1. HPDF Block Diagram

The structural block diagram of HPDF is shown in *Figure 34-1. HPDF block diagram*.

Figure 34-1. HPDF block diagram



The HPDF interface communicates with the external Σ-Δ modulator by the pins and internal signal in *Table 34-1. HPDF pins definition*.

Table 34-1. HPDF pins definition

PINs	Type	Description
EXTRG[1:0]	External trigger input	Input pin of external trigger signal source, the trigger signal sources are EXTI11 and EXTI15, which are used as the trigger signal of inserted group HPDF_ ITRG[24] and HPDF_ITRG[25]
CKOUT	Clock out	The clock output signal of HPDF module, provides clock signal to external Σ-Δ modulator.
CKINx	Clock input	External Σ-Δ modulator provides clock signal to serial interface.

PINs	Type	Description
DATAINx	Data input	The external Σ-Δ modulator transmits 1 bit data stream to the serial channel by this pin.

Table 34-2. HPDF internal signal

Break name	Break destination
HPDF_BREAK[0]	TIMER0 break0 / TIMER14 break0 / TIMER19 break0
HPDF_BREAK[1]	TIMER0 break1 / TIMER15 break0 / TIMER19 break1
HPDF_BREAK[2]	TIMER7 break0 / TIMER16 break0
HPDF_BREAK[3]	TIMER7 break1

34.3.2. HPDF on-off control

When the HPDF module is started normally, the HPDF module can be enabled globally by setting HPDFEN to 1 in the HPDF_CH0CTL register. Then set the CHEN bit in HPDF_CHxCTL and the FLTEN bit in HPDF_FLTyCTL0 to 1 to enable the input channel and channel digital filter respectively. In addition, as long as the input channel is enabled, the input channel will immediately start receiving serial data.

HPDF can enter stop mode by clearing FLTEN during operation. After entering stop mode, the ongoing conversion tasks of the HPDF module will immediately stop, and the configuration of the registers remains unchanged (except for the HPDF_FLTySTAT and HPDF_FLTyTMSTAT registers are reset).

In stop mode, the HPDF system clock will automatically stop. The HPDFEN bit must be cleared before the system clock is stopped to enter stop mode.

Low power mode

HPDF module optimizes the reduction of power consumption. In the normal working mode, the filter and integrator will automatically enter the idle state to achieve the purpose of reducing power consumption when there is no conversion task.

34.3.3. HPDF clock

The clock of HPDF includes the system clock and the serial clock. The system clock is used to drive the internal modules, and the serial clock used by the serial interface.

System clock

The system clock $f_{HPDFCLK}$ of HPDF is used to drive channel transceiver, digital filter, integrator, threshold monitor, malfunction monitor, extremum detector and control module. The HPDF system clock source can be configured by the HPDFSEL bit in the RCU_CFG1 register of the RCU chapter.

Serial input clock

The serial interface of HPDF can receive clock signal from external sigma delta modulator by CKINx pin, so as to receive the serial data stream from sigma delta modulator.

Using external input clock in serial interface is limited by clock frequency. If the standard SPI interface is used, the system clock $f_{HPDFCLK} \geq 4f_{CKIN}$. If the Manchester coding interface is used, the system clock $f_{HPDFCLK} \geq 6f_{CKIN}$ is required.

Serial output clock

HPDF supports the function of outputting serial clock, which can drive sigma delta modulator connected with it. The source of the serial output clock can be selected by CKOUTSEL bit in HPDF_CH0CTL register. When CKOUTSEL=0, the serial output clock source is the HPDF system clock. When CKOUTSEL=1, the serial output clock source is the audio clock. And the configuration of the audio clock can refer to the HPDFAUDIOSEL[1:0] bit field in the RCU_CFG1 of the RCU chapter.

After the serial output clock source is determined, the output clock frequency division can be controlled by configuring the CKOUTDIV [7:0] bit field in the HPDF_CH0CTL register. When CKOUTDIV[7:0] ≠ 0, the value of the serial output clock divider is CKOUTDIV[7:0]+1. When CKOUTDIV[7:0] = 0, the serial output clock is disabled and the pin of CKOUT remains low.

In addition, after clearing HPDFEN, the signal of serial output clock can also be stopped. When the serial output clock source is the system clock (CKOUTSEL = 0), if clear HPDFEN, the serial output clock stopped after 4 system clocks. When the serial output clock source is the audio clock (CKOUTSEL = 1), if clear HPDFEN, the serial output clock stopped after one system clock and three audio clocks.

The serial output clock source can only be modified when HPDFEN = 0. In order to avoid the burr signal on the pin of CKOUT, the software can only modify the value of the CKOUTSEL bit in the HPDF_CH0CTL register after the serial output clock stopped.

The frequency range of the serial output clock is 0-20MHz.

34.3.4. Multiplex serial data channel

HPDF has eight multiplexing serial data channels, which support SPI code and Manchester code. The interface type supported can be selected for the current channel by configuring the SITYP[1:0] bit field in the HPDF_CHxCTL register.

SPI interface

Under the standard SPI interface, sigma delta modulator sends 1-bit data stream to the serial channel by the pin of DATAINx. The clock signal between HPDF and sigma delta modulator can be output by CKOUT pin or input by CKINx pin.

The data sampling point in SPI communication is determined by the SITYP[1:0] bit field and

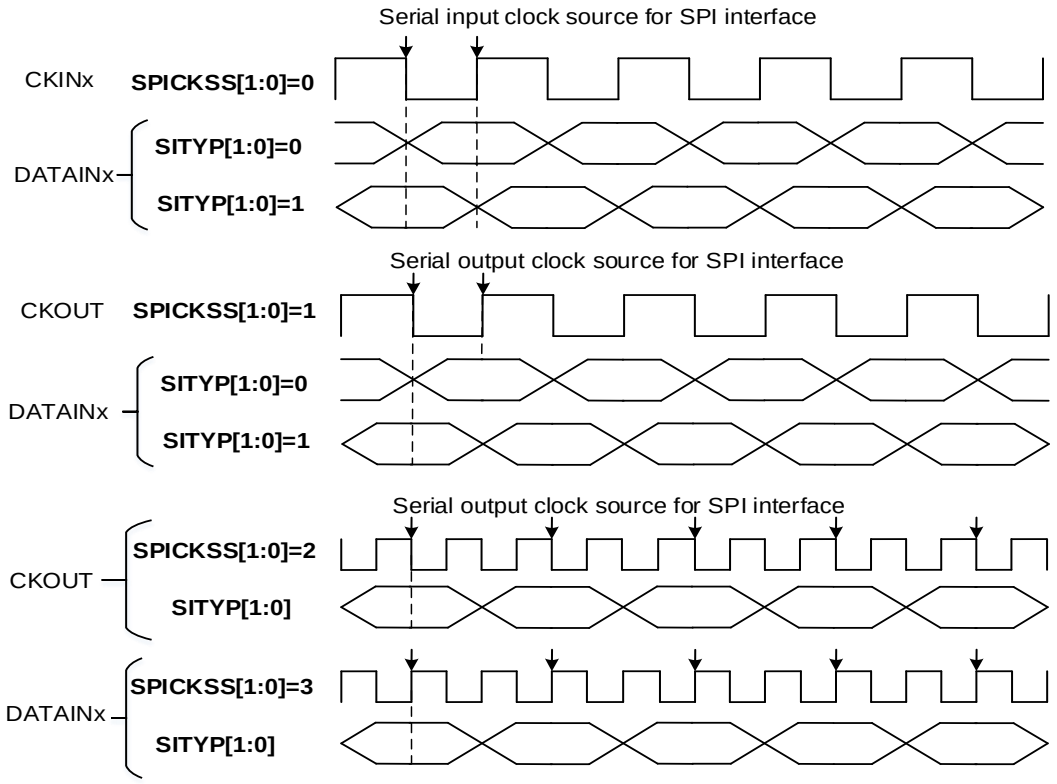
SPICKSS[1:0] bit field in HPDF_CHxCTL register. The data sampling points in SPI communication are shown in the table.

Table 34-3. SPI interface clock configuration

SPICKSS[1:0]	Clock source	SITYP[1:0]	Sampling point	Description
00	CKINx signal	00	rising edge	Data is sampled at the rising edge of the external serial input clock signal
		01	falling edge	Data is sampled at the falling edge of the external serial input clock signal
01	CKOUT signal	00	rising edge	The data is sampled at the rising edge of the internal serial output clock signal
		01	falling edge	The data is sampled at the falling edge of the internal serial output clock signal
10	CKOUT/2 signal (Generated at the rising edge of CKOUT)	xx	Falling edge of each second CKOUT signal	The external sigma delta modulator divides the CKOUT signal into 2 frequencies to generate the serial input communication clock. The data is sampled at the falling edge of every second CKOUT.
11	CKOUT/2 signal (Generated at the falling edge of CKOUT)	xx	Rising edge of each second CKOUT signal	The external sigma delta modulator divides the CKOUT signal into 2 frequencies to generate the serial input communication clock. The data is sampled at the rising edge of every second CKOUT.

According to ***Table 34-3. SPI interface clock configuration***, the sequence diagram of SPI data transmission is shown in the figure below.

Figure 34-2. The sequence diagram of SPI data transmission



Note: if SPI data interface is adopted, the frequency range of clock source is 0-20MHz and less than $f_{HPDFCLK}/4$.

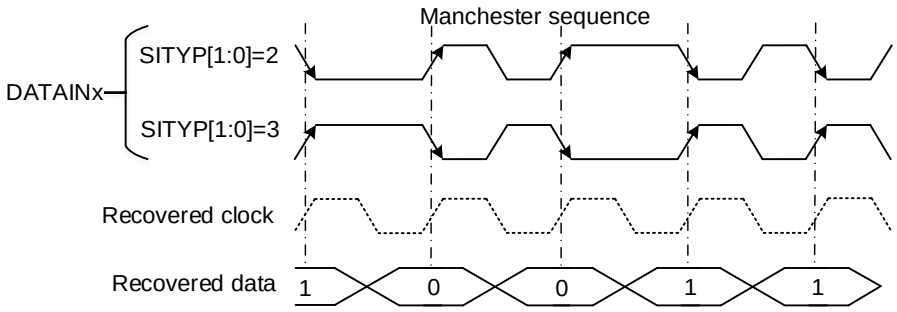
Manchester interface

HPDF has eight multiplexing serial data channels using Manchester encoding format. Two encoding formats can be configured by SITYP[1:0] bit field in HPDF_CHxCTL:

1. When SITYP[1:0] = 2, Manchester code: rising edge = logic 0, falling edge = logic 1.
2. When SITYP[1:0] = 3, Manchester code: rising edge = logic 1, falling edge = logic 0.

When Manchester code is used, the data stream between the external sigma delta modulator and HPDF is only transmitted by the DATAINx pin. After the HPDF module Manchester decoding, the clock signal and data are recovered from the serial data stream. The recovered clock signal frequency must be between 0-10MHz and less than $f_{HPDFCLK}/6$. The timing chart of Manchester data transmission is shown in the figure below.

Figure 34-3. The sequence diagram of Manchester data transmission



In order to receive and decode Manchester data correctly, configure the CKOUTDIV[7:0] frequency divider according to the expected flow rate of Manchester data. The value of CKOUTDIV[7:0] is calculated with reference to the following format:

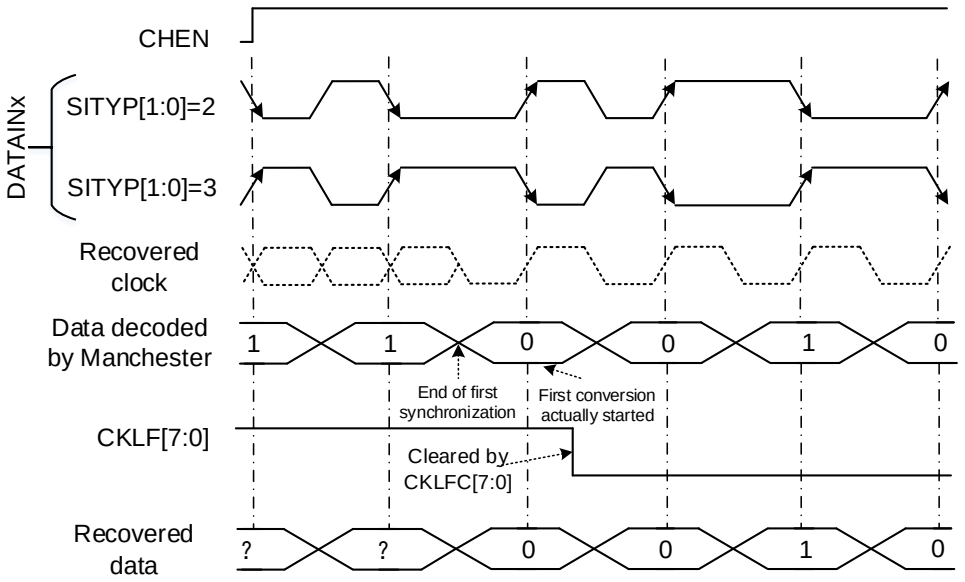
$$((CKOUTDIV+1)xT_{SYSCLK})<T_{Manchester_clock}<(2xCKOUTDIV\times T_{SYSCLK}) \quad (34\text{-}1)$$

Serial communication coding synchronization

After the serial channel is enabled, the data can only be received correctly after successful synchronization. The synchronization of SPI code occurs after the first detection of clock input signal by SPI data stream. If the channel uses Manchester coding, the first synchronization occurs when the channel receives data stream changes from 1-0 or 0-1.

Before the transceiver of the serial channel synchronizes, the clock loss flag bit of channel is set to 1. After successful synchronization, the clock loss flag bit can be cleared by CKLFC[7:0]. When the transceiver of the serial channel is not synchronized, the clock loss flag bit cannot be cleared by the CKLFC[7:0]. Therefore, it is possible to determine whether the serial channel is successfully synchronized by querying the CKLF[7:0] bit circularly. The following figure shows the timing chart of the first synchronization of Manchester code.

Figure 34-4. Manchester synchronous sequence diagram



External serial clock frequency measurement

The measuring of a channel serial clock input frequency provides a real data rate from an external ΣΔ modulator, which is important for application purposes.

An external serial clock input frequency can be measured by a timer counting HPDF clocks ($f_{HPDFCLK}$) during one conversion duration. The counting starts at the first input data clock after a conversion trigger (regular or injected) and finishes by last input data clock before conversion ends (end of conversion flag is set). Each conversion duration (time between first serial sample and last serial sample) is updated in counter CNVCNT[27:0] in register HPDF_FLTxCT when the conversion finishes (ICEF=1 or RCEF=1). The user can then compute the data rate according to the digital filter settings (SFO, SFOR, IOR, FAST). The external serial frequency measurement is stopped only if the filter is bypassed (SFOR=0, only integrator is active, CNVCNT[27:0]=0 in HPDF_FLTxCT register).

In case of parallel data input the measured frequency is the average input data rate during one conversion.

Note: When conversion is interrupted (by disabling/enabling the selected channel) the interruption time is also counted in CNVCNT[27:0]. Therefore it is recommended to not interrupt the conversion for correct conversion duration result.

Conversion times:

injected conversion or regular conversion with FAST = 0 (or first conversion if FAST=1):

for Sincx filters:

$$T = CNVCNT / f_{HPDFCLK} = [SFOR * (IOR-1 +SFO) + SFO] / f_{CKIN}$$

for FastSinc filter:

$$T = CNVCNT / f_{HPDFCLK} = [SFOR * (IOR-1 +4) + 2] / f_{CKIN})$$

regular conversion with FAST = 1 (except first conversion):

for Sincx and FastSinc filters:

$$T = CNVCNT / f_{HPDFCLK} = [SFOR * IOR-1] / f_{CKIN})$$

in case if FOSR = FOSR[9:0]+1 = 1 (filter bypassed, active only integrator):

$$T = IOR / f_{CKIN} \text{ (but CNVCNT=0)}$$

where:

- f_{CKIN} is the channel input clock frequency (on given channel CKINx pin) or input data rate (in case of parallel data input).
- SFOR is the filter oversampling ratio: SFOR = SFOR[9:0]+1 (see HPDF_FLTxSFCFG register).
- IOR is the integrator oversampling ratio: IOR = IOR[7:0]+1 (see HPDF_FLTxSFCFG register).

- SFO is the filter order: SFO = SFO[2:0] (see HPDF_FLTxSFCFG register)

Clock loss detection

Clock loss detection is to detect whether the channel serial input clock (CKINx signal) is lost, so as to ensure whether there is any error in the data of serial channel conversion (or threshold monitor and malfunction monitor). If a clock signal loss event occurs, the given data should be discarded. When using the clock loss detection function, you must configure the ckout signal source as the system clock.

The clock loss detection function can be enabled or disabled by the CKLEN bit in HPDF_CHxCTL register. When the enable clock loss detection function and the clock loss interrupt CKLIE occur, if a clock loss event occurs, the clock loss flag bit (CKLF) will be set to 1 and a clock loss interrupt will be generated. The corresponding interrupt flag bit can be cleared by setting CKLFC[7:0] bit field.

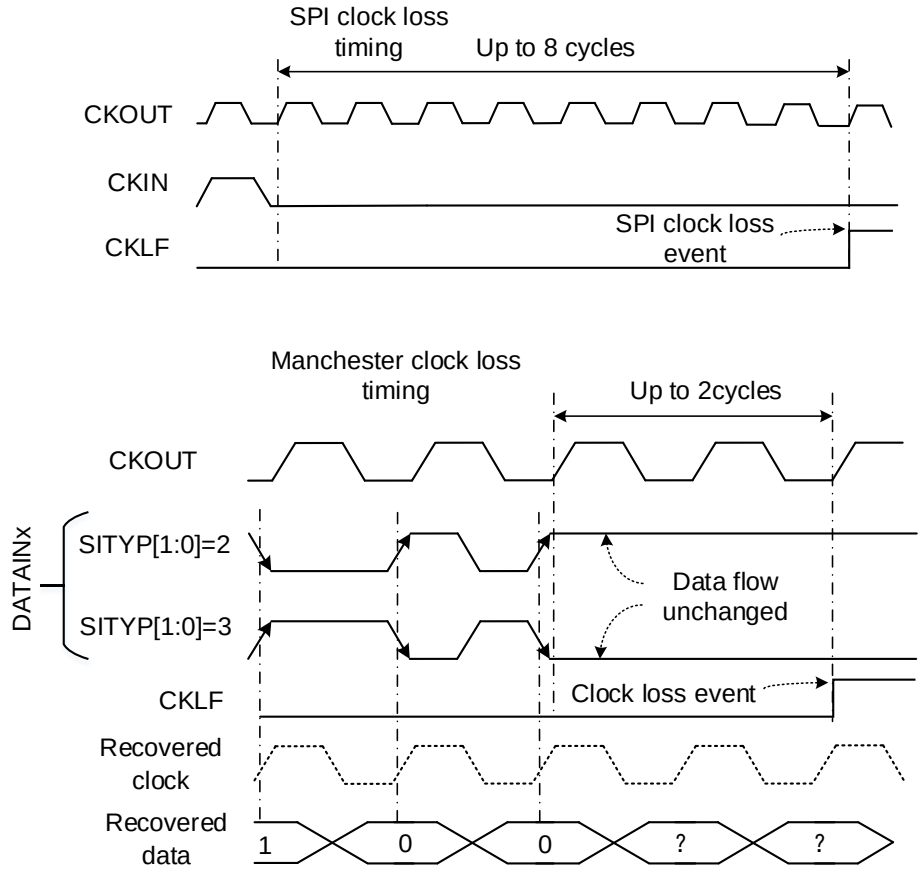
When the transceiver of the serial interface has not been synchronized, the clock loss flag bit is set to 1 and cannot be cleared by the corresponding CKLFC[7:0]. Therefore, the correct steps to use the clock loss function are as follows:

1. Enable the channel CHEN = 1.
2. Check the clock loss flag cyclically and write 1 to the CKLFC of the channel. When it is confirmed that the CKLF bit is cleared, the serial channel transceiver synchronization is successful.
3. Enable clock loss detection function CKLEN = 1. To detect possible clock loss, enable clock loss interrupt CKLIE = 1.

If the SPI interface is used in the serial channel, the external serial input clock (CKINx signal) is compared with the serial output clock (ckout signal) when the clock loss detection function is used. The external serial input clock signal must be inverted at least once every 8 CKOUT signal cycles, otherwise a clock loss event will occur.

If the serial channel uses the Manchester interface, the clock loss detection starts after the first successful synchronization of the Manchester code, and the external serial input data (DATAINx signal) is compared with the serial output clock (CKOUT signal). The serial input data must change every 2 ckout signal cycles, otherwise clock loss event will be generated. The timing of clock loss is shown in the figure below.

Figure 34-5. Clock loss detection timing diagram



Note: the maximum rate of Manchester encoded data stream must be less than the clock output CKOUT signal.

Channel pin redirection

Channel pin redirection means that the pins of serial channel 0 can be configured as the pins of channel 1, that is, channel 0 can read information from the DATAIN1 and CKIN1 pins. Pin redirection is used to sampling audio data of PDM microphone. The audio signal of PDM microphone includes data and clock signal. The data is divided into left/right channel data. The left channel data is sampled at the rising edge of clock signal, and the right channel data is sampled at the falling edge of clock signal.

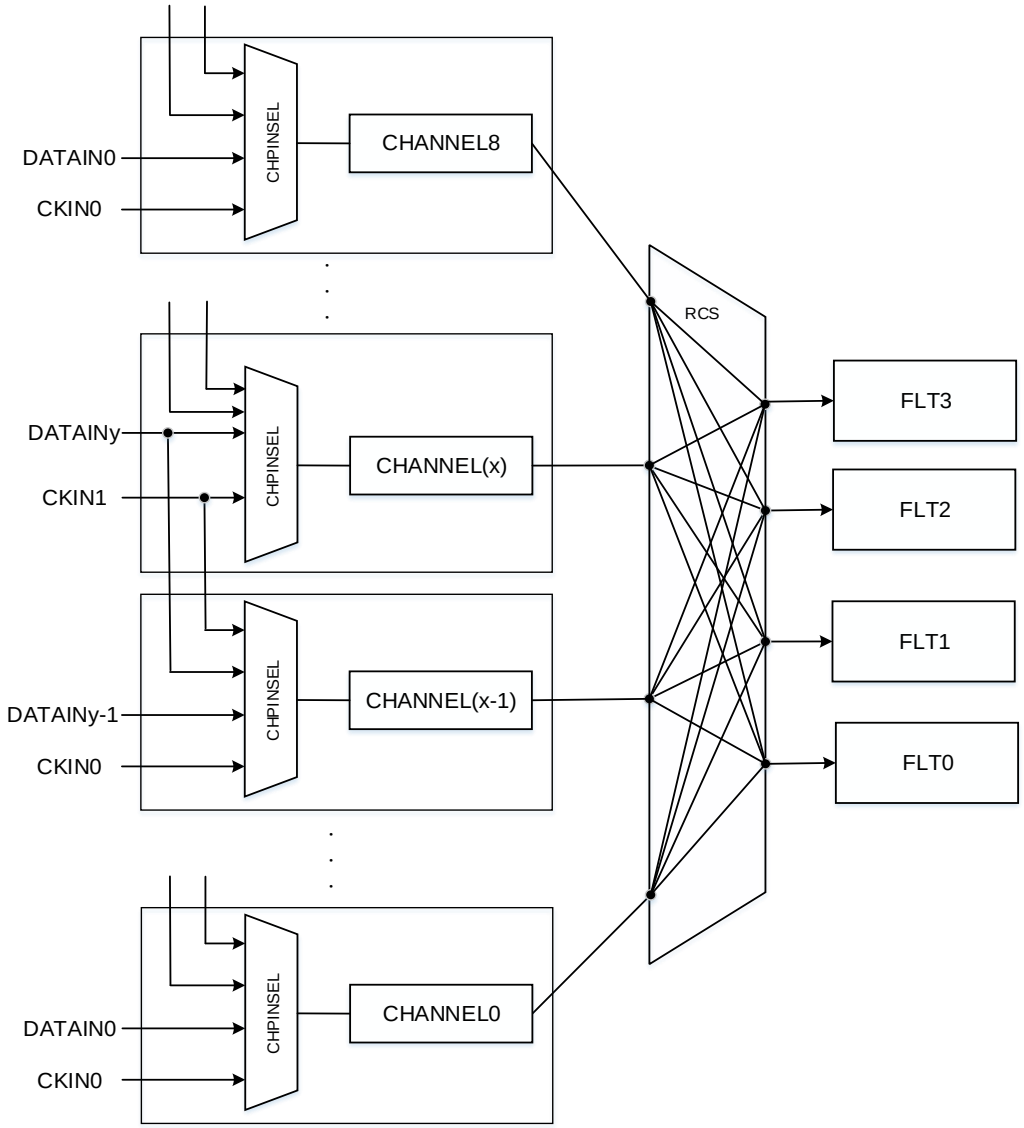
When PDM microphone data stream is input into serial channel, its configuration process is as follows:

1. Select the HPDF serial channel 1 of PDM microphone data stream input.
2. Write 0 to CHPINSEL bit of channel 1 in HPDF_CHxCTL register, and input pin of channel 1 is own pin, DATAINx and CKINx. When SITYP[1:0] = 2b’00, the serial data stream is sampled at the rising edge of the clock signal, that is, the input of channel 1 is the left channel data.
3. Set the CHPINSEL to 1 in channel 0, and the DATAINx and CKINx pins will be used for channel 0. When SITYP[1:0] = 2b’01, the serial data stream is sampled at the falling edge of the clock signal, that is, the input of channel 0 is the right channel data.

4. Configure channelx with corresponding filters to filter the left and right channel data of PDM microphone.

The channel pin redirection diagram of HPDF module is shown in ***Figure 34-6. Channel pins redirection.***

Figure 34-6. Channel pins redirection



Pulses skipper

Pulse skipper refers to that the serial input data stream enters the filter after skipping a specified number of clock pulses, so as to discard a certain number of bit bits. This operation will cause the final output sample (and the next sample) from the filter to be calculated from the subsequent input data compared to the data stream that was not skipped.

The number of pulses to be skipped is determined by the PLSK[5:0] bit field in the HPDF_CHxPS register. Write the value to PLSK[5:0] bit field, and the specified channel will start to perform the pulse skipper function. Read the of PLSK[5:0], indicating the number of

remaining pulse skipper not executed. For a single write operation of PLSK[5:0], the maximum number of pulse skipper executed is 63. More pulse skipper can be obtained by writing to PLSK[5:0] bit field several times.

Serial input interface configuration

The configuration steps of serial input interface of HPDF module are as follows:

1. Configure clock output prescaler: by configuring the CKOUTDIV[7:0] bit field in the HPDF_CH0CTL register, the coefficient of prescaler is CKOUTDIV[7:0] + 1.
2. Configure the serial interface type and input clock phase: configure the serial interface type as SPI code or Manchester code, and determine the clock input sampling edge by the SITYP[1:0] bit field in HPDF_CHxCTL register.
3. Configure input clock source: select the clock source of serial interface as serial input clock or serial output clock by configuring SPICKSS[1:0] in HPDF_CHxCTL register.
4. Configure data offset correction and shift right bits: DTRS[4:0] defines the bits of the final data shift right in HPDF_CHxCFG register. After data shift, perform offset calibration defined by CALOFF [23:0] bit field.
5. Enable short circuit detection and clock loss detection function: enable short circuit detection and clock loss detection function by setting MMEN and CKLEN to 1.
6. Set the threshold monitor filter and malfunction monitor: the filter parameters of the threshold monitor, the malfunction signal allocation of the malfunction monitor and the counter threshold are all configured by the HPDF_CHxCFG1 register.

34.3.5. Parallel data input

HPDF module can select parallel data as the data input source of the channel. The CMSD[1:0] bit field in HPDF_CHxCTL is configured to determine whether the channel data input source is from serial data or parallel data. Each channel provides a 32-bit parallel data input register (HPDF_CHxPDI), which can write two 16 bit parallel data by CPU/DMA. The register has two 16-bit data in signed format.

Input from internal ADC

For parallel ADC data input (CMSD[1:0]), The ADC[x] result is assigned to channel x input. The end of conversion event from ADC[x] causes the data of channel x to be updated (the parallel data from ADC[x] is used as the next sampling for the digital filter). When the end of conversion event occurs, data from ADC[x] is written to the HPDF_CHxPDI register (DATAIN0[15:0]).

Data packing mode setting (DPM[1:0] in register HPDF_CHxCTL) has no effect on ADC data input.

CPU / DMA write parallel data

There are two ways to write parallel data: CPU direct write and DMA write. When using DMA

to write parallel data, DMA should be configured as memory to memory mode, and its target address is the address of HPDF_CHxPDI.

Note: DMA writing parallel data is different from DMA reading final conversion data from HPDF module. The latter needs to be configured in peripheral to memory mode.

Parallel data packed mode

The data stored in HPDF_CHxPDI register will be processed by channel filter. There are three modes of parallel data stored in the HPDF_CHxPDI register. In different data packed modes, the number of filter samples allowed to load depends on the value of DPM [1:0] bit field in the HPDF_CHxCTL register. The different data encapsulation modes are as follows:

1. Standard mode (DPM[1:0] = 2'b00):

   In this mode, the upper 16 bits in the HPDF_CHxPDI register are write protected, and the 16-bit data written by CPU / DMA is stored in the low 16 bit DATAIN0[15:0] bit field. CPU / DMA is configured as a 16-bit access mode. When writing 16-bit data once, the channel filter must perform an input sampling to clear the HPDF_CHxPDI register.

2. Interleaving mode (DPM[1:0] = 2'b01):

   In this mode, the CPU/DMA is configured as a 32-bit access mode, and the data is stored in the DATAIN0[15:0] bit domain of the lower 16 bits and the DATAIN1[15:0] bit domain of the higher 16 bits. When writing 32-bit data once, the channel filter must perform two input samples to clear the HPDF_CHxPDI register. The channel filter samples the DATAIN0[15:0] bit domain for the first time and the DATAIN1[15:0] bit domain for the second time.

3. Dual channel mode (DPM[1:0] = 2'b10):

   In this mode, the CPU / DMA is configured as a 32-bit access mode, and the data is stored in the DATAIN0[15:0] bit domain of the lower 16 bits and the DATAIN1[15:0] bit domain of the higher 16 bits. The data in the DATAIN0[15:0] bit field is used for the current channel x, and the data in the DATAIN1[15:0] bit field is automatically copied to the lower 16 bits of the parallel data input register of the channel x+1, and the data is used for the channel x+1. CPU / DMA writes data once, digital filter performs two sampling, the first is channel x sampling, the second is channel x+1 sampling.

In HPDF module, only even channel (channel0) supports dual channel mode. If odd channel (channel1) is configured as dual channel mode, the parallel data input register HPDF_CHxPDI of this channel is write protected. If channel x is even and configured as dual channel mode, odd channel x+1 must be configured as standard mode.

The operation mode of HPDF_CHxPDI register is as follows:

Table 34-4. Parallel data packed mode

Channel	Packed mode					
	Standard mode		Interleaving mode		Dual channel mode	
	DATAIN1	DATAIN0	DATAIN1	DATAIN0	DATAIN1	DATAIN0
Channel0	Write protect	CH0 sampling	CH0 second sampling	CH0 first sampling	CH1 sampling	CH0 sampling
Channel1	Write protect	CH1 sampling	CH1 second sampling	CH1 first sampling	Write protect	CH1 sampling
Channel2	Write protect	CH2 sampling	CH2 second sampling	CH2 first sampling	CH3 sampling	CH2 sampling
Channel3	Write protect	CH3 sampling	CH3 second sampling	CH3 first sampling	Write protect	CH3 sampling
Channel4	Write protect	CH4 sampling	CH4 second sampling	CH4 first sampling	CH5 sampling	CH4 sampling
Channel5	Write protect	CH5 sampling	CH5 second sampling	CH5 first sampling	Write protect	CH5 sampling
Channel6	Write protect	CH6 sampling	CH6 second sampling	CH6 first sampling	CH7 sampling	CH6 sampling
Channel7	Write protect	CH7 sampling	CH7 second sampling	CH7 first sampling	Write protect	CH7 sampling

CPU / DMA should write to HPDF_CHxPDI register after the channel is enabled, because after the channel is enabled, the channel conversion will be started, and the data in HPDF_CHxPDI register will be discarded before the channel conversion is started.

34.3.6. Regular group conversion

HPDF module has two multiplexing channels, which can be used for regular group conversion or inserted group conversion respectively. If the channel is disabled (CHEN = 0), enabling the channel conversion will cause the channel to remain in the conversion state. The channel can be restored only by enabling the channel (CHEN=1) or disabling the HPDF module (HPDFEN=0).

The regular group selects only one of the two channels, which is determined by the RCS bit in the HPDF_FLTyCTL0 register. At the same time, only one regular conversion can be executed or pending. If an existing regular conversion request has not been completed, the new regular conversion start request is ignored. The priority of regular conversion is lower than that of inserted group conversion and can be interrupted by inserted group conversion request.

The conversion time of regular group: t = CTCNT[27:0] / $f_{HPDFCLK}$.

Conversion start mode

Regular group conversion can only be achieved by software startup. There are two modes of

software startup, the specific methods are as follows:

1. General software startup: write 1 to SRCs bit in HPDF_FLTyCTL0 register.
2. Software synchronous start: Set the RCSYN bit in HPDF_FLTyCTL0 register and start the regular conversion of HPDF_FLT0 by general software startup. Then HPDF_FLTy also starts the regular conversion synchronously.

Conversion mode

Regular group transformation supports continuous mode and fast mode.

Continuous mode

Set the RCCM bit to 1 in HPDF_FLTyCTL0 register to enable continuous mode. In continuous mode, after the software starts the regular group conversion, the conversion regular group channel conversion is repeated. When the RCCM bit is cleared, the regular conversion in continuous mode stops immediately.

Fast mode

Enable fast mode by setting FAST bit to 1 in HPDF_FLTyCTL0. In fast mode, it can improve the data rate in continuous mode. Because in continuous mode, if the data is continuously converted from one channel, there is no need to fill the filter with new data, because the data in the filter is valid data sampled from the previous continuous mode. The increase in data rate is determined by the order of the selected filter.

After the continuous conversion is started, the time for the first conversion of the fast mode to the non open fast mode is the same, and then the subsequent conversion will be completed at a shorter time interval.

34.3.7. Inserted group conversion

The conversion channel of the inserted group must select at least one of the two channels. You can select which channel to convert into the inserted group by the ICGSEL[7:0] bit field in the HPDF_FLTyICGS register. ICGSEL[x]=1 means channel x is the inserted group channel.

The priority of the inserted group is higher than that of the regular group. The ongoing regular group conversion will be interrupted by the inserted group conversion request. Wait for the inserted group to complete the conversion and restart the interrupted regular conversion. At the same time, only one inserted conversion is in execution or pending state. If an existing inserted conversion request has not been completed, the new inserted conversion start request will be ignored.

The conversion time of the inserted group t=CTCNT[27:0] * $f_{HPDFCLK}$.

Conversion start mode

The conversion of the inserted group can be achieved through channel software startup and

trigger startup.

1. General software startup: write 1 to the SICC bit in HPDF_FLTyCTL0 register.
2. Software synchronous startup: Set the ICSYN bit in HPDF_FLTyCTL0 register to start synchronously. When using general software to start the inserted group conversion of HPDF_FLT0, the channel 1 that enables the synchronous start function also starts the inserted conversion.
3. Trigger startup: When the ICTSSEL[4:0] bit field in the HPDF_FLTyCTL0 register is written with a value other than 0, it indicates that trigger start is enabled and trigger signal source is selected at the same time. The effective edge of the trigger is determined by the ICTEEN[1:0] bit field.

The trigger signals of the inserted group are shown in the following table:

Table 34-5. Trigger signal of inserted group

Trigger signal	Signal source
HPDF_ITRG0	TIMER0_TRGO0
HPDF_ITRG1	TIMER0_TRGO1
HPDF_ITRG2	TIMER7_TRGO0
HPDF_ITRG3	TIMER7_TRGO1
HPDF_ITRG4	TIMER2_TRGO0
HPDF_ITRG5	TIMER3_TRGO0
HPDF_ITRG6	TIMER15_CH1
HPDF_ITRG7	TIMER5_TRGO0
HPDF_ITRG8	TIMER6_TRGO0
HPDF_ITRG[9~10]	Reserved
HPDF_ITRG11	Reserved
HPDF_ITRG12	Reserved
HPDF_ITRG[13~23]	Reserved
HPDF_ITRG24	EXTI11
HPDF_ITRG25	EXTI15
HPDF_ITRG26	-
HPDF_ITRG27	-
HPDF_ITRG28	-
HPDF_ITRG[29~30]	Reserved
HPDF_ITRG31	HPDF_ITRG

Scan conversion mode

By setting the SCMOD bit in HPDF_FLTyCTL0 register, the scan conversion mode for inserted group conversion can be enabled. In the scan mode, when the inserted group conversion is triggered, all channels in the inserted group will be converted sequentially starting from the lowest channel.

If the scan mode is disabled, each time the inserted group conversion is triggered, only one

channel in the inserted group will be converted, and the next trigger will select another channel. And writing to the ICGSEL[7:0] bit field will use the lowest channel as the selected conversion channel.

Conversion request priority

The conversion of the inserted group has a higher priority than the regular group. The regular conversion that is already in progress will be immediately interrupted by the inserted conversion request. When the inserted conversion sequence ends, if RCCM remains at 1, the continuous regular conversion will start again. The value of the RCHPDT bit indicates that the interrupted regular conversion is delayed.

If an inserted conversion is pending or already in progress, you cannot start other inserted conversions: as long as ICPF=1, any request to launch an inserted conversion (software or trigger start) will be ignored. The regular conversion is the same.

When the inserted conversion is in progress (ICPF=1), write 1 to the SRCS bit of HPDF_FLTyCTL0 to request regular conversion. When the inserted sequence is completed, the priority indicates the next step to perform regular conversion, and the delayed start is indicated by the RCHPDT bit.

34.3.8. Digital filter

The digital filter of the HPDF module is of $Sinc^X$ type. The input data stream is filtered by $Sinc^X$, thereby reducing the output data rate and increasing the output data resolution. Configure the order and oversampling rate (decimation filtering) of the $Sinc^X$ filter by the SFO[2:0] and SFOR[9:0] bits in the HPDF_FLTySFCFG register. The user can configure the order and oversampling rate of the $Sinc^X$ filter according to the desired resolution. The relationship between the maximum output resolution of $Sinc^X$ filtering and oversampling filtering is as follows:

Table 34-6. The relationship between the maximum output resolution and oversampling filtering of SincX filtering

SFOR	Sinc	$Sinc^2$	FastSinc	$Sinc^3$	$Sinc^4$	$Sinc^5$
x	±x	$\pm x^2$	$\pm 2x^2$	$\pm x^3$	$\pm x^4$	$\pm x^5$
4	±4	±16	±32	±64	±256	±1024
8	±8	±64	±64	±512	±4096	±32768
32	±32	±1024	±2048	±32768	±1048576	±33554432
64	±64	±4096	±8192	±262144	±16777216	±1073741824
128	±128	±16384	±32768	±2097152	±268435456	-
256	±256	±65536	±131072	±16777216	Under full-scale input conditions, the result will overflow	
1024	±1024	±1048576	±2097152	±1073741824		

Note: The maximum output resolution in this table comes from the peak data value of the filter output.

34.3.9. Integrator

The integrator performs further oversampling rate (decimation rate) and resolution improvement on the data from the digital filter. The integrator performs a simple summation operation on a given number of data samples from the filter. The output data of the integrator comes from the sum of the output samples of the filter, and the number of output samples is determined by the oversampling rate of the integration. The oversampling rate (decimation filtering) of the integrator can be configured by IOR[7:0] in HPDF_FLTySFCFG register. The relationship between the maximum output resolution, oversampling rate, and Sinc filter order of the integrator is as follows:

Table 34-7. Relationship between the maximum output resolution and IOR, SFOR, SFO of the integrator

Filter type	Integrator maximum output resolution
Sinc	±(SFOR×IOR)
$Sinc^2$	±($SFOR^2$×IOR)
FastSinc	±(2$SFOR^2$×IOR)
$Sinc^3$	±($SFOR^3$ ×IOR)
$Sinc^4$	±($SFOR^4$×IOR)
$Sinc^5$	±($SFOR^5$×IOR)

34.3.10. Threshold monitor

The threshold monitor of the HPDF module is used to monitor the serial input data of the channel or the final output data after the channel conversion. When the data reaches the threshold set by the threshold monitor (maximum or minimum threshold), an interrupt or break event will be generated. The maximum threshold is determined by the HTVAL[23:0] bits in HPDF_FLTyTMHT register, and the minimum threshold is determined by the LTVAL[23:0] bits in HPDF_FLTyTMLT register.

The HPDF module has four threshold monitor. By configuring the TMCHEN[7:0] bit field in HPDF_FLTyCTL1 register, it determines whether the analog threshold monitor x monitors the input channel. For example, TMCHEN[1]=1 in HPDF_FLT0CTL1 register means threshold monitor 0 monitors channel 1.

Threshold monitor working mode

The working mode of the threshold monitor is divided into standard mode and fast mode. Fast mode is to configure the serial input data of the threshold monitor monitoring channel and compare it with the threshold. Standard mode is to configure the final data output after the threshold monitor monitor channel conversion (stored in the inserted group data register HPDF_FLTyIDATA or the regular group data register HPDF_FLTyRDATA). The fast mode of the threshold monitor can be enabled by the TMFM bit in HPDF_FLTyCTL0. The

characteristics in both cases are as follows:

Table 34-8. Features of threshold monitor working mode

Mode	Enable Bit	Channel Data Source	Analog Input Data Source	Input Data Resolution	Detailed Description
Standard mode	TMFM=0	Serial data stream, Parallel data	HPDF final data output	24bit	The threshold monitor monitors the final data output after the channel conversion. Slow response time, not suitable for overcurrent/overvoltage detection
Fast mode	TMFM=1	Serial data stream	Serial data stream	16bit	The input data is provided in continuous mode, and the threshold monitor directly monitors the serial input data, regardless of rules or injection conversion. Fast response time, suitable for overcurrent/overvoltage detection.

In fast mode, the threshold monitor uses only the upper 16 bits of the threshold (maximum threshold HTVAL[23:0] or minimum threshold LTVAL[23:0]) to compare with the serial input data of the channel, that is, only the upper 16 bits of HTVAL[23:0] and LTVAL[23:0] define the threshold, because the resolution of the threshold monitor filter is 16 bits.

In non-fast mode of threshold monitor, the final data of right shift and offset calibration will be compared with HTVAL[23:0] and LTVAL[23:0].

Threshold monitor fast mode

In fast mode, the filter of the threshold monitor will be used, and the oversampling rate (decimation rate) and order of the threshold monitor filter can be set in HPDF_CHxCFG1 register.

The configuration of the threshold monitor is flexible. An threshold monitor can be configured to monitor multiple channels by the TMCHEN[7:0] bit field in HPDF_FLTyCTL1 register. In this case, when multiple channels send out requests, the threshold monitor will preferentially process requests with small channel numbers, and then process requests with large channel numbers. Each threshold monitor has a status register HPDF_FLTyTMSTAT. When the monitored channel exceeds the threshold, the corresponding flag in the HTF[7:0] or LTF[7:0] bit field will be set. If HTF[0]=2b’01, it means that channel 0 occured an event that exceeds

the upper threshold.

After each channel sends a comparison request, it will be executed within 8 HPDF clock cycles. Therefore, the bandwidth of each channel is limited to 8 HPDF clock cycles (if TMCHEN[7:0]=255). Since the maximum sampling frequency of the input channel is $f_{HPDFCLK}/4$, at this input clock speed, the threshold monitor filter cannot be bypassed (TMFOR=0). Therefore, the user must correctly configure the threshold monitor filter parameters and the number of channels monitored based on the input sampling clock speed and $f_{HPDFCLK}$.

In fast mode, reading the TMDATA[15:0] bit field in HPDF_CHxTMFDT register to get the threshold monitor filter data for the given channel x. The number of serial samples required for a result of the threshold monitor filter output (at the serial input clock frequency f_{CKIN}) is as follows:

1. First conversion:

   FastSinc filter: Number of samples is equal to (TMSFO×4+ 2+1).

   $Sinc^X$ filter (x=1..5): Number of samples is equal to (((TMSFO[1:0]+1)×TMFOR)+ TMSFO+1).

2. Subsequent conversions other than the first conversion:

   FastSinc and $Sinc^X$ filter (x=1…5): Number of samples is equal to (TMSFO[1:0]+1)×(IOR[7:0]+1).

Threshold monitor flag

The global state of the threshold monitor is the TMEOF flag in HPDF_FLTySTAT register. When TMEOF=1, it indicates that at least one threshold monitor event has occurred, that is, an event that exceeds the (upper/lower limit) threshold is generated. If the threshold monitor event interrupt TMIE=1 in HPDF_FLTyCTL1 register is enabled, an threshold monitor interrupt can be generated. When all HTF[7:0] and LTF[7:0] are cleared, the TMEOF bit is cleared.

The HPDF_FLTyTMSTAT register defines the error event flag of the channel exceeding the threshold. The HTF[1:0] bit field indicates whether the maxmum threshold HTVAL[23:0] has been exceeded on the channel x. The LTF[7:0] bit field indicates whether the minimum threshold LTVAL[23:0] value has been exceeded on channel x. Clear the threshold event flag by writing "1" to the corresponding HTFC[7:0] or LTFC[7:0] bit in the HPDF_FLTyTMFC register.

As is shown in ***Table 34-2. HPDF internal signal***, there are 4 break output signals in the HPDF module. The break output signals are assigned to threshold monitor threshold event by setting the HTBSD[3:0] and LTBSD[3:0] bit fields in the HPDF_FLTyTMHT register and the HPDF_FLTyTMLT register.

34.3.11. Malfunction monitor

The purpose of the malfunction monitor is to be able to send a signal with an extremely fast response time when the analog signal reaches a saturation value and remains in this state for a given time. This feature can be used to detect short-circuit or open-circuit faults (e.g. overcurrent/overvoltage). The broken output signals can be assigned to the malfunction monitor event, which can be configured by the MMBSD[3:0] bit field in the HPDF_CHxCFG1 register. The broken output signal is the same as the threshold monitor.

The input data of the malfunction monitor comes from the serial input data of the channel.When the channel input data source is parallel data, the malfunction monitor function is prohibited.There is an up counter on each input channel to record how many consecutive 0 or 1 on the output of the serial data receiver. When the counter reaches the threshold value of the malfunction monitor (MMCT[7:0] bits in the HPDF_CHxCFG1 register), a malfunction event occurs. If a 0-1 or 1-0 change is encountered when monitoring the data stream, the value of the counter will be automatically cleared and counted again.

The user can enable the malfunction monitor function by setting the MMEN bit in HPDF_CHxCTL register. When a malfunction event occurs on the channel, the corresponding malfunction monitor flag MMF[7:0] is set. The corresponding flag can be cleared by MMFC[7:0] in HPDF_FLTyINTC. If channel x is disabled (CHEN=0), the hardware will also clear the malfunction monitor flag.

34.3.12. Extremes monitor

The extremes monitor is used to sample the minimum and maximum values (peak to peak) of the final output data word. An extremes monitor can be configured to sample the extreme values of multiple channels through the EMCS[7:0] bit field in HPDF_FLTyCTL1 register.

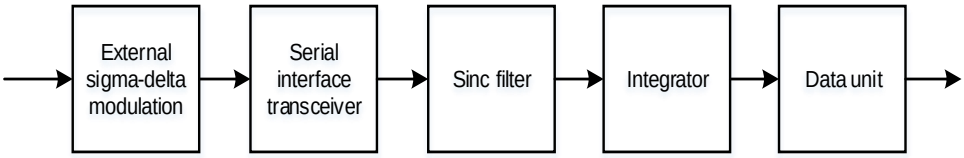
If the sampling final output data word is higher than the value in the maximum value register of the extremes monitor (MAXVAL[23:0] bits in the HPDF_FLTyEMMAX register), the value of this register is updated to the current final output data. If the sampling final output data word is smaller than the value in the minimum value register of the extremes monitor (MINVAL[23:0] bits in HPDF_FLTyEMMIN register), the value of this register is updated to the current final output data. The values of the MAXDC bit and the MINDC bit indicate which channel the maximum / minimum value comes from.

When reading the HPDF_FLTyEMMAX or HPDF_FLTyEMMIN register, the maximum or minimum value is updated with the reset value.

34.3.13. Data unit

The data unit is the last part of data processing in the entire HPDF module, and the flow of data processing by the HPDF module is shown in the following figure.

Figure 34-7. HPDF module external input data processing flow



The output data rate depends on the serial data stream rate, filter and integrator settings. The maximum output data rate is shown in the table below.

Table 34-9. Maximum output rate

Input source	Conversion mode	Filter type	Maximum output data rate (samples/second)
Serial input	Non-fast mode (FAST=0)	$Sinc^X$	$\frac{f_{CKIN}}{SFOR\times(IOR-1+SFO)+(SFO+1)}$
	Non-fast mode (FAST=0)	FastSinc	$\frac{f_{CKIN}}{SFOR\times(IOR-1+4)+(2+1)}$
	Fast mode (FAST=1)	FastSinc and $Sinc^X$	$\frac{f_{CKIN}}{SFOR\times IOR}$
Parallel input	Non-fast mode (FAST=0)	$Sinc^X$	$\frac{f_{DATA}}{SFOR\times(IOR-1+SFO)+(SFO+1)}$
	Non-fast mode (FAST=0)	FastSinc	$\frac{f_{DATA}}{SFOR\times(IOR-1+4)+(2+1)}$
	Fast mode (FAST=1)	FastSinc and $Sinc^X$	$\frac{f_{DATA}}{SFOR\times IOR}$

Note: f_{DATA} is the parallel data rate of the CPU / DMA input. When the filter is bypassed, $f_{DATA} \leqslant f_{HPDFCLK}$ must be satisfied.

Signed data format

Signed data in HPDF module: parallel data register, regular and inserted group data register, threshold monitor vlaue, extreme monitor value, and offset calibration are all signed formats. The most significant bit of the output data indicates the sign of the value, and the data is in two's complement format.

Since all operations in digital processing are performed on 32-bit signed registers, the following conditions must be met in order for the result not to overflow:

1. When using $Sinc^X$ filter (x=1...5): $(SFOR^{SFO})\times IOR \leq 2^{31}$.
2. When using FastSinc filter: $2\times(SFOR^2)\times IOR \leq 2^{31}$.

Data right bit shift

Since the final data width is 24 bits and the data from the processing path can be up to 32 bits, the right shift of the final data is performed in this module. For each selected input channel, the number of bits shifted to the right can be configured in the DTRS[4:0] bit field in

the HPDF_CHxCFG0 register. The result is round to the nearest value by abandoning the lowest bit.

Data offset calibration

In the HPDF module, each channel has a data offset calibration value, which is stored in the CALOFF[23:0] bit field in HPDF_CHxCFG0 register. When offset calibration is performed, the offset calibration value is subtracted from the output data of the channel to obtain the final data output by the HPDF module.

Data offset calibration occurs after the right shift of the data.

34.3.14. HPDF interrupt

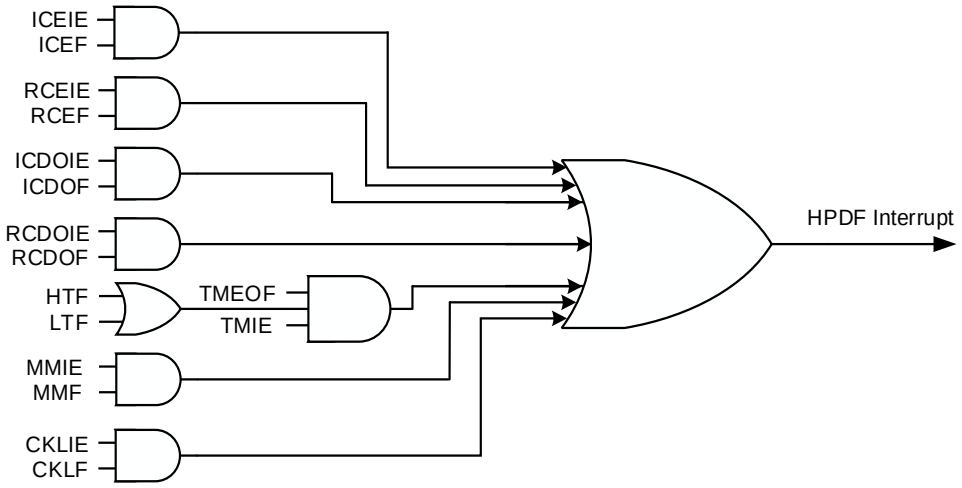
HPDF interrupt events can be divided into channel conversion interrupt events, threshold monitor interrupt events, malfunction monitor interrupt events, and channel clock loss interrupt events. The specific interrupt event description is as ***Table 34-10. HPDF interrupt event***.

Table 34-10. HPDF interrupt event

Interrupt event	description	Clear	Enable interrupt
ICEF	end of inserted conversion	Read HPDF_FLTyIDATA register	ICEIE
RCEF	end of regular conversion	Read HPDF_FLTyRDATA register	RCEIE
ICDOF	inserted conversion data overflow	Write 1 to the ICDOFC bit	ICDOIE
RCDOF	regular conversion data overflow	Write 1 to RCDOFC bit	RCDOIE
TMEOF HTF[7:0] LTF[7:0]	threshold monitor events	Write 1 to HTFC[7:0] bit field Write 1 to LTFC[7:0] bit field	TMIE
MMF	malfunction event	Write 1 to MMFC[7:0] bits	MMIE
CKLF	Channel clock loss event	Write 1 to CKLFC[7:0] bits	CKLIE

HPDF interrupt logic is as ***Figure 34-8. HPDF interrupt logic diagram*** shown.

Figure 34-8. HPDF interrupt logic diagram



34.4. Register definition

HPDF base address: 0x4001 7000

34.4.1. HPDF channel x registers (x=0…7)

Channel x control register (HPDF_CHxCTL)

Address offset: 0x00 + 0x20 * x, (x = 0…7)

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HPDFEN	CKOUTSEL	CKOUTDM	Reserved					CKOUTDIV[7:0]							
rw	rw	rw						rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DPM[1:0]		CMSD[1:0]		Reserved			CHPINSEL	CHEN	CKLEN	MMEN	Reserved	SPICKSS[1:0]		SITYP[1:0]	
rw		rw					rw	rw	rw	rw		rw		rw	

Bits	Fields	Descriptions
31	HPDFEN	Global enable for HPDF interface 0: HPDF disabled 1: HPDF enabled If HPDFEN=0, the HPDF_FLTySTAT register and HPDF_FLTyTMSTAT register is set to reset state. This bit is only available in HPDF_CH0CTL.
30	CKOUTSEL	Serial clock output source selection 0: Serial clock output source is from CK_HPDF clock 1: Serial clock output source is from CK_HPDFAUDIO clock This bit can be configured only when HPDFEN=0. This bit is only available in HPDF_CH0CTL.
29	CKOUTDM	Serial clock output duty mode 0: Serial clock output duty mode disable 1: Serial clock output duty mode enable, the duty is 1:1 This bit can be configured only when HPDFEN=0. This bit is only available in HPDF_CH0CTL.
28:24	Reserved	Must be kept at reset value.
23:16	CKOUTDIV[7:0]	Serial clock output divider 0: Output clock generation is disabled (CKOUT signal is set to low state)

		1~255: The value of division for the serial clock output is CKOUTDIV+1. CKOUTDIV also defines the threshold for a clock loss detection. This value can only be modified when HPDFEN=0.During a HPDF clock cycle after HPDFEN=0, the CKOUT is set to low state. This bit is only available in HPDF_CH0CTL.
15:14	DPM[1:0]	Data packing mode for HPDF_CHxPDI register 00: Standard mode 01: Interleaved mode 10: Dual mode 11: Reserved For a detailed introduction of data encapsulation mode, please refer to ***Parallel data packed mode.*** These bits can be configured only when CHEN=0.
13:12	CMSD[1:0]	Channel x multiplexer select input data source 00: Input data source for channel x is taken from serial inputs 01: Input data source for channel x is taken from the internal analog-to-digital converter ADC output register update 10: Input data source for channel x is taken from internal HPDF_CHxPDI register 11: Reserved The HPDF_CHxPDI register is write protected when these bits are reset. These bits can be configured only when CHEN=0.
11:9	Reserved	Must be kept at reset value.
8	CHPINSEL	Channel inputs pins selection 0: Channel inputs select pins of the current channel x 1: Channel inputs select pins of the next channel. This bit can be configured only when CHEN=0.
7	CHEN	Channel x enable 0: Channel x disabled 1: Channel x enabled If channel x is enabled, then serial data will be received based on the given channel settings.
6	CKLEN	Clock loss detector enable 0: Clock loss detector disabled 1: Clock loss detector enabled
5	MMEN	Malfunction monitor enable 0: malfunction monitor is no effect 1: malfunction monitor is effect
4	Reserved	Must be kept at reset value.
3:2	SPICKSS[1:0]	SPI clock source select

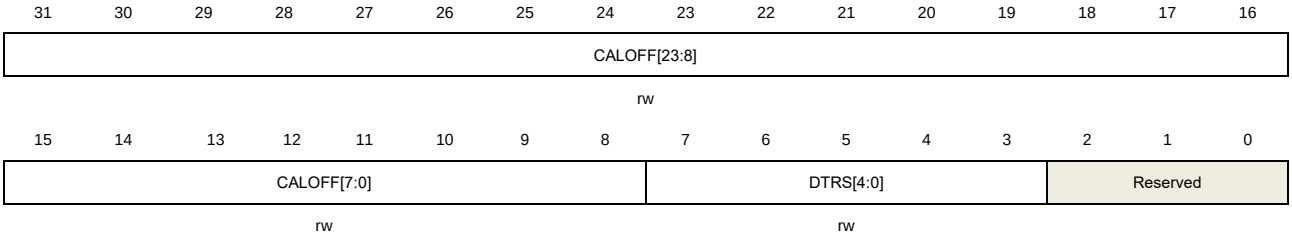
		00: External CKINx input is selected for SPI clock source, sampling point is determined by SITYP[1:0] 01: Internal CKOUT output is selected for SPI clock source, sampling point is determined by SITYP[1:0] 10: Internal CKOUT is selected for SPI clock source, sampling point on each second CKOUT falling edge. 11: Internal CKOUT output is selected for SPI clock source, sampling point on each second CKOUT rising edge. These bits can be configured only when CHEN=0.
1:0	SITYP[1:0]	Serial interface type 00: SPI interface, sample data on rising edge 01: SPI interface, sample data on falling edge 10: Manchester coded input: rising edge = logic 0, falling edge = logic 1 11: Manchester coded input: rising edge = logic 1, falling edge = logic 0 These bits can only be configured when CHEN=0.

Channel x configuration register 0 (HPDF_CHxCFG0)

Address offset: 0x04 + 0x20 * x, (x = 0…7)

Reset value: 0x0000 0000

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



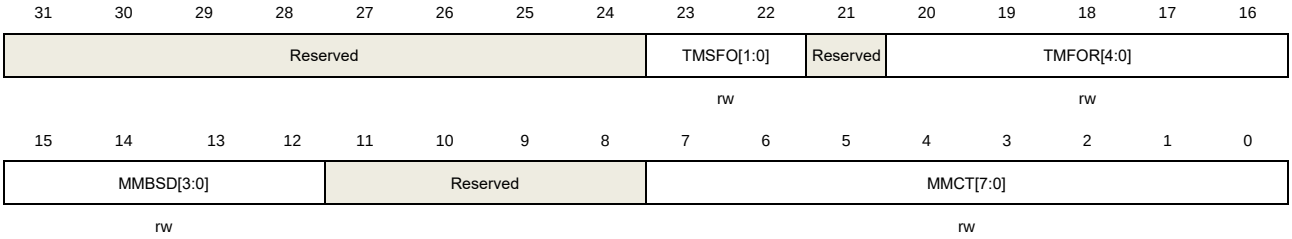
Bits	Fields	Descriptions
31:8	CALOFF[23:0]	24-bit calibration offset Calibration offset must be performed for each conversion result of the channel. These bits can be set by software.
7:3	DTRS[4:0]	Data right bit-shift 0-31: The number of bits that determine the right shift of data Bit-shift is performed before offset correction. The data shift rounds the result to the nearest integer value and the sign is preserved. These bits can be configured only when CHEN=0 (in HPDF_CHxCTL register).
2:0	Reserved	Must be kept at reset value.

Channel x configuration register 1 (HPDF_CHxCFG1)

Address offset: 0x08 + 0x20 * x, (x = 0…7)

Reset value: 0x0000 0000

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



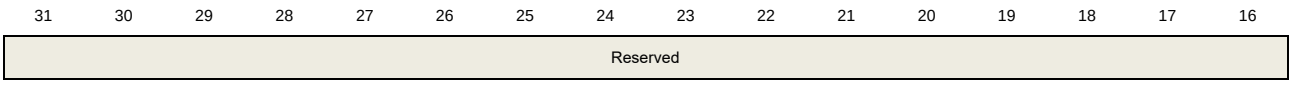
Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:22	TMSFO[1:0]	Threshold monitor Sinc filter order selection 00: FastSinc filter 01: Sinc^1 filter 10: Sinc^2 filter 11: Sinc^3 filter These bits can be configured only when CHEN=0 (in HPDF_CHxCTL register).
21	Reserved	Must be kept at reset value
20:16	TMFOR[4:0]	Threshold monitor filter oversampling rate (decimation rate) 0 - 31: The filter decimation rate equal to TMFOR[4:0]+ 1 If TMFOR=0, the filter is bypassed. These bits can be configured only when CHEN=0 (in HPDF_CHxCTL register).
15:12	MMBSD[3:0]	Malfunction monitor break signal distribution MMBSD[i] = 0: Break i signal not is distributed to malfunction monitor on channel x MMBSD[i] = 1: Break i signal is distributed to malfunction monitor on channel x
11:8	Reserved	Must be kept at reset value.
7:0	MMCT[7:0]	Malfunction monitor counter threshold These bits be used determine the count value of malfunction monitor counter threshold. The count value is written by software. If the count value is reached, then an event of malfunction monitor occurs on a given channel.

Channel x threshold monitor filter data register (HPDF_CHxTMFDT)

Address offset: 0x0C + 0x20 * x, (x = 0…7)

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
TMDATA[15:0]															
r															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	TMDATA[15:0]	Threshold monitor data The data is come from the threshold monitor filter and continuously converted (no trigger) for this channel.

Channel x parallel data input register (HPDF_CHxPDI)

Address offset: 0x10 + 0x20 * x, (x = 0…7)
Reset value: 0x0000 0000

This register has to be accessed by half-word (16-bit) and word (32-bit).

This register contains 16-bit input data to be processed by HPDF filter module.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATAIN1[15:0]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATAIN0[15:0]															
rw															

Bits	Fields	Descriptions
31:16	DATAIN1[15:0]	Data input for channel x or channel x+1 Data can be written by CPU/DMA. If DPM[1:0]=0 (standard mode), DATAIN1[15:0] is write protected. If DPM[1:0]=1 (interleaved mode), second channel x data sample is stored into DATAIN1[15:0]. First channel x data sample is stored into DATAIN0[15:0]. Both samples are read sequentially by HPDF_FLTy filter. If DPM[1:0]=2 (dual mode): For channel 0: sample in DATAIN1[15:0] is automatically copied into DATAIN0[15:0] of channel 1. For channel 1: DATAIN1[15:0] is write protected. The more details refer to ***Parallel data packed mode*** DATAIN1[15:0] is a signed format data.
15:0	DATAIN0[15:0]	Data input for channel x Data can be written by CPU / DMA. If DPM[1:0]=0 (standard mode), channel x data sample is stored into DATAIN0[15:0]. If DPM[1:0]=1 (interleaved mode), first channel x data sample is stored into

DATAIN0[15:0]. Second channel x data sample is stored into DATAIN1[15:0]. Both samples are read sequentially by HPDF_FLTy filter.

If DPM[1:0]=2 (dual mode):

For channel 0: Channel x data sample is stored into DATAIN0[15:0].

For channel 1: DATAIN0[15:0] is write protected.

The more details refer to ***Parallel data packed mode***

DATAIN0[15:0] is a signed format data.

Channel x pulse skip register (HPDF_CHxPS)

Address offset: 0x14 + 0x20 * x, (x = 0…7)

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	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										PLSK[5:0]					
										rw					

Bits	Fields	Descriptions
31:6	Reserved	Must be kept at reset value.
5:0	PLSK[5:0]	Pulses to skip for input data skipping function 0-63: Defines the number of serial input samples that will be skipped. Skipping function is take effect immediately after writing to this field. Read PLSK[5:0] to return the remaining value of the pulses which will be skipped. The value of PLSK[5:0] can be updated even PLSK[5:0] is not zero.

34.4.2. HPDF filter y registers (y=0…3)

Filter y control register 0 (HPDF_FLTyCTL0)

Address offset: 0x100 + 0x80 * y, (y = 0…3)

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	TMFM	FAST	Reserved		RCS[2:0]			Reserved		RCDMAEN	Reserved	RCSYN	RCCM	SRCS	Reserved
	rw	rw			rw					rw		rw	rw	rt_w	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	ICTEEN[1:0]		ICTSSEL[4:0]					Reserved		ICDMAEN	SCMOD	ICSYN	Reserved	SICC	FLTEN

rw rw rw rw rw rt_w rw

Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30	TMFM	Threshold monitor fast mode 0: Threshold monitor watch on the final data after performing offset correction and right shift 1: Threshold monitor watch on serial input data stream
29	FAST	Fast conversion mode for regular conversions 0: Fast conversion mode disabled 1: Fast conversion mode enabled If fast mode is enabled, the normal conversion in continuous mode (except for the first conversion) is performed faster than the conversion in standard mode. This bit has no effect on conversions which are not continuous. This bit can be configured only when FLTEN=0.
28:27	Reserved	Must be kept at reset value
26:24	RCS[2:0]	Regular conversion channel selection 0: Channel 0 is selected as the regular conversion channel 1: Channel 1 is selected as the regular conversion channel … 7: Channel 7 is selected as the regular conversion channel When RCPF=1, writing this bit takes effect when the next regular conversion begins.
23:22	Reserved	Must be kept at reset value
21	RCDMAEN	DMA channel enabled to read data for the regular conversion 0: Disable the DMA channel to read regular data 1: Enable the DMA channel to read regular data This bit can be configured only when FLTEN=0.
20	Reserved	Must be kept at reset value.
19	RCSYN	Regular conversion synchronously with HPDF_FLT0 0: Do not launch a regular conversion synchronously with HPDF_FLT0 1: Launch a regular conversion synchronously in HPDF_FLTy when a regular conversion is launched in HPDF_FLT0 If RCSYN=1 in HPDF_FLT0CTL0 register, the regular conversion channel will be Launched synchronously which selected in HPDF_FLTyCTL0. This bit can be configured only when FLTEN=0.
18	RCCM	Regular conversions continuous mode 0: The regular channel is converted just once for each conversion request 1: The regular channel is converted repeatedly after each conversion request Writing "0" to this bit will immediately stop continuous mode during a continuous

		regular conversion.
17	SRCS	Start regular channel conversion by software 0: No effect 1: Make a request to start regular channel conversion If RCPF=1, invalid write to SRCS, and if RCSYN=1, write '1' to SRCS, launch a regular conversion synchronously. This bit is always read as '0'.
16:15	Reserved	Must be kept at reset value.
14:13	ICTEEN[1:0]	Inserted conversions trigger edge enable 00: Disable trigger detection 01: Each rising edge on the trigger signal makes a request to start an inserted conversion 10: Each falling edge on the trigger signal makes a request to start an inserted conversion 11: The edge (rising edges and falling edges) on the trigger signal make requests to start inserted conversions This bit can be configured only when FLTEN=0.
12:8	ICTSSEL[4:0]	Inserted conversions trigger signal selection 0x00~0x1F: The value indicates that different trigger signals are selected to start the conversion The maximum delay from the generation of trigger signal to the start of synchronous trigger is 1 $f_{HPDFCLK}$ clock cycle, and the delay of asynchronous trigger is 2-3 $f_{HPDFCLK}$ clock cycles. This bit can be configured only when FLTEN=0.
7:6	Reserved	Must be kept at reset value.
5	ICDMAEN	DMA channel enabled to read data for the inserted channel group 0: Disable DMA channel to read inserted conversions data 1: Enable DMA channel to read inserted conversions data This bit can be configured only when FLTEN=0.
4	SCMOD	Scan conversion mode of inserted conversions 0: One channel conversion is performed from the inserted channel group and next the channel is selected from this group. 1: The series of conversions for the inserted group channels is executed, starting over with the lowest selected channel. If SCMOD=0, writing ICGSEL will resets the channel selection to the lowest selected channel. This bit can be configured only when FLTEN=0.
3	ICSYN	Inserted conversion synchronously 0: Do not launch an inserted conversion synchronously with HPDF_FLT0

		1: Launch an inserted conversion synchronously in HPDF_FLTy when an inserted conversion is launched by trigger SICC. This bit can be configured only when FLTEN=0.
2	Reserved	Must be kept at reset value
1	SICC	Start inserted group channel conversion 0: No effect. 1: Makes a request to convert the channels in the inserted conversion group. If ICPF=1 already, invalid write to SICC. If RCSYN=1, write '1' to SICC, launch an inserted conversion synchronously. This bit is always read as '0'.
0	FLTEN	HPDF_FLTy enable 0: HPDF_FLTy is disabled. 1: HPDF_FLTy is enabled. If HPDF_FLTy is enabled, then HPDF_FLTy starts operating according to its setting. If HPDF_FLTy is disabled, all conversions of given HPDF_FLTy are stopped immediately and all HPDF_FLTy functions are stopped. Meanwhile HPDF_FLTySTAT register and HPDF_FLTyTMSTAT register is set to the reset state.

Filter y control register 1 (HPDF_FLTyCTL1)

Address offset: 0x104 + 0x80 * y, (y = 0…3)

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								TMCHEN[7:0]							
								rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EMCS[7:0]								Reserved	CKLIE	MMIE	TMIE	RCDOIE	ICDOIE	RCEIE	ICEIE
rw									rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:16	TMCHEN[7:0]	Threshold monitor channel enable These bits select the input channel to be guarded continuously by the threshold monitor. TMCHEN[x] = 0: Threshold monitor y is disabled on channel x TMCHEN[x] = 1: Threshold monitor y is enabled on channel x
15:8	EMCS[7:0]	Extremes monitor channel selection These bits select the input channels to be taken by the extremes monitor. EMCS[x] = 0: Extremes monitor y does not monitor data from channel x

		EMCS[x] = 1: Extremes monitor y monitor data from channel x
7	Reserved	Must be kept at reset value
6	CKLIE	Clock loss interrupt enable 0: Detection of channel input clock loss interrupt is disabled 1: Detection of channel input clock loss interrupt is enabled This bit is only available in HPDF_FLT0CTL1 register.
5	MMIE	Malfunction monitor interrupt enable 0: malfunction monitor interrupt is disabled 1: malfunction monitor interrupt is enabled This bit is only available in HPDF_FLT0CTL1 register.
4	TMIE	Threshold monitor interrupt enable 0: Threshold monitor interrupt is disabled 1: Threshold monitor interrupt is enabled
3	RCDOIE	Regular conversion data overflow interrupt enable 0: Regular conversion data overflow interrupt is disabled 1: Regular conversion data overflow interrupt is enabled
2	ICDOIE	Inserted conversion data overflow interrupt enable 0: Inserted data overflow interrupt is disabled 1: Inserted data overflow interrupt is enabled
1	RCEIE	Regular conversion end interrupt enable 0: Regular conversion end interrupt is disabled 1: Regular conversion end interrupt is enabled
0	ICEIE	Inserted conversion end interrupt enable 0: Inserted conversion end interrupt is disabled 1: Inserted conversion end interrupt is enabled

Filter y status register (HPDF_FLTySTAT)

Address offset: 0x108 + 0x80 * y, (y = 0…3)

Reset value: 0x0003 0000

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

All the bits of HPDF_FLTySTAT are automatically reset when FLTEN=0.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MMF[7:0]								CKLF[7:0]							
r								r							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	RCPF	ICPF	Reserved								TMEOF	RCDOF	ICDOF	RCEF	ICEF
	r	r									r	r	r	r	r

Bits	Fields	Descriptions
31:24	MMF[7:0]	Malfunction monitor flag MMF[x]=0: No malfunction event occurred on channel x MMF[x]=1: Malfunction event occurred on channel x This bit is set by hardware. It can be cleared by software using the corresponding MMFC[x] bit in the HPDF_ FLTyINTC register. MMF[x] is cleared also by hardware when CHEN[x] = 0 (given channel is disabled). This bit is only available in HPDF_FLT0STAT register.
23:16	CKLF[7:0]	Clock loss flag CKLF[x]=0: Clock signal is not lost on channel x CKLF[x]=1: Clock signal is lost on channel x When CHEN=0 or the serial interface is not synchronized, the state is maintained by the hardware. After the synchronization of serial interface is completed, if the clock of channel x is lost, the corresponding bit in CKLF[7:0] bit field are set by hardware. By setting the CKLFC[7:0] bit field in HPDF_FLTyINTC, the corresponding bit in the CKLF[7:0] bit field can be cleared. This bit is only available in HPDF_FLT0STAT register.
15	Reserved	Must be kept at reset value
14	RCPF	Regular conversion in progress flag 0: No request of regular conversion has been generated 1: The regular conversion is in progress or a request for a regular conversion is pending If RCPF=1, a request to start a regular conversion is ignored. When write 1 to SRCS bit, the RCPF will be setted 1 immediately.
13	ICPF	Inserted conversion in progress flag 0: No request to the inserted group conversion has been generated (neither by software nor by trigger). 1: The inserted group conversion is in progress or a request for a inserted conversion is pending. If ICPF=1, a request to start an inserted conversion is ignored. When write 1 to SICC bit, the ICPF will be setted 1 immediately.
12:5	Reserved	Must be kept at reset value
4	TMEOF	Threshold monitor event occurred flag 0: No Threshold monitor event occurred. 1: Threshold monitor event occurred which detected data crosses the threshold This bit is set by hardware. It is cleared by clearing HTF[7:0] and LTF[7:0] in HPDF_FLTyTMSTAT register.
3	RCDOF	Regular conversion data overflow flag 0: No regular conversion data overflow has occurred 1: A regular conversion data overflow has occurred

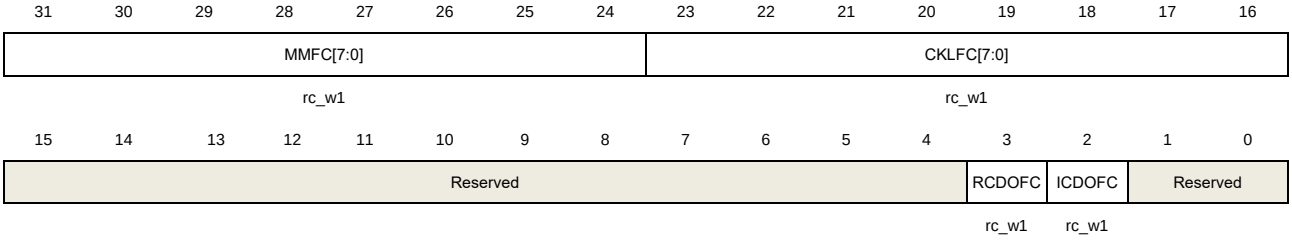
		If RCDOF=1, it means that a regular conversion finished while RCEF has already been set. RDATA is not affected by overflows. This bit is set by hardware. It can be cleared by setting the RCDOFC in the HPDF_FLTyINTC register.
2	ICDOF	Inserted conversion data overflow flag 0: No inserted conversion data overflow has occurred 1: An inserted conversion data overflow has occurred If RCDOF=1, it means that an inserted conversion finished while ICEF has already been set. FLTyIDATA is not affected by overflows This bit is set by hardware. It can be cleared by software setting the ICDOFC bit in the HPDF_FLTyINTC register.
1	RCEF	Regular conversion end flag 0: No regular conversion has completed 1: A regular conversion has completed If RCEF=1, it means that the data may be read. This bit is set by hardware. It is cleared when the software or DMA reads HPDF_FLTyRDATA register.
0	ICEF	Inserted conversion end flag 0: No inserted conversion has completed 1: An inserted conversion has completed If ICEF=1, it means that its data may be read. This bit is set by hardware. It is cleared when the software or DMA reads HPDF_FLTyIDATA register.

Filter y interrupt flag clear register (HPDF_FLTyINTC)

Address offset: 0x10C + 0x80 * y, (y = 0…3)
Reset value: 0x0000 0000

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

Note: The bits of HPDF_FLTyINTC are always read as '0'.



Bits	Fields	Descriptions
31:24	MMFC[7:0]	Clear the malfunction monitor flag MMFC[x]=0: No effect

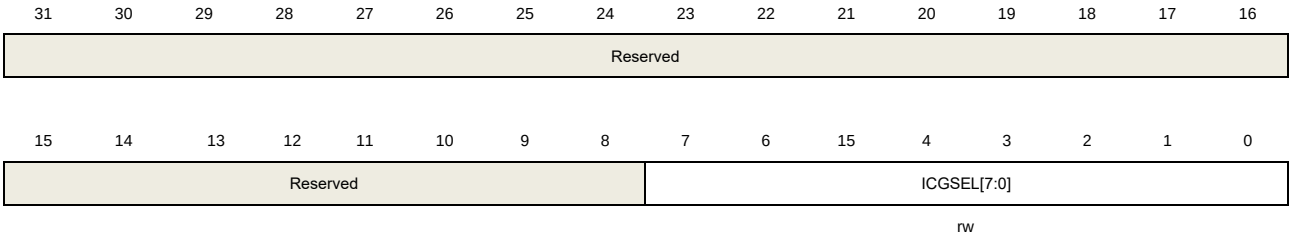
		MMFC[x]=1: Clear the malfunction monitor flag on channel x This bit is only available in HPDF_FTL0INTC register.
23:16	CKLFC[7:0]	Clear the clock loss flag CKLFC[x]=0: No effect CKLFC[x]=1: Clear the clock loss flag on channel x When the serial transceiver is not yet synchronized, the clock loss flag is set and cannot be cleared by CKLFC[7:0]. This bit is only available in HPDF_FTL0INTC register.
15:4	Reserved	Must be kept at reset value.
3	RCDOFC	Clear the regular conversion data overflow flag 0: No effect 1: Clear the RCDOF bit in the HPDF_FLTySTAT register
2	ICDOFC	Clear the inserted conversion data overflow flag 0: No effect 1: Clear the ICDOF bit in the HPDF_FLTySTAT register
1:0	Reserved	Must be kept at reset value.

Filter y inserted channel group selection register (HPDF_FLTyICGS)

Address offset: 0x110 + 0x80 * y, (y = 0…3)

Reset value: 0x0000 0001

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



Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	ICGSEL[7:0]	Injected channel group selection ICGSEL[x]=0: Channel x is not belongs to the inserted group ICGSEL[x]=1: Channel x belongs to the inserted group If SCMOD=1, each of the selected channels is converted, one after another. The priority conversion with lowest channel number. If SCMOD=0, then only one channel is converted from the selected channels, and the channel selection is moved to the next channel. When SCMOD=0, Writing ICGSEL will reset the channel selection to the lowest selected channel. At least one channel must always be selected for the inserted group. All writes that

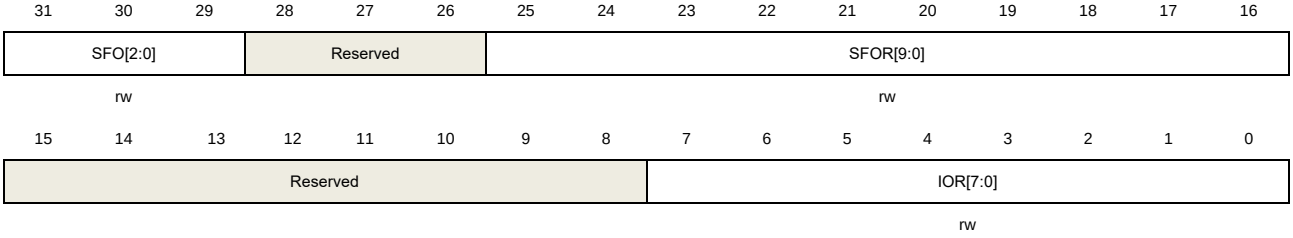
make ICGSEL[7:0]=0 are ignored.

Filter y sinc filter configuration register (HPDF_FLTySFCFG)

Address offset: 0x114 + 0x80 * y, (y = 0…3)

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:29	SFO[2:0]	Sinc filter order. 000: FastSinc filter type 001: $Sinc^1$ filter type 010: $Sinc^2$ filter type 011: $Sinc^3$ filter type 100: $Sinc^4$ filter type 101: $Sinc^5$ filter type 110~111: Reserved This bit can only be configured when FLTEN=0 in HPDF_FLTyCTL0 register.
28:26	Reserved	Must be kept at reset value.
25:16	SFOR[9:0]	Sinc filter oversampling ratio (decimation rate) 0 ~1023: Sinc filter oversampling ratio (decimation rate) SFOR= SFOR[9:0] +1. If SFOR [9:0] = 0 (SFOR=1), the filter will be bypass. This bit can only be configured when FLTEN=0 in HPDF_FLTyCTL0 register.
15:8	Reserved	Must be kept at reset value.
7:0	IOR[7:0]	Integrator oversampling ratio 0~255: Integrator oversampling ratio IOR=IOR[7:0]+1. The output data rate from the integrator will be decreased by this value. If IOR[7:0] = 0 (IOR=1), the integrator will be bypass. This bit can only be configured when FLTEN=0 in HPDF_FLTyCTL0 register.

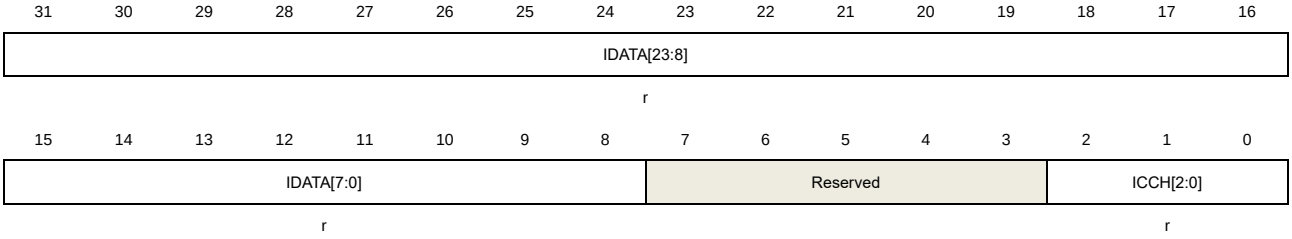
Filter y inserted group conversion data register (HPDF_FLTyIDATA)

Address offset: 0x118 + 0x80 * y, (y = 0…3)

Reset value: 0x0000 0000

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

Note: Half-word accesses may be used to read only the MSB of conversion data. DMA can be used to read the data from this register. Reading this register also clears ICEF bit.



Bits	Fields	Descriptions
31:8	IDATA[23:0]	Inserted group conversion data When each a channel in the inserted group is converted, the resulting data is stored in this field. The data is valid when ICEF=1. Reading this register clears the corresponding ICEF (in HPDF_FLTySTAT register).
7:3	Reserved	Must be kept at reset value.
2:0	ICCH[2:0]	Inserted channel most recently converted When each a channel in the inserted group is converted, ICCH is updated to indicate which channel was converted. Therefore, IDATA[23:0] holds the data that corresponds to the channel indicated by ICCH.

Filter y regular channel conversion data register (HPDF_FLTyRDATA)

Address offset: 0x11C + 0x80 * y, (y = 0…3)
Reset value: 0x0000 0000

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

Note: Half-word accesses may be used to read only the MSB of conversion data. Reading this register also clears RCEF bit.



Bits	Fields	Descriptions
31:8	RDATA[23:0]	Regular channel conversion data When each regular conversion finishes, its data is stored in these bits. The data is valid when RCEF=1. Reading this register clears the corresponding RCEF (in HPDF_FLTySTAT register).

7:5	Reserved	Must be kept at reset value.
4	RCHPDT	Regular channel pending data Regular data in RDATA[23:0] was delayed due to an inserted channel trigger during the conversion
3	Reserved	Must be kept at reset value.
2:0	RCCH[2:0]	Regular channel most recently converted When each regular conversion finishes, RCCH[2:0] is updated to indicate which channel was converted. Thus RDATA[23:0] holds the data that corresponds to the channel indicated by RCCH[2:0].

Filter y threshold monitor high threshold register (HPDF_FLTyTMHT)

Address offset: 0x120 + 0x80 * y, (y = 0…3)

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HTVAL[23:8]															
rw															

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

Bits	Fields	Descriptions
31:8	HTVAL[23:0]	Threshold monitor high threshold value These bits are written by software to determine the high threshold for the threshold monitor. If TMFM=1, the higher 16 bits determine the 16-bit threshold as compared with the threshold monitor filter output. Bits HTVAL[7:0] are ignored.
7:4	Reserved	Must be kept at reset value.
3:0	HTBSD[3:0]	High threshold event break signal distribution HTBSD[i] = 0: Break signal is not distributed to high threshold event HTBSD[i] = 1: Break signal i is distributed to high threshold event

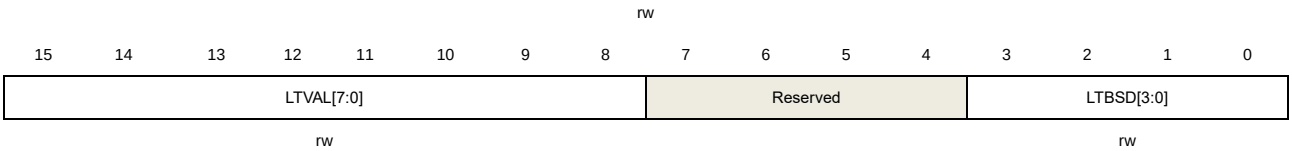
Filter y threshold monitor low threshold register (HPDF_FLTyTMLT)

Address offset: 0x124 + 0x80 * y, (y = 0…3)

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LTVAL[23:8]															



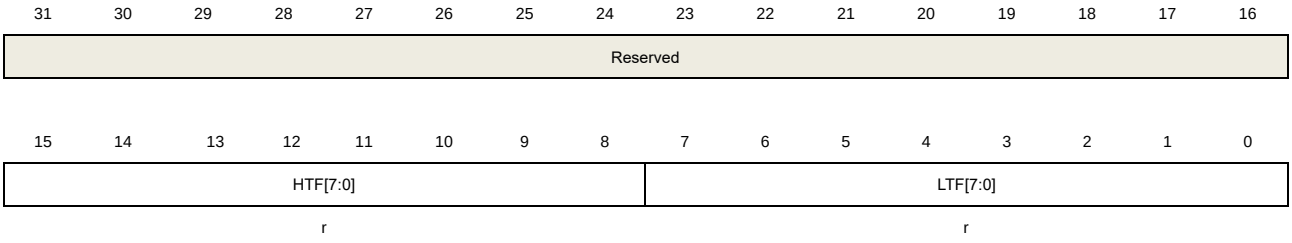
Bits	Fields	Descriptions
31:8	LTVAL[23:0]	Threshold monitor low threshold value. These bits are written by software to determine the low threshold for the threshold monitor. If TMFM=1, the higher 16 bits determine the 16-bit threshold as compared with the threshold monitor filter output. Bits LTVAL[7:0] are ignored.
7:4	Reserved	Must be kept at reset value.
3:0	LTBSD[3:0]	Low threshold event break signal distribution. LTBSD[i] = 0: Break signal is not distributed to low threshold event. LTBSD[i] = 1: Break signal i is distributed to low threshold event.

Filter y threshold monitor status register (HPDF_FLTyTMSTAT)

Address offset: 0x128 + 0x80 * y, (y = 0…3)
Reset value: 0x0000 0000

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

Note: All the bits of HPDF_FLTyTMSTAT are automatically reset when FLTEN=0.



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:8	HTF[7:0]	Threshold monitor high threshold flag HTF[x]=0: No high threshold error on channel x HTF[x]=1: High threshold error on channel x It is set by hardware. It can be cleared by software setting the corresponding HTFC[7:0] bit in the HPDF_FLTyTMFC register.
7:0	LTF[7:0]	Threshold monitor low threshold flag LTF[x]=1: No low threshold error on channel x LTF[x]=1: Low threshold error on channel x It is set by hardware. It can be cleared by software setting the corresponding

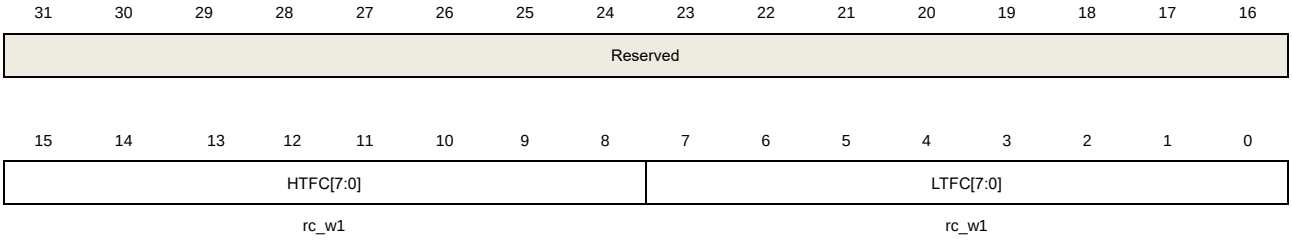
LTFC[7:0] bit in the HPDF_FLTyTMFC register.

Filter y threshold monitor flag clear register (HPDF_FLTyTMFC)

Address offset: 0x12C + 0x80 * y, (y = 0…3)

Reset value: 0x0000 0000

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



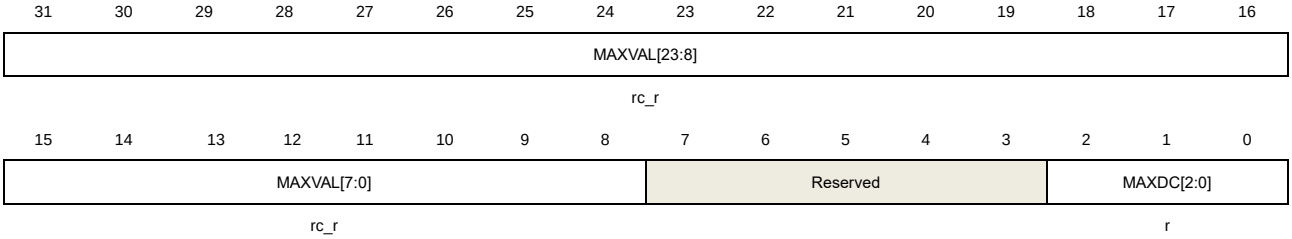
Bits	Fields	Descriptions
31:10	Reserved	Must be kept at reset value.
15:8	HTFC[7:0]	Clear the threshold monitor high threshold flag HTFC[y]=0: No effect HTFC[y]=1: Clear the threshold monitor high threshold flag on channel x
7:0	LTFC[7:0]	Clear the threshold monitor low threshold flag LTFC[y]=0: No effect LTFC[y]=1: Clear the threshold monitor low threshold flag on channel x

Filter y extremes monitor maximum register (HPDF_FLTyEMMAX)

Address offset: 0x130 + 0x80 * y, (y = 0…3)

Reset value: 0x8000 0000

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



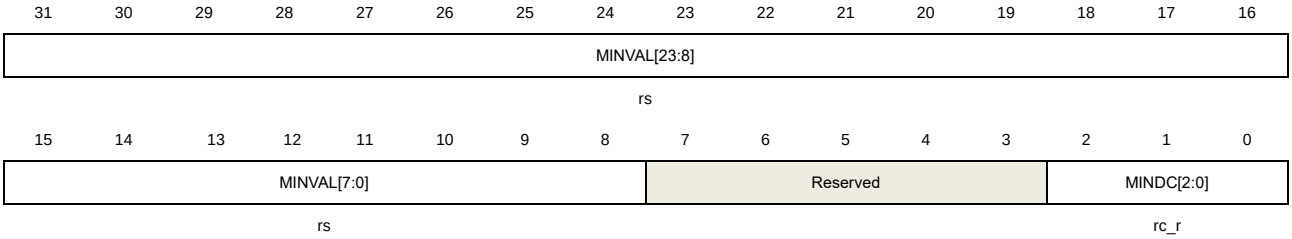
Bits	Fields	Descriptions
31:8	MAXVAL[23:0]	Extremes monitor maximum value These bits are set by hardware and indicate the highest value of channel converted by HPDF_FLTy. These bits can be reset by reading of this register.

7:3	Reserved	Must be kept at reset value.
2:0	MAXDC[2:0]	Extremes monitor maximum data channel. This bits indicate the channel on which the data is stored into MAXVAL[23:0]. It can be cleared by reading of this register.

Filter y extremes monitor minimum register (HPDF_FLTyEMMIN)

Address offset: 0x134 + 0x80 * y, (y = 0…3)
Reset value: 0x7FFF FF00

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

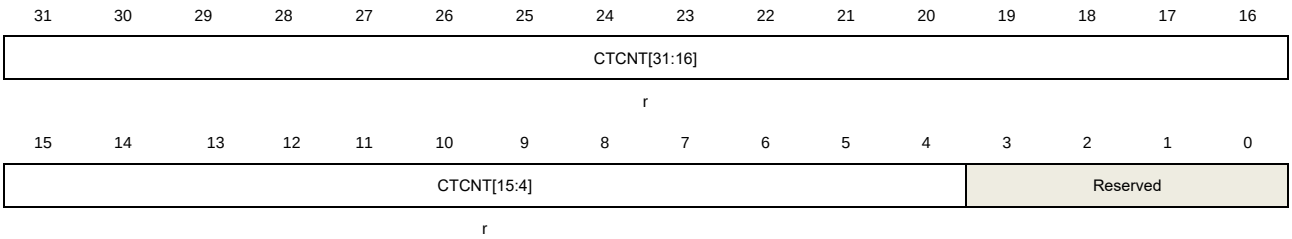


Bits	Fields	Descriptions
31:8	MINVAL[23:0]	Extremes monitor minimum value These bits are set by hardware and indicate the lowest value converted by HPDF_FLTy. These bits can be reset by reading of this register.
7:3	Reserved	Must be kept at reset value.
2:0	MINDC[2:0]	Extremes monitor minimum data channel This bit indicate the channel on which the data is stored into MINVAL[23:0]. It can be cleared by reading of this register.

Filter y conversion timer register (HPDF_FLTyCT)

Address offset: 0x138 + 0x80 * y, (y = 0…3)
Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:4	CTCNT[27:0]	conversion time counted by HPDFCLK

t = CNVCNT[27:0] / $f_{HPDFCLK}$

The timer has an input clock (system clock $f_{HPDFCLK}$) from the HPDF clock. The conversion time measurement starts at the beginning of each conversion and stops when the conversion is complete (the interval between the first and last serial sample). Conversion time measurement stops and CNVCNT[27:0] = 0 only when the filter is bypassed (FOSR[9:0] = 0). The times counted are:

if FAST=0 (or first conversion in continuous mode if FAST=1):

for Sincx filters: t = [SFOR * (IOR-1 + SFO) + SFO] / f_{CKIN}.

for FastSinc filter: t = [SFOR * (IOR-1 + 4) + 2] / f_{CKIN}.

if FAST=1 in continuous mode (except first conversion):

t = [SFOR * IOR-1] / f_{CKIN}.

in case if FOSR = SFOR [9:0]+1 = 1: CNVCNT = 0 (counting is stopped, conversion time: t = IOR / f_{CKIN})

where f_{CKIN} is the channel input clock frequency (on a given channel CKINx pin) or the input data rate in the case of parallel data input (from CPU/DMA writes)

Note: The timer will also counts this interrupt time when a conversion is interrupted (for example by disabling/enabling the selected channel).

3:0	Reserved	Must be kept at reset value.

35. Filter arithmetic accelerator (FAC)

35.1. Overview

The filter arithmetic accelerator unit consist of multiplier, accumulator and address generation logic, so as to index vector elements stored in local memory. Circular buffering is valid for both input and output, which allows to realize finite impulse response (FIR) filters and infinite impulse response (IIR) filters. The unit support CPU to be free from frequent or lengthy filtering operations, compared with software implementation, it can accelerate calculations and the processing speed of time critical tasks.

35.2. Characteristics

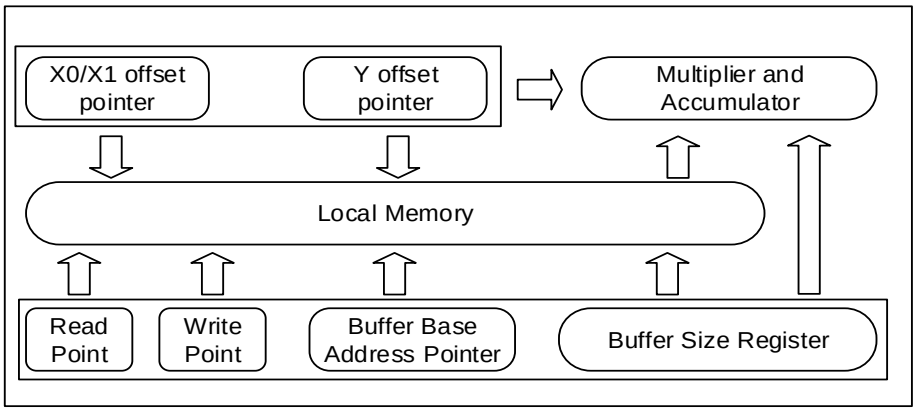
- Fixed or float multiplier and accumulator.
- 256 x 32-bit local memory.
- 16-bit fixed-point or 32-bit float point input and output.
- Up to three buffers, two input buffers and one output buffer.
- Buffer can be circular.
- FIR and IIR can be realized.
- Vector functions support convolution, Dot product, correlation functions.
- Data can be read and written through DMA.

35.3. Function overview

35.3.1. General description

The unit can be configured based on fixed point multiplier and accumulator or flaot point multiplier and accumulator. Two 16-bit input signed datas or 32-bit input float datas from local memory are taken to MAC, they are multiplied together and added to the accumulator. A set of pointers, which could be incremented, decremented and loaded or reset by hardware, index the address of the input data in memory. Built-in sequencer arranges control the pointer and MAC operations to perform the requested operation in order.

Figure 35-1 FAC structure diagram



The processor or DMA controller load two input vectors into the local memory to calculate a dot product, select and start the requested operation. The elements of input vector is extracted from memory, multiplied together and then accumulated the multiplier output together. After processing the requested operation, the local memory is used to store the contents of the accumulator, the processor or DMA can access the corresponding address.

The finite impulse response filter operation repeatedly calculate the dot product, which refer to the coefficient data and a input sample data, along with discarding the least recent sample and adding a new sample.

The IIR filter calculates the product of the feedback coefficient and the previous output value, and adds the calculated result to the FIR convolution result to obtain the final filtered output.

35.3.2. Local memory and buffers

The unit contains 256 x 32 bit memory which can be read and write. X0 buffer and X1 buffer save the input values,and Y buffer saves the output values.

The locations of the buffers are specific as follows: x0_base, the X0 buffer base address, x1_base, the X1 buffer base address, y_base, the Y buffer base address.

The sizes of the buffers are specific as follows: x0_buf_size, the number of word allocated to the X0 buffer, x1_buf_size, the number of word addresses allocated to the X1 buffer, y_buf_size, the number of word addresses allocated to the Y buffer. Above parameters could be configured in corresponding register.

Through using the initialization functions,the X0 buffer, X1 buffer and Y buffer can be initialized. The data is transferred to the target buffer, which is indicated by a write pointer. The write pointer increase along with each new write, if the pointer arrives the end of the buffer space, the pointer returns to the base address. Thus, the vector element is loaded before operation, it can also be used to load the filter coefficients and initialize the filter.

Buffer configuration registers configure the buffer sizes and base address. The filter function specifies required buffer size for each function, while the base address in the local memory could be configured optionally, therefore, considering that all buffers address from 0x00 to

0xFF, in other words, base address add buffer size should be less than 256. The location and size of the buffers lack of constraint, even they can overlap completely. Do not overlap the buffer of filter function to avoid abnormal operation.

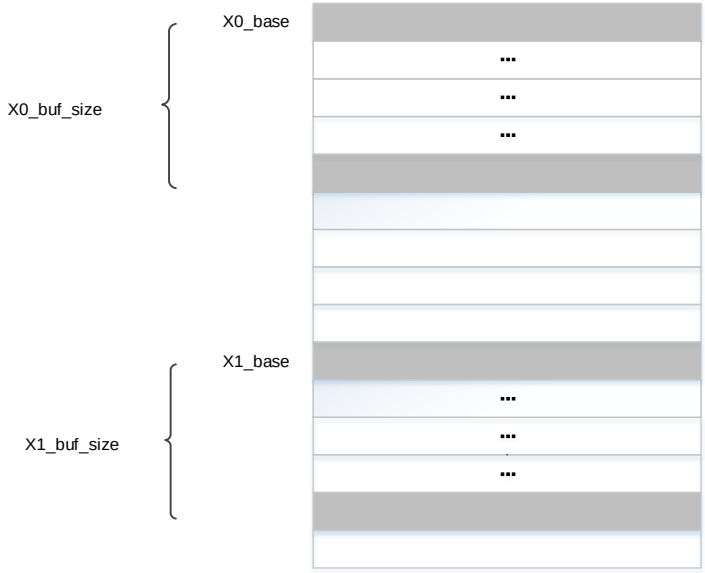
An optional headroom is added to the buffer size if circular buffer operation is required,. Moreover, for regulating the DMA or CPU activity, it is necessary to set watermark level. The headroom value and watermark level should be selected according to the actual application.

Usually, for more data throughput, the input buffer always has data, the headroom is slightly greater than the watermark level, so as to allow interrupt or DMA latency. On the other side, if the input data providing speed is less than the unit processing speed, the input buffer could be empty and wait for writing the next data. Therefore, the watermark level can be equal to headroom, so as to ensure that the input does not overflow.

35.3.3. Input buffers

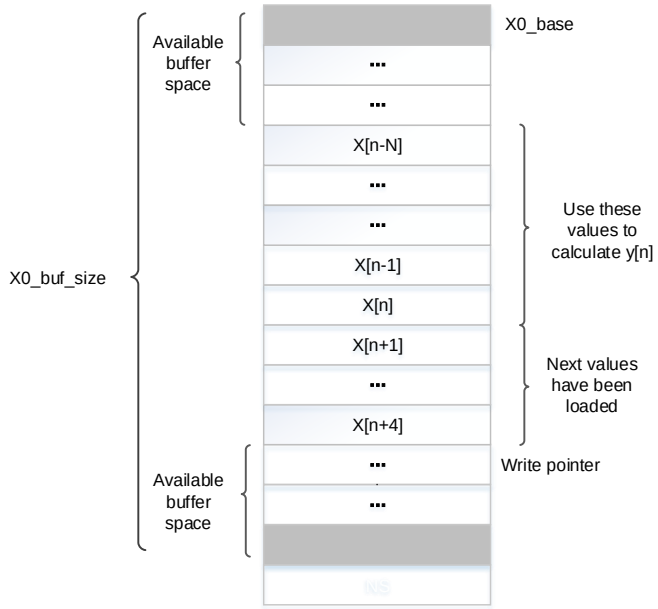
X0 buffer and X1 buffer store the input data of MAC, Every multiplication operation multiplies two data together, one data from X0 buffer and the other from X1 buffer. The read address offset is generated by the pointer of the control unit, which is relative to the buffer base address.

Figure 35-2 Input buffer area



If the X0 buffer works in circular mode, new data will continue to be transferred to the input buffer in case that space is available. For digital filters, preloading the buffer is optional. When the operation is started, if no input samples are written to X0 buffer, the buffer is flagged as empty. DMA or CPU are required to load new samples, The request will not disappear until there are enough samples to begin operation.

Figure 35-3 Circular input buffer area



The X1 buffer can only work as a not circular buffer. Unless the contents of the buffer do not follow operation change, X1 buffer usually needs to be pre-loaded. In addition, X1 buffer could store the filter coefficients for filter functions.

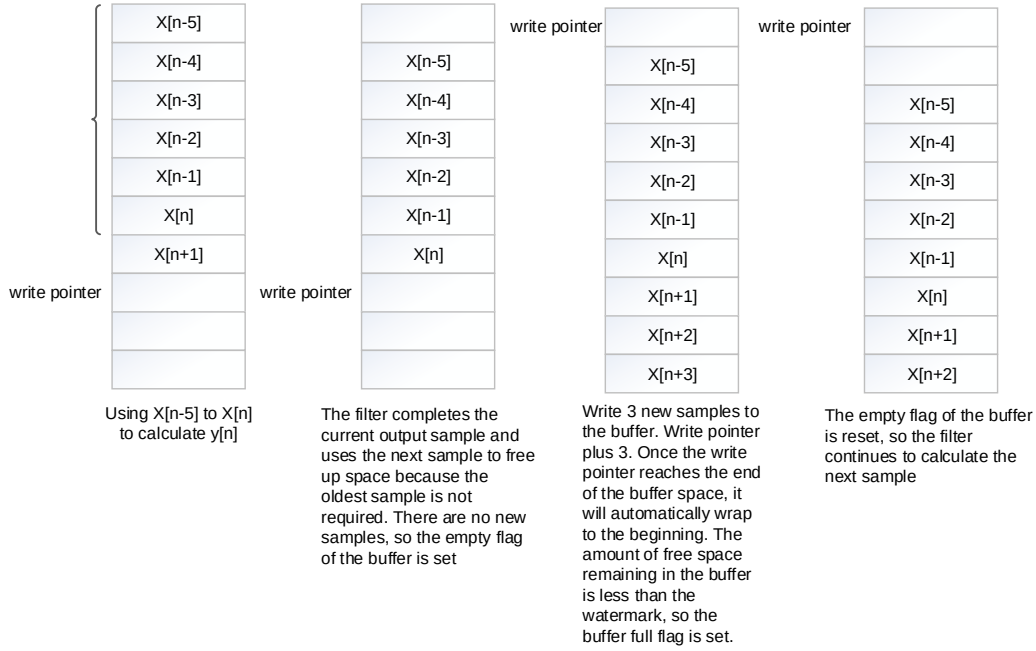
For a circular buffer, the allocated space of the buffer (x0_buf_size) should be greater than element numbers in use, therefore, new values are always available in the buffer. ***Figure 35-3 Circular input buffer*** show the buffer layout for a filter operation. For calculating an output sample y[n], a set of input samples is called, from x[n-N] to x[n], length is N+1. Once calculating is finished, call input samples x[n-N+1] to x[n+1], and then start calculating y[n+1]. The least-recent input sample (x[n-N]) is discarded, and a new sample(x[n+1]) is added.

It is necessary to ensure that the new sample x[n+1] in the buffer space is available if required. If x[n+1] is not available, the execution will be suspended and the buffer is flagged as empty, unless a new sample is added. If a timer or other peripheral control the flow of samples, considering that the source provide it is slower than the filter sample processing, the buffer work in empty states usually.

The watermark threshold is configured in the X0_WBFF bitfield of the FAC_X0BCFG register, If the amount of free space in X0 buffer is less than the watermark threshold, the X0 buffer is regarded as full state. Interrupts are generated if the full flag is not set, more data are requested for the buffer while FAC in enable,. Under one interrupt, the watermark permit that transferring several data without considering overflow risk. However,, the OFEF error flag is set if overflow occur, on the same time, the write data is ignored and the write pointer is not incremented.

Figure 35-4 Circular input buffer operation shows the change process of X0 buffer during an 6-tap FIR filtering processing,while the watermark is set to 3.

Figure 35-4 Circular input buffer operation

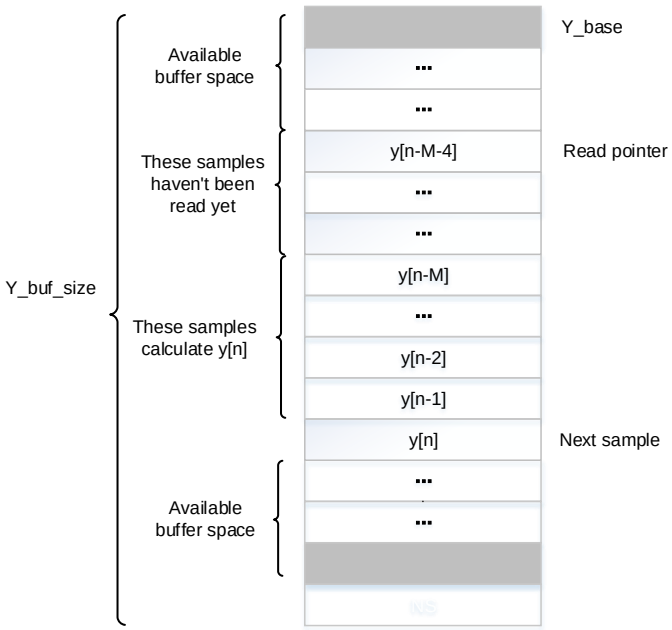


35.3.4. Output buffer

The output of an accumulation is held in Y buffer, the buffer space will be released when the output value is read by processor or DMA.

The read pointer points to the address where the data needs to be read during the read operation, the read data is taken out from the read pointer address when a read command occurs, meanwhile the read pointer is incremented. While pointer reaches the end of the Y buffer space, it returns back to base address.

Figure 35-5 Circular output buffer



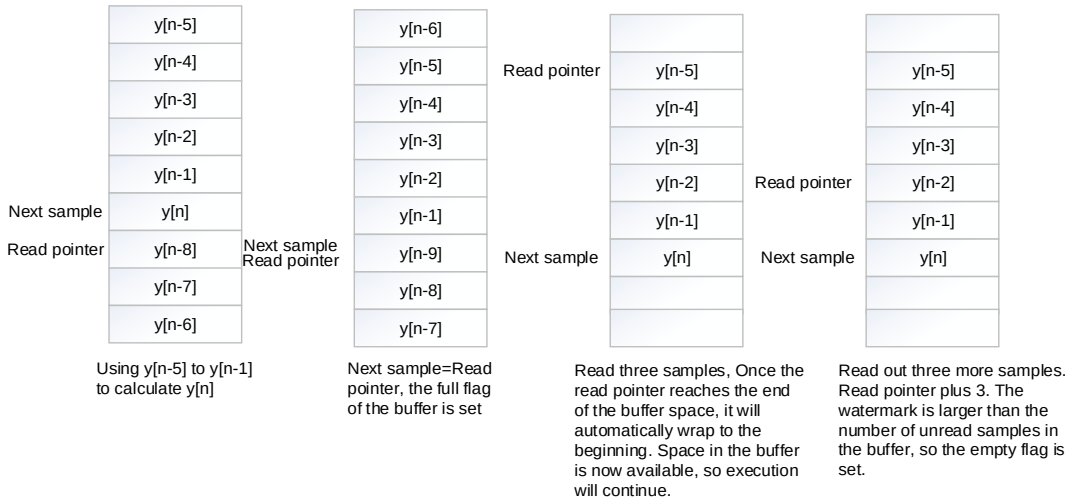
The Y buffer could work in circular buffer mode. The buffer full flag is set if the address for next output data is equal to the address which is indicated by the read pointer, and operation is stalled until the output data is read.

For IIR filters, the next output sample y[n] is calculated by M previous output samples y[n-M] to y[n-1], the least recent sample y[n-M] is discarded when a new sample is added.

If the watermark threshold, which is programmed in the Y_WBFF field of the FAC_YBCFG register, is no less than the number of unread data, the Y buffer is regarded as empty. In case that the empty flag is not set, if interrupt or DMA is enabled, the requests are generated to read data from the Y buffer. Without risk of underflow, several data could be transferred under one interrupt for watermark. However, in case of underflow occurring, UFEF flag is set and the read pointer stops incrementing, while the content of memory, which is addressed at the read pointer, is returned for read operation.

Figure 35-6 Circular output buffer area shows the change process of Y buffer during an 5-tap IIR filtering processing,while the watermark is set to 3.

Figure 35-6 Circular output buffer area



35.3.5. Initialization functions

The FAC unit is initialized by writing the proper value in the FUN bitfield of the FAC_PARACFG register when the EXE bit is setting. The IPP and IPQ bitfields contain values that need to be preloaded, while IPR bitfield is not used. The EXE bit is reset by hardware automatically, as the Initialization function completed.

DMA requests and interrupts should be disabled during initialization, since flow control is not required, data can be transferred to FAC memory through DMA transfer or software.

X0 buffer loading function

This loading function pre-loads values from the address X0_BASE, and the write data is loaded into the X0 buffer from FAC_WDATA register, at the same time, the write address is increasing. When N values have been loaded into the X0 buffer, the write pointer finally points

to the address X0_BASE + N. The parameter IPP contains N, the number of values, which is loaded into the X0 buffer, while IPQ and IPR are not used. This function is completed when the N write operations to the FAC_WDATA register are completed.

X1 buffer loading function：

This loading function pre-loads values from the address X1_BASE, and the write data is loaded into the X1 buffer from FAC_WDATA register, at the same time, the write address is increasing. In IIR filter, N feed-forward and M feed-back coefficients are loaded into X1 buffer, The coefficients sum is N+M. In FIR filter, since feedback coefficients are absent, M is equal to 0, N feed-forward coefficients are loaded into X1 buffer.

The parameter IPP contains N feed-forward coefficients and the parameter IPQ contains M feed-back coefficients, Both IPP and IPQ are loaded into X1 buffer, where the starting address of IPP is X1_BASE and the starting address of IPQ is X1_BASE+N. IPR is not used in loading X1 buffer. This function is completed when the N + M write operations to the FAC_WDATA register are completed.

Y buffer loading function：

This loading function pre-loads values from the address Y_BASE, and the write data is loaded into the Y buffer from FAC_WDATA register, at the same time, the write address is increasing the write pointer finally points to the address Y_BASE + N. By this function, the feedback storage element of the IIR filter can be preloaded. The parameter IPP contains N, the number of values, which is loaded into the Y buffer, while IPQ and IPR are not used. This function is completed when the N write operations to the FAC_WDATA register are completed.

35.3.6. Filter functions

Writing the appropriate value in the FUN bitfield of the FAC_PARACFG register can trigger FIR or IIR filter functions when EXE bit is setting. The IPP, IPQ, and IPR fields contain the suitable parameter values for each filter function, The filter function runs all the time except that the software resets the EXE bit.

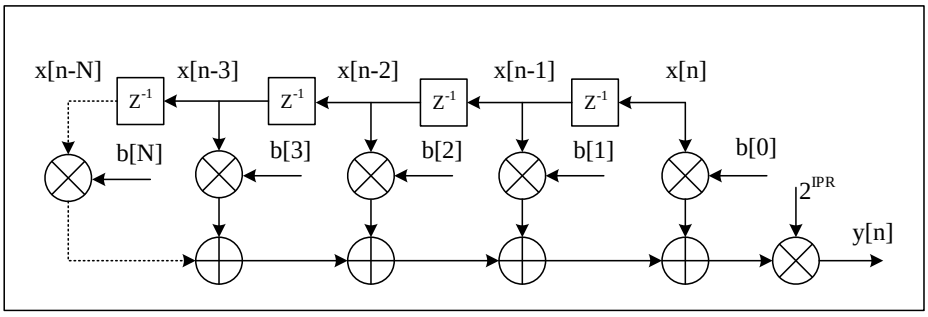
FIR filter: $\underline{Y} = \underline{B} * \underline{X}$

$$y_n = 2^{IPR} \sum_{k=0}^{N} (b_k \times x_{n-k}) \quad (35\text{-}1)$$

vector $\underline{B}$ contains N + 1 filter coefficients, vector $\underline{X}$ contains indefinite length input samples, The elements of $\underline{Y}$ is calculated as the dot product, $y_n = \underline{B}.\underline{X}_n$, and $\underline{X}_n = [x_{n-N},...,x_n]$. This function conforms to a finite impulse response (FIR) filter.

FIR filter structure is shown as ***Figure 35-7 The structure of FIR fliter***.

Figure 35-7 The structure of FIR fliter function



X0 buffer is a circular buffer and composed of the elements of vector X, the length of buffer is N + 1 + d(d is the length of headroom). X1 buffer is composed of the elements of vector B, the length of buffer is N + 1. Y buffer is a circular buffer and composed of the output values(y_n), the length of buffer is d. The length of the parameter IPP is N+1, The vector B is in the range[2:127]. The parameter IPR is the gain, applied to the accumulator output by multiplied 2^{IPR}, where IPR is in the range [0:7]. Parameter IPQ is not used.

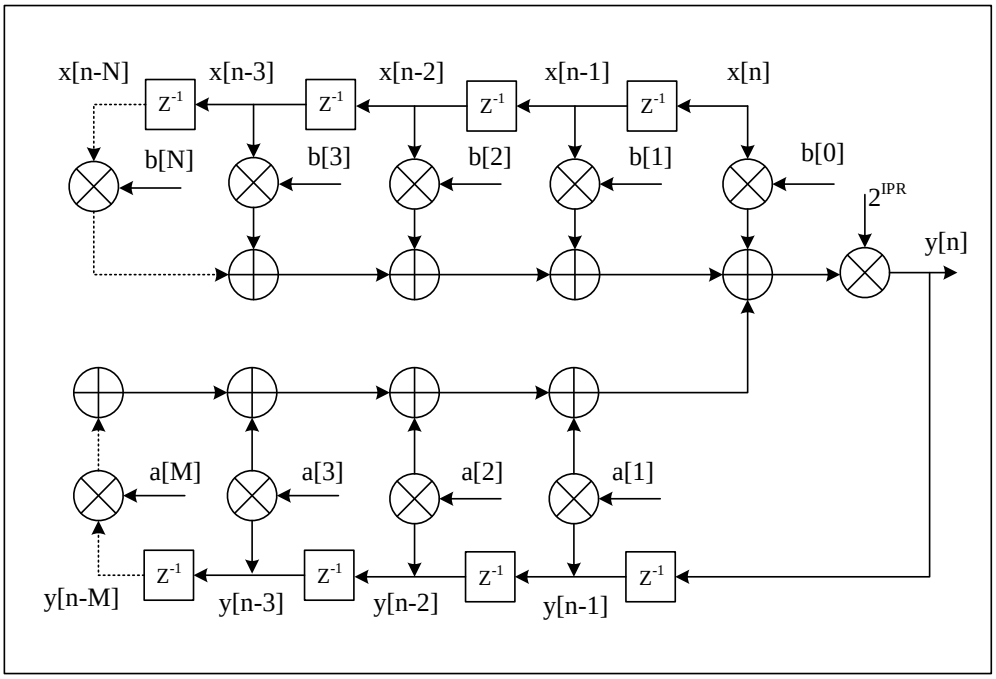
IIR filter: Y = B*X+A*Y

$$y_n = 2^{IPR}\left(\sum_{k=0}^{N}(x_{n-k}\times b_k) + \sum_{k=1}^{M}\left(y_{n-k}\times a_k\right)\right) \quad (35\text{-}2)$$

The infinite impulse response (IIR) filter output vector Y which is the convolution of a coefficient vector B (length is N+1) and a vector X (length is indefinite), add the convolution of vector Y with vector A (length is M). The elements of Y are calculated by B.X_n + A.Y_{n-1}, where X_n = [x_{n-N},...,x_n] is composed of the N+1 elements, while Y_{n-1}= [y_{n-M},...,y_{n-1}] is composed of the M elements.

IIR filter structure is shown as***Figure 35-8 The structure of IIR fliter***.

Figure 35-8 The structure of IIR fliter



X0 buffer is a circular buffer and composed of the elements of vector X, the length of buffer is N + 1 + d. X1 buffer is composed of the elements of vector B and A, the length of buffer is M + N + 1. Y buffer is a circular buffer and composed of the output values(y_n), the length of buffer is M + d. The length of parameter IPP is N+1, the coefficient vector B is in the range [2:64]. The parameter IPR is the gain, applied to the accumulator output by multiplied 2^{IPR}, where IPR is in the range [0:7]. The length of parameter IPQ is M, the coefficient vector A is in the range [1:63].

35.3.7. Fixed point data format

The FAC operates input and output values in fixed signed integer format (q1.15), 1 is integer sign and 15 is fractional. The numeric range is from -1 (0x8000) to 1 - 2^{-15} (0x7FFF).

The accumulator value format is q4.22, the format value contains 26 bits, 4 are integer sign and 22 are fractional, support accumulation sums in the range from -8 (0x2000000) to +7.99999976(0x1FFFFFF). In steps of 6dB, a programmable gain can be used to the accumulator output, from 0dB to 42dB.

If the numeric range is exceeded, the accumulator content is not saturated. If value is more than +7.99999976 or less than -8, triggered wrap is harmless, since subsequent accumulations undo the wrapping. However, if wrapping occurs, STEF flag in the FAC_STAT register is set, and then, if STEIE flag in the FAC_CTL register is set, corresponding interrupt is generated.

When the CPEN bit of FAC_CTL register is set, the data output of accumulator could be saturated optionally after the programmable gain application. Any value is forced output as 1 - 2^{-15} or -1 while the value exceeds the q1.15 numeric range. While using the gain, If CPEN

bit is not set, the unused accumulator bits are truncated simply.

35.3.8. Float point data format

The operation data and calculation result data format is given in ***Table 35-1 IEEE 32-Bit Single Precision Floating-Point Format***. They must meet IEEE 32 bit Single Precision Floating-Point.

Table 35-1 IEEE 32-Bit Single Precision Floating-Point Format

S [31]	E [30:23]	M [22:0]	Value (V)
0	0	0	Zero (V = 0)
1	0	0	Negative Zero (V = -0)
0 + ve 1 - ve	0	non zero	De-normalized (V=$(-1)^{s}*2^{(-126)}*$ (0.M))
0 + ve 1 - ve	1 to 254	0 to 0x7FFFFF	Normal Range(V=$(-1)^{s}*2^{(E-127)}*$ (1.M))
0	254	0x7FFFFF	Positive Max (V = +Max)
1	254	0x7FFFFF	Negative Max (V = -Max)
0	max=255	0	Positive Infinity (V = +Infinity)
1	max=255	0	Negative Infinity (V = -Infinity)
x	max=255	non zero	Not A Number (V = NaN)

The treatment of the various IEEE floating-point numerical formats for this FAC is given as below:

De-Normalized Numbers: A de-normalized operand (E=0, M!=0) input is treated as zero (E=0, M=0).

Overflow: Overflow occurs when an operation generates a value that is too large to represent in the given floating-point format. Under such cases, a positive or negtive Infinity value is returned, and STEF flag in the FAC_STAT register is set.

Not a Number (NaN): An NaN operand (E=max, M!=0) input is treated as infinity (E=max, M=0).

Note: configure FLTEN bit in FAC_CTL register as set, input and output data in 32bit IEEE data format, not support clip and gain of output.

35.3.9. FIR filters

The FAC supports FIR filters, and the number of taps or coefficients is N. FIR filters require a minimum length of local memory of 2N + 1: N input samples ,1 output sample and N coefficients,. the maximum size for N is 127 while the local memory size is 256,. A small amount of additional space is allocated for maximum throughput, d0 is for input sample buffer and d1 is for output sample buffer, so as to guarantee the filter does not stop to wait for a new input sample or reading the output sample. 2N + d0 + d1 is the required local memory.

X0_BUF_SIZE is equal to N + d0, X1_BUF_SIZE is equal to N and Y_BUF_SIZE is equal to d1 (Y_BUF_SIZE could be equal to 1 if no extra space is needed).

Even though the user can arbitrarily allocate the buffer base address, it is necessary to avoid that the X1 buffer overlap with the other buffer, otherwise, the coefficients are overwritten. For example, X1_BASE is equal to 0, X0_BASE is equal to N, and Y_BASE is equal to 2N+d0. However, the X0 and Y buffers may overlap if the memory space is limited, such as X1_BASE is equal to 0, X0_BASE is equal to N, and Y_BASE is equal to N. The output sample would replace the oldest input sample. Since Y_BUF_SIZE = X0_BUF_SIZE = N + d0, the buffers hold in sync still.

Note: The X0_WBFF field of X0BCFG register must be programmed not more than $\log_2(d0)$, or else, before writing N input samples, the buffer is flagged as full, then the samples are no longer needed. In the same way, the Y_WBEF field of YBCFG register must be programmed not more than $\log_2(d1)$.

The X1 buffer must preload the filter coefficients. Any number of samples, which is up to N, could be preloaded into the X0 buffer. Since feedback path is not needed for the FIR filter, pre-loading the Y buffer is unnecessary.

The FAC_CTL register should be programmed depend on polling, interrupt and DMA, which is used for writing data to FAC memory and reading data from FAC memory. In polling method, software should confirm that the X0BFF bit is reset or YBEF bit is reset before writing to WDATA or reading from RDATA. In interrupt method, the interrupt request is launched when the X0BFF bit is reset for writing, or the YBEF bit is reset for reading. In DMA method, DMA requests are launched on the DMA write channel or read channel, while the X0BFF bit is reset or the YBEF bit is reset.

Writing the following values in the FAC_PARACFG register, thus the filter is started.

FUN= 8 (FIR filter); IPP = N (number of coefficients); IPQ = “any value”; IPR= Gain; EXE = 1.

If the number of values preloaded in the X0 buffer is less than $N + d - 2^{X0_WBFF}$, the X0BFF flag is in low state. If the WIE bit is set, writing 2^{X0_WBFF} samples into X0 buffer through the FAC_WDATA register is triggered by the write interrupt request. When 2^{X0_WBFF} values have been written to FAC_WDATA register, The interrupt handler check the X0BFF flag unless the X0BFF bit in FAC_STAT register is set. In the same way, if DWEN bit is set in the FAC_CTL register, constantly generate DMA write channel request, unless the X0BFF bit in FAC_STAT register is set.

When samples (at least N) have been written into the X0 buffer, First output sample is calculated by the filter. When writing 2^{Y_WBEF} output samples into the Y buffer,the YBEF bit in the FAC_STAT register is reset, the interrupt is request to read 2^{Y_WBEF} samples from the buffer, if the RIE bit is set in the FAC_CTL register. The interrupt handler should repeatedly check the YBEF flag after every 2^{Y_WBEF} values have been read from FAC_RDATA register unless the YBEF bit in FAC_STAT register is set. In the same way, if DREN bit is set in the FAC_CTL register, constantly generate DMA read channel request, unless the YBEF bit in FAC_STAT register is set. Resetting the EXE bit could halt the filter operation, or else, the

filter will continue to operate.

35.3.10. IIR filters

The FAC supports IIR filters with length N (the number of coefficients or feed-forward taps) and M (the number of feedback coefficients, which can be configured from 1 to N-1).

The minimum memory required for an IIR filter is 2N + 2M, which include N feed-forward coefficients and M feed-back coefficients, N input samples and M output samples. In case that M is equal to N-1, the maximum filter length N = 64 can be implemented.

For maximum throughput, there is an additional space d0 for input buffer size, and d1 for output buffer size, should be allowed to configured, so the total memory requirement 2M + 2N + d0 + d1. X0_BUF_SIZE = N + d0, X1_BUF_SIZE = N + M and Y_BUF_SIZE = M + d1. The buffer base address must not overlap, even if it is allocated anywhere, such as: X1_BASE = 0, X0_BASE = N+M and Y_BASE = 2N+M+d0.

Note: The WBFF field of X0BCFG register must be programmed not more than $\log_2(d0)$, or else, before writing N input samples, the full flag of the buffer is set, and then no more sample points are needed. In the same way, the WBEF field of YBCFG register must be programmed not more than $\log_2(d1)$.

By using the Load X1 buffer function, the X1 buffer must preload the filter coefficients (N feed-forward and M feedback). Any number of samples, which is up to a maximum of N, could be preloaded into the X0 buffer. In the same way, any number of samples, which is up to a maximum of M, could be preloaded into the Y buffer.

Writing the following values in the FAC_PARACFG register, thus the filter is started.

FUN= 9 (IIR filter); IPP = N (number of feed-forward coefficients);

IPQ = M (number of feed-back coefficients); IPR= Gain; EXE = 1.

If the number of values, which have been preloaded in the X0 buffer, is less than N + d – 2^{X0_WBFF}, the X0BFF flag is held in low state. If the WIE bit is set in the FAC_CTL register, When 2^{X0_WBFF} values have been written to FAC_WDATA register, The interrupt handler check the X0BFF flag unless the X0BFF bit in FAC_STAT register is set.In the same way, if DWEN bit is set in the FAC_CTL register, constantly generate DMA write channel request, unless the X0BFF bit in FAC_STAT register is set.

When samples (at least N) have been written in the X0 buffer, first output sample is calculated by the filter, which is calculated by using the X0 buffer first N samples and the Y buffer first M samples. The address at where first output sample write into Y buffer is Y_BASE+M.

When writing 2^{Y_WBEF} output samples in the Y buffer, the YBEF bit in the FAC_STAT register is reset. the interrupt request to read 2^{Y_WBEF} samples from the buffer, if the RIE bit is set in the FAC_CTL register. The interrupt handler should repeatedly check the YBEF flag after every 2^{Y_WBEF} values have been read from FAC_RDATA register unless the YBEF bit in FAC_STAT register is set. In the same way, if DREN bit is set in the FAC_CTL register,

constantly generate DMA read channel request, unless the YBEF bit in FAC_STAT register is set. Resetting the EXE bit could halt the filter operation, or else, the filter will continue to operate.

35.4. Register definition

FAC base address: 0x4802 4800

35.4.1. FAC X0 buffer configure register (FAC_X0BCFG)

Address offset: 0x00
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit) and can only be modified if EXE = 0 in the FAC_PARACFG register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						X0_WBFF[1:0]		Reserved							
						rw									

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X0B_SIZE[7:0]								X0B_ADDR[7:0]							
rw								rw							

Bits	Fields	Descriptions
31:26	Reserved	Must be kept at reset value.
25:24	X0_WBFF[1:0]	Buffer full flag of watermark. If free spaces number in the buffer is less than 2^{X0_WBFF}, the flag is set. 00: Threshold = 1(if DMA write requests are enabled) 01: Threshold = 2 10: Threshold = 4 11: Threshold = 8 Under one interrupt, if several data would be transferred into the buffer, the threshold should be set more than 1.
23:16	Reserved	Must be kept at reset value.
15:8	X0B_SIZE[7:0]	X0 buffer size, the number of feed-forward taps.
7:0	X0B_ADDR[7:0]	X0 buffer base address

35.4.2. FAC X1 buffer configure register (FAC_X1BCFG)

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

X1B_SIZE[7:0]	X1B_ADDR[7:0]
rw	rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:8	X1B_SIZE[7:0]	X1 buffer size When FAC is running (EXE = 1), this bitfield can not be changed.
7:0	X1B_ADDR[7:0]	X1 buffer base address When FAC is running (EXE = 1), this bitfield can be changed. For example, When changing the coefficient value, the filter should be paused, because changing the factor during the calculation will affect the results.

35.4.3. FAC Y buffer configure register (FAC_YBCFG)

Address offset: 0x08

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit) and can only be modified if EXE = 0 in the FAC_PARACFG register.

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

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

Bits	Fields	Descriptions
31:26	Reserved	Must be kept at reset value.
25:24	Y_WBEF[1:0]	Buffer empty flag of watermark If the number of unread data in the buffer is less than 2^{Y_WBEF}, the flag is set. 00: Threshold = 1 01: Threshold = 2 10: Threshold = 4 11: Threshold = 8 Under one interrupt, if several data would be transferred from the buffer, setting the threshold more than 1. If DMA read command is enabled, threshold should be set to 1.
23:16	Reserved	Must be kept at reset value.
15:8	YB_SIZE[7:0]	Y buffer size The minimum buffer size is the watermark threshold + 1 for FIR filters. the minimum buffer size is the watermark threshold + the number of feedback taps

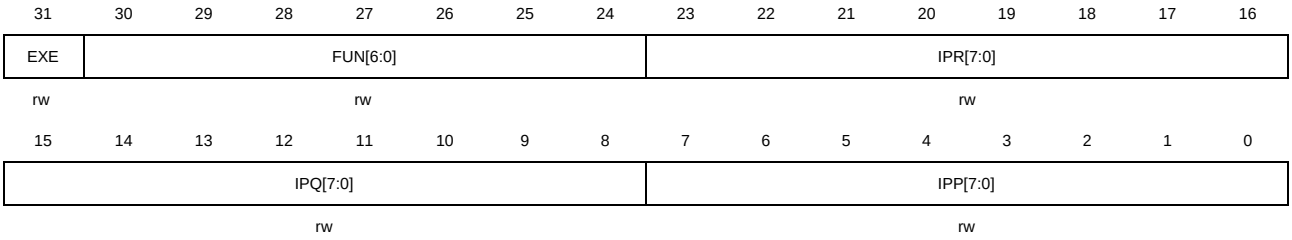
		for IIR filters.
7:0	YB_ADDR[7:0]	Y buffer base address

35.4.4. FAC Parameter configure register (FAC_PARACFG)

Address offset: 0x0C

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31	EXE	Execution 0: Stop execution 1: Start execution This bit is set in order to execute the function of the FUN bitfield, FAC will stop any ongoing functions through software reset. This bit is reset by hardware for initialization functions.
30:24	FUN[6:0]	Function 0000001: Load X0 buffer 0000010: Load X1 buffer 0000011: Load Y buffer 0001000: FIR filter 0001001: IIR filter others: Reserved This register can only be modified when EXE = 0 in the FAC_PARACFG register.
23:16	IPR[7:0]	Input parameter IPR This register can only be modified when EXE = 0 in the FAC_PARACFG register.
15:8	IPQ[7:0]	Input parameter IPQ This register can only be modified when EXE = 0 in the FAC_PARACFG register.
7:0	IPP[7:0]	Input parameter IPP This register can only be modified when EXE = 0 in the FAC_PARACFG register.

35.4.5. FAC Control register (FAC_CTL)

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															RST
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPEN	FLTEN	Reserved				DWEN	DREN	Reserved		GSTEIE	STEIE	UFEIE	OFEIE	WIE	RIE
rw	rw					rw	rw			rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:17	Reserved	Must be kept at reset value.
16	RST	Reset FAC unit 0: Reset disable 1: Reset enable The write pointer and read pointer, EXE bit, FMAC_CTL register and FMAC_PARACFG register will be reset when RST is 1.
15	CPEN	Clipping enable 0: Clipping disable, the value of accumulator output out of range is truncated 1: Clipping enable, the value of accumulator output out of range is saturated to the maximum positive value or maximum negative value.
14	FLTEN	Floating point format enable 0: Input data and result support fixed point data format q1.15 1: Input data and result support floating point data format This bit can only be modified if EXE = 0 in the FAC_PARACFG register.
13:10	Reserved	Must be kept at reset value.
9	DWEN	DMA write channel enable 0: DMA request is not generated 1: DMA request is generated while the X0 buffer is not full. This bit can only be modified if EXE = 0 in the FAC_PARACFG register.
8	DREN	DMA read channel enable 0: DMA request is not generated 1: DMA request is generated while the Y buffer is not empty. This bit can only be modified if EXE = 0 in the FAC_PARACFG register.
7:6	Reserved	Must be kept at reset value.
5	GSTEIE	Gain saturation error interrupt enable 0: No interrupts are generated. 1: An interrupt request is generated while the GSTEF flag is set Software set and clear this bit.
4	STEIE	Saturation error interrupt enable

		0: No interrupts are generated. 1: An interrupt request is generated while the STEF flag is set Software set and clear this bit.
3	UFEIE	Underflow error interrupt enable 0: No interrupts are generated. 1: An interrupt request is generated while the UFEF flag is set Software set and clear this bit.
2	OFEIE	Overflow error interrupt enable 0: No interrupts are generated. 1: An interrupt request is generated while the OFEF flag is set Software set and clear this bit.
1	WIE	Write interrupt enable 0: No interrupts are generated. 1: An interrupt request is generated if the X0BFF flag is set Software set and clear this bit.
0	RIE	Read interrupt enable 0: No interrupts are generated. 1: An interrupt request is generated if the YBEF flag is set Software set and clear this bit.

35.4.6. FAC Status register (FAC_STAT)

Address offset: 0x14

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				GSTEF	STEF	UFEF	OFEF	Reserved						X0BFF	YBEF
				r	r	r	r							r	r

Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11	GSTEF	Gain saturation error flag, it is set when gain exceed range 0: No gain saturation error detected 1: Gain saturation error detected.
10	STEF	Saturation error flag 0: No saturation error detected 1: Saturation error detected.

		Saturation occurs when the cumulative result exceeds the range of the value
9	UFEF	Underflow error flag 0: No underflow error detected 1: Underflow error detected. When there is no valid data available in the Y buffer, underflow occurs when reading from FAC_RDATA
8	OFEF	Overflow error flag 0: No overflow error detected 1: Overflow error detected When there is no free space in X1 buffer, overflow occurs when writinging to FAC_WDATA.
7:2	Reserved	Must be kept at reset value.
1	X0BFF	X0 buffer full flag 0: X0 buffer is not full. 1: X0 buffer is full. Hardware or a reset will set and clear this bit.
0	YBEF	Y buffer empty flag 0: Y buffer is not full. 1: Y buffer is full. Hardware or a reset will set and clear this bit.

35.4.7. FAC write data register (FAC_WDATA)

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

When FLTEN =1, float point data format is selected;

Bits	Fields	Descriptions
31:0	WDATA[31:0]	Write data When a write command is performed on the register, the write data is transferred to the address offset which is pointed to by the write pointer. After each write of data is completed, The pointer address is incremented.

When FLTEN =0, fix point data format is selected;

Bits	Fields	Descriptions

31:16	Reserved	Must be kept at reset value.
15:0	WDATA[15:0]	Write data When a write command is performed on the register, the write data is transferred to the address offset which is pointed to by the write pointer. After each write of data is completed, The pointer address is incremented.

35.4.8. FAC read data register (FAC_RDATA)

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
RDATA															
r															

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

When FLTEN =1, float point data format is selected;

Bits	Fields	Descriptions
31:0	RDATA[31:0]	Read data When a read command is performed on the register, The contents in the Y buffer which is pointed to by the read pointer are the read data. When each read data is completed, The pointer address is incremented.

When FLTEN =0, fix point data format is selected;

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	RDATA[15:0]	Read data When a read command is performed on the register, The contents in the Y buffer which is pointed to by the read pointer are the read data. When each read data is completed, The pointer address is incremented.

36. Document appendix

36.1. List of abbreviations used in registers

Table 36-1. List of abbreviations used in register

abbreviations for registers	Descriptions
read/write (rw)	Software can read and write to this bit.
read-only (r)	Software can only read this bit.
write-only (w)	Software can only write to this bit. Reading this bit returns the reset value.
read/clear write 1 (rc_w1)	Software can read as well as clear this bit by writing 1. Writing 0 has no effect on the bit value.
read/clear write 0 (rc_w0)	Software can read as well as clear this bit by writing 0. Writing 1 has no effect on the bit value.
toggle (t)	Software can toggle this bit by writing 1. Writing 0 has no effect.
read-only/set by write 1 (rt_w1)	Software can only read to this bit. Writing 1 trigges the event but has no effect on the bit value.
read/set (rs)	Software can read as well as set this bit to 1. Writing 0 has no effect on the bit value.
read/clear by read (rc_r)	Software can read this bit. Reading this bit automatically clears it to '0'. Writing '0' has no effect on the bit value.
read/set by read (rs_r)	Software can read this bit, and set this bit by reading. Writing has no effect on the bit value.
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/clear write (rc_w)	Software can read as well as clear this bit by writing. Writing 0 or 1 has the same effect to this bit.
read-only/write trigger (rt_w)	Software can read this bit. Writing 0 or 1 triggers an event but has no effect on the bit value.

36.2. List of terms

Table 36-2. 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	Writing 0 has no effect IAP is the ability to re-program the Flash memory of a

Glossary	Descriptions
programming)	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.

36.3. Available peripherals

For availability of peripherals and their number across all MCU series types,refer to the corresponding device data datasheet.

37. Revision history

Table 37-1. Revision history

Revision No.	Description	Date
1.0	Initial Release	Nov.08, 2024
1.1	1. Change the description of the lower 2 bits in ***System configuration register 3 (SYSCFG_CFG3)***, ***System configuration register 4 (SYSCFG_CFG4)***, and ***System configuration register 5 (SYSCFG_CFG5) registers***. 2. Change the description of the bit 0 in ***SYSCFG TCMSRAM control and status register(SYSCFG_TCMSRAMCS)***. 3. Add the description: "When TCMSRAM is set to write protection, operations on it will generate a hardfault" in chapter ***TCMSRAM write protection***. 4. Delete the Asymmetric PWM mode chapter and its related content. 5. Change the ***Figure 25-58. Trigger to DAC when TRIG0M = 1 and TRIG1M = 1*** in chapter ***25.4.10. Double source trigger***. 6. Change the ***Table 25-17. Filtering signals mapping in windowing mode***, ***Figure 25-36. ISO0 = 0 and CHOP =1 in delayed IDLE*** and ***Figure 25-37. ISO0 = 1 and CHOP =1 in delayed IDLE*** in chapter ***25.4.2. Slave_TIMERx(0..7) unit***. 7. Change the description of ***Bunch mode entry*** in chapter ***25.4.4. Bunch mode***. 8. Change the description of TRGHALFM in *HRTIMER Slave_TIMERx control register 1 (HRTIMER_STxCTL1)*. 9. Change the description in ***Fault blank*** chapterfrom "When the number of fault channel event is equal to FLTyCNT[3:0], the fault is takeing effect" to "When the number of fault channel event is equal to (FLTyCNT[3:0]+1), the fault is takeing effect". 10. Change the ***Figure 17-1. ADC module block diagram***. 11. Change the ***Table 17-2. ADC input pins definition***. 12. Add "Note" description in chapter ***17.4.2. Dual clock domain architecture***. 13. Change the ***Figure 33-3. Four regions of bank0 address mapping*** in chapter ***33.3.3 External device address mapping***. 14. Change ***Figure 4-1. The system reset circuit***.	Feb.19, 2025

	15. Change the description of ***4.2.1. Overview*** in chapter ***4. Reset and clock unit*** chapter, change the ***Figure 4-2. Clock tree***. 16. IRC40K is changed to IRC32K in the documentation. 17. Change the ***Table 24-2. External trigger mapping*** in chapter ***24.4.7. External trigger mapping***. 18. Change the ***Table 6-1. Tfigger input bit fields selection*** in chapter ***6.4. Internal connect***. 19. Add “Note” description to ***Bus integration state*** in chapter ***39.3.9. Errors and states***. 20. Change description of ***Counter clock source*** in chapter ***29.3.10. Communication parameters***. 21. Change description of BAUDPSC[9:0] bit field in ***Bit timing register (CAN_BT)*** and ***FD bit timing register (CAN_FDBT)***. 22. Change bit field description of ***Receive mailbox public filter register (CAN_RMPUBF)***. 23. Change the port reset value of the GPIO in chapter ***7.4. Register definition***. 24. Change the ***Table 5-2. Interrupt vector table***. 25. Change the ***Figure 31-1 QSPI diagram***. 26. Change the description of DLYSCEN bit in ***Device configuration register (QSPI_DCFG)***. 27. Add description “When the operation is executed successfully and the operation end interrupt is enabled (ENDIE = 1), the ENDF in FMC_STAT register is set” in chapter ***2.3.7. Mass erase***, ***2.3.8. Main flash programming***, ***2.3.9. OTP programming*** and ***Status register (FMC_STAT)***. 28. Change the ***Figure 3-1. Power supply overview*** and add the ***Figure 3-2. 1.1V domain supply overview***.	
1.2	1. Modify the description of the DQS signal in ***Table 31-1. QSPI signal description***. 2. Change HRTIMER_MTCARL to HRTIMER_MTCAR in ***Table 25-4. Alternate mode selection***. 3. Change HRTIMER_STxCARL to HRTIMER_STxCAR in ***Table 25-8. Compare values for the two alternate modes***. 4. Delete the description stating that differential mode channels cannot be sampled in single-ended mode in Section ***17.4.4. Single-enden and differential input channels***. 5. In the LPTIMER section, change "internal 16 MHz RC" to	Aug.08, 2025

	"internal 8 MHz RC". 6. In the HRTIMER chapter on ***External event counter***, change " external event is only valid when the value of EXEVXCNT[5:0] is grater or equal to EXEVXCNTTHR[5:0]" to " external event is only valid when the value of EXEVXCNT[5:0] is grater or equal to EXEVXCNTTHR[5:0]. + 1". 7. In the HRTIMER chapter on ***External event counter***, change EXEVXCNT[5:0] to EXEVXCNTTHR[5:0] in the description: " The EXEVXCEN bit should be set after writing EXEVXCNT[5:0] bits, and the EXEVXCNT[5:0] bits can be changed when External event X counter is enabled. And the new value of EXEVXCNT[5:0] will be take effect in next period. The EXEVSEL[3:0] can not be changed when external event X counter is enabled". 8. In the HRTIMER chapter on ***25.4.8. Trigger to ADC***, change ROVM to ADCROVM, modify Figure 25 56. ADC trigger division in center aligned mode, and supplement the description of ADC trigger events. 9. Modify the ADCROVM bit field description in the ***HRTIMER Slave_TIMERx control register 1 (HRTIMER_STxCTL1)*** in the HRTIMER chapter. 10. In the ***HRTIMER Slave TIMERx counter reset and channel 0/1 set/reset request add register (HRTIMER_STxCNTRSTA)***, modify the bit names and bit field descriptions for Slave_TIMER5 - Slave_TIMER7 (bits 20, 21, 28, 29). 11. Modify ***Figure 25-44. External event X counter when EXEVXCEN bit is 1 and EXEVXCNTTHR[5:0] = 2*** in the HRTIMER chapter. 12. Modify the ALTM[1:0] bit field descriptions in the ***HRTIMER Master_TIMER control register 0 (HRTIMER_MTCTL0)*** and ***HRTIMER Slave TIMERx control register 0 (HRTIMER_STxCTL0)***. 13. Change the FLTENPROT bit field description in the ***HRTIMER Slave TIMERx fault control register (HRTIMER_STxFLTCTL)*** from "FLTyEN(y=0..4)" to "FLTyEN(y=0..7)". 14. Change DTGCKDIV[2:0] to DTGCKDIV[3:0] in the ***HRTIMER Slave_TIMERx dead-time control register (HRTIMER_STxDTCTL)***. 15. Modify the names of bits 0-4 in the ***HRTIMER trigger source 1 to ADC add register***	

	(HRTIMER_ADCTRIGS1A), ***HRTIMER trigger source 2 to ADC add register (HRTIMER_ADCTRIGS2A)***, and ***HRTIMER trigger source 3 to ADC add register (HRTIMER_ADCTRIGS3A)***. 16. Change bit 7 from BMCH1DTI to BMCH0DTI in the ***HRTIMER Slave TIMERx channel output control register (HRTIMER_STxCHOCTL)***. 17. Modify the bit field descriptions for bits 1-3 in the ***HRTIMER Slave TIMERx counter reset register (HRTIMER_STxCNTRST)***. 18. Modify the pin configuration description in Section ***27.3.11: Half-duplex communication mode***. 19. Add a new register: ***HRTIMER fault recovery control register (HRTIMER_FLTRECCTL)*** in the HRTIMER section. 20. Modify the description of the ***Counter clock*** subsection in the HRTIMER section. 21. Modify the FLT0RST bit field description in the HRTIMER Fault Input Configuration Register 2 (HRTIMER_FLTINCFG2). 22. Modify the LXTALDRI bit field description in the ***Backup domain control register (RCU_BDCTL)***. 23. Modify the captions for ***Figure 25 3. Counter clock when divided by 16*** and ***Figure 25 9 Reset event resynchronization when prescaling ratio is 64***. 24. Add a subsection on ***Fault resume*** in Section ***25.4.7. Fault Input***. 25. Add a "Note:" in Section ***25.4.10. Double source trigger*** to explain the scope and limitations of slope compensation. 26. Modify the description of FLTxRST in Section ***25.4.7. Fault Input***. 27. Change the second paragraph in Section ***25.4.7. Fault Input*** to "When a fault event occurs, the output (STxCHy_O, x = 0..7, y = 0,1) is preset to a level." 28. Modify ***Table 25 23. Source of counter reset*** in Section ***25.4.7. Fault Input***. 29. Update the text description in ***Figure 25 52. Fault counter when FLTxRST bit is 1, FLTRECCTL bit is 1 and FLTxCNT[3:0] = 0x03***. 30. In the ADC section, remove the LXTAL clock and specify that only the IRC32K clock can be used in the ***Sample and keep mode***. 31. Modify the IMSEL[1:0] bit field description in the FFT	

	Control and status register (FFT_CSR). 32. Modify ***Figure 3-6. Waveform*** of the BOR and ***Figure 3-10. waveform of VUVD*** in the PMU section. 33. Modify ***Figure 23-31. BREAK0 function logic diagram***, ***Figure 23-115. BREAK0 function logic diagram*** and ***Figure 23-137. BREAK0 function logic diagram*** in the TIMER section. 34. Modify the ADDMSK2[2:0] bit field description in the ***Slave address register 1 (I2C_SADDR1)*** and the reset value of the ***Control register 2 (I2C_CTL2)*** in the I2C section. 35. Add a description in the "Note:" for ***Single bank selection(DBS = 0)*** in the FMC section.	

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.

© 2025 GigaDevice Semiconductor Inc. – All rights reserved