GigaDevice Semiconductor Inc.

GD32H77x_78x
Arm® Cortex®-M7 32-bit MCU

For GD32H779xx, GD32H77Exx, GD32H77Dxx, GD32H77Rxx,

GD32H789xx, GD32H78Exx, GD32H78Dxx

User Manual

Revision 1.0

(Nov. 2026)

Table of Contents

Table of Contents **2**
List of Figures **48**
List of Tables **65**
1. System and memory architecture **74**
1.1. Arm® Cortex®-M7 processor **74**
1.2. System architecture **75**
1.2.1. Bus matrix Region 0 80
1.2.2. Bus matrix Region 1 80
1.2.3. Bus matrix Region 2 81
1.3. Memory map **82**
1.3.1. On-chip SRAM memory 90
1.3.2. Non-volatile memory overview 93
1.4. Boot configuration **93**
1.5. System configuration controller (SYSCFG) **95**
1.6. Timer break input lock **95**
1.7. AXI interconnect matrix (AXIIM) **95**
1.7.1. Characteristics 95
1.7.2. Function overview 95
1.8. System configuration registers **98**
1.8.1. Peripheral mode configuration register 0 (SYSCFG_PMCFG0) 98
1.8.2. EXTI sources selection register 0 (SYSCFG_EXTISS0) 100
1.8.3. EXTI sources selection register 1 (SYSCFG_EXTISS1) 102
1.8.4. EXTI sources selection register 2 (SYSCFG_EXTISS2) 103
1.8.5. EXTI sources selection register 3 (SYSCFG_EXTISS3) 104
1.8.6. Lockup control register (SYSCFG_LKCTL) 105
1.8.7. Bus timeout enable register (SYSCFG_BUSTO) 107
1.8.8. I/O compensation control register (SYSCFG_CPSCTL) 109
1.8.9. I/O compensation cell code configuration register (SYSCFG_CPSCCCFG) 110
1.8.10. Timer input selection register 0 (SYSCFG_TIMERCISEL0) 110
1.8.11. Timer input selection register 1 (SYSCFG_TIMERCISEL1) 111
1.8.12. Timer input selection register 3 (SYSCFG_TIMERCISEL3) 112
1.8.13. Timer input selection register 4 (SYSCFG_TIMERCISEL4) 112
1.8.14. Timer input selection register 5 (SYSCFG_TIMERCISEL5) 113
1.8.15. Timer input selection register 6 (SYSCFG_TIMERCISEL6) 115
1.8.16. CPU ICACHE error status register(SYSCFG_CPUICAC) 117
1.8.17. CPU DCACHE error status register (SYSCFG_CPUDCAC) 117

1.8.18. CACHE and FPU interrupt enable register (SYSCFG_CFPUINTEN) 118
1.8.19. SRAM configuration register 0 (SYSCFG_SRAMCFG0) 119
1.8.20. Bus timeout states register (SYSCFG_BUSTOSTAT) 119
1.8.21. ESC configuration register 0 (SYSCFG_ESCCFG0) 121
1.8.22. ESC configuration register 1 (SYSCFG_ESCCFG1) 122
1.8.23. ESC configuration register 2 (SYSCFG_ESCCFG2) 123
1.8.24. TIMERx configuration register 0 (SYSCFG_TIMERxCFG0, x=0, 7) 123
1.8.25. TIMERx configuration register 1 (SYSCFG_TIMERxCFG1, x=0, 7) 126
1.8.26. TIMERx configuration register 2 (SYSCFG_TIMERxCFG2, x=0, 7) 128
1.8.27. TIMERx configuration register 0 (SYSCFG_TIMERxCFG0, x=1, 2, 3, 4, 22, 23, 30, 31) .. 129
1.8.28. TIMERx configuration register 1 (SYSCFG_TIMERxCFG1, x=1, 2, 3, 4, 22, 23, 30, 31) .. 132
1.8.29. TIMERx configuration register 2 (SYSCFG_TIMERxCFG2, x=1, 2, 3, 4, 22, 23, 30, 31) .. 134
1.8.30. TIMERx configuration register 0 (SYSCFG_TIMERxCFG0, x=14, 40, 41, 42, 43, 44) 136
1.8.31. TIMERx configuration register 1 (SYSCFG_TIMERxCFG1, x=14, 40, 41, 42, 43, 44) 137
1.8.32. TIMERx configuration register 2 (SYSCFG_TIMERxCFG2, x=14, 40, 41, 42, 43, 44) 139
1.8.33. Peripheral mode configuration register 1 (SYSCFG_PMCFG1) 140
1.8.34. User configuration register (SYSCFG_USERCFG) 141

1.9. AXI interconnect registers 141

1.9.1. AXI peripheral ID4 register (AXI_PERIPH_ID4) 142
1.9.2. AXI peripheral ID0 register (AXI_PERIPH_ID0) 142
1.9.3. AXI peripheral ID1 register (AXI_PERIPH_ID1) 142
1.9.4. AXI peripheral ID2 register (AXI_PERIPH_ID2) 143
1.9.5. AXI peripheral ID3 register (AXI_PERIPH_ID3) 143
1.9.6. AXI component ID0 register (AXI_COMP_ID0) 144
1.9.7. AXI component ID1 register (AXI_COMP_ID1) 144
1.9.8. AXI component ID2 register (AXI_COMP_ID2) 145
1.9.9. AXI component ID3 register (AXI_COMP_ID3) 145
1.9.10. AXI Master Port x bus matrix issuing functionality control register (AXI_MPxBM_ISS_CTL) 145
1.9.11. AXI Master Port x bus matrix functionality control register (AXI_MPxBM_CTL) 146
1.9.12. AXI Master Port x long burst functionality control register (AXI_MPx_LB_CTL) 146
1.9.13. AXI Master Port x issuing functionality control register (AXI_MPx_ISS_CTL) 147
1.9.14. AXI Slave Port x functionality control register (AXI_SPx_CTL) 147
1.9.15. AXI Slave Port x AHB issuing functionality control register (AXI_SPx_AHBISS_CTL) 148
1.9.16. AXI Slave Port x read QOS control register (AXI_SPx_RDQOS_CTL) 149
1.9.17. AXI Slave Port x write QOS control register (AXI_SPx_WRQOS_CTL) 149
1.9.18. AXI Slave Port x issuing functionality control register (AXI_SPx_ISS_CTL) 149

1.10. Device electronic signature 151

1.10.1. CNVM size information 151
1.10.2. Memory density information 151
1.10.3. Unique device ID (96 bits) 152

2. Hardware semaphore (HWSEM) 153

2.1. Overview 153
2.2. Characteristics 153
2.3. Function overview 153
2.3.1. Block diagram 153
2.3.2. Semaphore x 154
2.3.3. Lock a semaphore 154
2.3.4. Unlock a semaphore 155
2.3.5. Unlock all semaphores 156
2.3.6. Interrupts 156
2.4. Register definition 158
2.4.1. Control register (HWSEM_CTLx)(x = 0…31) 158
2.4.2. Read lock register (HWSEM_RLKx)(x = 0...31) 158
2.4.3. Interrupt enable register (HWSEM_INTEN) 159
2.4.4. Interrupt flag clear register (HWSEM_INTC) 160
2.4.5. Status register (HWSEM_STAT) 160
2.4.6. Interrupt flag register (HWSEM_INTF) 160
2.4.7. Unlock register (HWSEM_UNLK) 161
2.4.8. Key register (HWSEM_KEY) 162
3. Interrupt / event controller (EXTI) 163
3.1. Overview 163
3.2. Characteristics 163
3.3. Interrupt function overview 163
3.4. External interrupt and event (EXTI) block diagram 170
3.5. External interrupt and event function overview 170
3.6. Register definition 173
3.6.1. Interrupt enable register 0 (EXTI_INTEN0) 173
3.6.2. Event enable register 0 (EXTI_EVEN0) 173
3.6.3. Rising edge trigger enable register 0 (EXTI_RTEN0) 173
3.6.4. Falling edge trigger enable register 0 (EXTI_FTEN0) 174
3.6.5. Software interrupt event register 0 (EXTI_SWIEV0) 174
3.6.6. Pending register 0 (EXTI_PD0) 175
3.6.7. Interrupt enable register 1 (EXTI_INTEN1) 175
3.6.8. Event enable register 1 (EXTI_EVEN1) 176
3.6.9. Rising edge trigger enable register 1 (EXTI_RTEN1) 176
3.6.10. Falling edge trigger enable register 1 (EXTI_FTEN1) 176
3.6.11. Software interrupt event register 1 (EXTI_SWIEV1) 177
3.6.12. Pending register 1 (EXTI_PD1) 177
4. Direct memory access controller (DMA) 179
4.1. Overview 179

4.2. Characteristics 179
4.3. Function overview 180
4.3.1. Block diagram 180
4.3.2. Peripheral handshake 182
4.3.3. Data process 182
4.3.4. Address generation 188
4.3.5. Circular mode 188
4.3.6. Switch-buffer mode 189
4.3.7. Transfer operation 189
4.3.8. Transfer finish 190
4.3.9. Channel configuration 192
4.4. Interrupts 193
4.4.1. Flag 193
4.4.2. Exception 194
4.4.3. Error 195
4.4.4. DMA request mapping 196
4.5. Register definition 197
4.5.1. Interrupt flag register 0 (DMA_INTF0) 197
4.5.2. Interrupt flag register 1 (DMA_INTF1) 198
4.5.3. Interrupt flag clear register 0 (DMA_INTC0) 199
4.5.4. Interrupt flag clear register 1 (DMA_INTC1) 200
4.5.5. Channel x control register (DMA_CHxCTL) 200
4.5.6. Channel x counter register (DMA_CHxCNT) 204
4.5.7. Channel x peripheral base address register (DMA_CHxPADDR) 204
4.5.8. Channel x memory 0 base address register (DMA_CHxM0ADDR) 205
4.5.9. Channel x memory 1 base address register (DMA_CHxM1ADDR) 206
4.5.10. Channel x FIFO control register (DMA_CHxFCTL) 206
5. Master direct memory access controller (MDMA) 208
5.1. Overview 208
5.2. Characteristics 208
5.3. Function overview 209
5.3.1. Data process 211
5.3.2. Address generation 214
5.3.3. Transfer modes 215
5.3.4. Transfer status 218
5.3.5. MDMA interrupts and errors 218
5.4. Register definition 221
5.4.1. Global interrupt flag register (MDMA_GINTF) 221
5.4.2. Channel x status register 0 (MDMA_CHxSTAT0) 221
5.4.3. Channel x status clear register (MDMA_CHxSTATC) 222
5.4.4. Channel x status register 1 (MDMA_CHxSTAT1) 223

5.4.5. Channel x control register 0 (MDMA_CHxCTL0) 224
5.4.6. Channel x configure register (MDMA_CHxCFG) 226
5.4.7. Channel x block transfer configure register (MDMA_CHxBTCFG) 230
5.4.8. Channel x source address register (MDMA_CHxSADDR) 230
5.4.9. Channel x destination address register (MDMA_CHxDADDR) 231
5.4.10. Channel x multi-block address update register (MDMA_CHxMBADDRU) 231
5.4.11. Channel x link address register (MDMA_CHxLADDR) 232
5.4.12. Channel x control register 1 (MDMA_CHxCTL1) 232
5.4.13. Channel x mask address register (MDMA_CHxMADDR) 233
5.4.14. Channel x mask data register (MDMA_CHxMDATA) 234

6. DMA request multiplexer (DMAMUX) 235

6.1. Overview 235

6.2. Characteristics 235

6.3. Block diagram 236

6.4. Signal description 236

6.5. Function overview 237

6.5.1. DMAMUX request multiplexer 237
6.5.2. DMAMUX request generator 240
6.5.3. Channel configurations 240
6.5.4. Interrupt 241
6.5.5. DMAMUX mapping 241

6.6. Register definition 251

6.6.1. Request multiplexer channel x configuration register (DMAMUX_RM_CHxCFG) 251
6.6.2. Request multiplexer channel interrupt flag register (DMAMUX_RM_INTF) 252
6.6.3. Request multiplexer channel interrupt flag clear register (DMAMUX_RM_INTC) 252
6.6.4. Request generator channel x configuration register (DMAMUX_RG_CHxCFG) 253
6.6.5. Request generator channel interrupt flag register (DMAMUX_RG_INTF) 254
6.6.6. Request generator channel interrupt flag clear register (DMAMUX_RG_INTC) 254

7. Non-volatile memory controller (NVMC) 256

7.1. Overview 256

7.2. Characteristics 256

7.3. Function overview 257

7.3.1. Non-volatile memory architecture 257
7.3.2. Error Checking and Correcting (ECC) 262
7.3.3. CNVM controller (CNVMC) 263
7.3.4. ECNVM controller (ECNVMC) 277
7.3.5. Data integrity enhancement features 287
7.3.6. Data integrity enhancement mode 288
7.3.7. CNVM bad block feature 289
7.3.8. OTA 289

7.3.9. Data Integrity Enhanced Area function limitations 292
7.3.10. Option bytes 293
7.3.11. Sector erase / program protection 299
7.3.12. Security protection 300
7.3.13. Dedicated code read protection area 302
7.3.14. Secure user area 304
7.3.15. Secure mode 306
7.3.16. OTP 307
7.4. Register definition 316
7.4.1. NVMC registers 316
7.4.2. CNVM register 351
7.4.3. ECNVM Register 364
7.4.4. OTP register 369
8. RAM ECC monitor unit (RAMECCMU) 385
8.1. Characteristics 385
8.2. Function overview 385
8.3. Register definition 387
8.3.1. RAMECCMU global interrupt register (RAMECCMU_INT) 387
8.3.2. RAMECCMU monitor x control register (RAMECCMU_MxCTL) 387
8.3.3. RAMECCMU monitor x status register (RAMECCMU_MxSTAT) 388
8.3.4. RAMECCMU monitor x failing address register (RAMECCMU_MxFADDR) 389
8.3.5. RAMECCMU monitor x failing data low register (RAMECCMU_MxFDL) 390
8.3.6. RAMECCMU monitor x failing data high register (RAMECCMU_MxFDH) 390
8.3.7. RAMECCMU monitor x failing ECC error code register (RAMECCMU_MxFECODE) 390
9. Power management unit (PMU) 392
9.1. Overview 392
9.2. Characteristics 392
9.3. Function overview 393
9.3.1. Backup domain 394
9.3.2. V_{DD} / V_{DDA} power domain 395
9.3.3. V_{CORE} power domain 399
9.3.4. USB Power Management 403
9.3.5. DSI Power domain 404
9.3.6. Power saving modes 405
9.4. Register definition 408
9.4.1. Control register 0 (PMU_CTL0) 408
9.4.2. Control and status register (PMU_CS) 410
9.4.3. Control register 1 (PMU_CTL1) 413
9.4.4. Control register 2 (PMU_CTL2) 414
9.4.5. Control register 3 (PMU_CTL3) 415

9.4.6. Control register 4 (PMU_CTL4) 417
9.4.7. Parameter register (PMU_PAR) 418
10. Reset and clock unit (RCU) 420
10.1. Reset control unit (RCTL) 420
10.1.1. Overview 420
10.1.2. Function overview 420
10.2. Clock control unit (CCTL) 421
10.2.1. Overview 421
10.2.2. Characteristics 426
10.2.3. Function overview 427
10.3. Register definition 433
10.3.1. Control register (RCU_CTL) 433
10.3.2. PLL0 register (RCU_PLL0) 435
10.3.3. Clock configuration register 0 (RCU_CFG0) 437
10.3.4. Clock interrupt register (RCU_INT) 439
10.3.5. AHB1 reset register (RCU_AHB1RST) 442
10.3.6. AHB2 reset register (RCU_AHB2RST) 444
10.3.7. AHB3 reset register (RCU_AHB3RST) 445
10.3.8. AHB4 reset register (RCU_AHB4RST) 447
10.3.9. APB1 reset register (RCU_APB1RST) 448
10.3.10. APB2 reset register (RCU_APB2RST) 452
10.3.11. APB3 reset register (RCU_APB3RST) 455
10.3.12. APB4 reset register (RCU_APB4RST) 455
10.3.13. AHB1 enable register (RCU_AHB1EN) 456
10.3.14. AHB2 enable register (RCU_AHB2EN) 458
10.3.15. AHB3 enable register (RCU_AHB3EN) 460
10.3.16. AHB4 enable register (RCU_AHB4EN) 462
10.3.17. APB1 enable register (RCU_APB1EN) 463
10.3.18. APB2 enable register (RCU_APB2EN) 467
10.3.19. APB3 enable register (RCU_APB3EN) 470
10.3.20. APB4 enable register (RCU_APB4EN) 471
10.3.21. AHB1 sleep mode enable register (RCU_AHB1SPEN) 472
10.3.22. AHB2 sleep mode enable register (RCU_AHB2SPEN) 474
10.3.23. AHB3 sleep mode enable register (RCU_AHB3SPEN) 476
10.3.24. AHB4 sleep mode enable register (RCU_AHB4SPEN) 477
10.3.25. APB1 sleep mode enable register (RCU_APB1SPEN) 479
10.3.26. APB2 sleep mode enable register (RCU_APB2SPEN) 482
10.3.27. APB3 sleep mode enable register (RCU_APB3SPEN) 486
10.3.28. APB4 sleep mode enable register (RCU_APB4SPEN) 486
10.3.29. Backup domain control register (RCU_BDCTL) 487
10.3.30. Reset source/clock register (RCU_RSTSCK) 489
10.3.31. PLL clock additional control register (RCU_PLLADDCTL) 491

10.3.32. PLL1 register (RCU_PLL1) 493
10.3.33. PLL2 register (RCU_PLL2) 495
10.3.34. Clock configuration register 1 (RCU_CFG1) 497
10.3.35. Clock configuration register 2 (RCU_CFG2) 500
10.3.36. Clock configuration register 3 (RCU_CFG3) 502
10.3.37. PLL all configuration register (RCU_PLLALL) 504
10.3.38. PLL0 fraction configuration register (RCU_PLL0FRA) 506
10.3.39. PLL1 fraction configuration register (RCU_PLL1FRA) 506
10.3.40. PLL2 fraction configuration register (RCU_PLL2FRA) 507
10.3.41. Additional clock control register 0 (RCU_ADDCTL0) 508
10.3.42. Additional clock control register 1(RCU_ADDCTL1) 509
10.3.43. Additional clock interrupt register (RCU_ADDINT) 510
10.3.44. Clock configuration register 4 (RCU_CFG4) 511
10.3.45. USB clock control register (RCU_USBCLKCTL) 512
10.3.46. PLLUSB configuration register (RCU_PLLUSBCFG) 514
10.3.47. APB2 additional reset register (RCU_ADDAPB2RST) 515
10.3.48. APB2 additional enable register (RCU_ADDAPB2EN) 516
10.3.49. APB2 additional sleep enable register (RCU_ADDAPB2SPEN) 518
10.3.50. Clock configuration register 5 (RCU_CFG5) 519
10.3.51. Clock configuration register 6 (RCU_CFG6) 521
10.3.52. Clock frequency monitor configuration register 0 (RCU_CKFMCFG0) 522
10.3.53. Clock frequency monitor configuration register 1 (RCU_CKFMCFG1) 523
10.3.54. Clock frequency monitor configuration register 2 (RCU_CKFMCFG2) 524

11. Clock trim controller (CTC) 526

11.1. Overview 526

11.2. Characteristics 526

11.3. Function overview 526

11.3.1. REF sync pulse generator 527
11.3.2. CTC trim counter 527
11.3.3. Frequency evaluation and automatically trim process 528
11.3.4. Software program guide 529

11.4. Register definition 531

11.4.1. Control register 0 (CTC_CTL0) 531
11.4.2. Control register 1 (CTC_CTL1) 532
11.4.3. Status register (CTC_STAT) 533
11.4.4. Interrupt clear register (CTC_INTC) 535

12. Clock phase delay module (CPDM) 537

12.1. Overview 537

12.2. Characteristics 537

12.3. Function overview 537

12.3.1. Overview 538
12.3.2. Operation process 539
12.4. Register definition 541
12.4.1. Control register (CPDM_CTL) 541
12.4.2. Configuration register (CPDM_CFG) 541
13. General-purpose and alternate-function I/Os (GPIO and AFIO) 543
13.1. Overview 543
13.2. Characteristics 543
13.3. Function overview 543
13.3.1. GPIO pin configuration 545
13.3.2. External interrupt/event lines 546
13.3.3. Alternate functions (AF) 546
13.3.4. Additional functions 546
13.3.5. Input configuration 546
13.3.6. Output configuration 547
13.3.7. Analog configuration 547
13.3.8. Alternate function (AF) configuration 548
13.3.9. GPIO locking function 548
13.3.10. GPIO single cycle toggle function 549
13.3.11. I/O compensation unit 549
13.3.12. Analog configuration for ADC 549
13.3.13. Input filtering 550
13.4. Register definition 554
13.4.1. Port control register (GPIOx_CTL, x=A...H, J, K) 554
13.4.2. Port output mode register (GPIOx_OMODE, x=A…H, J, K) 556
13.4.3. Port output speed register (GPIOx_OSPD, x=A...H, J, K) 557
13.4.4. Port pull-up/down register (GPIOx_PUD, x=A…H, J, K) 559
13.4.5. Port input status register (GPIOx_ISTAT, x=A…H, J, K) 561
13.4.6. Port output control register (GPIOx_OCTL, x=A…H, J, K) 561
13.4.7. Port bit operate register (GPIOx_BOP, x=A...H, J, K) 562
13.4.8. Port configuration lock register (GPIOx_LOCK, x=A...H, J, K) 562
13.4.9. Alternate function selected register 0 (GPIOx_AFSEL0, x=A...H, J, K) 563
13.4.10. Alternate function selected register 1 (GPIOx_AFSEL1, x=A…H, J, K) 564
13.4.11. Bit clear register (GPIOx_BC, x=A…H, J, K) 566
13.4.12. Port bit toggle register (GPIOx_TG, x=A…H, J, K) 566
13.4.13. Input filtering register (GPIOx_IFL, x=A...H, J, K) 567
13.4.14. Input filtering type register (GPIOx_IFTP, x=A...H, J, K) 567
13.4.15. Alternate function selected register 2 (GPIOx_AFSEL2, x=B,C,D,E) 569
14. Trigger selection controller (TRIGSEL) 571
14.1. Overview 571

14.2. Characteristics 571
14.3. Function overview 571
14.4. Internal connect 572
14.5. Register definition 584
14.5.1. Trigger selection for EXTOUT register 0 (TRIGSEL_EXTOUT_0) 584
14.5.2. Trigger selection for EXTOUT register 1 (TRIGSEL_EXTOUT_1) 584
14.5.3. Trigger selection for EXTOUT register 2 (TRIGSEL_EXTOUT_2) 585
14.5.4. Trigger selection for EXTOUT register 3 (TRIGSEL_EXTOUT_3) 586
14.5.5. Trigger selection for ADC0 register (TRIGSEL_ADC0) 586
14.5.6. Trigger selection for ADC1 register (TRIGSEL_ADC1) 587
14.5.7. Trigger selection for ADC2 register (TRIGSEL_ADC2) 588
14.5.8. Trigger selection for DAC0_OUT0 register (TRIGSEL_DAC0OUT0) 589
14.5.9. Trigger selection for DAC0_OUT1 register (TRIGSEL_DAC0OUT1) 589
14.5.10. Trigger selection for TIMER0_BRKIN register (TRIGSEL_TIMER0BRKIN) 590
14.5.11. Trigger selection for TIMER7_BRKIN register (TRIGSEL_TIMER7BRKIN) 591
14.5.12. Trigger selection for TIMER14_BRKIN register (TRIGSEL_TIMER14BRKIN) 592
14.5.13. Trigger selection for TIMER15_BRKIN register (TRIGSEL_TIMER15BRKIN) 592
14.5.14. Trigger selection for TIMER16_BRKIN register (TRIGSEL_TIMER16BRKIN) 593
14.5.15. Trigger selection for TIMER40_BRKIN register (TRIGSEL_TIMER40BRKIN) 594
14.5.16. Trigger selection for TIMER41_BRKIN register (TRIGSEL_TIMER41BRKIN) 594
14.5.17. Trigger selection for TIMER42_BRKIN register (TRIGSEL_TIMER42BRKIN) 595
14.5.18. Trigger selection for TIMER43_BRKIN register (TRIGSEL_TIMER43BRKIN) 596
14.5.19. Trigger selection for TIMER44_BRKIN register (TRIGSEL_TIMER44BRKIN) 596
14.5.20. Trigger selection for CAN0 register (TRIGSEL_CAN0) 597
14.5.21. Trigger selection for CAN1 register (TRIGSEL_CAN1) 597
14.5.22. Trigger selection for CAN2 register (TRIGSEL_CAN2) 598
14.5.23. Trigger selection for LPDTS register (TRIGSEL_LPDTS) 599
14.5.24. Trigger selection for TIMER0_ETI register (TRIGSEL_TIMER0ETI) 599
14.5.25. Trigger selection for TIMER1_ETI register (TRIGSEL_TIMER1ETI) 600
14.5.26. Trigger selection for TIMER2_ETI register (TRIGSEL_TIMER2ETI) 600
14.5.27. Trigger selection for TIMER3_ETI register (TRIGSEL_TIMER3ETI) 601
14.5.28. Trigger selection for TIMER4_ETI register (TRIGSEL_TIMER4ETI) 602
14.5.29. Trigger selection for TIMER7_ETI register (TRIGSEL_TIMER7ETI) 602
14.5.30. Trigger selection for TIMER22_ETI register (TRIGSEL_TIMER22ETI) 603
14.5.31. Trigger selection for TIMER23_ETI register (TRIGSEL_TIMER23ETI) 603
14.5.32. Trigger selection for TIMER30_ETI register (TRIGSEL_TIMER30ETI) 604
14.5.33. Trigger selection for TIMER31_ETI register (TRIGSEL_TIMER31ETI) 605
14.5.34. Trigger selection for EDOUT register (TRIGSEL_EDOUT) 605
14.5.35. Trigger selection for HPDF register (TRIGSEL_HPDF) 606
14.5.36. Trigger selection for TIMER0_ITI14 register (TRIGSEL_TIMER0ITI14) 606
14.5.37. Trigger selection for TIMER1_ITI14 register (TRIGSEL_TIMER1ITI14) 607
14.5.38. Trigger selection for TIMER2_ITI14 register (TRIGSEL_TIMER2ITI14) 608
14.5.39. Trigger selection for TIMER3_ITI14 register (TRIGSEL_TIMER3ITI14) 608

14.5.40. Trigger selection for TIMER4_ITI14 register (TRIGSEL_TIMER4ITI14) 609
14.5.41. Trigger selection for TIMER7_ITI14 register (TRIGSEL_TIMER7ITI14) 609
14.5.42. Trigger selection for TIMER14_ITI14 register (TRIGSEL_TIMER14ITI14) 610
14.5.43. Trigger selection for TIMER22_ITI14 register (TRIGSEL_TIMER22ITI14) 611
14.5.44. Trigger selection for TIMER23_ITI14 register (TRIGSEL_TIMER23ITI14) 611
14.5.45. Trigger selection for TIMER30_ITI14 register (TRIGSEL_TIMER30ITI14) 612
14.5.46. Trigger selection for TIMER31_ITI14 register (TRIGSEL_TIMER31ITI14) 612
14.5.47. Trigger selection for TIMER40_ITI14 register (TRIGSEL_TIMER40ITI14) 613
14.5.48. Trigger selection for TIMER41_ITI14 register (TRIGSEL_TIMER41ITI14) 614
14.5.49. Trigger selection for TIMER42_ITI14 register (TRIGSEL_TIMER42ITI14) 614
14.5.50. Trigger selection for TIMER43_ITI14 register (TRIGSEL_TIMER43ITI14) 615
14.5.51. Trigger selection for TIMER44_ITI14 register (TRIGSEL_TIMER44ITI14) 615
14.5.52. Trigger selection for RDCM register (TRIGSEL_RDCM) 616
14.5.53. Trigger selection for TIMER0_CHBRKIN register 0 (TRIGSEL_TIMER0CHBRKIN_0) 617
14.5.54. Trigger selection for TIMER0_CHBRKIN register 1 (TRIGSEL_TIMER0CHBRKIN_1) 618
14.5.55. Trigger selection for TIMER7_CHBRKIN register 0 (TRIGSEL_TIMER7CHBRKIN_0) 618
14.5.56. Trigger selection for TIMER7_CHBRKIN register 1 (TRIGSEL_TIMER7CHBRKIN_1) 619
14.5.57. Trigger selection for ENDAT register (TRIGSEL_ENDAT) 620
14.5.58. Trigger selection for AFMT register (TRIGSEL_AFMT) 621
14.5.59. Trigger selection for TFMT register (TRIGSEL_TFMT) 621
14.5.60. Trigger selection for BISS register (TRIGSEL_BISS) 622
14.5.61. Trigger selection for HDSL register (TRIGSEL_HDSL) 622

15. TIMER (TIMERx) 624

15.1. Advanced timer (TIMERx, x=0, 7) 626
15.1.1. Overview 626
15.1.2. Characteristics 626
15.1.3. Block diagram 627
15.1.4. Function overview 627
15.1.5. Registers definition (TIMERx, x=0, 7) 680

15.2. General level0 timer (TIMERx, x=1,2,3,4,22,23,30,31) 756
15.2.1. Overview 756
15.2.2. Characteristics 756
15.2.3. Block diagram 756
15.2.4. Function overview 757
15.2.5. Registers definition (TIMERx, x=1,2,3,4,22,23,30,31) 780

15.3. General level3 timer (TIMERx, x=14,40,41,42,43,44) 813
15.3.1. Overview 813
15.3.2. Characteristics 813
15.3.3. Block diagram 813
15.3.4. Function overview 814
15.3.5. Register definition (TIMERx, x=14,40,41,42,43,44) 846

15.4. General level4 timer (TIMERx, x=15,16) 881

15.4.1. Overview 881
15.4.2. Characteristics 881
15.4.3. Block diagram 881
15.4.4. Function overview 882
15.4.5. Register definition (TIMERx, x=15,16) 900
15.5. Basic timer (TIMERx, x=5,6,50,51) 926
15.5.1. Overview 926
15.5.2. Characteristics 926
15.5.3. Block diagram 926
15.5.4. Function overview 926
15.5.5. Registers definition (TIMERx, x=5,6,50,51) 930
16. Real time clock (RTC) 938
16.1. Overview 938
16.2. Characteristics 938
16.3. Function overview 939
16.3.1. Block diagram 939
16.3.2. Clock source and prescalers 940
16.3.3. Shadow registers introduction 940
16.3.4. Configurable and field maskable alarm 940
16.3.5. Configurable periodic auto-wakeup counter 941
16.3.6. RTC initialization and configuration 941
16.3.7. Calendar reading 942
16.3.8. Resetting the RTC 944
16.3.9. RTC shift function 944
16.3.10. RTC reference clock detection 945
16.3.11. RTC smooth digital calibration 945
16.3.12. Time-stamp function 947
16.3.13. Tamper detection 948
16.3.14. Calibration clock output 949
16.3.15. Alarm output 950
16.3.16. RTC pin configuration 950
16.3.17. RTC power saving mode management 951
16.3.18. RTC interrupts 951
16.4. Register definition 953
16.4.1. Time register (RTC_TIME) 953
16.4.2. Date register (RTC_DATE) 953
16.4.3. Control register (RTC_CTL) 954
16.4.4. Status register (RTC_STAT) 957
16.4.5. Prescaler register (RTC_PSC) 959
16.4.6. Wakeup timer register (RTC_WUT) 959
16.4.7. Alarm 0 time and date register (RTC_ALRM0TD) 960

16.4.8. Alarm 1 time and date register (RTC_ALRM1TD) 961
16.4.9. Write protection key register (RTC_WPK) 962
16.4.10. Sub second register (RTC_SS) 962
16.4.11. Shift function control register (RTC_SHIFTCTL) 963
16.4.12. Time of time stamp register (RTC_TTS) 964
16.4.13. Date of time stamp register (RTC_DTS) 964
16.4.14. Sub second of time stamp register (RTC_SSTS) 965
16.4.15. High resolution frequency compensation register (RTC_HRFC) 965
16.4.16. Tamper register (RTC_TAMP) 966
16.4.17. Alarm 0 sub second register (RTC_ALRM0SS) 968
16.4.18. Alarm 1 sub second register (RTC_ALRM1SS) 969
16.4.19. Configuration register (RTC_CFG) 970
16.4.20. Backup registers (RTC_BKPx) (x=0..31) 971

17. Watchdog timer (WDGT) 972

17.1. Free watchdog timer (FWDGT) 972
17.1.1. Overview 972
17.1.2. Characteristics 972
17.1.3. Function overview 972
17.1.4. Register definition 975

17.2. Window watchdog timer (WWDGT) 979
17.2.1. Overview 979
17.2.2. Characteristics 979
17.2.3. Function overview 979
17.2.4. Register definition 982

18. Debug (DBG) 984

18.1. Overview 984

18.2. JTAG / SW function overview 984
18.2.1. Switch JTAG or SW interface 984
18.2.2. Pin assignment 984
18.2.3. JTAG 985
18.2.4. Debug reset 987
18.2.5. JEDEC-106 ID code 987

18.3. Debug hold function overview 987
18.3.1. Debug support for power saving mode 987
18.3.2. Debug support for TIMER, I2C, WWDGT, FWDGT, RTC and CAN 987

18.4. Register definition 988
18.4.1. ID code register (DBG_ID) 988
18.4.2. Control register0 (DBG_CTL0) 988
18.4.3. Control register1 (DBG_CTL1) 989
18.4.4. Control register2 (DBG_CTL2) 990
18.4.5. Control register3 (DBG_CTL3) 992

18.4.6. Control register4 (DBG_CTL4) 993
19. Cyclic redundancy checks management unit (CRC) 995
19.1. Overview 995
19.2. Characteristics 995
19.3. Function overview 996
19.4. Register definition 998
19.4.1. Data register (CRC_DATA) 998
19.4.2. Free data register (CRC_FDATA) 998
19.4.3. Control register (CRC_CTL) 999
19.4.4. Initialization data register (CRC_IDATA) 999
19.4.5. Polynomial register (CRC_POLY) 1000
20. True random number generator (TRNG) 1001
20.1. Overview 1001
20.2. Characteristics 1001
20.3. Function overview 1001
20.3.1. LFSR 1002
20.3.2. Post processing 1002
20.3.3. Conditioning 1002
20.3.4. Output FIFO 1002
20.3.5. Health tests 1003
20.3.6. NIST mode state 1004
20.3.7. Operation flow 1005
20.3.8. Error flags 1005
20.3.9. Low power usage 1005
20.4. Register definition 1007
20.4.1. Control register (TRNG_CTL) 1007
20.4.2. Status register (TRNG_STAT) 1009
20.4.3. Data register (TRNG_DATA) 1010
20.4.4. Health tests configure register (TRNG_HTCFG) 1010
21. Cryptographic Acceleration Unit (CAU) 1012
21.1. Overview 1012
21.2. Characteristics 1012
21.3. CAU data type and initialization vectors 1012
21.3.1. Data type 1012
21.3.2. Initialization vectors 1014
21.4. Cryptographic acceleration processor 1014
21.4.1. AES cryptographic acceleration processor 1015
21.5. Operating modes 1024

21.6. CAU DMA interface 1026
21.7. CAU interrupts 1026
21.8. CAU suspended mode 1026
21.9. Register definition 1028
21.9.1. Control register (CAU_CTL) 1028
21.9.2. Status register 0 (CAU_STAT0) 1030
21.9.3. Data input register (CAU_DI) 1030
21.9.4. Data output register (CAU_DO) 1031
21.9.5. DMA enable register (CAU_DMAEN) 1031
21.9.6. Interrupt enable register (CAU_INTEN) 1032
21.9.7. Status register 1 (CAU_STAT1) 1033
21.9.8. Interrupt flag register (CAU_INTF) 1033
21.9.9. Key registers (CAU_KEY0..3(H / L)) 1034
21.9.10. Initial vector registers (CAU_IV0..1(H / L)) 1036
21.9.11. GCM or CCM mode context switch register x (CAU_GCMCCMCTXSx) (x=0..7) 1038
21.9.12. GCM mode context switch register x (CAU_GCMCTXSx) (x=0..7) 1038
21.9.13. CMAC mode context switch register x (CAU_CMACCTXSx) (x=0..3) 1040
22. Hash Acceleration Unit (HAU) 1041
22.1. Overview 1041
22.2. Characteristics 1041
22.3. HAU data type 1041
22.4. HAU core 1043
22.4.1. Automatic data padding 1043
22.4.2. Digest computing 1044
22.4.3. Hash mode 1045
22.4.4. HMAC mode 1045
22.5. HAU suspended mode 1046
22.5.1. Transfer data by CPU 1046
22.5.2. Transfer data by DMA 1046
22.6. HAU interrupt 1047
22.6.1. Input FIFO interrupt 1047
22.6.2. Calculation completion interrupt 1047
22.7. Register definition 1048
22.7.1. Control register (HAU_CTL) 1048
22.7.2. Data input register (HAU_DI) 1049
22.7.3. Configuration register (HAU_CFG) 1050
22.7.4. Data output register (HAU_DO0..7) 1051
22.7.5. Interrupt enable register (HAU_INTEN) 1053
22.7.6. Status and flag register (HAU_STAT) 1054
22.7.7. Context switch register x (HAU_CTXSx) (x = 0...53) 1054

23. Public Key Cryptographic Acceleration Unit (PKCAU) 1056
23.1. Overview 1056
23.2. Characteristics 1056
23.3. Function overview 1056
23.3.1. Operands 1057
23.3.2. RSA algorithm 1057
23.3.3. ECC algorithm 1059
23.3.4. Integer arithmetic operations 1060
23.3.5. Elliptic curve operations in Fp domain 1070
23.3.6. PKCAU operation process 1076
23.3.7. Processing times 1077
23.3.8. Status, errors and interrupts 1078
23.4. Register definition 1080
23.4.1. Control register (PKCAU_CTL) 1080
23.4.2. Status register (PKCAU_STAT) 1081
23.4.3. Status clear register (PKCAU_STATC) 1082
24. Real-time decryption (RTDEC) 1084
24.1. Overview 1084
24.2. Characteristics 1084
24.3. Function overview 1084
24.3.1. RTDEC block diagram 1085
24.3.2. RTDEC real-time decryption introduction 1085
24.3.3. Decryption with AES-128 in counter mode 1086
24.3.4. RTDEC configuration 1087
24.3.5. RTDEC error management 1089
24.4. RTDEC interrupts 1090
24.5. Registers definition 1091
24.5.1. Area x configuration register (RTDEC_AREx_CFG) 1091
24.5.2. Area x start address register (RTDEC_AREx_SADDR) 1092
24.5.3. Area x end address register (RTDEC_AREx_EADDR) 1093
24.5.4. Area x nonce register 0 (RTDEC_AREx_NONCE0) 1093
24.5.5. Area x nonce register 1 (RTDEC_AREx_NONCE1) 1094
24.5.6. Area x key register 0 (RTDEC_AREx_KEY0) 1094
24.5.7. Area x key register 1 (RTDEC_AREx_KEY1) 1095
24.5.8. Area x key register 2 (RTDEC_AREx_KEY2) 1095
24.5.9. Area x key register 3 (RTDEC_AREx_KEY3) 1095
24.5.10. Interrupt flag register (RTDEC_INTF) 1096
24.5.11. Interrupt flag clear register (RTDEC_INTC) 1097
24.5.12. Interrupt enable register (RTDEC_INTEN) 1097

25. Analog-to-digital converter (ADC) 1099
25.1. Overview 1099
25.2. Characteristics 1099
25.3. Pins and internal signals 1100
25.4. Function overview 1101
25.4.1. Foreground calibration function 1101
25.4.2. Dual clock domain architecture 1102
25.4.3. ADC enable 1102
25.4.4. Single-ended and differential input channels 1102
25.4.5. Routine and inserted sequence 1103
25.4.6. Operation modes 1104
25.4.7. Inserted sequence management 1108
25.4.8. Convert data latch 1108
25.4.9. Conversion result threshold monitor function 1110
25.4.10. Data storage mode 1110
25.4.11. Sample time configuration 1112
25.4.12. External trigger configuration 1113
25.4.13. DMA request 1113
25.4.14. Overflow detection 1113
25.4.15. ADC internal channels 1114
25.4.16. Battery voltage monitoring 1115
25.4.17. Using HPDF to managing the conversion results 1115
25.4.18. Programmable resolution (DRES) 1116
25.4.19. On-chip hardware oversampling 1116
25.5. ADC sync mode 1119
25.5.1. Free mode 1120
25.5.2. Routine parallel mode 1120
25.5.3. Inserted parallel mode 1121
25.5.4. Routine follow-up mode 1122
25.5.5. Trigger rotation mode 1122
25.5.6. Combined routine parallel & inserted parallel mode 1123
25.5.7. Combined routine parallel & trigger rotation mode 1124
25.5.8. Use DMA in ADC sync mode 1125
25.6. ADC interrupts 1125
25.7. Register definition 1127
25.7.1. Status register (ADC_STAT) 1127
25.7.2. Control register 0 (ADC_CTL0) 1128
25.7.3. Control register 1 (ADC_CTL1) 1131
25.7.4. Inserted channel data offset register x(ADC_IOFFx) (x=0…3) 1134
25.7.5. Watchdog high threshold register0 (ADC_WDHT0) 1134
25.7.6. Watchdog low threshold register0 (ADC_WDLT0) 1134

25.7.7. Routine sequence register 0 (ADC_RSQ0) 1135
25.7.8. Routine sequence register 1 (ADC_RSQ1) 1135
25.7.9. Routine sequence register 2 (ADC_RSQ2) 1136
25.7.10. Routine sequence register 3 (ADC_RSQ3) 1137
25.7.11. Routine sequence register 4 (ADC_RSQ4) 1137
25.7.12. Routine sequence register 5 (ADC_RSQ5) 1138
25.7.13. Routine sequence register 6 (ADC_RSQ6) 1138
25.7.14. Routine sequence register 7 (ADC_RSQ7) 1139
25.7.15. Routine sequence register 8 (ADC_RSQ8) 1140
25.7.16. Inserted sequence register 0 (ADC_ISQ0) 1140
25.7.17. Inserted sequence register 1 (ADC_ISQ1) 1141
25.7.18. Inserted sequence register 2 (ADC_ISQ2) 1141
25.7.19. Latch data register x (ADC_LDATAx) (x=0...3) 1142
25.7.20. Routine data register (ADC_RDATA) 1142
25.7.21. Inserted data register (ADC_IDATA) 1143
25.7.22. Latch data control register (ADC_LDCTL) 1143
25.7.23. Oversample control register (ADC_OVSAMPCTL) 1144
25.7.24. Watchdog 1 Channel Selection Register (ADC_WD1SR) 1146
25.7.25. Watchdog 2 Channel Selection Register (ADC_WD2SR) 1146
25.7.26. Watchdog high threshold register1 (ADC_WDHT1) 1147
25.7.27. Watchdog low threshold register1 (ADC_WDLT1) 1148
25.7.28. Watchdog high threshold register2 (ADC_WDHT2) 1148
25.7.29. Watchdog low threshold register2 (ADC_WDLT2) 1149
25.7.30. Differential mode control register (ADC_DIFCTL) 1149
25.7.31. Summary status register (ADC_SSTAT) 1150
25.7.32. Sync control register (ADC_SYNCCTL) 1151
25.7.33. Sync routine data register0 (ADC_SYNCDATA0) 1153
25.7.34. Sync routine data register1 (ADC_SYNCDATA1) 1153

26. Digital-to-analog converter (DAC) 1154

26.1. Overview 1154

26.2. Characteristics 1154

26.3. Function description 1156

26.3.1. DAC enable 1156
26.3.2. DAC output buffer 1156
26.3.3. DAC data configuration 1156
26.3.4. DAC trigger 1156
26.3.5. DAC conversion 1156
26.3.6. DAC noise wave 1157
26.3.7. DAC output voltage 1158
26.3.8. DMA request 1158
26.3.9. DAC concurrent conversion 1158
26.3.10. DAC output buffer calibration 1159

26.3.11. DAC modes 1160
26.3.12. DAC low-power modes 1161
26.4. DAC register 1163
26.4.1. DACx control register 0 (DAC_CTL0) 1163
26.4.2. DACx software trigger register (DAC_SWT) 1165
26.4.3. DACx_OUT0 12-bit right-aligned data holding register (DAC_OUT0_R12DH) 1166
26.4.4. DACx_OUT0 12-bit left-aligned data holding register (DAC_OUT0_L12DH) 1166
26.4.5. DACx_OUT0 8-bit right-aligned data holding register (DAC_OUT0_R8DH) 1167
26.4.6. DACx_OUT1 12-bit right-aligned data holding register (DAC_OUT1_R12DH) 1167
26.4.7. DACx_OUT1 12-bit left-aligned data holding register (DAC_OUT1_L12DH) 1168
26.4.8. DACx_OUT1 8-bit right-aligned data holding register (DAC_OUT1_R8DH) 1168
26.4.9. DACx concurrent mode 12-bit right-aligned data holding register (DACC_R12DH) 1169
26.4.10. DACx concurrent mode 12-bit left-aligned data holding register (DACC_L12DH) 1169
26.4.11. DACx concurrent mode 8-bit right-aligned data holding register (DACC_R8DH) 1170
26.4.12. DACx_OUT0 data output register (DAC_OUT0_DO) 1170
26.4.13. DACx_OUT1 data output register (DAC_OUT1_DO) 1171
26.4.14. DACx status register 0 (DAC_STAT0) 1171
26.4.15. DACx calibration register (DAC_CALR) 1172
26.4.16. DACx mode control register (DAC_MDCR) 1173
26.4.17. DACx sample and keep sample time register 0 (DAC_SKSTR0) 1174
26.4.18. DACx sample and keep sample time register 1 (DAC_SKSTR1) 1174
26.4.19. DACx sample and keep keep time register (DAC_SKKTR) 1175
26.4.20. DACx sample and keep refresh time register (DAC_SKRTR) 1175
27. Comparator (CMP) 1177
27.1. Overview 1177
27.2. Characteristic 1177
27.3. Function overview 1177
27.3.1. CMP clock 1178
27.3.2. CMP I/O configuration 1178
27.3.3. CMP operating mode 1180
27.3.4. CMP Window mode 1180
27.3.5. CMP hysteresis 1180
27.3.6. CMP register write protection 1180
27.3.7. CMP output blanking 1181
27.3.8. CMP voltage scaler function 1181
27.3.9. CMP digital filter 1181
27.3.10. CMP interrupt 1182
27.4. Register definition 1183
27.4.1. CMP status register (CMP_STAT) 1183
27.4.2. CMP interrupt flag clear register (CMP_IFC) 1184
27.4.3. CMP alternate select register (CMP_SR) 1184

27.4.4. CMP0 control/status register (CMP0_CS) 1185
27.4.5. CMP1 control/status register (CMP1_CS) 1187
28. VREF 1191
28.1. Overview 1191
28.2. Characteristics 1191
28.3. Function overview 1191
28.4. Register definition 1193
28.4.1. Control and status register (VREF_CS) 1193
28.4.2. Calibration register (VREF_CALIB) 1193
29. Low power digital temperature sensor (LPDTS) 1195
29.1. Overview 1195
29.2. Characteristics 1195
29.3. Block diagram 1195
29.4. Function overview 1196
29.4.1. LPDTS internal signals 1196
29.4.2. Operating modes 1196
29.4.3. Temperature measurement principles 1196
29.4.4. Sampling time 1197
29.4.5. Trigger input 1198
29.4.6. On-off control and ready flag 1199
29.4.7. LPDTS low-power modes 1199
29.4.8. LPDTS interrupts 1199
29.5. Register definition 1201
29.5.1. Configuration register (LPDTS_CFG) 1201
29.5.2. Sensor T0 data register 1 (LPDTS_SDATA) 1202
29.5.3. Ramp data register (LPDTS_RDATA) 1202
29.5.4. Interrupt threshold register (LPDTS_IT) 1203
29.5.5. Temperature data register (LPDTS_DATA) 1203
29.5.6. Temperature sensor status register (LPDTS_STAT) 1203
29.5.7. Interrupt enable register (LPDTS_INTEN) 1205
29.5.8. Interrupt clear flag register (LPDTS_INTC) 1206
29.5.9. Option register (LPDTS_OP) 1207
30. TFT-LCD interface (TLI) 1208
30.1. Overview 1208
30.2. Characteristics 1208
30.3. Block diagram 1208
30.4. Signal description 1209

30.5. Function overview 1209
30.5.1. LCD display timing 1209
30.5.2. Pixel DMA function 1210
30.5.3. Pixel formats 1211
30.5.4. Layer window and blending function 1211
30.5.5. Layer configuration reload 1212
30.5.6. Dithering function 1213
30.6. Interrupts 1213
30.7. Register definition 1214
30.7.1. Synchronous pulse size register (TLI_SPSZ) 1214
30.7.2. Back-porch size register (TLI_BPSZ) 1214
30.7.3. Active size register (TLI_ASZ) 1215
30.7.4. Total size register (TLI_TSZ) 1215
30.7.5. Control register (TLI_CTL) 1216
30.7.6. Reload layer register (TLI_RL) 1217
30.7.7. Background color register (TLI_BGC) 1218
30.7.8. Interrupt enable register (TLI_INTEN) 1218
30.7.9. Interrupt flag register (TLI_INTF) 1219
30.7.10. Interrupt flag clear register (TLI_INTC) 1219
30.7.11. Line mark register (TLI_LM) 1220
30.7.12. Current pixel position register (TLI_CPPOS) 1220
30.7.13. Status register (TLI_STAT) 1221
30.7.14. Layer x control register (TLI_LxCTL) (x = 0,1) 1221
30.7.15. Layer x horizontal position parameters register (TLI_LxHPOS) (x = 0,1) 1222
30.7.16. Layer x vertical position parameters register (TLI_LxVPOS) (x = 0,1) 1223
30.7.17. Layer x color key register (TLI_LxCKEY) (x = 0,1) 1223
30.7.18. Layer x packeted pixel format register (TLI_LxPPF) (x = 0,1) 1224
30.7.19. Layer x specified alpha register (TLI_LxSA) (x = 0,1) 1224
30.7.20. Layer x default color register (TLI_LxDC) (x = 0,1) 1225
30.7.21. Layer x blending register (TLI_LxBLEND) (x = 0,1) 1225
30.7.22. Layer x frame base address register (TLI_LxFBADDR) (x = 0,1) 1226
30.7.23. Layer x frame line length register (TLI_LxFLLEN) (x = 0,1) 1226
30.7.24. Layer x frame total line number register (TLI_LxFTLN) (x = 0,1) 1227
30.7.25. Layer x look up table register (TLI_LxLUT) (x = 0,1) 1227
31. DSI host controller (DSI) 1229
31.1. Overview 1229
31.2. Characteristics 1229
31.3. Block diagram 1231
31.4. Function overview 1231
31.4.1. High Speed Protocol Handler 1231
31.4.2. Low Speed Protocol Handler 1231

31.4.3. Channelizer 1231
31.4.4. Lane Manager 1232
31.4.5. Error Handler 1232
31.4.6. DPI Interface 1232
31.4.7. AHB slave Interface 1232
31.4.8. DCS Generic FIFO logics 1233
31.4.9. DPI Data FIFO 1233
31.4.10. D-PHY 1233
31.4.11. RAM 1233
31.5. DPI stream to DSI packets 1234
31.6. Contention and Recovery Schemes 1236
31.7. Fault mode recovery schemes 1237
31.8. Pixel to Byte encoding formats 1237
31.8.1. RGB666 data Transmission and storage 1237
31.8.2. RGB565 data transmission and storage 1238
31.8.3. RGB888 or RGB666 loosely packed data transmission and storage 1239
31.9. DSI TX reset and clock generation 1240
31.9.1. Reset Generation 1240
31.9.2. Clocks 1241
31.10. DPHY 1241
31.10.1. Overview of MIPI D-PHY Devices 1241
31.10.2. Overview of D-PHY TX 1243
31.10.3. DPHY TX features 1243
31.10.4. ACS lane swap feature 1244
31.10.5. Functional Description of DPHY Pads 1245
31.10.6. DPHY PLL 1245
31.11. Programing procedure 1249
31.11.1. DSI Host Initializations by the processor 1249
31.11.2. DPI 1249
31.11.3. DCS/Generic through AHB interface 1251
31.11.4. DSI Host Operations in dual channel mode 1255
31.11.5. Blanking Time management 1256
31.12. TLI DMA 1257
31.13. Tearing effect 1258
31.14. WMS hardware unpacking 1258
31.14.1. WMS automatic unpacking: Using HS command mode to send WMS as an example 1258
31.14.2. WMS automatic unpacking: Using LP command mode to send WMS as an example 1259
31.15. Register definition 1260
31.15.1. DSI Host device ready register (DSI_DEVRDY) 1260
31.15.2. DSI Host interrupt status register0 (DSI_INTSTAT0) 1261

31.15.3. DSI Host interrupt enable register 0 (DSI_INTEN0) 1264
31.15.4. DSI Host function program register (DSI_FUNCPRG) 1267
31.15.5. DSI Host high speed transmission timeout register (DSI_HSTXTO) 1268
31.15.6. DSI Host low power reception timeout register (DSI_LPRXTO) 1268
31.15.7. DSI Host turn around timeout register (DSI_TATO) 1269
31.15.8. DSI Host device reset timer register (DSI_DEVRSTTM) 1269
31.15.9. DSI Host DPI resolution register (DSI_DPIRES) 1269
31.15.10. DSI Host DCS word count register (DSI_DCSWC) 1270
31.15.11. DSI Host horizontal sync active count configuration register (DSI_HSACFG) 1270
31.15.12. DSI Host horizontal back porch count configuration register (DSI_HBPCFG) 1271
31.15.13. DSI Host horizontal front porch count configuration register (DSI_HFPCFG) 1271
31.15.14. DSI Host horizontal active area count configuration register (DSI_HACFG) 1272
31.15.15. DSI Host vertical sync active count configuration register (DSI_VSACFG) 1272
31.15.16. DSI Host vertical back porch count configuration register (DSI_VBPCFG) 1272
31.15.17. DSI Host vertical front porch count configuration register (DSI_VFPCFG) 1273
31.15.18. DSI Host data lane high low switch count configuration register (DSI_DLCFG) 1273
31.15.19. DSI Host DPI control register (DSI_DPICTL) 1274
31.15.20. DSI Host Dphy PLL lock count register (DSI_DPHY_PLLLK) 1274
31.15.21. DSI Host initialization count configuration register (DSI_INITCFG) 1275
31.15.22. DSI Host maximum return packet control register(DSI_MAXRPKTCTL) 1275
31.15.23. DSI Host Video mode format register (DSI_VMFMT) 1276
31.15.24. DSI Host clock EOT control register (DSI_CKEOTCTL) 1276
31.15.25. DSI Host polarity control register (DSI_PLCTL) 1277
31.15.26. DSI Host clock lane switching configuration register (DSI_CKLSWCFG) 1277
31.15.27. DSI Host low power byte clock configuration register (DSI_LPBCLKCFG) 1278
31.15.28. DSI Host Dphy parameter configuration register (DSI_DPHY_PCFG) 1278
31.15.29. DSI Host Dphy clock lane timing parameter configuration register (DSI_DPHY_CLTMPCFG) 1278
31.15.30. DSI Host Dphy reset enable to DFE register (DSI_DPHY_RSTENDFE) 1279
31.15.31. DSI Host Dphy trim control register 0 (DSI_DPHY_TRIMCTL0) 1279
31.15.32. DSI Host Dphy trim control register 1 (DSI_DPHY_TRIMCTL1) 1281
31.15.33. DSI Host Dphy trim control register 2 (DSI_DPHY_TRIMCTL2) 1282
31.15.34. DSI Host deskew control register (DSI_DSKCTL) 1284
31.15.35. DSI Host periodic deskew timer register(DSI_DSKTM) 1284
31.15.36. DSI Host deskew status register (DSI_DSKSTAT) 1285
31.15.37. DSI Host error auto recovery control register (DSI_EARCTL) 1285
31.15.38. DSI Host MIPI direction and DPI status register (DSI_MIPIDIRDPISTAT) 1286
31.15.39. DSI Host Polarity Swap register (DSI_PLSWAP) 1287
31.15.40. DSI Host frequency mode control register (DSI_FMCTL) 1287
31.15.41. DSI Host Tx data width configuration register (DSI_TXDTWCFG) 1288
31.15.42. DSI Host LP control register (DSI_LPCTL) 1288
31.15.43. DSI Host LP data register (DSI_LPDATA) 1289
31.15.44. DSI Host HS control register (DSI_HSCTL) 1289
31.15.45. DSI Host HS data register (DSI_HSDATA) 1290

31.15.46. DSI Host FIFO status register (DSI_FIFOSTAT) 1290
31.15.47. DSI Host interrupt status register1 (DSI_INTSTAT1) 1292
31.15.48. DSI Wrapper control register (DSI_WCTL) 1292
31.15.49. DSI Wrapper interrupt enable register (DSI_WINTEN) 1293
31.15.50. DSI Wrapper interrupt and status register (DSI_WINTSTAT) 1294
31.15.51. DSI Wrapper interrupt flag clear register (DSI_WINTC) 1295
31.15.52. DSI Wrapper RGB data format register (DSI_WRGBDF) 1295
31.15.53. DSI Wrapper command mode data source register (DSI_WCMDDS) 1296
31.15.54. DSI Wrapper DMA request control register (DSI_WDMARCTL) 1297
31.15.55. DSI Wrapper WMS write control register (DSI_WWMS_WCTL) 1297
31.15.56. DSI Wrapper WMS interrupt mask register (DSI_WWMS_INTMSK) 1298
31.15.57. DSI Wrapper WMS header message register (DSI_WWMS_HEAD) 1298
31.15.58. DSI Wrapper WMS size control register (DSI_WWMS_SIZE) 1299
31.15.59. DSI Wrapper WMS status register 0 (DSI_WWMS_STAT0) 1299
31.15.60. DSI Wrapper WMS status register 1 (DSI_WWMS_STAT1) 1300
31.15.61. DSI Host interrupt enable register 1 (DSI_INTEN1) 1300

32. Fast Fourier Transform (FFT) 1302

32.1. Overview 1302

32.2. Characteristics 1302

32.3. Functional description 1302

32.4. Data format 1303

32.5. Radix-2 FFT 1304

32.6. Butterfly unit 1304

32.7. IFFT mode 1305

32.8. FFT SRAM 1305

32.9. FFT loop mode 1305

32.10. Operation guide 1306

32.11. FFT interrupts 1307

32.12. Register definition 1308

32.12.1. Control and status register (FFT_CSR) 1308
32.12.2. Real start address register(FFT_RESADDR) 1310
32.12.3. Image start address register (FFT_IMSADDR) 1310
32.12.4. Window start address register (FFT_WSADDR) 1310
32.12.5. Output start address register (FFT_OSADDR) 1311
32.12.6. Loop length register (FFT_LOOPLEN) 1311

33. High-Performance Digital Filter (HPDF) 1313

33.1. Overview 1313

33.2. Characteristics 1313

33.3. Function overview 1314
33.3.1. HPDF Block Diagram 1314
33.3.2. HPDF on-off control 1315
33.3.3. HPDF clock 1315
33.3.4. Multiplex serial data channel 1316
33.3.5. Parallel data input 1324
33.3.6. Regular group conversion 1326
33.3.7. Inserted group conversion 1327
33.3.8. Digital filter 1329
33.3.9. Integrator 1330
33.3.10. Threshold monitor 1330
33.3.11. Sum threshold monitor 1332
33.3.12. Malfunction monitor 1334
33.3.13. Extremes monitor 1335
33.3.14. Data unit 1335
33.3.15. HPDF interrupt 1337
33.4. Register definition 1338
33.4.1. HPDF channel x registers (x=0, …, 7) 1338
33.4.2. HPDF filter y registers (y=0, …, 3) 1343
34. Trigonometric Math Unit (TMU) 1362
34.1. Overview 1362
34.2. Characteristics 1362
34.3. Block diagram 1362
34.4. Function overview 1363
34.4.1. Data format and configuration 1363
34.4.2. Floating-point data format 1364
34.4.3. Mode configuration 1365
34.4.4. TMU operation pending 1377
34.4.5. Zero-overhead mode 1377
34.4.6. Interrupt and DMA requests 1378
34.5. Registers definition 1379
34.5.1. Control and status register (TMU_CS) 1379
34.5.2. Input data register (TMU_IDATA) 1382
34.5.3. Output data register (TMU_ODATA) 1382
35. Filter arithmetic accelerator (FAC) 1384
35.1. Overview 1384
35.2. Characteristics 1384
35.3. Function overview 1384
35.3.1. General description 1384
35.3.2. Local memory and buffers 1385

35.3.3. Input buffers 1386
35.3.4. Output buffer 1388
35.3.5. Initialization functions 1389
35.3.6. Filter functions 1390
35.3.7. Fixed point data format 1392
35.3.8. Float point data format 1393
35.3.9. FIR filters 1393
35.3.10. IIR filters 1395
35.4. Register definition 1397
35.4.1. FAC X0 buffer configure register (FAC_X0BCFG) 1397
35.4.2. FAC X1 buffer configure register (FAC_X1BCFG) 1397
35.4.3. FAC Y buffer configure register (FAC_YBCFG) 1398
35.4.4. FAC Parameter configure register (FAC_PARACFG) 1399
35.4.5. FAC Control register (FAC_CTL) 1399
35.4.6. FAC Status register (FAC_STAT) 1401
35.4.7. FAC write data register (FAC_WDATA) 1402
35.4.8. FAC read data register (FAC_RDATA) 1403
36. Image processing accelerator (IPA) 1404
36.1. Overview 1404
36.2. Characteristics 1404
36.3. Block diagram 1405
36.4. Signal description 1405
36.5. Function overview 1406
36.5.1. Conversion operation 1407
36.5.2. Foreground and background LUT 1408
36.5.3. Foreground and background pixel channel extension (PCE) 1408
36.5.4. Foreground channel scaling 1412
36.5.5. Blending 1413
36.5.6. Destination pixel channel compression (PCC) 1414
36.5.7. Rotation 1415
36.5.8. Inter-timer 1415
36.5.9. Line mark 1416
36.5.10. Transfer flow 1416
36.5.11. Configuration 1417
36.6. Interrupts 1421
36.7. Register definition 1424
36.7.1. Control register (IPA_CTL) 1424
36.7.2. Interrupt flag register (IPA_INTF) 1426
36.7.3. Interrupt flag clear register (IPA_INTC) 1427
36.7.4. Foreground memory base address register (IPA_FMADDR) 1428

36.7.5. Foreground line offset register (IPA_FLOFF) 1428
36.7.6. Background memory base address register (IPA_BMADDR) 1429
36.7.7. Background line offset register (IPA_BLOFF) 1429
36.7.8. Foreground pixel control register (IPA_FPCTL) 1430
36.7.9. Foreground pixel value register (IPA_FPV) 1432
36.7.10. Background pixel control register (IPA_BPCTL) 1432
36.7.11. Background pixel value register (IPA_BPV) 1434
36.7.12. Foreground LUT memory base address register (IPA_FLMADDR) 1435
36.7.13. Background LUT memory base address register (IPA_BLMADDR) 1435
36.7.14. Destination pixel control register (IPA_DPCTL) 1436
36.7.15. Destination pixel value register (IPA_DPV) 1437
36.7.16. Destination memory base address register (IPA_DMADDR) 1440
36.7.17. Destination line offset register (IPA_DLOFF) 1441
36.7.18. Image size register (IPA_IMS) 1441
36.7.19. Line mark register (IPA_LM) 1442
36.7.20. Inter-timer control register (IPA_ITCTL) 1442
36.7.21. Bilinear scaling control register (IPA_BSCTL) 1443
36.7.22. Scaling destination image size register (IPA_DIMS) 1443
36.7.23. Foreground even frame/UV memory base address register (IPA_EF_UV_MADDR) 1444
36.7.24. Color space conversion coefficient configure register 0 (IPA_CSCC_CFG0) 1445
36.7.25. Color space conversion coefficient configure register 1 (IPA_CSCC_CFG1) 1445
36.7.26. Color space conversion coefficient configure register 2 (IPA_CSCC_CFG2) 1446

37. Encoder Divided-Output controller (EDOUT) 1447

37.1. Overview 1447

37.2. Characteristics 1447

37.3. Function overview 1447

37.4. Z-phase output mode 1448

37.5. Operation guidance 1449

37.5.1. EDOUT initialization 1449
37.5.2. EDOUT update processing 1450
37.5.3. EDOUT working example 1450

37.6. Register definition 1452

37.6.1. Control register (EDOUT_CTL) 1452
37.6.2. Enable register (EDOUT_ENABLE) 1452
37.6.3. Location register (EDOUT_LOC) 1453
37.6.4. Output counter register (EDOUT_OCNT) 1453
37.6.5. Location counter register (EDOUT_LCNT) 1454
37.6.6. Z-phase configuration register (EDOUT_ZCR) 1455

38. Encoder Interface Module (EDIM) 1456

38.1. A-format encoder interface module (AFMT) 1456

38.1.1. Overview 1456
38.1.2. Characteristics 1456
38.1.3. Block diagram 1457
38.1.4. Function overview 1457
38.1.5. Registers definition 1466
38.2. T-format encoder interface module (TFMT) 1473
38.2.1. Overview 1473
38.2.2. Characteristics 1473
38.2.3. Block diagram 1473
38.2.4. Function overview 1474
38.2.5. Registers definition 1478
38.3. EnDat 2.2 encoder interface module (ENDAT) 1486
38.3.1. Overview 1486
38.3.2. Characteristics 1486
38.3.3. Block diagram 1487
38.3.4. Function overview 1488
38.3.5. Registers definition 1497
38.4. BiSS-C encoder interface module (BISS) 1511
38.4.1. Overview 1511
38.4.2. Characteristics 1511
38.4.3. Block diagram 1511
38.4.4. Function overview 1512
38.4.5. Registers definition 1521
38.5. HIPERFACE DSL encoder interface module (HDSL) 1536
38.5.1. Overview 1536
38.5.2. Characteristics 1536
38.5.3. Block diagram 1536
38.5.4. Function overview 1537
38.5.5. Registers definition 1543
39. Resolver-to-digital module (RDCM) 1585
39.1. Overview 1585
39.2. Characteristics 1585
39.3. Block diagram 1586
39.4. Function overview 1587
39.4.1. RDCM clock 1587
39.4.2. Sinusoidal excitor 1587
39.4.3. High-pass filter 1589
39.4.4. Synthetic Reference Generation 1589
39.4.5. Phase sensitive demodulator 1592
39.4.6. Group delay summary 1595

39.4.7. Nonorthogonal phase mismatch compensation 1596
39.4.8. Type II tracking loop 1597
39.4.9. Data output rate 1598
39.4.10. Timestamp support 1598
39.4.11. Result data 1599
39.4.12. Fault Detection 1600
39.4.13. DMA mode 1602
39.4.14. Overflow detection 1603
39.4.15. Interrupts 1603
39.4.16. Simulation mode 1604
39.5. Registers definition 1605
39.5.1. Control and status register (RDCM_CS) 1605
39.5.2. Control register (RDCM_CTL) 1608
39.5.3. Exciter configuration register 0 (RDCM_EXCFG0) 1611
39.5.4. Exciter configuration register 1 (RDCM_EXCFG1) 1611
39.5.5. High pass filter configuration register 0 (RDCM_HFCFG0) 1612
39.5.6. High pass filter configuration register 1 (RDCM_HFCFG1) 1612
39.5.7. Reference frequency mixer configuration register 0 (RDCM_RFMCFG0) 1613
39.5.8. Reference frequency mixer configuration register 1 (RDCM_RFMCFG1) 1613
39.5.9. Zero crossing threshold configuration register 0 (RDCM_ZTCFG0) 1614
39.5.10. Zero crossing threshold configuration register 1 (RDCM_ZTCFG1) 1614
39.5.11. Mismatch counter average configuration register (RDCM_MCACFG) 1615
39.5.12. Demodulator configuration register (RDCM_DEMCFG) 1616
39.5.13. CIC3 shift and gain configuration register 0 (RDCM_CSGCFG0) 1617
39.5.14. CIC3 shift and gain configuration register 1 (RDCM_CSGCFG1) 1618
39.5.15. CIC3 configuration register 0 (RDCM_CICCFG0) 1618
39.5.16. CIC3 configuration register 1 (RDCM_CICCFG1) 1619
39.5.17. CIC3 decimation counter register (RDCM_CDCNT) 1620
39.5.18. Nonorthogonal phase mismatch configuration register (RDCM_NPMCFG) 1620
39.5.19. Tracking gain configuration register (RDCM_TRKGAINCFG) 1621
39.5.20. Tracking integrator configuration register (RDCM_TRKINTCFG) 1621
39.5.21. Tracking compensation configuration register (RDCM_TRKCMPCFG) 1622
39.5.22. Velocity smooth filter configuration register (RDCM_VSFCFG) 1622
39.5.23. High pass filter data register (RDCM_HFDATA) 1623
39.5.24. Reference frequency mixer data register (RDCM_RFMDATA) 1623
39.5.25. Demodulator data register (RDCM_DEMDATA) 1623
39.5.26. Tracking data register (RDCM_TRKDATA) 1624
39.5.27. Monitor data register (RDCM_MONDATA) 1624
39.5.28. Timestamp control register (RDCM_TSCTL) 1625
39.5.29. Timestamp data register (RDCM_TSDATA) 1625
39.5.30. FIFO control and status register (RDCM_FIFOCS) 1626
39.5.31. FIFO data register (RDCM_FIFODATA) 1627
39.5.32. Envelop threshold configuration register 0 (RDCM_ETHCFG0) 1627

39.5.33. Envelop threshold configuration register 1 (RDCM_ETHCFG1) 1628
39.5.34. Monitor signal threshold configuration register (RDCM_MTHCFG) 1628
39.5.35. Tracking error signal threshold configuration register (RDCM_TRKERRTHCFG) 1629
39.5.36. Threshold match counter register (RDCM_THCNT) 1630
39.5.37. Simulation data input register (RDCM_SIMDI) 1630
39.5.38. Data input software selection register (RDCM_DISWSEL) 1631
39.5.39. PWM configuration register (RDCM_PWMCFG) 1631
39.5.40. Position data register (RDCM_POSDATA) 1632
39.5.41. Velocity data register (RDCM_VELDATA) 1632

40. Universal synchronous / asynchronous receiver / transmitter (USART) ... 1634

40.1. Overview 1634

40.2. Characteristics 1634

40.3. Function overview 1636

40.3.1. USART frame format 1637
40.3.2. Baud rate generation 1637
40.3.3. USART transmitter 1638
40.3.4. USART receiver 1639
40.3.5. Use DMA for data buffer access 1641
40.3.6. Hardware flow control 1642
40.3.7. Multi-processor communication 1644
40.3.8. LIN mode 1645
40.3.9. Synchronous mode 1646
40.3.10. IrDA SIR ENDEC mode 1646
40.3.11. Half-duplex communication mode 1648
40.3.12. Internal loopback mode 1648
40.3.13. Smartcard (ISO7816-3) mode 1648
40.3.14. ModBus communication 1650
40.3.15. Receive / Transmitter FIFO 1651
40.3.16. Wakeup from deep-sleep mode 1652
40.3.17. USART interrupts 1652

40.4. Register definition 1655

40.4.1. Control register 0 (USART_CTL0) 1655
40.4.2. Control register 1 (USART_CTL1) 1658
40.4.3. Control register 2 (USART_CTL2) 1661
40.4.4. Baud rate generator register (USART_BAUD) 1664
40.4.5. Prescaler and guard time configuration register (USART_GP) 1665
40.4.6. Receiver timeout register (USART_RT) 1666
40.4.7. Command register (USART_CMD) 1666
40.4.8. Status register (USART_STAT) 1667
40.4.9. Interrupt status clear register (USART_INTC) 1672
40.4.10. Receive data register (USART_RDATA) 1673
40.4.11. Transmit data register (USART_TDATA) 1674

40.4.12. USART coherence control register (USART_CHC) 1674
40.4.13. USART FIFO control and status register (USART_FCS) 1675
41. Serial peripheral interface/Inter-IC sound (SPI/I2S) 1679
41.1. Overview 1679
41.2. Characteristics 1679
41.2.1. SPI characteristics 1679
41.2.2. I2S characteristics 1679
41.3. SPI function overview 1680
41.3.1. SPI block diagram 1680
41.3.2. SPI signal description 1681
41.3.3. SPI clock timing and data format 1682
41.3.4. SPI clock delay mode 1683
41.3.5. RxFIFO and TxFIFO 1685
41.3.6. NSS function 1688
41.3.7. SPI operation modes 1690
41.3.8. DMA function 1699
41.3.9. CRC function 1699
41.3.10. SPI interrupts 1700
41.4. I2S function overview 1704
41.4.1. I2S block diagram 1704
41.4.2. I2S signal description 1705
41.4.3. I2S audio standards 1705
41.4.4. I2S clock 1712
41.4.5. RxFIFO and TxFIFO 1713
41.4.6. Operation 1714
41.4.7. DMA function 1716
41.4.8. I2S interrupts 1716
41.5. Register definition 1719
41.5.1. Control register 0 (SPI_CTL0) 1719
41.5.2. Control register 1 (SPI_CTL1) 1721
41.5.3. Configuration register 0 (SPI_CFG0) 1721
41.5.4. Configuration register 1 (SPI_CFG1) 1724
41.5.5. Interrupt register (SPI_INT) 1726
41.5.6. Status register (SPI_STAT) 1728
41.5.7. Interrupt/Status flags clear register (SPI_STATC) 1730
41.5.8. Data Transfer register (SPI_TDATA) 1731
41.5.9. Data Receive register (SPI_RDATA) 1731
41.5.10. CRC polynomial register (SPI_CRCPOLY) 1732
41.5.11. TX CRC register (SPI_TCRC) 1732
41.5.12. RX CRC register (SPI_RCRC) 1733
41.5.13. Underrun Data register (SPI_URDATA) 1733

41.5.14. I2S control register (SPI_I2SCTL) 1734
41.5.15. Quad_SPI mode control register (SPI_QCTL) 1736
41.5.16. RX clock delay register (SPI_RXDLYCK) 1737
42. Inter-integrated circuit interface (I2C) 1738
42.1. Overview 1738
42.2. Characteristics 1738
42.3. Function overview 1738
42.3.1. Clock requirements 1739
42.3.2. I2C communication flow 1740
42.3.3. Noise filter 1743
42.3.4. I2C timings configuration 1743
42.3.5. I2C reset 1745
42.3.6. Data transfer 1745
42.3.7. I2C slave mode 1747
42.3.8. I2C master mode 1752
42.3.9. SMBus support 1757
42.3.10. SMBus mode 1760
42.3.11. Wakeup from power saving mode 1761
42.3.12. Use DMA for data transfer 1762
42.3.13. I2C error and interrupts 1762
42.3.14. I2C debug mode 1763
42.4. Register definition 1764
42.4.1. Control register 0 (I2C_CTL0) 1764
42.4.2. Control register 1 (I2C_CTL1) 1766
42.4.3. Slave address register 0 (I2C_SADDR0) 1768
42.4.4. Slave address register 1 (I2C_SADDR1) 1769
42.4.5. Timing register (I2C_TIMING) 1770
42.4.6. Timeout register (I2C_TIMEOUT) 1771
42.4.7. Status register (I2C_STAT) 1772
42.4.8. Status clear register (I2C_STATC) 1775
42.4.9. PEC register (I2C_PEC) 1776
42.4.10. Receive data register (I2C_RDATA) 1776
42.4.11. Transmit data register (I2C_TDATA) 1777
42.4.12. Control register 2 (I2C_CTL2) 1777
43. Controller area network (CAN) 1779
43.1. Overview 1779
43.2. Characteristics 1779
43.3. Function overview 1780
43.3.1. Mailbox descriptor 1781
43.3.2. Rx FIFO descriptor 1786

43.3.3. Communication modes 1791
43.3.4. Power saving modes 1793
43.3.5. Data transmission 1794
43.3.6. Data reception 1798
43.3.7. Data reception in Pretended Networking mode 1805
43.3.8. CAN FD operation 1808
43.3.9. Errors and states 1811
43.3.10. Communication parameters 1814
43.3.11. Interrupts 1817
43.4. Example for a typical configuration flow of CAN 1817
43.5. CAN registers 1820
43.5.1. Control register 0 (CAN_CTL0) 1820
43.5.2. Control register 1 (CAN_CTL1) 1822
43.5.3. Timer register (CAN_TIMER) 1824
43.5.4. Receive mailbox public filter register (CAN_RMPUBF) 1824
43.5.5. Error register 0 (CAN_ERR0) 1825
43.5.6. Error register 1 (CAN_ERR1) 1826
43.5.7. Interrupt enable register (CAN_INTEN) 1829
43.5.8. Status register (CAN_STAT) 1829
43.5.9. Control register 2 (CAN_CTL2) 1830
43.5.10. CRC for classical frame register (CAN_CRCC) 1833
43.5.11. Receive FIFO public filter register (CAN_RFIFOPUBF) 1833
43.5.12. Receive FIFO identifier filter matching number register (CAN_RFIFOIFMN) 1834
43.5.13. Bit timing register (CAN_BT) 1834
43.5.14. Receive FIFO/mailbox private filter x register (CAN_RFIFOMPFx)(x=0..31) 1835
43.5.15. Pretended Networking mode control register 0 (CAN_PN_CTL0) 1835
43.5.16. Pretended Networking mode timeout register (CAN_PN_TO) 1837
43.5.17. Pretended Networking mode status register (CAN_PN_STAT) 1837
43.5.18. Pretended Networking mode expected identifier 0 register (CAN_PN_EID0) 1838
43.5.19. Pretended Networking mode expected DLC register (CAN_PN_EDLC) 1839
43.5.20. Pretended Networking mode expected data low 0 register (CAN_PN_EDL0) 1840
43.5.21. Pretended Networking mode expected data low 1 register (CAN_PN_EDL1) 1840
43.5.22. Pretended Networking mode identifier filter / expected identifier 1 register (CAN_PN_IFEID1) 1841
43.5.23. Pretended Networking mode data 0 filter / expected data high 0 register (CAN_PN_DF0EDH0) 1842
43.5.24. Pretended Networking mode data 1 filter / expected data high 1 register (CAN_PN_DF1EDH1) 1843
43.5.25. Pretended Networking mode received wakeup mailbox x control status information register (CAN_PN_RWMxCS)(x=0..3) 1843
43.5.26. Pretended Networking mode received wakeup mailbox x identifier register (CAN_PN_RWMxI)(x=0..3) 1844
43.5.27. Pretended Networking mode received wakeup mailbox x data 0 register

(CAN_PN_RWMxD0)(x=0..3) 1845
43.5.28. Pretended Networking mode received wakeup mailbox x data 1 register (CAN_PN_RWMxD1)(x=0..3) 1845
43.5.29. FD control register (CAN_FDCTL) 1846
43.5.30. FD bit timing register (CAN_FDBT) 1847
43.5.31. CRC for classical and FD frame register (CAN_CRCCFD) 1848
44. OSPI I/O manager(OSPIM) 1849
44.1. Overview 1849
44.2. Characteristics 1849
44.3. Function overview 1849
44.3.1. OSPIM block diagram 1849
44.3.2. OSPIM matrix 1849
44.3.3. OSPIM multiplexer 1850
44.4. Register definition 1850
44.4.1. Control register (OSPIM_CTL) 1851
44.4.2. Port configuration register (OSPIM_PCFGx) (x = 0, 1) 1851
45. Octal-SPI interface(OSPI) 1853
45.1. Overview 1853
45.2. Characteristics 1853
45.3. Function overview 1854
45.3.1. OSPI block diagram 1854
45.3.2. OSPI regular command format 1857
45.3.3. HyperBusTM mode 1861
45.3.4. Wrap support 1864
45.3.5. Refresh support 1864
45.4. Operating modes 1864
45.4.1. Indirect mode 1864
45.4.2. Status polling mode 1866
45.4.3. Memory-mapped mode 1866
45.5. OSPI configuration 1867
45.5.1. OSPI system configuration 1867
45.5.2. OSPI device configuration 1867
45.5.3. OSPI regular command configuration 1867
45.5.4. OSPI HyperBusTM mode configuration 1869
45.6. Data sampling shift 1869
45.7. OSPI send instruction only once 1870
45.8. Busy 1870
45.9. Error management 1870

45.10. OSPI interrupt 1871

45.11. Register definition 1872

45.11.1. Control register (OSPI_CTL) 1872
45.11.2. Device configuration register 0 (OSPI_DCFG0) 1874
45.11.3. Device configuration register 1 (OSPI_DCFG1) 1876
45.11.4. Device configuration register 3 (OSPI_DCFG3) 1877
45.11.5. Status register (OSPI_STAT) 1878
45.11.6. Status clear register (OSPI_STATC) 1879
45.11.7. Data length register (OSPI_DTLEN) 1879
45.11.8. Address register(OSPI_ADDR) 1880
45.11.9. Data register (OSPI_DATA) 1881
45.11.10. Status mask register (OSPI_STATMK) 1881
45.11.11. Status match register(OSPI_STATMATCH) 1882
45.11.12. Interval register (OSPI_INTERVAL) 1882
45.11.13. Transfer configuration register (OSPI_TCFG) 1883
45.11.14. Timing configuration register (OSPI_TIMCFG) 1885
45.11.15. Instruction register (OSPI_INS) 1886
45.11.16. Alternate bytes register (OSPI_ALTE) 1886
45.11.17. Low power timeout register (OSPI_LPOUT) 1887
45.11.18. Wrap transfer configuration register (OSPI_WPTCFG) 1887
45.11.19. Wrap timing configuration register (OSPI_WPTIMCFG) 1890
45.11.20. Wrap instruction register (OSPI_WPINS) 1890
45.11.21. Wrap alternate byte register (OSPI_WPALTE) 1890
45.11.22. Write transfer configuration register (OSPI_WTCFG) 1891
45.11.23. Write timing configuration register (OSPI_WTIMCFG) 1893
45.11.24. Write instruction register (OSPI_WINS) 1894
45.11.25. Write alternate byte register (OSPI_WALTE) 1894
45.11.26. HyperBus™ latency configuration register (OSPI_HBLCFG) 1894

46. Secure digital input/output interface (SDIO) 1896

46.1. Overview 1896

46.2. Characteristics 1896

46.3. SDIO function overview 1897

46.3.1. SDIO bus topology 1897
46.3.2. SDIO operation modes 1899
46.3.3. Block diagram 1900
46.3.4. SDIO pins and internal signals 1900
46.3.5. General description 1901
46.3.6. SDIO adapter 1902
46.3.7. AHB slave interface 1907
46.3.8. AHB master interface 1908
46.3.9. MDMA request 1909
46.3.10. AHB and SDIO_CLK clock relation 1910

46.3.11. Hardware flow control 1910
46.4. Card function overview 1911
46.4.1. Card registers 1911
46.4.2. Commands 1912
46.4.3. Responses 1924
46.4.4. Data packets format 1928
46.4.5. Two status fields of the card 1930
46.5. Programming sequence 1938
46.5.1. Card identification 1938
46.5.2. Boot operation 1939
46.5.3. No data commands 1942
46.5.4. Single block or multiple block write 1942
46.5.5. Single block or multiple block read 1943
46.5.6. Stream write and stream read (e•MMC only) 1944
46.5.7. Erase 1946
46.5.8. Bus width selection 1947
46.5.9. Protection management 1947
46.5.10. Card Lock/Unlock operation 1948
46.5.11. Sleep 1950
46.5.12. CMD12 send timing 1951
46.6. Specific operations 1953
46.6.1. SD I/O specific operations 1953
46.6.2. UHS-I 1957
46.7. Register definition 1959
46.7.1. Power control register (SDIO_PWRCTL) 1959
46.7.2. Clock control register (SDIO_CLKCTL) 1960
46.7.3. Command argument register (SDIO_CMDAGMT) 1962
46.7.4. Command control register (SDIO_CMDCTL) 1962
46.7.5. Command index response register (SDIO_RSPCMDIDX) 1964
46.7.6. Response register (SDIO_RESPx x = 0..3) 1965
46.7.7. Data timeout register (SDIO_DATATO) 1965
46.7.8. Data length register (SDIO_DATALEN) 1966
46.7.9. Data control register (SDIO_DATACTL) 1967
46.7.10. Data counter register (SDIO_DATACNT) 1969
46.7.11. Status register (SDIO_STAT) 1969
46.7.12. Interrupt clear register (SDIO_INTC) 1971
46.7.13. Interrupt enable register (SDIO_INTEN) 1973
46.7.14. ACK timeout register (SDIO_ACKTO) 1974
46.7.15. FIFO data register (SDIO_FIFO) 1975
46.7.16. IDMA control register (SDIO_IDMACTL) 1975
46.7.17. IDMA buffer size register (SDIO_IDMASIZE) 1976
46.7.18. IDMA buffer 0 base address register (SDIO_IDMAADDR0) 1977

46.7.19. IDMA buffer 1 base address register (SDIO_IDMAADDR1) 1977
47. Management data input / output (MDIO) 1979
47.1. Overview 1979
47.2. Characteristics 1979
47.3. Pins and internal signals 1979
47.4. Function overview 1980
47.4.1. Frame structure 1980
47.4.2. Typical Usage Sequence 1982
47.5. Register definition 1984
47.5.1. Control register (MDIO_CTL) 1984
47.5.2. Received frame information register (MDIO_RFRM) 1984
47.5.3. Received data register (MDIO_RDATA) 1985
47.5.4. Received address register (MDIO_RADDR) 1985
47.5.5. Transfer data register (MDIO_TDATA) 1986
47.5.6. Configuration register (MDIO_CFG) 1986
47.5.7. Status register (MDIO_STAT) 1987
47.5.8. Interrupt enable register (MDIO_INTEN) 1988
47.5.9. Pin value register (MDIO_PIN) 1989
47.5.10. Timeout register (MDIO_TO) 1989
48. External memory controller (EXMC) 1991
48.1. Overview 1991
48.2. Characteristics 1991
48.3. Function overview 1991
48.3.1. Block diagram 1991
48.3.2. Bus interface 1992
48.3.3. AXI error 1992
48.3.4. Basic regulation of EXMC access 1993
48.3.5. External device address mapping 1994
48.3.6. NOR/PSRAM controller 1998
48.3.7. NAND flash controller 2019
48.3.8. SDRAM controller 2023
48.4. Register definition 2035
48.4.1. NOR/PSRAM controller registers 2035
48.4.2. NAND flash controller registers 2040
48.4.3. SDRAM controller registers 2045
49. Digital camera interface (DCI) 2053
49.1. Overview 2053
49.2. Characteristics 2053

49.3. Block diagram 2053
49.4. Signal description 2054
49.5. Function overview 2054
49.5.1. DCI hardware synchronization mode 2054
49.5.2. Embedded synchronization mode 2055
49.5.3. CCIR656 mode 2055
49.5.4. Capture data using snapshot or continuous capture modes 2058
49.5.5. Window function 2058
49.5.6. Pixel formats, data padding and DMA 2058
49.6. Interrupts 2059
49.7. Register definition 2061
49.7.1. Control register (DCI_CTL) 2061
49.7.2. Status register0 (DCI_STAT0) 2062
49.7.3. Status register1 (DCI_STAT1) 2063
49.7.4. Interrupt enable register (DCI_INTEN) 2064
49.7.5. Interrupt flag register (DCI_INTF) 2065
49.7.6. Interrupt flag clear register (DCI_INTC) 2066
49.7.7. Synchronization codes register (DCI_SC) 2067
49.7.8. Synchronization codes unmask register (DCI_SCUMSK) 2068
49.7.9. Cropping window start position register (DCI_CWSPOS) 2068
49.7.10. Cropping window size register (DCI_CWSZ) 2069
49.7.11. DATA register (DCI_DATA) 2069
50. General parallel slave interface(GPSI) 2070
50.1. Overview 2070
50.2. Characteristics 2070
50.3. Function overview 2070
50.3.1. Block diagram 2070
50.3.2. Pins and Signals 2071
50.3.3. Clock 2072
50.3.4. GPSI optional control signals 2072
50.3.5. Data Transfer and DMA 2077
50.4. Interrupt requests 2079
50.5. Register definition 2079
50.5.1. Control register (GPSI_CTL) 2079
50.5.2. Status register (GPSI_STAT) 2081
50.5.3. Interrupt status register (GPSI_INTF) 2082
50.5.4. Interrupt enable register (GPSI_INTEN) 2082
50.5.5. Masked interrupt status register (GPSI_MINTF) 2083
50.5.6. Interrupt clear register (GPSI_INTC) 2083
50.5.7. Data register (GPSI_DATA) 2084

51. Receiver of Sony/Philips Digital Interface (RSPDIF) 2085
51.1. Overview 2085
51.2. Characteristics 2085
51.3. Function overview 2085
51.3.1. RSPDIF block diagram 2085
51.3.2. S/PDIF protocol 2087
51.3.3. RSPDIF 2089
51.3.4. RSPDIF synchronization process 2093
51.3.5. RSPDIF state machine 2095
51.3.6. RSPDIF data reception management 2098
51.3.7. RSPDIF clock management 2101
51.3.8. DMA function 2102
51.3.9. Status、error and interrupt 2103
51.4. Register definition 2108
51.4.1. Control register (RSPDIF_CTL) 2108
51.4.2. Interrupt enable register (RSPDIF_INTEN) 2110
51.4.3. Status register (RSPDIF_STAT) 2111
51.4.4. Status flag clear register (RSPDIF_STATC) 2114
51.4.5. RX data register (RSPDIF_DATA) 2114
51.4.6. RX Channel status register (RSPDIF_CHSTAT) 2117
51.4.7. RX data threshold register (RSPDIF_DTH) 2117
52. Serial Audio Interface (SAI) 2119
52.1. Overview 2119
52.2. Characteristics 2119
52.3. Function overview 2121
52.3.1. Block diagram 2121
52.3.2. Clock divider 2122
52.3.3. Operating mode 2123
52.3.4. SAI synchronization mode 2124
52.3.5. Frame configuration 2125
52.3.6. Slot configuration 2127
52.3.7. Data configuration 2129
52.3.8. Internal FIFO 2129
52.3.9. PDM interface 2130
52.3.10. AC'97 link controller 2137
52.3.11. SPDIF Output 2139
52.3.12. Stereo/Mono 2140
52.3.13. Mute 2140
52.3.14. Compander 2141
52.3.15. Output drive 2144
52.3.16. IO management 2145

52.3.17. DMA interface 2145
52.3.18. Enable/Disable 2145
52.3.19. Error flags 2146
52.3.20. Interrupts 2148
52.4. Register definition 2150
52.4.1. Synchronize configuration register (SAI_SYNCFG) 2150
52.4.2. Block x configuration register0 (SAI_BxCFG0) (x = 0,1) 2150
52.4.3. Block x configuration register1 (SAI_BxCFG1) (x = 0,1) 2153
52.4.4. Block x frame configuration register (SAI_BxFCFG) (x = 0,1) 2155
52.4.5. Block x slot configuration register (SAI_BxSCFG) (x = 0,1) 2157
52.4.6. Block x interrupt enable register (SAI_BxINTEN) (x = 0,1) 2158
52.4.7. Block x status register (SAI_BxSTAT) (x = 0,1) 2159
52.4.8. Block x interrupt flag clear register (SAI_BxINTC) (x = 0,1) 2161
52.4.9. Block x data register (SAI_BxDATA) (x = 0,1) 2162
52.4.10. PDM control register (SAI_PDMCTL) 2163
52.4.11. PDM configuration register (SAI_PDMCFG) 2164
53. Universal serial bus High-Speed interface (USBHS) 2166
53.1. Overview 2166
53.2. Characteristics 2166
53.3. Block diagram 2167
53.4. Signal description 2167
53.5. Function overview 2168
53.5.1. USBHS PHY selection, clocks and working modes 2168
53.5.2. USB host function 2171
53.5.3. USB device function 2174
53.5.4. OTG function overview 2175
53.5.5. Data FIFO 2177
53.5.6. DMA function 2180
53.5.7. Operation guide 2181
53.6. Interrupts 2187
53.7. Register definition 2189
53.7.1. Global registers 2189
53.7.2. Host control and status registers 2218
53.7.3. Device control and status registers 2232
53.7.4. Power and clock control register (USBHS_PWRCLKCTL) 2261
54. Ethernet (ENET) 2263
54.1. Overview 2263
54.2. Characteristics 2263
54.2.1. Block diagram 2264

54.2.2. MAC 802.3 Ethernet packet description 2265
54.2.3. Ethernet signal description 2266
54.3. Function overview 2268
54.3.1. Interface configuration 2268
54.3.2. MAC function overview 2272
54.3.3. DMA controller description 2283
54.3.4. MAC statistics counters: MSC 2308
54.3.5. Wake up management: WUM 2308
54.3.6. Precision time protocol: PTP 2311
54.3.7. Example for a typical configuration flow of Ethernet 2316
54.3.8. Ethernet interrupts 2317
54.4. Register definition 2320
54.4.1. MAC configuration register (ENET_MAC_CFG) 2320
54.4.2. MAC frame filter register (ENET_MAC_FRMF) 2322
54.4.3. MAC hash list high register (ENET_MAC_HLH) 2324
54.4.4. MAC hash list low register (ENET_MAC_HLL) 2325
54.4.5. MAC PHY control register (ENET_MAC_PHY_CTL) 2325
54.4.6. MAC PHY data register (ENET_MAC_PHY_DATA) 2326
54.4.7. MAC flow control register (ENET_MAC_FCTL) 2326
54.4.8. MAC VLAN tag register (ENET_MAC_VLT) 2328
54.4.9. MAC remote wakeup frame filter register (ENET_MAC_RWFF) 2329
54.4.10. MAC wakeup management register (ENET_MAC_WUM) 2329
54.4.11. MAC debug register (ENET_MAC_DBG) 2330
54.4.12. MAC interrupt flag register (ENET_MAC_INTF) 2332
54.4.13. MAC interrupt mask register (ENET_MAC_INTMSK) 2333
54.4.14. MAC address 0 high register (ENET_MAC_ADDR0H) 2334
54.4.15. MAC address 0 low register (ENET_MAC_ADDR0L) 2334
54.4.16. MAC address 1 high register (ENET_MAC_ADDR1H) 2335
54.4.17. MAC address 1 low register (ENET_MAC_ADDR1L) 2336
54.4.18. MAC address 2 high register (ENET_MAC_ADDR2H) 2336
54.4.19. MAC address 2 low register (ENET_MAC_ADDR2L) 2337
54.4.20. MAC address 3 high register (ENET_MAC_ADDR3H) 2337
54.4.21. MAC address 3 low register (ENET_MAC_ADDR3L) 2338
54.4.22. MAC flow control threshold register (ENET_MAC_FCTH) 2338
54.4.23. MSC control register (ENET_MSC_CTL) 2339
54.4.24. MSC receive interrupt flag register (ENET_MSC_RINTF) 2340
54.4.25. MSC transmit interrupt flag register (ENET_MSC_TINTF) 2341
54.4.26. MSC receive interrupt mask register (ENET_MSC_RINTMSK) 2342
54.4.27. MSC transmit interrupt mask register (ENET_MSC_TINTMSK) 2343
54.4.28. MSC transmitted good frames after a single collision counter register (ENET_MSC_SCCNT) 2344
54.4.29. MSC transmitted good frames after more than a single collision counter register (ENET_MSC_MSCCNT) 2344

54.4.30. MSC transmitted good frames counter register (ENET_MSC_TGFCNT) 2345
54.4.31. MSC received frames with CRC error counter register (ENET_MSC_RFCECNT) 2345
54.4.32. MSC received frames with alignment error counter register (ENET_MSC_RFAECNT) .. 2345
54.4.33. MSC received good unicast frames counter register (ENET_MSC_RGUFCNT) 2346
54.4.34. PTP time stamp control register (ENET_PTP_TSCTL) 2346
54.4.35. PTP subsecond increment register (ENET_PTP_SSINC) 2349
54.4.36. PTP time stamp high register (ENET_PTP_TSH) 2349
54.4.37. PTP time stamp low register (ENET_PTP_TSL) 2350
54.4.38. PTP time stamp update high register (ENET_PTP_TSUH) 2350
54.4.39. PTP time stamp update low register (ENET_PTP_TSUL) 2351
54.4.40. PTP time stamp addend register (ENET_PTP_TSADDEND) 2351
54.4.41. PTP expected time high register (ENET_PTP_ETH) 2352
54.4.42. PTP expected time low register (ENET_PTP_ETL) 2352
54.4.43. PTP time stamp flag register (ENET_PTP_TSF) 2353
54.4.44. PTP PPS control register (ENET_PTP_PPSCTL) 2353
54.4.45. DMA bus control register (ENET_DMA_BCTL) 2354
54.4.46. DMA transmit poll enable register (ENET_DMA_TPEN) 2356
54.4.47. DMA receive poll enable register (ENET_DMA_RPEN) 2356
54.4.48. DMA receive descriptor table address register (ENET_DMA_RDTADDR) 2357
54.4.49. DMA transmit descriptor table address register (ENET_DMA_TDTADDR) 2358
54.4.50. DMA status register (ENET_DMA_STAT) 2358
54.4.51. DMA control register (ENET_DMA_CTL) 2362
54.4.52. DMA interrupt enable register (ENET_DMA_INTEN) 2365
54.4.53. DMA missed frame and buffer overflow counter register (ENET_DMA_MFBOCNT) 2367
54.4.54. DMA receive state watchdog counter register (ENET_DMA_RSWDC) 2367
54.4.55. DMA current transmit descriptor address register (ENET_DMA_CTDADDR) 2368
54.4.56. DMA current receive descriptor address register (ENET_DMA_CRDADDR) 2368
54.4.57. DMA current transmit buffer address register (ENET_DMA_CTBADDR) 2369
54.4.58. DMA current receive buffer address register (ENET_DMA_CRBADDR) 2369

55. EtherCAT® SubDevice controller (ESC) 2371

55.1. Overview 2371

55.2. Characteristics 2371
55.2.1. Block diagram 2371
55.2.2. EtherCAT SubDevice Controller Function Blocks 2372

55.3. Function overview 2373
55.3.1. FMMU 2373
55.3.2. SyncManager 2374
55.3.3. Distributed Clocks 2375
55.3.4. EtherCAT State Machine 2375
55.3.5. EEPROM 2376
55.3.6. Interrupts 2377
55.3.7. LED 2378

55.3.8. Station management interface(SMI) 2378
55.3.9. AHB direct / indirect access 2379
55.3.10. Register protection in BUSY state 2381
55.3.11. OPB transmission timeout function 2381
55.4. ESC Core Control Register definition 2382
55.4.1. ESC CCTL data register (ESC_CCTL_DATA) 2382
55.4.2. ESC CCTL command register (ESC_CCTL_CMD) 2382
55.4.3. ESC PRAM FIFO data read register (ESC_PRAM_FIFO_DR) 2383
55.4.4. ESC PRAM address and length read register (ESC_PRAM_ALR) 2383
55.4.5. ESC PRAM command read register (ESC_PRAM_CR) 2384
55.4.6. ESC PRAM FIFO data write register (ESC_PRAM_FIFO_DW) 2385
55.4.7. ESC PRAM address and length write register (ESC_PRAM_ALW) 2385
55.4.8. ESC PRAM command write register (ESC_PRAM_CW) 2386
55.4.9. ESC Control register (ESC_CTL) 2387
55.4.10. ESC Sync0 register (ESC_SYNC0) 2388
55.4.11. ESC Sync1 register (ESC_SYNC1) 2389
55.5. ESC Register definition 2389
55.5.1. ESC Type register (ESC_TYPE) 2389
55.5.2. ESC Revision register (ESC_REVISION) 2390
55.5.3. ESC Build register (ESC_BUILD) 2390
55.5.4. ESC FMMU numbers register (ESC_FMMUS) 2390
55.5.5. ESC SyncManagers numbers register (ESC_SYNCMS) 2391
55.5.6. ESC RAM size register (ESC_RAMSIZE) 2391
55.5.7. ESC Port descriptor register (ESC_PORT_DESC) 2391
55.5.8. ESC Features supporter register (ESC_FEATURE) 2392
55.5.9. ESC Configured station address register (ESC_STATION_ADDR) 2394
55.5.10. ESC Configured station alias register (ESC_STATION_ALS) 2394
55.5.11. ESC Write enable register (ESC_REG_WEN) 2394
55.5.12. ESC Write protection register (ESC_REG_WP) 2395
55.5.13. ESC Write enable register (ESC_WEN) 2395
55.5.14. ESC Write protection register (ESC_WP) 2396
55.5.15. ESC Reset register (ESC_RST_ECAT) 2396
55.5.16. ESC Reset PDI register (ESC_RST_PDI) 2397
55.5.17. ESC DL control register (ESC_DL_CTL) 2398
55.5.18. ESC Physical read / write offset register (ESC_PHYSICAL_RW_OFFSET) 2400
55.5.19. ESC DL status register (ESC_DL_STAT) 2400
55.5.20. ESC AL control register (ESC_AL_CTL) 2402
55.5.21. ESC AL status register (ESC_AL_STAT) 2402
55.5.22. ESC AL status code register (ESC_AL_STAT_CODE) 2403
55.5.23. ESC RUN LED override register (ESC_RUN_LED_OVRD) 2403
55.5.24. ESC ERR LED override register (ESC_ERR_LED_OVRD) 2404
55.5.25. ESC PDI control register (ESC_PDI_CTL) 2405
55.5.26. ESC Configuration register (ESC_CFG) 2405

55.5.27. ESC PDI Information register (ESC_PDI_INFO) 2406
55.5.28. ESC PDI configuration register (ESC_PDI_CFG) 2407
55.5.29. ESC Sync / Latch configuration register (ESC_SL_CFG) 2407
55.5.30. ESC PDI extended configuration register (ESC_PDI_EXT_CFG) 2408
55.5.31. ESC Event mask register (ESC_EVT_MSK) 2410
55.5.32. ESC PDI AL event register (ESC_PDI_AL_EVT) 2410
55.5.33. ESC Event request register (ESC_EVT_REQ) 2411
55.5.34. ESC AL event request register (ESC_AL_EVT_REQ) 2412
55.5.35. RX error port X register (RX_PORTX_ERR) (X = 0,1,2,3) 2414
55.5.36. Forwarded RX error port X register (FWD_PORTX_ERR) (X = 0,1,2,3) 2415
55.5.37. ESC processing unit error register (ESC_PU_ERR) 2415
55.5.38. ESC PDI error counter register (ESC_PDI_ERR) 2416
55.5.39. ESC PORTx lost link register (ESC_PORTX_LOST_LINK) (x = 0,1,2,3) 2416
55.5.40. ESC Watchdog divider register (ESC_WDG_DIV) 2417
55.5.41. ESC Watchdog time PDI register (ESC_WDG_TIME_PDI) 2417
55.5.42. ESC Watchdog time process data register (ESC_WDG_TIME_PDATA) 2417
55.5.43. ESC Watchdog status process data register (ESC_WDG_STAT_PDATA) 2418
55.5.44. ESC Watchdog counter process data register (ESC_WDG_CNT_PDATA) 2418
55.5.45. ESC Watchdog counter PDI register (ESC_WDG_CNT_PDI) 2419
55.5.46. ESC EEPOM configuration register (ESC_EEPROM_CFG) 2419
55.5.47. ESC EEPOM PDI access register (ESC_EEPROM_PDI_ACC) 2419
55.5.48. ESC EEPOM control / status register (ESC_EEPROM_CS) 2420
55.5.49. ESC EEPROM address register (ESC_EEPROM_ADDR) 2422
55.5.50. ESC EEPROM data register (ESC_EEPROM_DATA) 2422
55.5.51. ESC MII management control / status register (ESC_MII_MNG_CS) 2423
55.5.52. ESC PHY address register (ESC_PHY_ADDR) 2424
55.5.53. ESC PHY register address register (ESC_PHY_REG_ADDR) 2425
55.5.54. ESC PHY data register (ESC_PHY_DATA) 2425
55.5.55. MII management ECAT access state register (MIIM_ECAT_ACC_STAT) 2426
55.5.56. MII management PDI access state register (MIIM_PDI_ACC_STAT) 2426
55.5.57. PHY Port X status register (PHY_PORTX_STAT) (X = 0,1,2,3) 2427
55.5.58. Logical start address FMMU X register (FMMUX_LOGIC_ADDR) (X = 0…F) 2427
55.5.59. Length FMMU X register (FMMUX_LENGTH) (X = 0…F) 2428
55.5.60. Start bit FMMU X in Logical address space register (FMMUX_STRA_BIT) (X = 0…F) .. 2428
55.5.61. Stop bit FMMU X in Logical address space register (FMMUX_STOP_BIT) (X = 0…F) ... 2429
55.5.62. Physical start address FMMU X register (FMMUX_ADDR) (X = 0…F) 2429
55.5.63. Physical start bit FMMU X register (FMMUX_PSBIT) (X = 0…F) 2429
55.5.64. Type FMMU X register (FMMUX_TYPE) (X = 0…F) 2430
55.5.65. Active FMMU X register (FMMUX_ACTIVE) (X = 0…F) 2430
55.5.66. Physical Start address SyncManager X register (SMX_START_ADDR) (X = 0…F) 2431
55.5.67. Length SyncManager X register (SMX_LENGTH) (X = 0…F) 2431
55.5.68. Control register SyncManager X register (SMX_CTL) (X = 0…F) 2431
55.5.69. Status register SyncManager X register (SMX_STAT) (X = 0…F) 2432
55.5.70. Activate SyncManager X register (SMX_ACTIVE) (X = 0…F) 2433

55.5.71. PDI Control SyncManager X register (SMX_PDI_CTL) (X = 0…F) 2434
55.5.72. ESC Receive time port 0 register (ESC_RCV_TIMEP0) 2435
55.5.73. ESC Receive time port 1 register (ESC_RCV_TIMEP1) 2435
55.5.74. ESC Receive time port 2 register (ESC_RCV_TIMEP2) 2436
55.5.75. ESC System time register (ESC_SYS_TIME) 2436
55.5.76. ESC Receive time ECAT processing unit register (ESC_RCVT_ECAT_PU) 2437
55.5.77. ESC System time offset register (ESC_SYS_TIME_OFFSET) 2437
55.5.78. ESC System time delay register (ESC_SYS_TIME_DELAY) 2437
55.5.79. ESC System time difference register (ESC_SYS_TIME_DIFF) 2438
55.5.80. ESC Speed counter start register (ESC_SPD_CNT_START) 2439
55.5.81. ESC Speed counter diff register (ESC_SPD_CNT_DIFF) 2439
55.5.82. ESC System time difference filter depth register (ESC_SYS_TIME_DIFF_FD) 2440
55.5.83. ESC Speed counter filter depth register (ESC_SPD_CNT_FD) 2440
55.5.84. ESC Cyclic unit control register (ESC_CYC_UNIT_CTL) 2440
55.5.85. ESC Register activation register (ESC_REG_ACT) 2441
55.5.86. ESC Pulse length of sync signals register (ESC_PULSE_LEN) 2442
55.5.87. ESC Activation status register (ESC_ACT_STAT) 2443
55.5.88. ESC SYNC0 status register (ESC_SYNC0_STAT) 2443
55.5.89. ESC SYNC1 status register (ESC_SYNC1_STAT) 2444
55.5.90. ESC Start time cyclic operation register (ESC_START_TIME_CO) 2444
55.5.91. ESC Next SYNC1 pulse register (ESC_NEXT_SYNC1_PULSE) 2445
55.5.92. ESC SYNC0 cycle time register (ESC_SYNC0_CYC_TIME) 2445
55.5.93. ESC SYNC1 cycle time register (ESC_SYNC1_CYC_TIME) 2446
55.5.94. ESC Latch0 control register (ESC_LATCH0_CTL) 2446
55.5.95. ESC Latch1 control register (ESC_LATCH1_CTL) 2447
55.5.96. ESC Latch0 status register (ESC_LATCH0_STAT) 2447
55.5.97. ESC Latch1 status register (ESC_LATCH1_STAT) 2448
55.5.98. ESC Latch0 time positive edge register (ESC_LATCH0_TIME_PE) 2448
55.5.99. ESC_ESC Latch0 time negative edge register (ESC_LATCH0_TIME_NE) 2449
55.5.100. ESC Latch1 time positive edge register (ESC_LATCH1_TIME_PE) 2450
55.5.101. ESC Latch1 time negative edge register (ESC_LATCH1_TIME_NE) 2450
55.5.102. ESC Buffer change event time register (ESC_BUF_CET) 2451
55.5.103. ESC PDI buffer start event time register (ESC_PDI_BUF_SET) 2451
55.5.104. ESC PDI buffer change event time register (ESC_PDI_BUF_CET) 2452
55.5.105. ESC Product ID register (ESC_PRODUCT_ID) 2452
55.5.106. ESC Vendor ID register (ESC_VENDOR_ID) 2452
55.5.107. ESC Digital I/O output data register (ESC_DIG_DATA) 2453
55.5.108. ESC General purpose outputs register (ESC_GP_OUTPUT) 2453
55.5.109. ESC General purpose inputs register (ESC_GP_INPUT) 2454
55.5.110. ESC User RAM register (ESC_USER_RAM) 2454
55.5.111. ESC Process data RAM register (ESC_PDRAM) 2454

56. Appendix 2456

56.1. List of abbreviations used in register 2456

56.2. List of terms 2456
56.3. Available peripherals 2457
57. Revision history 2458

List of Figures

Figure 1-1. The structure of the Cortex®-M7 processor 75
Figure 1-2. The system architecture of GD32H77x devices 77
Figure 1-3. The system architecture of GD32H78x devices 78
Figure 1-4. The system architecture of GD32H77R devices 79
Figure 1-5. Bus matrix Region 0 80
Figure 1-6. Bus matrix Region 1 81
Figure 1-7. Bus matrix Region 2 81
Figure 1-8. Block diagram of AXI SRAM controller 91
Figure 1-9. Block diagram of RAM shared by ITCM/DTCM/AXI SRAM 92
Figure 1-10. Block diagram of AXI interconnect matrix 96
Figure 2-1. HWSEM diagram 153
Figure 2-2. HWSEM state machine 154
Figure 2-3. First try lock success 156
Figure 2-4. Second try lock success 157
Figure 2-5. Second try lock fail 157
Figure 2-6. HWSEM interrupt logic 157
Figure 3-1. Block diagram of EXTI 170
Figure 4-1. Block diagram of DMA 180
Figure 4-2. Data stream for three transfer modes 181
Figure 4-3. Handshake mechanism 182
Figure 4-4. Data packing/unpacking when PWIDTH = '00' 187
Figure 4-5. Data packing/unpacking when PWIDTH = '01' 187
Figure 4-6. Data packing/unpacking when PWIDTH = '10' 188
Figure 4-7. DMA operation of switch-buffer mode 189
Figure 4-8. System connection of DMA0 and DMA1 196
Figure 5-1. Block diagram of MDMA 209
Figure 5-2. Connections of the four modes 210
Figure 5-3. Word, halfword, byte order exchange 212
Figure 5-4. Data padding and alignment (source greater than destination) 212
Figure 5-5. Data padding and alignment (source less than destination) 213
Figure 5-6. Data packing / unpacking 214
Figure 5-7. MDMA interrupt logic 220
Figure 6-1. Block diagram of DMAMUX 236
Figure 6-2. Synchronization mode 238
Figure 6-3. Event generation 239
Figure 7-1. NVMC block diagram 258
Figure 7-2. ECNVM read and write clock diagram of GD32H77x Devices 259
Figure 7-3. ECNVM read and write clock diagram of GD32H78x Devices 259
Figure 7-4. Process of sector erase operation 266
Figure 7-5. Process of mass erase operation 268

Figure 7-6. Process of typical chip erase operation 270
Figure 7-7. Process of protection-removed chip erase operation 272
Figure 7-8. Process of program operation 274
Figure 7-9. Process of page erase operation 279
Figure 7-10. Process of sector erase operation 281
Figure 7-11. Process of mass erase operation 283
Figure 7-12. Process of program operation 285
Figure 7-13. OTA upgrade operation. 291
Figure 7-14. Memory architecture in standard mode and secure mode 306
Figure 7-15. OTP interrupt mapping diagram 315
Figure 8-1. Block architecture of RAMECCMU 385
Figure 9-1. Power supply overview 393
Figure 9-2. Waveform of the Backup domain voltage thresholds 395
Figure 9-3. Waveform of the POR / PDR 396
Figure 9-4. Waveform of the BOR 397
Figure 9-5. Waveform of the LVD threshold 397
Figure 9-6. Waveform of the VAVD threshold 398
Figure 9-7. Temperature thresholds 399
Figure 9-8. SMPS supplies for V_{CORE} power domain 400
Figure 9-9. Bypass 400
Figure 9-10. Power on and power down timing requirements for V_{DD} and V_{CORE} in bypass mode 401
Figure 9-11. Waveform of the VOVD0 401
Figure 9-12. Waveform of the VOVD1 402
Figure 9-13. waveform of VUVD0 402
Figure 9-14. waveform of VUVD1 403
Figure 9-15. Connection diagram when USB regulator is enabled 403
Figure 9-16. Connection diagram when USB regulator is disabled 403
Figure 9-17. Schematic diagram of MIPI_LDO power supply 404
Figure 10-1. The system reset circuit 421
Figure 10-2. Clock tree of GD32H77x 422
Figure 10-3. Clock tree of GD32H78x 423
Figure 10-4. HXTAL clock source 427
Figure 10-5. HXTAL clock source in bypass mode 427
Figure 11-1. CTC overview 527
Figure 11-2. CTC trim counter 528
Figure 12-1. Schematic diagram of CPDM 538
Figure 12-2. CPDM delay line length configuration flowchart 539
Figure 12-3. CPDM output clock phase configuration flowchart 540
Figure 13-1. Basic structure of a standard I/O port bit 545
Figure 13-2. Input configuration 546
Figure 13-3. Output configuration 547
Figure 13-4. Analog configuration 548
Figure 13-5. Alternate function configuration 548

Figure 13-6. Analog configuration for ADC 550
Figure 13-7. Filtering using the sampling window 551
Figure 13-8. Input filtering clock cycle 553
Figure 14-1. TRIGSEL main composition example 572
Figure 15-1. Advanced timer block diagram 627
Figure 15-2. Normal mode, internal clock divided by 1 628
Figure 15-3. Counter timing diagram with prescaler division change from 1 to 2 629
Figure 15-4. Timing diagram of up counting mode, PSC=0/2 630
Figure 15-5. Timing diagram of up counting mode, change TIMERx_CAR ongoing 631
Figure 15-6. Timing diagram of down counting mode, PSC=0/2 632
Figure 15-7. Timing diagram of down counting mode, change TIMERx_CAR ongoing 632
Figure 15-8. Timing diagram of center-aligned counting mode 634
Figure 15-9. Repetition counter timing diagram of center-aligned counting mode 635
Figure 15-10. Repetition counter timing diagram of up counting mode 635
Figure 15-11. Repetition counter timing diagram of down counting mode 636
Figure 15-12. Input capture logic for channel 0 637
Figure 15-13. Input capture logic for multi mode channel 0 637
Figure 15-14. Output compare logic (when MCHxMSEL = 2'b00, x=0, 1, 2, 3) 639
Figure 15-15. Output compare logic (when MCHxMSEL = 2'b11, x=0,1,2,3) 639
Figure 15-16. Output-compare in three modes 641
Figure 15-17. Timing diagram of EAPWM 642
Figure 15-18. Timing diagram of CAPWM 642
Figure 15-19. Channel x output PWM with (CHxVAL < CHxCOMVAL_ADD) 644
Figure 15-20. Channel x output PWM with (CHxVAL = CHxCOMVAL_ADD) 644
Figure 15-21. Channel x output PWM with (CHxVAL > CHxCOMVAL_ADD) 645
Figure 15-22. Channel x output PWM with CHxVAL or CHxCOMVAL_ADD exceeds CARL 645
Figure 15-23. Channel x output PWM duty cycle changing with CHxCOMVAL_ADD 645
Figure 15-24. Four Channels outputs in Composite PWM mode 646
Figure 15-25. Channel x output PWM with(CHxVAL < CHxCOMVAL_ADD / CHxCOMVAL_ADD < CHxVAL) 648
Figure 15-26. Channel x output PWM with(CHxVAL = CHxCOMVAL_ADD) 650
Figure 15-27. Channel x output PWM with CHxVAL or CHxCOMVAL_ADD exceeds CARL 650
Figure 15-28. CHx_O output with a pulse in edge-aligned mode (CHxOMPSEL!=2'b00) 651
Figure 15-29. CHx_O output with a pulse in center-aligned mode (CHxOMPSEL!=2'b00) 652
Figure 15-30. Complementary output with dead time insertion 658
Figure 15-31. BREAK0 function logic diagram 659
Figure 15-32. BREAK1 function logic diagram 659
Figure 15-33. Output behavior of the channel in response to BREAK0 (the break input high active and IOS=1) 659
Figure 15-34. Output behavior of the channel outputs with the BREAK0 and BREAK1 661
Figure 15-35. Channel BREAK function logic diagram 662
Figure 15-36. BRKINx (x=0...2) pins logic with BREAK0 function 663

Figure 15-37. Example of counter operation in decoder interface mode 664
Figure 15-38. Example of decoder interface mode with CI0FE0 polarity inverted 665
Figure 15-39. Quadrature decoder signal disconnection detection block diagram 665
Figure 15-40. Example of counter operation in non-quadrature decoder mode 0 / 1 with CH1P=0 666
Figure 15-41. Example of counter operation in non-quadrature decoder mode 2 / 3 (CH0P / CH1P=0) 667
Figure 15-42. Hall sensor is used for BLDC motor 669
Figure 15-43. Hall sensor timing between two timers 670
Figure 15-44. Restart mode 671
Figure 15-45. Pause mode 671
Figure 15-46. Event mode 672
Figure 15-47. Single pulse mode TIMERx_CHxCV=0x04, TIMERx_CAR=0x60 673
Figure 15-48. delayable single pulse mode with TIMERx_CHxCV=0x00, TIMERx_CAR=0x60 674
Figure 15-49. Trigger mode of TIMER0 controlled by enable signal of TIMER2 675
Figure 15-50. Trigger mode of TIMER0 controlled by update signal of TIMER2 675
Figure 15-51. Pause mode of TIMER0 controlled by enable signal of TIMER2 676
Figure 15-52. Pause mode of TIMER0 controlled by O0CPRE signal of TIMER2 677
Figure 15-53. Trigger TIMER0 and TIMER2 by the CI0 signal of TIMER2 678
Figure 15-54. General Level 0 timer block diagram 757
Figure 15-55. Normal mode, internal clock divided by 1 758
Figure 15-56. Counter timing diagram with prescaler division change from 1 to 2 759
Figure 15-57. Timing chart of up counting mode, PSC=0/2 760
Figure 15-58. Timing chart of up counting, change TIMERx_CAR ongoing 760
Figure 15-59. Timing chart of down counting mode, PSC=0/2 761
Figure 15-60. Timing chart of down counting mode, change TIMERx_CAR ongoing 762
Figure 15-61. Timing chart of center-aligned counting mode 763
Figure 15-62. Input capture logic 764
Figure 15-63. Output compare logic (x=0,1,2,3) 765
Figure 15-64. Output-compare under three modes 766
Figure 15-65. Timing chart of EAPWM 767
Figure 15-66. Timing chart of CAPWM 767
Figure 15-67. Channel x output PWM with (CHxVAL < CHxCOMVAL_ADD) 769
Figure 15-68. Channel x output PWM with (CHxVAL = CHxCOMVAL_ADD) 769
Figure 15-69. Channel x output PWM with (CHxVAL > CHxCOMVAL_ADD) 770
Figure 15-70. Channel x output PWM with CHxVAL or CHxCOMVAL_ADD exceeds CARL 770
Figure 15-71. Channel x output PWM duty cycle changing with CHxCOMVAL_ADD 771
Figure 15-72. Four Channels outputs in Composite PWM mode 771
Figure 15-73. CHx_O output with a pulse in edge-aligned mode (CHxOMPSEL!=2'b00) 772
Figure 15-74. CHx_O output with a pulse in center-aligned mode (CHxOMPSEL!=2'b00) 773
Figure 15-75. Restart mode 775
Figure 15-76. Pause mode 776

Figure 15-77. Event mode 776
Figure 15-78. Single pulse mode TIMERx_CHxCV = 0x04, TIMERx_CAR=0x60 777
Figure 15-79. delayable single pulse mode TIMERx_CHxCV=0x00, TIMERx_CAR=0x60 .. 778
Figure 15-80. General level3 timer block diagram 814
Figure 15-81. Normal mode, internal clock divided by 1 815
Figure 15-82. Counter timing diagram with prescaler division change from 1 to 2 816
Figure 15-83. Timing diagram of up counting mode, PSC=0/2 817
Figure 15-84. Timing diagram of up counting mode, change TIMERx_CAR on the go 817
Figure 15-85. Timing diagram of down counting mode, PSC=0/2 818
Figure 15-86. Timing diagram of down counting mode, change TIMERx_CAR ongoing .. 819
Figure 15-87. Timing diagram of center-aligned counting mode 820
Figure 15-88. Repetition counter timing diagram of center-aligned counting mode 821
Figure 15-89. Repetition counter timing diagram of up counting mode 821
Figure 15-90. Repetition counter timing diagram of down counting mode 822
Figure 15-91. Input capture logic for channel 0 823
Figure 15-92. Input capture logic for multi mode channel 0 823
Figure 15-93. Output compare logic (when MCHxMSEL = 2'b00, x=0) 824
Figure 15-94. Output compare logic (when MCHxMSEL = 2'b11, x=0) 825
Figure 15-95. Output compare logic (x=1) 825
Figure 15-96. Output-compare in three modes 827
Figure 15-97. Timing diagram of EAPWM 828
Figure 15-98. Timing diagram of CAPWM 828
Figure 15-99. Channel x output PWM with (CHxVAL < CHxCOMVAL_ADD) 830
Figure 15-100. Channel x output PWM with (CHxVAL = CHxCOMVAL_ADD) 830
Figure 15-101. Channel x output PWM with (CHxVAL > CHxCOMVAL_ADD) 831
Figure 15-102. Channel x output PWM with CHxVAL or CHxCOMVAL_ADD exceeds CARL 831
Figure 15-103. Channel x output PWM duty cycle changing with CHxCOMVAL_ADD 831
Figure 15-104. CHx_O output with a pulse in edge-aligned mode (CHxOMPSEL!=2'b00) 832
Figure 15-105. CHx_O output with a pulse in center-aligned mode (CHxOMPSEL!=2'b00) 833
Figure 15-106. Complementary output with dead-time insertion 838
Figure 15-107. BREAK0 function logic diagram 839
Figure 15-108. Output behavior of the channel in response to BREAK0 (the break input high active and IOS=1) 839
Figure 15-109. BRKIN0 pin logic with BREAK0 function 841
Figure 15-110. Restart mode 842
Figure 15-111. Pause mode 842
Figure 15-112. Event mode 842
Figure 15-113. Single pulse mode TIMERx_CHxCV = 0x04 TIMERx_CAR=0x60 843
Figure 15-114. delayable single pulse mode TIMERx_CHxCV=0x00, TIMERx_CAR=0x60 844
Figure 15-115. General level4 timer block diagram 882
Figure 15-116. Normal mode, internal clock divided by 1 883
Figure 15-117. Counter timing diagram with prescaler division change from 1 to 2 883

Figure 15-118. Timing diagram of up counting mode, PSC=0/2 884
Figure 15-119. Timing diagram of up counting mode, change TIMERx_CAR on the go .. 885
Figure 15-120. Repetition timechart for up-counter 886
Figure 15-121. Input capture logic for channel 0 887
Figure 15-122. Input capture logic for multi mode channel 0 887
Figure 15-123. Output compare logic (when MCHxMSEL = 2'b00, x=0) 888
Figure 15-124. Output compare logic (when MCHxMSEL = 2'b11, x=0) 889
Figure 15-125. Output-compare in three modes 890
Figure 15-126. PWM mode timechart 891
Figure 15-127. Complementary output with dead-time insertion 894
Figure 15-128. BREAK0 function logic diagram 895
Figure 15-129. Output behavior of the channel in response to BREAK0 (the break input high active and IOS=1) 895
Figure 15-130. BRKIN0 pin logic with BREAK0 function 897
Figure 15-131. Single pulse mode TIMERx_CHxCV = 0x04 TIMERx_CAR=0x60 898
Figure 15-132. Basic timer block diagram 926
Figure 15-133. Normal mode, internal clock divided by 1 927
Figure 15-134. Counter timing diagram with prescaler division change from 1 to 2 927
Figure 15-135. Timing chart of up counting mode, PSC=0/2 (TIMERx, x=5,6) 928
Figure 15-136. Timing chart of up counting mode, change TIMERx_CAR ongoing (TIMERx, x=5,6) 928
Figure 16-1. Block diagram of RTC 939
Figure 17-1. Free watchdog block diagram 973
Figure 17-2. Window watchdog timer block diagram 979
Figure 17-3. Window watchdog timing diagram 980
Figure 18-1. Block diagram of JTAG unit 985
Figure 19-1. Block diagram of CRC calculation unit 996
Figure 20-1. TRNG block diagram 1001
Figure 21-1. DATAM No swapping and Half-word swapping 1013
Figure 21-2. DATAM Byte swapping and Bit swapping 1014
Figure 21-3. CAU diagram 1015
Figure 21-4. AES ECB encryption 1016
Figure 21-5. AES ECB decryption 1017
Figure 21-6. AES CBC encryption 1018
Figure 21-7. AES CBC decryption 1019
Figure 21-8. Counter block structure 1019
Figure 21-9. AES CTR encryption/decryption 1020
Figure 22-1. DATAM No swapping and Half-word swapping 1042
Figure 22-2. DATAM Byte swapping and Bit swapping 1042
Figure 22-3. HAU block diagram 1043
Figure 23-1. PKCAU module block diagram 1057
Figure 23-2. Flow chart of RSA algorithm 1058
Figure 23-3. Flow chart of ECDSA sign 1059
Figure 23-4. Flow chart of ECDSA verification 1060

Figure 23-5. Arithmetic addition 1061
Figure 23-6. Arithmetic subtraction 1062
Figure 23-7. Arithmetic multiplication 1062
Figure 23-8. Arithmetic comparison 1063
Figure 23-9. Modular reduction 1063
Figure 23-10. Modular addition 1064
Figure 23-11. Modular subtraction 1065
Figure 23-12. Montgomery parameter calculation 1065
Figure 23-13. Mutual mapping between Montgomery domain and natural domain 1066
Figure 23-14. Montgomery multiplication 1067
Figure 23-15. Modular exponentiation of normal mode 1067
Figure 23-16. Modular exponentiation of fast mode 1068
Figure 23-17. Modular inversion 1068
Figure 23-18. RSA CRT exponentiation 1069
Figure 23-19. Point on elliptic curve Fp check 1071
Figure 23-20. ECC scalar multiplication of normal mode 1072
Figure 23-21. ECC scalar multiplication of fast mode 1072
Figure 23-22. ECDSA sign 1074
Figure 23-23. ECDSA verification 1075
Figure 24-1. RTDEC block diagram 1085
Figure 24-2. Decryption flow with AES-128 in CTR mode 1087
Figure 24-3. RTDEC interrupt mapping diagram 1090
Figure 25-1. ADC module block diagram 1101
Figure 25-2. Single operation mode 1104
Figure 25-3. Continuous operation mode 1105
Figure 25-4. Scan operation mode, continuous disable 1106
Figure 25-5. Scan operation mode, continuous enable 1106
Figure 25-6. Discontinuous operation mode 1107
Figure 25-7. Auto-insertion, CTN = 1 1108
Figure 25-8. Triggered insertion 1108
Figure 25-9. Data latch 1109
Figure 25-10. Sequence data from ADC_LDATAx 1109
Figure 25-11. Data storage mode of 14-bit resolution 1111
Figure 25-12. Data storage mode of 12-bit resolution 1111
Figure 25-13. Data storage mode of 10-bit resolution 1112
Figure 25-14. Data storage mode of 8-bit resolution 1112
Figure 25-15. 6-bit data alignment 1112
Figure 25-16. Schematic diagram of handshake signal between HPDF and ADC module 1116
Figure 25-17. 20-bit to 16-bit result truncation (for 12bit ADC) 1117
Figure 25-18. Numerical example with 5-bits shift and rounding (for 12bit ADC) 1118
Figure 25-19. 14bit ADC oversampling with 10bits right shift 1118
Figure 25-20. Numerical example for 14bit ADC oversampling with 10bits right shift 1118
Figure 25-21. ADC sync block diagram 1120

Figure 25-22. Routine parallel mode on 16 channels 1121
Figure 25-23. Inserted parallel mode on 4 channels 1121
Figure 25-24. Routine follow-up mode on 1 channel in continuous operation mode 1122
Figure 25-25. trigger rotation: DISIC=0, IL=1 1123
Figure 25-26. trigger rotation: DISIC=1, IL=1 1123
Figure 25-27. Routine parallel & trigger rotation mode: SYNCM = 0010 1124
Figure 25-28. Trigger occurs during inserted conversion: SYNCM = 0010 1124
Figure 26-1. DAC block diagram 1155
Figure 26-2. DAC LFSR algorithm 1157
Figure 26-3. DAC triangle noise wave 1158
Figure 26-4. DAC sample and keep 1161
Figure 27-1. CMP block diagram 1178
Figure 27-2. CMP hysteresis 1180
Figure 27-3. The CMP outputs signal blanking 1181
Figure 27-4. The CMP output signal filtering 1182
Figure 28-1.Precision reference connection 1191
Figure 29-1. LPDTS block diagram 1195
Figure 29-2. Method for different REF_CLK 1197
Figure 30-1. TLI module block diagram 1209
Figure 30-2. Display timing diagram 1209
Figure 30-3. Block diagram of Blending 1212
Figure 31-1. Functional block diagram of DSI Host 1231
Figure 31-2. DPI Timing conveyed over DSI Link 1234
Figure 31-3. Non Burst Transfer with Start and End 1235
Figure 31-4. Non Burst with Sync Event 1235
Figure 31-5. DSI Packet flow for Burst communication 1236
Figure 31-6. Transmission of RGB666 stream 1238
Figure 31-7. Transmission of RGB565 stream 1239
Figure 31-8. Transmission of RGB888/RGB666 loosely packed stream 1240
Figure 31-9. Reset Block Diagram 1241
Figure 31-10. Overview of all functions 1242
Figure 31-11. MIPI Link diagram for four data lanes 1243
Figure 31-12. Block diagram showing Analog and Digital portion of D-PHY TX 1244
Figure 31-13. PLL Block diagram 1247
Figure 31-14. DPHY Data to Clock Timing Definitions 1248
Figure 31-15. DPI transfer flow 1250
Figure 31-16. DCS/Generic transfer flow 1254
Figure 31-17. DCS/Generic LP Write 1254
Figure 31-18. DCS/Generic read 1255
Figure 31-19. Diagram of TLI DMA and DSI controller 1257
Figure 32-1. FFT module block diagram 1302
Figure 32-2 8 point DIT FFT flow diagram 1304
Figure 32-3. 8 points DIT FFT flow diagram 1305
Figure 33-1. HPDF block diagram 1314

Figure 33-2. The sequence diagram of SPI data transmission 1318
Figure 33-3. The sequence diagram of Manchester data transmission 1319
Figure 33-4. Manchester synchronous sequence diagram 1319
Figure 33-5. Clock loss detection timing diagram 1322
Figure 33-6. Channel pins redirection 1323
Figure 33-7. HPDF module external input data processing flow 1335
Figure 33-8. HPDF interrupt logic diagram 1337
Figure 34-1. TMU block diagram 1362
Figure 35-1 FAC structure diagram 1385
Figure 35-2 Input buffer area 1386
Figure 35-3 Circular input buffer area 1387
Figure 35-4 Circular input buffer operation 1388
Figure 35-5 Circular output buffer 1388
Figure 35-6 Circular output buffer area 1389
Figure 35-7 The structure of FIR filter function 1391
Figure 35-8 The structure of IIR filter 1392
Figure 36-1. IPA block diagram 1405
Figure 36-2. Pixel extension from 'RGB888' to 'ARGB8888' 1410
Figure 36-3. Pixel extension from 'RGB565' to 'ARGB8888' 1410
Figure 36-4. Pixel extension from 'ARGB1555' or 'ARGB4444' to 'ARGB8888' 1410
Figure 36-5. Sampling method of decimation filter 1412
Figure 36-6. Bilinear Scaling diagram 1413
Figure 36-7. Pixel compression 1415
Figure 36-8. Inter timer operation 1416
Figure 36-9. System connection of IPA 1423
Figure 37-1. Block diagram of EDOUT 1447
Figure 37-2. ABZ-phase output waveforms 1448
Figure 37-3. Example of the settings of EDOUT and the AB-Phase and Z-Phase output waveforms 1451
Figure 38-1. AFMT module block diagram 1457
Figure 38-2. Timing of individual transmission mode 1459
Figure 38-3. Timing of broadcast transmission mode 1460
Figure 38-4. Command transmission process (software trigger) 1463
Figure 38-5. Command transmission process (external hardware trigger) 1464
Figure 38-6. TFMT module block diagram 1473
Figure 38-7. ENDAT module block diagram 1487
Figure 38-8. ENDAT module communication process 1492
Figure 38-9. BISS module block diagram 1511
Figure 38-10. Point-to-point connection 1512
Figure 38-11. Simple BiSS frame 1513
Figure 38-12. Typical reset sequence 1514
Figure 38-13. Basic operation for process data 1515
Figure 38-14. Broadcast BiSS command communication 1518
Figure 38-15. Addressed BiSS command communication 1518

Figure 38-16. Register communication for reading 1519
Figure 38-17. Register communication for writing 1519
Figure 38-18. HDSL master block diagram 1536
Figure 38-19. HDSL master system start procedure 1538
Figure 38-20. Interrupt condition 1540
Figure 38-21. Interrupt condition for safety 1541
Figure 39-1. RDCM block diagram 1586
Figure 39-2. PWM waveform generator block diagram 1588
Figure 39-3. I/Q synthetic reference generation block diagram 1590
Figure 39-4. Phase sensitive demodulator block diagram 1592
Figure 39-5. Compensation filter block diagram 1595
Figure 39-6. Type II tracking loop block diagram 1597
Figure 39-7. FIFO structure 1599
Figure 40-1. USART module block diagram 1636
Figure 40-2. USART character frame (8 bits data and 1 stop bit) 1637
Figure 40-3. USART transmit procedure 1639
Figure 40-4. Oversampling method of a receive frame bit (OSB = 0) 1640
Figure 40-5. Configuration step when using DMA for USART transmission 1641
Figure 40-6. Configuration step when using DMA for USART reception 1642
Figure 40-7. Hardware flow control between two USARTs 1643
Figure 40-8. Hardware flow control 1643
Figure 40-9. Break frame occurs during idle state 1645
Figure 40-10. Break frame occurs during a frame 1645
Figure 40-11. Example of USART in synchronous mode 1646
Figure 40-12. 8-bit format USART synchronous waveform (CLEN = 1) 1646
Figure 40-13. IrDA SIR ENDEC module 1647
Figure 40-14. IrDA data modulation 1647
Figure 40-15. ISO7816-3 frame format 1648
Figure 40-16. USART receive FIFO structure 1651
Figure 40-17. USART transmitter FIFO structure 1652
Figure 40-18. USART interrupt mapping diagram 1654
Figure 41-1. Block diagram of SPI 1680
Figure 41-2. SPI timing diagram in normal mode 1682
Figure 41-3. SPI3 / 4 timing diagram in Quad-SPI mode (CKPL = 1, CKPH = 1, LF = 0) 1683
Figure 41-4. SPI data frame right-aligned diagram 1683
Figure 41-5. SPI data and clock transmission path diagram 1684
Figure 41-6. SPI master rx delay configuration diagram 1684
Figure 41-7. SPI slave rx delay configuration diagram 1685
Figure 41-8. NSS signal delay timing diagram (MSSD[3:0] = 0011 (3 × Tclk), MDFD = 0011 (3 x Tclk)) 1689
Figure 41-9. NSS interlaced pulses timing diagram (MSSD[3:0] = 0011 (3 x Tclk), MDFD = 0011 (3 x Tclk)) 1690
Figure 41-10. A typical full-duplex connection 1691
Figure 41-11. A typical simplex connection (Master: Receive, Slave: Transmit) 1692

Figure 41-12. A typical simplex connection (Master: Transmit only, Slave: Receive) 1692
Figure 41-13. A typical bidirectional connection 1692
Figure 41-14. Timing diagram of TI master mode with discontinuous transfer 1695
Figure 41-15. Timing diagram of TI master mode with continuous transfer 1695
Figure 41-16. Timing diagram of TI slave mode 1696
Figure 41-17. Timing diagram of quad write operation in Quad-SPI mode 1697
Figure 41-18. Timing diagram of quad read operation in Quad-SPI mode 1698
Figure 41-19. Block diagram of I2S 1704
Figure 41-20. I2S Philips standard timing diagram (DTLEN = 00, CHLEN = 0, CKPL = 0) 1705
Figure 41-21. I2S Philips standard timing diagram (DTLEN = 00, CHLEN = 0, CKPL = 1) 1705
Figure 41-22. I2S Philips standard timing diagram (DTLEN = 10, CHLEN = 1, CKPL = 0) 1706
Figure 41-23. I2S Philips standard timing diagram (DTLEN = 10, CHLEN = 1, CKPL = 1) 1706
Figure 41-24. I2S Philips standard timing diagram (DTLEN = 01, CHLEN = 1, CKPL = 0) 1706
Figure 41-25. I2S Philips standard timing diagram (DTLEN = 01, CHLEN = 1, CKPL = 1) 1706
Figure 41-26. I2S Philips standard timing diagram (DTLEN = 00, CHLEN = 1, CKPL = 0) 1706
Figure 41-27. I2S Philips standard timing diagram (DTLEN = 00, CHLEN = 1, CKPL = 1) 1706
Figure 41-28. MSB justified standard timing diagram (DTLEN = 00, CHLEN = 0, CKPL = 0) 1707
Figure 41-29. MSB justified standard timing diagram (DTLEN = 00, CHLEN = 0, CKPL = 1) 1707
Figure 41-30. MSB justified standard timing diagram (DTLEN = 10, CHLEN = 1, CKPL = 0) 1707
Figure 41-31. MSB justified standard timing diagram (DTLEN = 10, CHLEN = 1, CKPL = 1) 1707
Figure 41-32. MSB justified standard timing diagram (DTLEN = 01, CHLEN = 1, CKPL = 0) 1707
Figure 41-33. MSB justified standard timing diagram (DTLEN = 01, CHLEN = 1, CKPL = 1) 1707
Figure 41-34. MSB justified standard timing diagram (DTLEN = 00, CHLEN = 1, CKPL = 0) 1708
Figure 41-35. MSB justified standard timing diagram (DTLEN = 00, CHLEN = 1, CKPL = 1) 1708
Figure 41-36. LSB justified standard timing diagram (DTLEN = 01, CHLEN = 1, CKPL = 0) 1708
Figure 41-37. LSB justified standard timing diagram (DTLEN = 01, CHLEN = 1, CKPL = 1) 1708

Figure 41-38. LSB justified standard timing diagram (DTLEN = 00, CHLEN = 1, CKPL = 0) 1709
Figure 41-39. LSB justified standard timing diagram (DTLEN = 00, CHLEN = 1, CKPL = 1) 1709
Figure 41-40. PCM standard short frame synchronization mode timing diagram (DTLEN = 00, CHLEN = 0, CKPL = 0) 1709
Figure 41-41. PCM standard short frame synchronization mode timing diagram (DTLEN = 00, CHLEN = 0, CKPL = 1) 1709
Figure 41-42. PCM standard short frame synchronization mode timing diagram (DTLEN = 10, CHLEN = 1, CKPL = 0) 1710
Figure 41-43. PCM standard short frame synchronization mode timing diagram (DTLEN = 10, CHLEN = 1, CKPL = 1) 1710
Figure 41-44. PCM standard short frame synchronization mode timing diagram (DTLEN = 01, CHLEN = 1, CKPL = 0) 1710
Figure 41-45. PCM standard short frame synchronization mode timing diagram (DTLEN = 01, CHLEN = 1, CKPL = 1) 1710
Figure 41-46. PCM standard short frame synchronization mode timing diagram (DTLEN = 00, CHLEN = 1, CKPL = 0) 1710
Figure 41-47. PCM standard short frame synchronization mode timing diagram (DTLEN = 00, CHLEN = 1, CKPL = 1) 1711
Figure 41-48. PCM standard long frame synchronization mode timing diagram (DTLEN = 00, CHLEN = 0, CKPL = 0) 1711
Figure 41-49. PCM standard long frame synchronization mode timing diagram (DTLEN = 00, CHLEN = 0, CKPL = 1) 1711
Figure 41-50. PCM standard long frame synchronization mode timing diagram (DTLEN = 10, CHLEN = 1, CKPL = 0) 1711
Figure 41-51. PCM standard long frame synchronization mode timing diagram (DTLEN = 10, CHLEN = 1, CKPL = 1) 1711
Figure 41-52. PCM standard long frame synchronization mode timing diagram (DTLEN = 01, CHLEN = 1, CKPL = 0) 1711
Figure 41-53. PCM standard long frame synchronization mode timing diagram (DTLEN = 01, CHLEN = 1, CKPL = 1) 1712
Figure 41-54. PCM standard long frame synchronization mode timing diagram (DTLEN = 00, CHLEN = 1, CKPL = 0) 1712
Figure 41-55. PCM standard long frame synchronization mode timing diagram (DTLEN = 00, CHLEN = 1, CKPL = 1) 1712
Figure 41-56. Block diagram of I2S clock generator 1712
Figure 42-1. I2C module block diagram 1739
Figure 42-2. Data validation 1740
Figure 42-3. START and STOP condition 1741
Figure 42-4. I2C communication flow with 10-bit address (Master Transmit) 1741
Figure 42-5. I2C communication flow with 7-bit address (Master Transmit) 1742
Figure 42-6. I2C communication flow with 7-bit address (Master Receive) 1742
Figure 42-7. I2C communication flow with 10-bit address (Master Receive when HEAD10R=0)

..... 1742
Figure 42-8. I2C communication flow with 10-bit address (Master Receive when HEAD10R=1) 1742
Figure 42-9. Data hold time 1743
Figure 42-10. Data setup time 1744
Figure 42-11. Data transmission 1746
Figure 42-12. Data reception 1746
Figure 42-13. I2C initialization in slave mode 1749
Figure 42-14. Programming model for slave transmitting when SS=0 1750
Figure 42-15. Programming model for slave transmitting when SS=1 1751
Figure 42-16. Programming model for slave receiving 1752
Figure 42-17. I2C initialization in master mode 1753
Figure 42-18. Programming model for master transmitting (N<=255) 1754
Figure 42-19. Programming model for master transmitting (N>255) 1755
Figure 42-20. Programming model for master receiving (N<=255) 1756
Figure 42-21. Programming model for master receiving (N>255) 1757
Figure 42-22. SMBus master transmitter and slave receiver communication flow 1760
Figure 42-23. SMBus master receiver and slave transmitter communication flow 1761
Figure 43-1. CAN module block diagram 1780
Figure 43-2. Transmitter delay 1811
Figure 43-3. CAN bit time 1814
Figure 45-1 OSPI octal communication mode block diagram 1855
Figure 45-2 OSPI quad communication mode block diagram 1856
Figure 45-3 OSPI dual-quad communication mode block diagram 1856
Figure 45-4 OSPI command format in octal mode 1857
Figure 45-5 OSPI command format in quad mode 1857
Figure 45-6 DTR read timing using DQS in octal mode 1859
Figure 45-7 CSN and SCK behavior 1860
Figure 45-8 HyperBus™ read operation with initial latency 1862
Figure 45-9 HyperBus™ read operation with additional latency 1862
Figure 45-10 Read operation page crossing with latency 1863
Figure 45-11 HyperBus™ write operation with initial latency 1863
Figure 45-12 HyperBus™ write operation with additional latency 1864
Figure 45-13 HyperBus™ write operation with no latency 1864
Figure 46-1. SDIO "no response" and "no data" operations 1898
Figure 46-2. SDIO multiple blocks read operation 1898
Figure 46-3. SDIO multiple blocks write operation 1898
Figure 46-4. SDIO sequential read operation 1899
Figure 46-5. SDIO sequential write operation 1899
Figure 46-6. SDIO block diagram 1900
Figure 46-7. CRU unit 1907
Figure 46-8. Hardware flow timing 1911
Figure 46-9. Command Token Format 1913
Figure 46-10. Response Token Format 1925

Figure 46-11. 1-bit data bus width 1929
Figure 46-12. 4-bit data bus width 1929
Figure 46-13. 8-bit data bus width 1929
Figure 46-14. 4-bit data bus DDR 1930
Figure 46-15. 8-bit data bus DDR 1930
Figure 46-16. Boot operation timing 1940
Figure 46-17. Alternative Boot operation timing 1941
Figure 46-18. CMD5 timing 1951
Figure 46-19. CMD12 stream timing 1953
Figure 46-20. Read wait control by stopping SDIO_CLK 1954
Figure 46-21. Read wait operation using SDIO_DAT[2] 1954
Figure 46-22. Function2 read cycle inserted during function1 multiple read cycle 1955
Figure 46-23. Read Interrupt cycle timing 1956
Figure 46-24. Write interrupt cycle timing 1956
Figure 46-25. Multiple block 4-Bit read interrupt cycle timing 1957
Figure 46-26. Multiple block 4-Bit write interrupt cycle timing 1957
Figure 46-27. CMD11 voltage switch sequence 1957
Figure 47-1. CFP Management Interface Architecture 1980
Figure 47-2. MDIO Architecture 1980
Figure 47-3. MDIO Frame Structure 1981
Figure 47-4. MDIO slave communication process 1983
Figure 48-1. The EXMC block diagram 1992
Figure 48-2. EXMC memory banks 1994
Figure 48-3. Four regions of bank0 address mapping 1995
Figure 48-4. NAND address mapping 1996
Figure 48-5. Diagram of bank2 common space 1997
Figure 48-6. SDRAM address mapping 1998
Figure 48-7. Mode 1 read access 2003
Figure 48-8. Mode 1 write access 2003
Figure 48-9. Mode A read access 2004
Figure 48-10. Mode A write access 2005
Figure 48-11. Mode 2/B read access 2006
Figure 48-12. Mode 2 write access 2007
Figure 48-13. Mode B write access 2007
Figure 48-14. Mode C read access 2009
Figure 48-15. Mode C write access 2009
Figure 48-16. Mode D read access 2011
Figure 48-17. Mode D write access 2011
Figure 48-18. Multiplex mode read access 2013
Figure 48-19. Multiplex mode write access 2013
Figure 48-20. Read access timing diagram under async-wait signal assertion 2015
Figure 48-21. Write access timing diagram under async-wait signal assertion 2015
Figure 48-22. Read timing of synchronous multiplexed burst mode 2017
Figure 48-23. Write timing of synchronous multiplexed burst mode 2018

Figure 48-24. Access timing of common memory space of NAND flash controller 2021
Figure 48-25. Access to none "NCE don't care" NAND Flash 2022
Figure 48-26. SDRAM controller block diagram 2025
Figure 48-27. Burst read operation 2029
Figure 48-28. Data sampling clock delay chain 2029
Figure 48-29. Burst write operation 2030
Figure 48-30. Read access when FIFO not hit (BRSTRD=1, CL=2, SDCLK=2, PIPED=2) 2031
Figure 48-31. Read access when FIFO hit (BRSTRD=1) 2031
Figure 48-32. Cross boundary read operation 2032
Figure 48-33. Cross boundary write operation 2032
Figure 48-34. Process for self-refresh entry and exit 2033
Figure 48-35. Process for power-down entry and exit 2034
Figure 49-1. DCI module block diagram 2053
Figure 49-2. Hardware synchronization mode 2054
Figure 49-3. Hardware synchronization mode: JPEG format supporting 2055
Figure 50-1. The GPSI block diagram 2071
Figure 50-2. Data sampling in receive mode waveform diagram(CKPL = 0, DPL/RPL = 0) 2073
Figure 50-3. Data sampling in receive mode waveform diagram(CKPL = 1, DPL/RPL = 0) 2073
Figure 50-4. Data output in transmit mode waveform diagram(CKPL = 0, DPL/RPL = 0) 2074
Figure 50-5. Data output in transmit mode waveform diagram(CKPL = 1, DPL/RPL = 0) 2074
Figure 50-6. Ready signal in receive mode waveform diagram(CKPL = 0, DPL/RPL = 0) 2075
Figure 50-7. Ready signal in transmit mode waveform diagram(CKPL = 0, DPL/RPL = 0) 2075
Figure 50-8. Data transmitted on the falling edge of the second clock pulse 2075
Figure 50-9. Data transmitted on the falling edge of the first clock pulse 2076
Figure 50-10. Bidirectional GPSI_DE/GPSI_RDY waveform diagram(CKPL = 0, DPL/RPL = 0, transmitter drive the GPSI_DE to low) 2076
Figure 50-11. Bidirectional GPSI_DE/GPSI_RDY waveform diagram(CKPL = 0, DPL/RPL = 0, receiver drive the GPSI_RDY to low) 2076
Figure 50-12. Bidirectional GPSI_DE/GPSI_RDY connection diagram 2077
Figure 51-1. RSPDIF block diagram 2086
Figure 51-2. S/PDIF block and S/PDIF sub-frame format 2088
Figure 51-3. Information bits coding example 2088
Figure 51-4. S/PDIF Preambles 2089
Figure 51-5. Noise filtering and rising/falling edge detection 2090
Figure 51-6. Waveforms of TWCNT 2091
Figure 51-7. Synchronization flowchart flow 2093
Figure 51-8. Synchronization process scheduling 2094
Figure 51-9. RSPDIF states 2095
Figure 51-10. RSPDIF_DATA register format 2099
Figure 51-11. Channel/user data format 2100

Figure 51-12. RSPDIF overrun error when RXSTEOMEN = 0 and RXDF[1:0] = 2'b0x 2104
Figure 51-13. RSPDIF overrun error when RXSTEOMEN = 0 and RXDF[1:0] = 2'b10 2105
Figure 51-14. RSPDIF overrun error when RXSTEOMEN = 1 and RXDF[1:0] = 2'b0x 2105
Figure 51-15. RSPDIF overrun error when RXSTEOMEN = 1 and RXDF[1:0] = 2'b10 2106
Figure 52-1. block diagram 2121
Figure 52-2. Clock divider logic 2122
Figure 52-3 FS active width 2125
Figure 52-4 FS polarity 2126
Figure 52-5 FS function 2126
Figure 52-6 Slot activation 2127
Figure 52-7 Slot distribution when FSFUNC = 0 2127
Figure 52-8 Slot distribution when FSFUNC = 1 2128
Figure 52-9 Slot partition convention 2128
Figure 52-10 Offset region treatment 2128
Figure 52-11 SD output management 2129
Figure 52-12 Data configuration 2129
Figure 52-13. PDM typical connection and timing 2130
Figure 52-14. PDM data processing diagram 2132
Figure 52-15. The start-up process of PDM data transmission 2132
Figure 52-16. Eight microphones, under different slot widths, BxDATA register data format 2133
Figure 52-17. Four microphones, under different slot widths, BxDATA register data format 2135
Figure 52-18. Dual microphones, under different slot widths, BxDATA register data format 2135
Figure 52-19 AC'97 slot partition 2137
Figure 52-20 AC'97 tag definition 2138
Figure 52-21 SPDIF data format 2139
Figure 52-22 Mute frame activation 2141
Figure 52-23 Compander data path 2142
Figure 52-24 Frame synchronization advanced detection 2147
Figure 52-25 Frame synchronization postponed detection 2147
Figure 53-1. USBHS block diagram 2167
Figure 53-2. Connection using internal embedded PHY with host or device mode 2169
Figure 53-3. Connection using internal embedded PHY with OTG mode 2170
Figure 53-4. Connection using external ULPI PHY 2171
Figure 53-5. State transition diagram of host port 2171
Figure 53-6. Host mode FIFO space in SRAM 2177
Figure 53-7. Host mode FIFO access register map 2178
Figure 53-8. Device mode FIFO space in SRAM 2179
Figure 53-9. Device mode FIFO access register map 2179
Figure 54-1. ENET module block diagram 2264
Figure 54-2. MAC / Tagged MAC frame format 2266
Figure 54-3. Station management interface signals 2268

Figure 54-4. Media independent interface signals 2270
Figure 54-5. Reduced media-independent interface signals 2272
Figure 54-6. Descriptor ring and chain structure 2284
Figure 54-7. Transmit descriptor in normal mode 2290
Figure 54-8. Transmit descriptor in enhanced mode 2295
Figure 54-9. Receive descriptor in normal mode 2299
Figure 54-10. Receive descriptor in enhanced mode 2305
Figure 54-11. Wakeup frame filter register 2311
Figure 54-12. System time update using the fine correction method 2313
Figure 54-13. MAC interrupt scheme 2318
Figure 54-14. Ethernet interrupt scheme 2319
Figure 54-15. Wakeup frame filter register 2329
Figure 55-1. EtherCAT system block diagram 2372
Figure 55-2. EtherCAT State Machine 2376
Figure 55-3. PDI Interrupt Masking and interrupt signals 2377
Figure 55-4. EtherCAT Interrupt Masking 2378
Figure 55-5. Station management interface signals 2379

List of Tables

Table 1-1. The interconnection relationship of the interconnect matrix 76
Table 1-2. Memory map of GD32H77x_78x devices 82
Table 1-3. Configuration of ITCM/DTCM/AXI SRAM 92
Table 1-4. Boot mode selection 93
Table 1-5. Details of Boot mode 93
Table 1-6. Configuration of ASIBs 96
Table 1-7. Configuration of AMIBs 97
Table 2-1. interrupt events 157
Table 3-1. NVIC exception types in Cortex®-M7 164
Table 3-2. Interrupt vector table 164
Table 3-3. EXTI source 171
Table 4-1. Transfer mode 181
Table 4-2. CNT configuration 184
Table 4-3. FIFO counter critical value configuration rules 185
Table 4-4. DMA interrupt events 193
Table 5-1. Transfer mode 209
Table 5-2. MDMA hardware request sources 210
Table 5-3. Source and destination address generation configuration 215
Table 5-4. Update mode of source and destination address 217
Table 5-5. Register link address 217
Table 5-6. MDMA error flags 218
Table 5-7. MDMA interrupt events 219
Table 6-1. Interrupt events 241
Table 6-2. Request multiplexer input mapping 242
Table 6-3. Trigger input mapping 247
Table 6-4. Synchronization input mapping 249
Table 7-1. GD32H77xxW/GD32H78xxW base address and size for 7.5MB ECNVM memory and 2MB CNVM 259
Table 7-2. GD32H77xxP/GD32H78xxP base address and size for 3.5MB ECNVM memory and 2MB CNVM 260
Table 7-3. GD32H77xxI/GD32H78xxI base address and size for 0MB ECNVM memory and 1.625MB CNVM 261
Table 7-4. The typical value configuration for WSCNT of GD32H77x Devices. 264
Table 7-5. The typical value configuration for WSCNT of GD32H78x Devices. 264
Table 7-6. CNVM interrupt requests 277
Table 7-7. ECNVM interrupt requests 287
Table 7-8. Option byte 294
Table 7-9. WP bit for sectors protected 299
Table 7-10. SPC protection level configuration 302
Table 7-11. DCRP area configuration 303

Table 7-12. Secure user area configuration 305
Table 7-13. System parameters 308
Table 7-14. DP CRC source code 314
Table 7-15. OTP interrupt requests 315
Table 8-1. RAMECC monitor x unit for Region 0 (x=0..4) 386
Table 8-2. RAMECC monitor x unit for Region 1 (x=0..2) 386
Table 9-1. supply mode 400
Table 9-2. Power saving mode summary 406
Table 10-1. Clock output 0 source select 432
Table 10-2. Clock output 1 source select 432
Table 13-1. GPIO configuration table 544
Table 14-1. Trigger input bit fields selection 572
Table 14-2. TRIGSEL input and output mapping 577
Table 15-1. Timers (TIMERx) are divided into five sorts 624
Table 15-2. Advanced timer channel description 627
Table 15-3.The Composite PWM pulse width 643
Table 15-4.The Asymmetric PWM pulse width 648
Table 15-5. Complementary outputs controlled by parameters (MCHxMSEL =2'b11) 655
Table 15-6. Output behavior of the channel in response to a BREAK0 and BREAK1 (the break input is high active) 660
Table 15-7. Break function input pins locked/ released conditions 662
Table 15-8. Counting direction in different quadrature decoder signals 664
Table 15-9. the counter operation in non-quadrature decoder mode 1 666
Table 15-10. the counter operation in non-quadrature decoder mode 2 / 3 668
Table 15-11. Examples of slave mode 670
Table 15-12.The Composite PWM pulse width 768
Table 15-13. Examples of slave mode 774
Table 15-14.The Composite PWM pulse width 829
Table 15-15. Complementary outputs controlled by parameters (MCHxMSEL =2'b11) 835
Table 15-16. Break function input pins locked/ released conditions 840
Table 15-17. Slave mode example table 841
Table 15-18. Complementary outputs controlled by parameters (MCHxMSEL =2'b11) 892
Table 15-19. Break function input pins locked/ released conditions 897
Table 16-1 RTC pin configuration and function 950
Table 16-2 RTC power saving mode management 951
Table 16-3 RTC interrupts control 951
Table 17-1. Min/max FWDGT timeout period at 32KHz (IRC32K) 974
Table 17-2. Min-max timeout value at 175 MHz (f_{PCLK3}) 981
Table 17-3. Min-max timeout value at 187.5 MHz (fPCLK3)(1) 981
Table 18-1. Pin assignment 984
Table 20-1. ALGO configurations 1003
Table 23-1. Parameters of RSA algorithm 1058
Table 23-2. Integer arithmetic operations 1060
Table 23-3. Range of parameters used by RSA CRT exponentiation operation 1070

Table 23-4. Elliptic curve operations in Fp domain 1070
Table 23-5. Range of parameters used by point on elliptic curve Fp check 1071
Table 23-6. Range of parameters used by ECC scalar multiplication 1073
Table 23-7. Range of parameters used by ECDSA sign 1074
Table 23-8. Range of parameters used by ECDSA verification 1076
Table 23-9. Modular exponentiation computation times 1077
Table 23-10. ECC scalar multiplication computation times 1078
Table 23-11. ECDSA signature average computation times 1078
Table 23-12. ECDSA verification average computation times 1078
Table 23-13. Montgomery parameters average computation times 1078
Table 23-14. PKCAU interrupt requests 1079
Table 24-1. RTDEC interrupt requests 1090
Table 25-1. ADC internal input signals 1100
Table 25-2. ADC input pins definition 1100
Table 25-3. ADC differential channel pin matching 1102
Table 25-4. Trigger source for routine channels for ADC0/ADC1/ADC2 1113
Table 25-5. Trigger source for inserted channels for ADC0/ADC1/ADC2 1113
Table 25-6. t_{CONV} timings depending on resolution for ADC0 and ADC1 1116
Table 25-7. t_{CONV} timings depending on resolution for ADC2 1116
Table 25-8. Some examples show the maximum output results for N and M combinations (grayed values indicates truncation, for 12bit ADC) 1118
Table 25-9. ADC sync mode table 1119
Table 26-1. DAC I/O description 1155
Table 26-2. DAC triggers and outputs summary 1155
Table 26-3. Triggers of DAC 1156
Table 26-4. Formula of sample and refresh time 1161
Table 27-1. CMP inputs and outputs summary 1179
Table 28-1. VREF modes 1192
Table 29-1. LPDTS signals 1196
Table 29-2. Sampling time configuration 1197
Table 29-3. Trigger configuration 1198
Table 29-4. Temperature sensor behavior in low-power modes 1199
Table 29-5. Temperature sensor behavior in low-power modes 1200
Table 30-1. Pins of display interface provided by TLI 1209
Table 30-2. Supported pixel formats 1211
Table 30-3. Status flags 1213
Table 30-4. Error flags 1213
Table 31-1. RAM Details 1234
Table 31-2. DSI host and peripheral initially detect contention 1236
Table 31-3. 32-bit data storage in RGB666 1237
Table 31-4. 32-bit data storage in RGB565 1238
Table 31-5. 32-bit data storage in RGB565 1239
Table 31-6. Reset Table 1240
Table 31-7. Clocks 1241

Table 31-8. Description of DPHY pads 1245
Table 31-9. PLL Specifications 1245
Table 31-10. Counter settings for different division ratios 1246
Table 31-11. DPHY Data to Clock Timing Specifications 1249
Table 32-1. IEEE 32-Bit Single Precision Floating-Point Format 1303
Table 32-2. 8 points input sequence bits reverse operation 1304
Table 33-1. HPDF pins definition 1314
Table 33-2. HPDF internal signal 1315
Table 33-3. SPI interface clock configuration 1317
Table 33-4. Parallel data packed mode 1326
Table 33-5. Trigger signal of inserted group 1328
Table 33-6. The relationship between the maximum output resolution and oversampling filtering of SincX filtering 1329
Table 33-7. Relationship between the maximum output resolution and IOR, SFOR, SFO of the integrator 1330
Table 33-8. Features of threshold monitor working mode 1331
Table 33-9. Features of sum threshold monitor working mode 1333
Table 33-10. Maximum output rate 1336
Table 33-11. HPDF interrupt event 1337
Table 34-1. Input data configuration 1363
Table 34-2. Output data configuration 1364
Table 34-3 IEEE754 32-bit single precision floating-point format 1364
Table 34-4. TMU mode configuration 1365
Table 34-5. Mode 0 description, when IFLTEN = 1 and OFLTEN = 1 1366
Table 34-6. Mode 0 description, when IFLTEN = 0 or OFLTEN = 0 1366
Table 34-7. Mode 1 description, when IFLTEN = 1 and OFLTEN = 1 1367
Table 34-8. Mode 1 description, when IFLTEN = 0 or OFLTEN = 0 1367
Table 34-9. Mode 2 description, when IFLTEN = 1 and OFLTEN = 1 1368
Table 34-10. Mode 2 description, when IFLTEN = 0 or OFLTEN = 0 1369
Table 34-11. Mode 3 description, when IFLTEN = 1 and OFLTEN = 1 1370
Table 34-12. Mode 3 description, when IFLTEN = 0 or OFLTEN = 0 1370
Table 34-13. Mode 4 description, when IFLTEN = 1 and OFLTEN = 1 1371
Table 34-14. Mode 4 description, when IFLTEN = 0 or OFLTEN = 0 1371
Table 34-15. Mode 5 description, when IFLTEN = 1 and OFLTEN = 1 1372
Table 34-16. Mode 5 description, when IFLTEN = 0 or OFLTEN = 0 1372
Table 34-17. Mode 6 description, when IFLTEN = 1 and OFLTEN = 1 1373
Table 34-18. Mode 6 description, when IFLTEN = 0 or OFLTEN = 0 1373
Table 34-19. Mode 7 description, when IFLTEN = 1 and OFLTEN = 1 1374
Table 34-20. Mode 7 description, when IFLTEN = 0 or OFLTEN = 0 1374
Table 34-21. Mode 8 description, when IFLTEN = 1 and OFLTEN = 1 1375
Table 34-22. Mode 8 description, when IFLTEN = 0 or OFLTEN = 0 1375
Table 34-23. Recommended scaling factors in mode 8 1376
Table 34-24. Mode 9 description, when IFLTEN = 1 and OFLTEN = 1 1376
Table 34-25. Mode 9 description, when IFLTEN = 0 or OFLTEN = 0 1376

Table 34-26. Recommended scaling factors in mode 9 1377
Table 35-1 IEEE 32-Bit Single Precision Floating-Point Format 1393
Table 36-1. Signals description of IPA 1406
Table 36-2. IPA conversion mode 1407
Table 36-3. Foreground and background CLUT pixel format 1408
Table 36-4. Foreground and background pixel format 1409
Table 36-5. Expected coefficients for YUV and YcbCr modes 1411
Table 36-6. Alpha channel value modulation 1411
Table 36-7. Destination pixel format 1414
Table 36-8. IPA interrupt events 1421
Table 38-1. The five protocols of EDIM 1456
Table 38-2. AFMT pins definition 1457
Table 38-3. A-format command 1458
Table 38-4. Received data in EDIM_AFMT_ENCnRDATAm register 1460
Table 38-5. Communication state flag 1461
Table 38-6. Encoder state flag 1462
Table 38-7. TFMT pins definition 1474
Table 38-8. Tamagawa format Data ID code 1474
Table 38-9. Received data in EDIM_TFMT_RDATAx register 1475
Table 38-10. Data ID code and error information 1476
Table 38-11. ENDAT pins definition 1487
Table 38-12. EnDat command 1488
Table 38-13. Received data in EDIM_ENDAT_RDATA0 and EDIM_ENDAT_RDATA1 registers 1490
Table 38-14. Timer values for the trigger period 1491
Table 38-15. Timeout detection threshold 1494
Table 38-16. BISS pins definition 1511
Table 38-17. Received data in EDIM_BISS_SnDATAm register 1515
Table 38-18. Received data in EDIM_BISS_SnDATAm register for example 1516
Table 38-19. HDSL pins definition 1537
Table 38-20. Quality monitoring events 1539
Table 38-21. Position data in POSn (n = 0…4) registers 1541
Table 39-1. RDCM pins definition 1587
Table 39-2. High-pass filter cutoff frequency 1589
Table 39-3. CIC3 filter properties (N = I/QCICDFAC[8:0] + 1) 1593
Table 39-4. FIR0 default coefficients 1595
Table 39-5. FIR1 default coefficients 1595
Table 39-6. Group delay of filter elements 1595
Table 40-1. Description of USART important pins 1636
Table 40-2. Configuration of stop bits 1637
Table 40-3. USART interrupt requests 1652
Table 41-1. SPI internal input/output signals 1681
Table 41-2. Quad-SPI internal input/output signals 1681
Table 41-3. MISO / MISO signal switching description 1682

Table 41-4. The maximum number of data frame stored in SPIX FIFO 1685
Table 41-5. NSS function in slave mode 1688
Table 41-6. NSS function in master mode 1689
Table 41-7. SPI operation modes 1690
Table 41-8. SPI interrupt requests 1703
Table 41-9. I2S bitrate calculation formulas 1713
Table 41-10. Audio sampling frequency calculation formulas 1713
Table 41-11. The maximum number of data frame stored in I2SX FIFO 1714
Table 41-12. Direction of I2S interface signals for each operation mode 1714
Table 41-13. I2S interrupt 1718
Table 42-1. Definition of I2C-bus terminology (refer to the I2C specification of Philips semiconductors) 1739
Table 42-2. Data setup time and data hold time 1745
Table 42-3. Communication modes to be shut down 1746
Table 42-4. I2C error flags 1762
Table 42-5. I2C interrupt events 1762
Table 43-1. Mailbox descriptor with 64 byte payload 1781
Table 43-2. Data bytes for DLC 1783
Table 43-3. Mailbox Rx CODE 1783
Table 43-4. Mailbox Tx CODE 1784
Table 43-5. Mailbox size 1786
Table 43-6. Rx FIFO descriptor 1787
Table 43-7. Mailbox arbitration value(32 bit) when local priority disabled 1796
Table 43-8. Mailbox arbitration value(35 bit) when local priority enabled 1796
Table 43-9. Rx mailbox matching 1803
Table 43-10. Rx FIFO matching 1804
Table 43-11. Interrupt events 1817
Table 43-12. Rx FIFO filter element number 1831
Table 44-1 OSPIM matrix mapping 1850
Table 45-1 OSPI signal description 1854
Table 45-2 Signal line description in HyperBus™ mode 1861
Table 45-3 Instruction/address phase description in HyperBus™ mode 1861
Table 45-4 OSPI interrupt requests 1871
Table 46-1. 2 SDIOs 1896
Table 46-2. SDIO operation modes for SD & SD I/O card 1899
Table 46-3. SDIO operation modes for *e*•MMC card 1899
Table 46-4. SDIO internal input/output signals 1900
Table 46-5. SDIO pins functions 1901
Table 46-6. SDIO CMD and data phase selection 1902
Table 46-7. SDIO connections to MDMA 1910
Table 46-8. Command format 1913
Table 46-9. Card command classes (CCCs) 1913
Table 46-10. Basic commands (class 0) 1915
Table 46-11. Block-Oriented read commands (class 2) 1918

Table 46-12. Stream read commands (class 1) and stream write commands (class 3) .. 1919
Table 46-13. Block-Oriented write commands (class 4) 1919
Table 46-14. Erase commands (class 5) 1920
Table 46-15. Block oriented write protection commands (class 6) 1921
Table 46-16. Lock card (class 7) 1921
Table 46-17. Application-specific commands (class 8) 1922
Table 46-18. I/O mode commands (class 9) 1923
Table 46-19. Switch function commands (class 10) 1924
Table 46-20. Response R1 1926
Table 46-21. Response R2 1926
Table 46-22. Response R3 1926
Table 46-23. Response R4 for e•MMC 1927
Table 46-24. Response R4 for SD I/O 1927
Table 46-25. Response R5 for *e*•MMC 1927
Table 46-26. Response R5 for SD I/O 1927
Table 46-27. Response R6 1928
Table 46-28. Response R7 1928
Table 46-29. Card status 1931
Table 46-30. SD status 1934
Table 46-31. Performance move field 1936
Table 46-32. AU_SIZE field 1936
Table 46-33. Maximum AU size 1937
Table 46-34. Erase size field 1937
Table 46-35. Erase timeout field 1937
Table 46-36. Erase offset field 1938
Table 46-37. Lock card data structure 1948
Table 46-38. CMD12 instructions 1952
Table 46-39. SDIO_RESPx register at different response type 1965
Table 47-1. MDIO pins definition 1979
Table 47-2. Frame Details for Different Frame Types (1) 1981
Table 47-3. Operation Code 1982
Table 48-1. EXMC bank mapping 1994
Table 48-2. SDRAM mapping 1998
Table 48-3. NOR flash interface signals description 1998
Table 48-4. PSRAM non-muxed signal description 1999
Table 48-5. EXMC bank 0 supports all transactions 1999
Table 48-6. NOR / PSRAM controller timing parameters 2001
Table 48-7. EXMC timing models 2001
Table 48-8. Mode 1 related registers configuration 2003
Table 48-9. Mode A related registers configuration 2005
Table 48-10. Mode 2/B related registers configuration 2007
Table 48-11. Mode C related registers configuration 2009
Table 48-12. Mode D related registers configuration 2011
Table 48-13. Multiplex mode related registers configuration 2013

Table 48-14. Timing configurations of synchronous multiplexed read mode 2017
Table 48-15. Timing configurations of synchronous multiplexed write mode 2018
Table 48-16. 8-bit or 16-bit NAND interface signal 2019
Table 48-17. Bank2 of EXMC support the memory and access mode 2020
Table 48-18. NAND flash programmable parameters 2021
Table 48-19. SDRAM command truth table 2026
Table 48-20. IO definition of SDRAM controller 2026

Table 49-1. Pins used by DCI 2054

Table 49-2. Typical one-line data composition 2056
Table 49-3. Coding for SAV and EAV 2056
Table 49-4. A frame image format definition 2057
Table 49-5. General case of progressive mode 2057
Table 49-6. Memory view in byte padding mode 2059
Table 49-7. Memory view in half-word padding mode 2059
Table 49-8. Status/Error and interrupt flags 2059
Table 50-1. GPSI input/output pins 2071
Table 50-2. GPSI internal input/output signals 2071
Table 50-3. GPSI_DCLK configured by CKPL bit 2072
Table 50-4. Data bytes transfer in 32-bit words(8-bit width) 2078
Table 50-5. data bytes transfer in 32-bit words(16-bit width) 2078
Table 50-6. GPSI interrupt requests 2079
Table 51-1. Sub-frame pattern 2087
Table 51-2. Preamble pattern 2089
Table 51-3. RSPDIF time interval 2089
Table 51-4. Transition encoder encoding rules 2092
Table 51-5. Thresholds calculation 2092
Table 51-6. Preamble transition sequence 2092
Table 51-7. Bi-phase decoding rules 2092
Table 51-8. RSPDIF state 2095
Table 51-9. RSPDIF_CTL register access permissions in different states 2097
Table 51-10. Mixing data and control flow 2100
Table 51-11. Minimum rspdif_ck frequency and audio sampling rate 2101
Table 51-12. Conditions of RSPDIF_symbol_ck generation 2102
Table 51-13. RSPDIF interrupt events 2106
Table 52-1. Commonly used audio sampling rate 2123
Table 52-2. External synchronization configuration 2125
Table 52-3 FIFO request generation conditions 2130
Table 52-4 Get the microphone data read register times under different configurations 2133
Table 52-5. TDM configuration 2136
Table 52-6 AC'97 transmitter tag definition 2138
Table 52-7 AC'97 receiver tag definition 2138
Table 52-8. SOPD mode 2139

Table 52-9 Parity bit calculation 2140
Table 52-10 Mute frame composition 2141
Table 52-11 A-law encoding 2143
Table 52-12 A-law decoding 2143
Table 52-13 Mu-law encoding 2144
Table 52-14 Mu-law decoding 2144
Table 52-15 Interrupt control 2148
Table 53-1. USBHS signal description 2167
Table 53-2. USBHS supported speeds 2168
Table 53-3. USBHS global interrupt 2187
Table 54-1. Ethernet signals (MII) 2266
Table 54-2. Ethernet signals (RMII) 2267
Table 54-3. Clock range 2269
Table 54-4. Rx interface signal encoding 2271
Table 54-5. Destination address filtering table 2277
Table 54-6. Source address filtering table 2278
Table 54-7. Error status decoding in Receive Descriptor0, only used for normal descriptor (DFM=0) 2302
Table 54-8. Supported time stamp snapshot with PTP register configuration 2349
Table 55-1. Alignment of valid data 2380
Table 56-1. List of abbreviations used in register 2456
Table 56-2. List of terms 2456
Table 57-1. Revision history 2458

1. System and memory architecture

The devices of GD32H77x_78x series are 32-bit general-purpose microcontrollers based on the Arm® Cortex®-M7 processor. The Arm® Cortex®-M7 processor includes 64-bit AMBA4 AXI, 32-bit AHB peripheral (AHBP) port, 32-bit AHB slave port for external master to access memories and APB interface for CoreSight debug components. 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®-M7 processor

The Arm® Cortex®-M7 processor is a highly efficient high-performance, embedded processor that features low interrupt latency, low-cost debug, and has backwards compatibility with existing Cortex-M profile processors. The processor has an in-order super-scalar pipeline that means many instructions can be dual-issued, including load/load and load/store instruction pairs because of multiple memory interfaces. The Cortex®-M7 is a high-performance processor, which features a 6-stage superscalar pipeline with branch prediction and an optional FPU capable of single-precision and double-precision operations. The instruction and data buses have been enlarged to 64-bit wide over the previous 32-bit buses.

The interfaces that the processor supports include:

- 64-bit AXI4 interface
- 32-bit AHB master interface
- 32-bit AHB slave interface
- 64-bit instruction TCM interface
- 2x32-bit data TCM interfaces

The processor contains the following external interfaces:

- AHBP interface
- AHBS interface
- AHBD interface
- External Private Peripheral Bus
- ATB interfaces
- TCM interface
- Cross Trigger interface
- AXIM interface

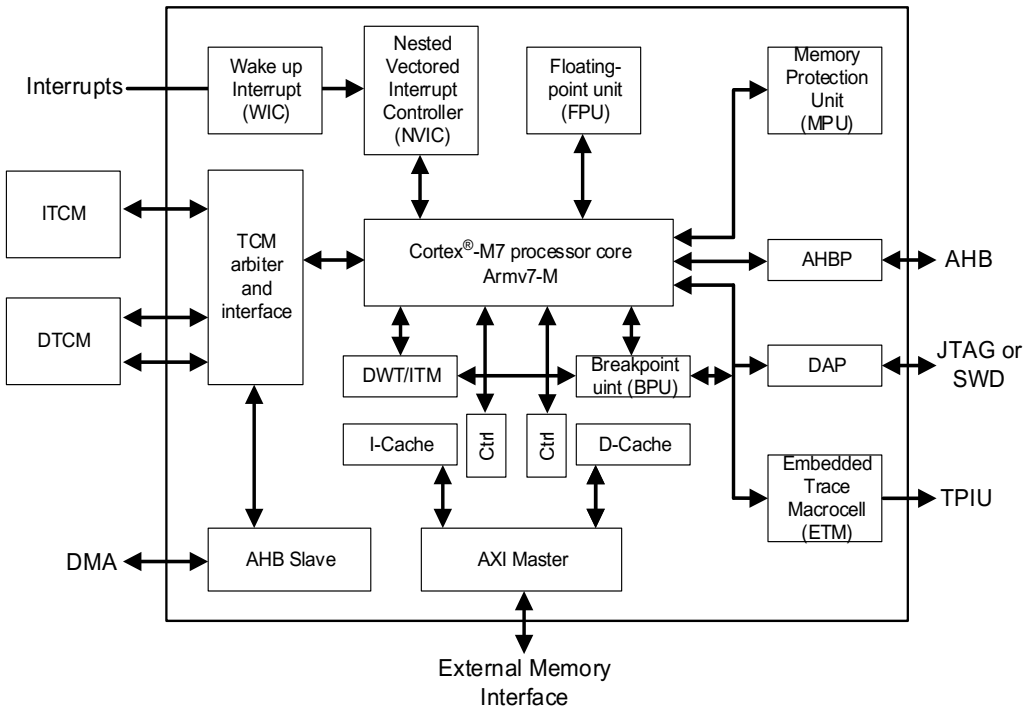
The Cortex®-M7 processor is based on the ARMv7-M architecture and supports a powerful and scalable instruction set including general data processing I/O control tasks, advanced data processing bit field manipulations, DSP and floating point instructions. Some system peripherals listed below are also provided by Cortex®-M7:

- Nested Vectored Interrupt Controller (NVIC)

- Flash Patch and Breakpoint (FPB)
- Data Watchpoint and Trace (DWT)
- Instrumentation Trace Macrocell (ITM)
- Embedded Trace Macrocell (ETM)
- JTAG or SWD Debug Port
- Trace Port Interface Unit (TPIU)
- Memory Protection Unit (MPU)
- Floating Point Unit (FPU), double-precision
- Load Store Unit (LSU)
- Data Processing Unit (DPU)
- Prefetch Unit (PFU)

Figure 1-1. The structure of the Cortex®-M7 processor shows the Cortex®-M7 processor block diagram. For more information, refer to the Arm® Cortex®-M7 Technical Reference Manual.

Figure 1-1. The structure of the Cortex®-M7 processor



1.2. System architecture

An interconnect matrix includes an AXI bus matrix and two AHB bus matrices, which enables parallel access paths between multiple masters and slaves in the system. The interconnection relationship of the interconnect matrix is shown in *Table 1-1. The interconnection relationship of the interconnect matrix*. In the following table, "1" indicates the corresponding master is able to access the corresponding slave through the interconnect

matrix, while the blank means the corresponding master cannot access the corresponding slave through the interconnect matrix.

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

Master interface / Slave interface	AXIM	AHBP	ITCM	DTCM	MDMA AXI	MDMA AHBS	DMA0 MEM	DMA0 PERIPH	DMA1 MEM	DMA1 PERIPH	IPA	TLI	SDIO0	SDIO1	ENET	USBHS0	USBHS1	FFT	EtherCAT
ITCM			1			1													
DTCM				1		1													
NVMC	1				1		1	1	1	1	1	1	1	1	1	1	1	1	1
AXI SRAM	1				1		1	1	1	1	1	1	1	1	1	1	1	1	1
RAM shared (ITCM/DTCM/AXI)	1				1		1	1	1	1	1	1	1	1	1	1	1	1	1
SRAM0	1				1		1	1	1	1	1			1	1	1	1	1	
SRAM1	1				1		1	1	1	1	1			1	1	1	1	1	
Backup RAM	1				1		1	1	1	1				1			1		
AHB1	1	1			1		1	1	1	1	1			1					
AHB2		1					1	1	1	1				1					
AHB3	1				1														
AHB4	1				1		1	1	1	1				1			1		
APB1		1					1	1	1	1				1					
APB2		1					1	1	1	1				1					
APB3	1				1														
APB4	1				1		1	1	1	1				1			1		
EXMC	1				1		1	1	1	1	1	1	1	1	1	1	1	1	1
EtherCAT		1			1		1	1	1	1	1			1					
OSPI	1				1		1	1	1	1	1	1	1	1	1	1	1	1	1

The system architecture of GD32H77x_78x devices are shown in ***Figure 1-2. The system architecture of GD32H77x devices***, ***Figure 1-3. The system architecture of GD32H78x devices***, ***Figure 1-4. The system architecture of GD32H77R devices***, and the work frequency is related to the voltage of power supply, please refer to the datasheet.

The GD32H78x series supports two operating modes: General Mode and High-Performance Mode.

In General Mode, the maximum frequency is 700 MHz, while in High-Performance Mode, the

maximum frequency is 750 MHz (provided that the operating temperature is within 85°C and the Vcore voltage operates at 0.97V).

Figure 1-2. The system architecture of GD32H77x devices

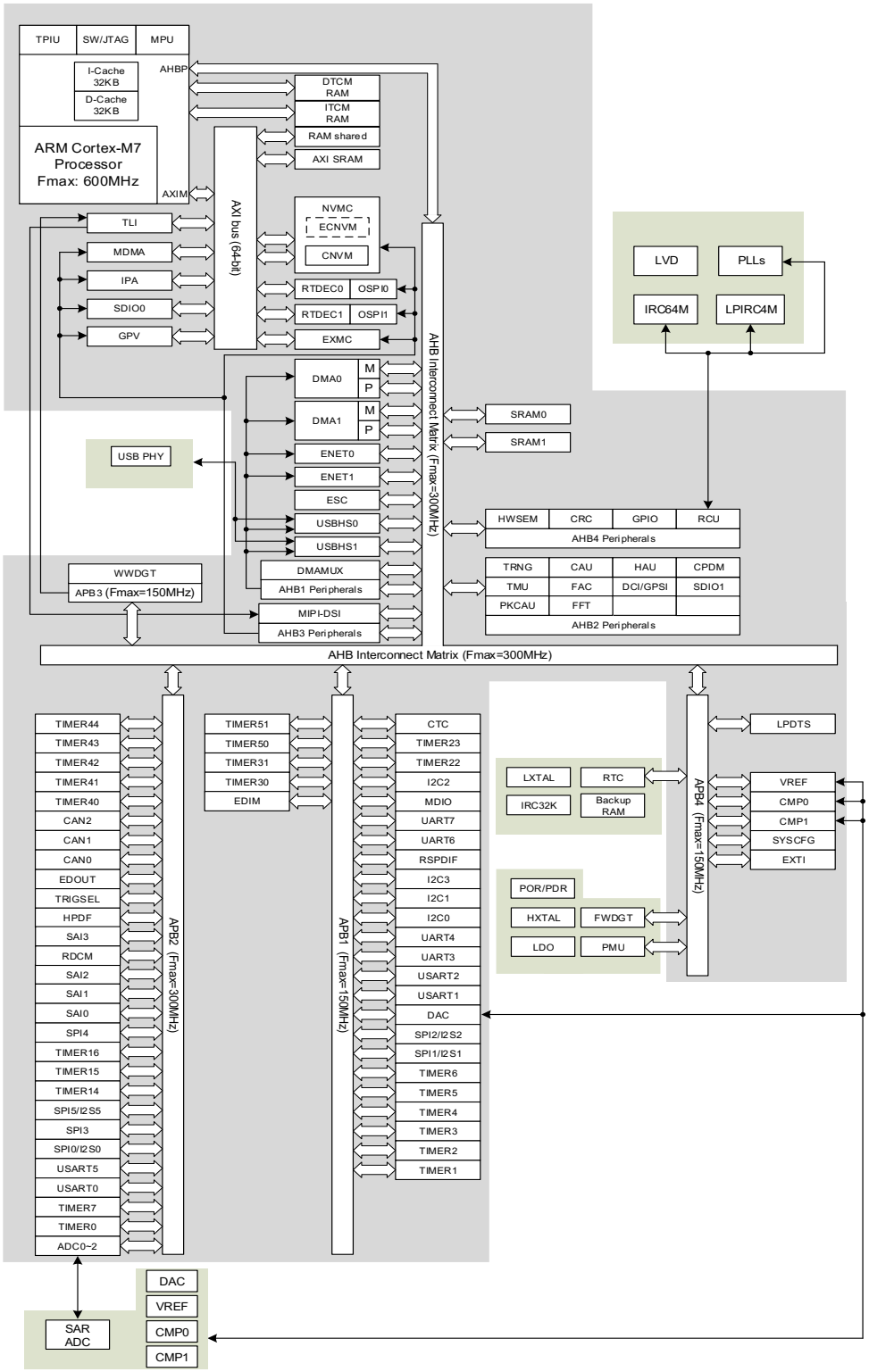
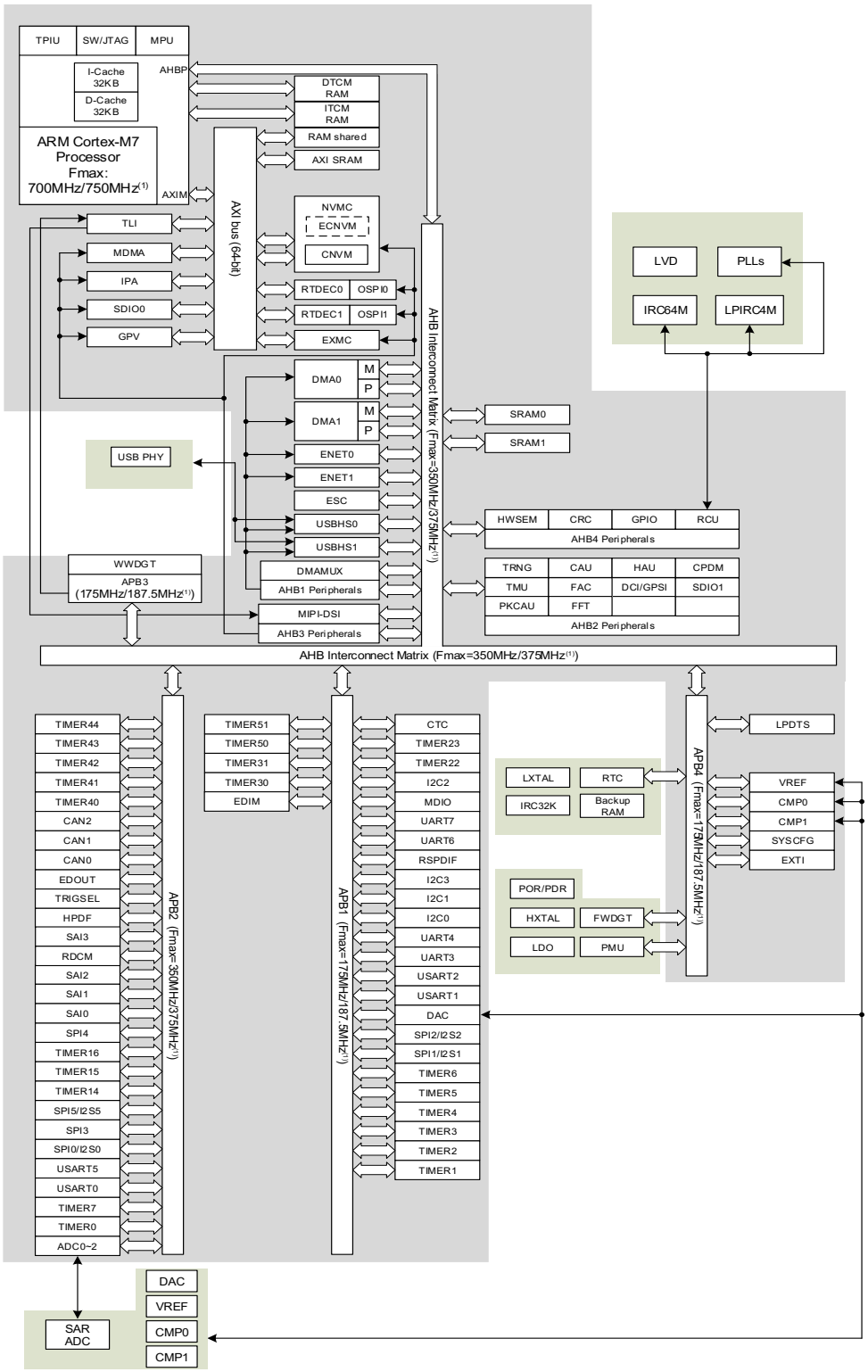
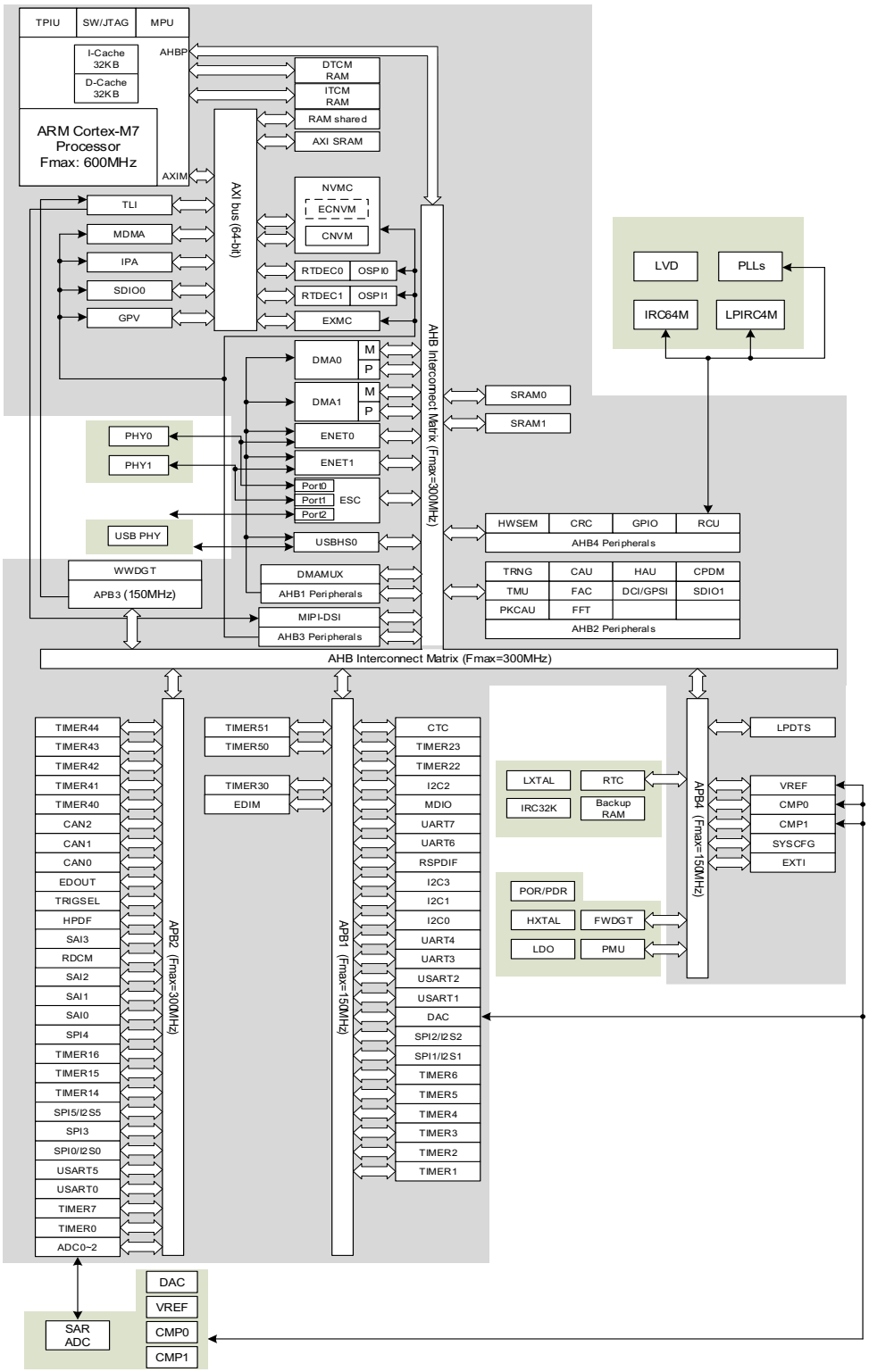


Figure 1-3. The system architecture of GD32H78x devices



Note: (1) Applicable to the high-performance mode of GD32H78x.

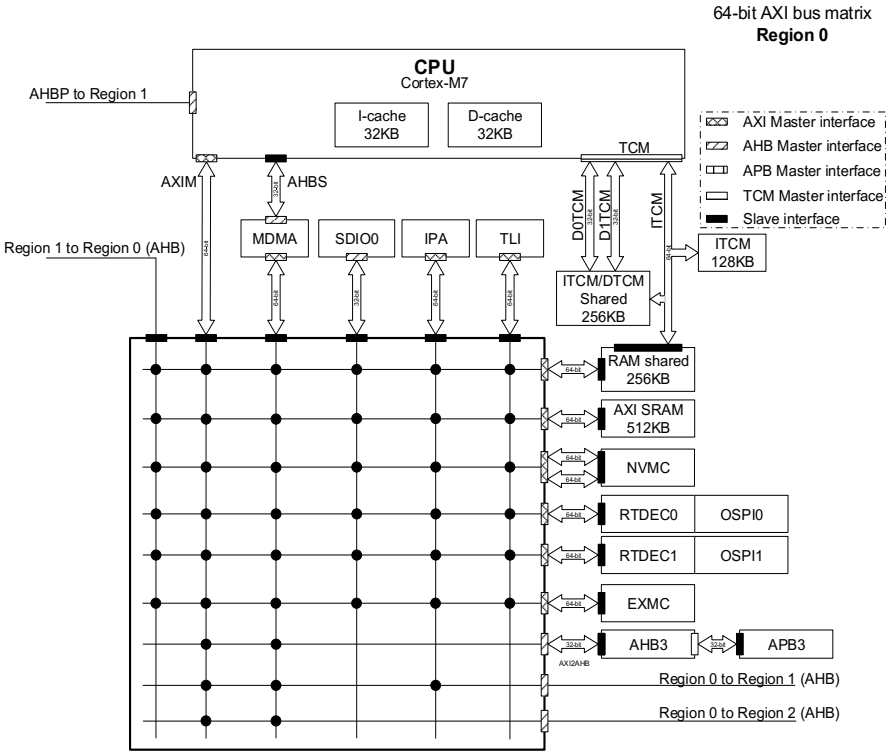
Figure 1-4. The system architecture of GD32H77R devices



1.2.1. Bus matrix Region 0

The 64-bit AXI bus matrix Region 0 is shown in *Figure 1-5. Bus matrix Region 0*.

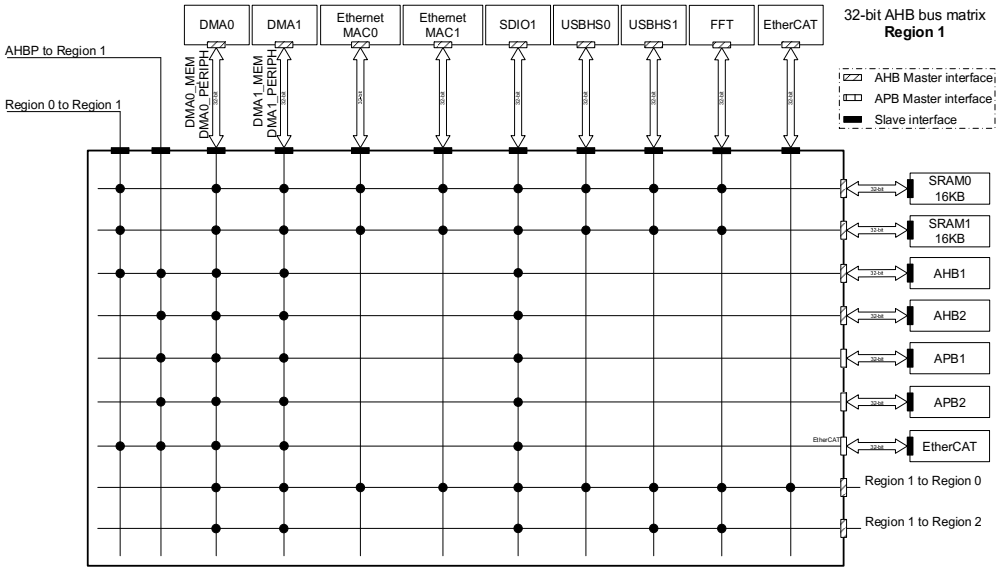
Figure 1-5. Bus matrix Region 0



1.2.2. Bus matrix Region 1

The 32-bit AHB bus matrix Region 1 is shown in *Figure 1-6. Bus matrix Region 1*.

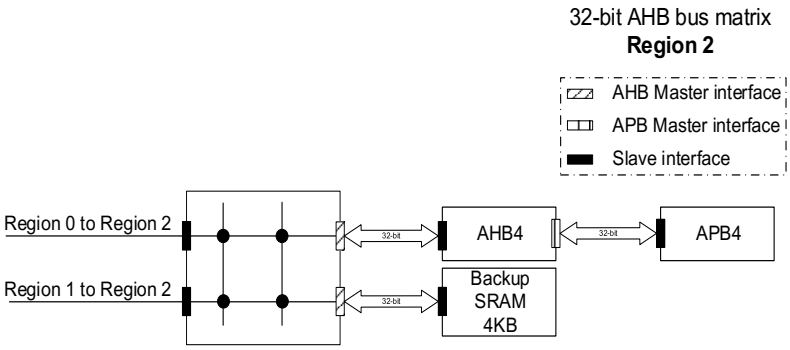
Figure 1-6. Bus matrix Region 1



1.2.3. Bus matrix Region 2

The 32-bit AHB bus matrix Region 2 is shown in ***Figure 1-7. Bus matrix Region 2***.

Figure 1-7. Bus matrix Region 2



As is shown above, there are 3 Regions of bus matrix, including I-Cache, D-Cache, TCM, MDMA, DMA0, DMA1, IPA, TLI, SDIO0, SDIO1, ENET, EtherCAT, FFT, USBHS0 and USBHS1. The AXI bus matrix in Region 0 and AHB bus matrix in Region 1 and Region 2 provide guarantee and arbitration for concurrent access from multiple masters to multiple slaves. Arbitration adopts round-robin scheduling algorithm with QoS function in Region 0 and round-robin scheduling algorithm in Region 1 and Region 2. The ITCM and DTCM are directly connected to the Cortex®-M7 core through TCM buses and the accesses of ITCM and DTCM are zero-wait states. ENET is the Ethernet. TLI is the TFT LCD interface. USBHS is the high-speed USB. And IPA is the image processing accelerator.

1.3. Memory map

The Arm® Cortex®-M7 processor is structured in Harvard architecture which can use separate buses to fetch instructions and load/store data. ***Table 1-2. Memory map of GD32H77x_78x devices*** shows the memory map of the GD32H77x_78x series devices, including Code, SRAM, peripheral, and other pre-defined regions. Almost each peripheral is allocated 1KB of space. This allows simplifying the address decoding for each peripheral.

Table 1-2. Memory map of GD32H77x_78x devices

Pre-defined Regions	Bus	Address	Peripherals
External RAM		0xD000 0000 - 0xDFFF FFFF	EXMC - SDRAM device 1
		0xC000 0000 - 0xCFFF FFFF	EXMC - SDRAM device 0 (EXMC Bank 0 Region 0-3)
		0xA000 1000 - 0xBFFF FFFF	Reserved
		0xA000 0000 - 0xA000 0FFF	Reserved
		0x9000 0000 - 0x9FFF FFFF	OSPI0
		0x8000 0000 - 0x8FFF FFFF	EXMC - NAND
		0x7000 0000 - 0x7FFF FFFF	OSPI1
		0x6000 0000 - 0x6FFF FFFF	EXMC - NOR/PSRAM/SRAM
	EtherCAT	0x5F00 4000 - 0x5FFF FFFF	Reserved
		0x5F00 0000 - 0x5F00 3FFF	ESC
Peripheral	AHB4	0x5802 6800 - 0x5EFF FFFF	Reserved
		0x5802 6400 - 0x5802 67FF	HWSEM
		0x5802 6000 - 0x5802 63FF	Reserved
		0x5802 5000 - 0x5802 5FFF	Reserved
		0x5802 4C00 - 0x5802 4FFF	CRC
		0x5802 4800 - 0x5802 4BFF	Reserved
		0x5802 4400 - 0x5802 47FF	RCU
		0x5802 2C00 - 0x5802 43FF	Reserved
		0x5802 2800 - 0x5802 2BFF	GPIOK
		0x5802 2400 - 0x5802 27FF	GPIOJ
		0x5802 2000 - 0x5802 23FF	Reserved
		0x5802 1C00 - 0x5802 1FFF	GPIOH
		0x5802 1800 - 0x5802 1BFF	GPIOG
		0x5802 1400 - 0x5802 17FF	GPIOF
		0x5802 1000 - 0x5802 13FF	GPIOE
		0x5802 0C00 - 0x5802 0FFF	GPIOD
		0x5802 0800 - 0x5802 0BFF	GPIOC
		0x5802 0400 - 0x5802 07FF	GPIOB
		0x5802 0000 - 0x5802 03FF	GPIOA
		0x5801 0000 - 0x5801 FFFF	Reserved

Pre-defined Regions	Bus	Address	Peripherals
	APB4	0x5800 7400 - 0x5800 FFFF	Reserved
		0x5800 7000 - 0x5800 73FF	Reserved
		0x5800 6C00 - 0x5800 6FFF	Reserved
		0x5800 6800 - 0x5800 6BFF	LPDTS
		0x5800 5C00 - 0x5800 67FF	Reserved
		0x5800 5800 - 0x5800 5BFF	PMU
		0x5800 5400 - 0x5800 57FF	Reserved
		0x5800 4C00 - 0x5800 53FF	Reserved
		0x5800 4800 - 0x5800 4BFF	FWDGT
		0x5800 4400-0x5800 47FF	Reserved
		0x5800 4100 - 0x5800 43FF	Reserved
		0x5800 4000 - 0x5800 40FF	RTC
		0x5800 3C00 - 0x5800 3FFF	VREF
		0x5800 3800 - 0x5800 3BFF	CMP0 - CMP1
		0x5800 3400 - 0x5800 37FF	Reserved
		0x5800 3000 - 0x5800 33FF	Reserved
		0x5800 2C00 - 0x5800 2FFF	Reserved
		0x5800 2800 - 0x5800 2BFF	Reserved
		0x5800 2400 - 0x5800 27FF	Reserved
		0x5800 2000 - 0x5800 23FF	Reserved
		0x5800 1C00 - 0x5800 1FFF	Reserved
		0x5800 1800-0x5800 1BFF	Reserved
		0x5800 1400 - 0x5800 17FF	Reserved
		0x5800 0800 - 0x5800 13FF	Reserved
		0x5800 0400 - 0x5800 07FF	SYSCFG
		0x5800 0000 - 0x5800 03FF	EXTI
	AHB3	0x5200 C000 - 0x57FF FFFF	Reserved
		0x5200 BC00 - 0x5200 BFFF	RTDEC1
		0x5200 B800 - 0x5200 BBFF	RTDEC0
		0x5200 B400 - 0x5200 B7FF	OSPIM
		0x5200 B000 - 0x5200 B3FF	CPDM(OSPI1)
		0x5200 A000 - 0x5200 AFFF	OSPI1
		0x5200 9400 - 0x5200 9FFF	Reserved
		0x5200 9100 - 0x5200 93FF	Reserved
		0x5200 9000 - 0x5200 90FF	RAMECCMU Region 0
		0x5200 8400 - 0x5200 8FFF	Reserved
		0x5200 8000 - 0x5200 83FF	CPDM(SDIO0)
		0x5200 7400 - 0x5200 7FFF	Reserved
		0x5200 7000 - 0x5200 73FF	SDIO0
		0x5200 6400 - 0x5200 6FFF	Reserved

Pre-defined Regions	Bus	Address	Peripherals
		0x5200 6000 - 0x5200 63FF	CPDM(OSPI0)
		0x5200 5000 - 0x5200 5FFF	OSPI0
		0x5200 4000 - 0x5200 4FFF	EXMC
		0x5200 3000 - 0x5200 3FFF	MIPI-DSI
		0x5200 2400 - 0x5200 2FFF	Reserved
		0x5200 2000 - 0x5200 23FF	NVMC
		0x5200 1000 - 0x5200 1FFF	IPA
		0x5200 0800 - 0x5200 0FFF	Reserved
		0x5200 0000 - 0x5200 07FF	MDMA
		0x5110 0000 - 0x51FF FFFF	Reserved
		0x5100 0000 - 0x510F FFFF	AXI interconnect matrix
		0x5001 1000 - 0x50FF FFFF	Reserved
		0x5001 0000 - 0x5001 0FFF	Reserved
	APB3	0x5000 4000 - 0x5000 FFFF	Reserved
		0x5000 3400 - 0x5000 3FFF	Reserved
		0x5000 3000 - 0x5000 33FF	WWDGT
		0x5000 2000 - 0x5000 2FFF	Reserved
		0x5000 1400 - 0x5000 1FFF	Reserved
		0x5000 1000 - 0x5000 13FF	TLI
		0x5000 0000 - 0x5000 0FFF	Reserved
	AHB2	0x4802 5800 - 0x4FFF FFFF	Reserved(AHB2)
		0x4802 5000 - 0x4802 57FF	Reserved
		0x4802 4800 - 0x4802 4FFF	FAC
		0x4802 4400 - 0x4802 47FF	TMU
		0x4802 4000 - 0x4802 43FF	FFT
		0x4802 3100 - 0x4802 3FFF	Reserved
		0x4802 3000 - 0x4802 30FF	RAMECCMU Region 1
		0x4802 2C00 - 0x4802 2FFF	Reserved(AHB2)
		0x4802 2800 - 0x4802 2BFF	CPDM(SDIO1)
		0x4802 2400 - 0x4802 27FF	SDIO1
		0x4802 1C00 - 0x4802 23FF	Reserved(AHB2)
		0x4802 1800 - 0x4802 1BFF	TRNG
		0x4802 1400 - 0x4802 17FF	HAU
		0x4802 1000 - 0x4802 13FF	CAU
		0x4802 0400 - 0x4802 0FFF	Reserved(AHB2)
		0x4802 0000 - 0x4802 03FF	DCI/GPSI
		0x4800 2000 - 0x4801 FFFF	Reserved(AHB2)
		0x4800 0000 - 0x4800 1FFF	PKCAU
	AHB1	0x400C 0000 - 0x47FF FFFF	Reserved(AHB1)
		0x4008 0000 - 0x400B FFFF	USBHS1

Pre-defined Regions	Bus	Address	Peripherals
		0x4004 0000 - 0x4007 FFFF	USBHS0
		0x4003 8C00 - 0x4003 FFFF	Reserved
		0x4003 8400 - 0x4003 8BFF	Reserved
		0x4003 8000 - 0x4003 83FF	Reserved
		0x4003 3000 - 0x4003 7FFF	Reserved
		0x4003 0000 - 0x4003 2FFF	Reserved
		0x4002 C000 - 0x4002 FFFF	Reserved
		0x4002 BC00 - 0x4002 BFFF	ENET1
		0x4002 B000 - 0x4002 BBFF	
		0x4002 A000 - 0x4002 AFFF	
		0x4002 8000 - 0x4002 9FFF	ENET0
		0x4002 6800 - 0x4002 7FFF	Reserved
		0x4002 6400 - 0x4002 67FF	Reserved
		0x4002 6000 - 0x4002 63FF	Reserved
		0x4002 5000 - 0x4002 5FFF	Reserved
		0x4002 4000 - 0x4002 4FFF	Reserved
		0x4002 3C00 - 0x4002 3FFF	Reserved
		0x4002 3800 - 0x4002 3BFF	Reserved
		0x4002 3400 - 0x4002 37FF	Reserved
		0x4002 3000 - 0x4002 33FF	Reserved
		0x4002 2C00 - 0x4002 2FFF	Reserved
		0x4002 2800 - 0x4002 2BFF	Reserved
		0x4002 2400 - 0x4002 27FF	Reserved
		0x4002 2000 - 0x4002 23FF	Reserved
		0x4002 1C00 - 0x4002 1FFF	Reserved
		0x4002 1800 - 0x4002 1BFF	Reserved
		0x4002 1400 - 0x4002 17FF	Reserved
		0x4002 1000 - 0x4002 13FF	Reserved
		0x4002 0C00 - 0x4002 0FFF	Reserved
		0x4002 0800 - 0x4002 0BFF	DMAMUX
		0x4002 0400 - 0x4002 07FF	DMA1
		0x4002 0000 - 0x4002 03FF	DMA0
	APB2	0x4001 F400 - 0x4001 FFFF	Reserved
		0x4001 F000 - 0x4001 F3FF	TIMER44
		0x4001 E000-0x4001 EFFF	Reserved
		0x4001 DC00 - 0x4001 DFFF	TIMER43
		0x4001 D800 - 0x4001 DBFF	TIMER42
		0x4001 D400 - 0x4001 D7FF	TIMER41
		0x4001 D000 - 0x4001 D3FF	TIMER40
		0x4001 C000 - 0x4001 CFFF	CAN2(4KB)

Pre-defined Regions	Bus	Address	Peripherals
		0x4001 B000 - 0x4001 BFFF	CAN1(4KB)
		0x4001 A000 - 0x4001 AFFF	CAN0(4KB)
		0x4001 8C00 - 0x4001 9FFF	Reserved
		0x4001 8900 - 0x4001 8BFF	Reserved
		0x4001 8800 - 0x4001 88FF	EDOUT
		0x4001 8400 - 0x4001 87FF	TRIGSEL
		0x4001 8000 - 0x4001 83FF	Reserved(APB2)
		0x4001 7C00 - 0x4001 7FFF	Reserved
		0x4001 7800 - 0x4001 7BFF	Reserved
		0x4001 7400 - 0x4001 77FF	Reserved
		0x4001 7000 - 0x4001 73FF	HPDF
		0x4001 6C00 - 0x4001 6FFF	Reserved
		0x4001 6800 - 0x4001 6BFF	SAI3
		0x4001 6400 - 0x4001 67FF	RDCM
		0x4001 6000 - 0x4001 63FF	SAI2
		0x4001 5C00 - 0x4001 5FFF	SAI1
		0x4001 5800 - 0x4001 5BFF	SAI0
		0x4001 5400 - 0x4001 57FF	Reserved
		0x4001 5000 - 0x4001 53FF	SPI4
		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	SPI5/I2S5
		0x4001 3400 - 0x4001 37FF	SPI3
		0x4001 3000 - 0x4001 33FF	SPI0/I2S0
		0x4001 2C00 - 0x4001 2FFF	ADC2
		0x4001 2800 - 0x4001 2BFF	ADC1
		0x4001 2400 - 0x4001 27FF	ADC0
		0x4001 2000 - 0x4001 23FF	Reserved
		0x4001 1C00 - 0x4001 1FFF	Reserved
		0x4001 1800 - 0x4001 1BFF	Reserved
		0x4001 1400 - 0x4001 17FF	USART5
		0x4001 1000 - 0x4001 13FF	USART0
		0x4001 0C00 - 0x4001 0FFF	Reserved
		0x4001 0800 - 0x4001 0BFF	Reserved
		0x4001 0400 - 0x4001 07FF	TIMER7
		0x4001 0000 - 0x4001 03FF	TIMER0
	APB1	0x4000 F800 - 0x4000 FFFF	Reserved

Pre-defined Regions	Bus	Address	Peripherals
		0x4000 F400 - 0x4000 F7FF	TIMER51
		0x4000 F000 - 0x4000 F3FF	TIMER50
		0x4000 EC00 - 0x4000 EFFF	TIMER31
		0x4000 E800 - 0x4000 EBFF	TIMER30
		0x4000 E400 - 0x4000 E7FF	TIMER23
		0x4000 E000 - 0x4000 E3FF	TIMER22
		0x4000 DC00 - 0x4000 DFFF	Reserved
		0x4000 C400 - 0x4000 DBFF	EDIM
		0x4000 C000 - 0x4000 C3FF	I2C2
		0x4000 9800 - 0x4000 BFFF	Reserved
		0x4000 9400 - 0x4000 97FF	MDIO
		0x4000 8800 - 0x4000 93FF	Reserved
		0x4000 8400 - 0x4000 87FF	CTC
		0x4000 8000 - 0x4000 83FF	Reserved
		0x4000 7C00 - 0x4000 7FFF	UART7
		0x4000 7800 - 0x4000 7BFF	UART6
		0x4000 7400 - 0x4000 77FF	DAC0
		0x4000 7000 - 0x4000 73FF	Reserved
		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	RSPDIF
		0x4000 3C00 - 0x4000 3FFF	SPI2/I2S2
		0x4000 3800 - 0x4000 3BFF	SPI1/I2S1
		0x4000 3400 - 0x4000 37FF	Reserved
		0x4000 3000 - 0x4000 33FF	Reserved
		0x4000 2C00 - 0x4000 2FFF	Reserved
		0x4000 2800 - 0x4000 2BFF	Reserved
		0x4000 2400 - 0x4000 27FF	Reserved
		0x4000 2000 - 0x4000 23FF	Reserved
		0x4000 1C00 - 0x4000 1FFF	Reserved
		0x4000 1800 - 0x4000 1BFF	Reserved

Pre-defined Regions	Bus	Address	Peripherals
		0x4000 1400 - 0x4000 17FF	TIMER6
		0x4000 1000 - 0x4000 13FF	TIMER5
		0x4000 0C00 - 0x4000 0FFF	TIMER4
		0x4000 0800 - 0x4000 0BFF	TIMER3
		0x4000 0400 - 0x4000 07FF	TIMER2
		0x4000 0000 - 0x4000 03FF	TIMER1
SRAM		0x3880 1000 - 0x3FFF FFFF	Reserved
		0x3880 0000 - 0x3880 0FFF	Backup SRAM
		0x3000 E000 - 0x387F FFFF	Reserved
		0x3000 C000 - 0x3000 DFFF	SRAM shared with FFT(8KB)
		0x3000 A000 - 0x3000 BFFF	SRAM shared with USBHS RAM(8KB)
		0x3000 9000 - 0x3000 9FFF	SRAM shared with ENET1 RAM(4KB)
		0x3000 8000 - 0x3000 8FFF	SRAM shared with ENET0 RAM(4KB)
		0x3000 4000 - 0x3000 7FFF	SRAM1(16KB)
		0x3000 0000 - 0x3000 3FFF	SRAM0(16KB)
		0x2410 0000 - 0x2FFF FFFF	Reserved
		0x240C 0000 - 0x240F FFFF	Reserved
		0x2408 0000 - 0x240B FFFF	RAM(256KB) shared (ITCM/AXI)
		0x2400 0000 - 0x2407 FFFF	AXI SRAM(512KB)
		0x2008 0000 - 0x23FF FFFF	Reserved
		0x2007 0000 - 0x2007 FFFF	Reserved
		0x2006 0000 - 0x2006 FFFF	Reserved
		0x2005 0000 - 0x2005 FFFF	Reserved
		0x2004 0000 - 0x2004 FFFF	Reserved
		0x2001 C000 - 0x2003 FFFF	DTCM RAM(from ITCM/DTCM)
		0x2001 8000 - 0x2001 BFFF	
		0x2001 0000 - 0x2001 7FFF	
		0x2000 D000 - 0x2000 FFFF	
		0x2000 C000 - 0x2000 CFFF	
		0x2000 8000 - 0x2000 BFFF	
		0x2000 5000 - 0x2000 7FFF	
		0x2000 2000 - 0x2000 4FFF	
		0x2000 1000 - 0x2000 1FFF	
		0x2000 0000 - 0x2000 0FFF	
Code		0x1FFF FC10 - 0x1FFF FFFF	Reserved
		0x1FFF FC00 - 0x1FFF FC0F	Reserved

Pre-defined Regions	Bus	Address	Peripherals	
		0x1FFF F818 - 0x1FFF FBFF	Reserved	
		0x1FFF F800 - 0x1FFF F817	Reserved	
		0x1FFF F000 - 0x1FFF F7FF	Reserved	
		0x1FFF EC00 - 0x1FFF EFFF	Reserved	
		0x1FFF C010 - 0x1FFF EBFF	Reserved	
		0x1FFF C000 - 0x1FFF C00F	Reserved	
		0x1FFF B000 - 0x1FFF BFFF	Reserved	
		0x1FFF 8000 - 0x1FFF AFFF	Reserved	
		0x1FFF 7A10 - 0x1FFF 7FFF	Reserved	
		0x1FFF 7800 - 0x1FFF 7A0F	Reserved	
		0x1FFF 7400 - 0x1FFF 77FF	Reserved	
		0x1FFF 7000 - 0x1FFF 73FF	Reserved	
		0x1FFF 1A00 -0x1FFF 6FFF	Reserved	
		0x1FFF 0000 - 0x1FFF 19FF	OTP	
		0x1FEF 0000 - 0x1FFE FFFF	System Memory	
		0x1002 0000 - 0x1FEE FFFF	Reserved	
		0x1001 0000 - 0x1001 FFFF	Reserved	
		0x1000 0000 - 0x1000 FFFF	Reserved	
		0x0A00 D000 - 0x0FFF FFFF	Reserved	
		0x0A00 C000 - 0x0A00 CFFF	Reserved	
		0x0A00 8000 - 0x0A00 BFFF	Reserved	
		0x0A00 0000 - 0x0A00 7FFF	Reserved	
		0x08C0 1000 - 0x09FF FFFF	Reserved	
		0x08C0 0000 - 0x08C0 0FFF	Reserved	
		0x0898 0000 - 0x08BF FFFF	Reserved	
		0x0880 0000 - 0x0897 FFFF	ECNVM	
		0x0840 0000 - 0x087F FFFF		
		0x083C 0000 - 0x083F FFFF		
		0x0830 0000 - 0x083B FFFF		
		0x0820 0000 - 0x082F FFFF		
		0x0810 0000 - 0x081F FFFF	CNVM	
		0x0808 0000 - 0x080F FFFF		
		0x0806 0000 - 0x0807 FFFF		
		0x0802 0000 - 0x0805 FFFF		
		0x0801 0000 - 0x0801 FFFF		
		0x0800 0000 - 0x0800 FFFF		
		0x0030 0000 - 0x07FF FFFF	Reserved	
		0x0010 0000 - 0x002F FFFF	Reserved	
		0x000A 0000 - 0x000F FFFF	Reserved	
		0x0006 0000 - 0x0009 FFFF	ITCM RAM	RAM shared

Pre-defined Regions	Bus	Address	Peripherals	
		0x0002 0000 - 0x0005 FFFF		
		0x0001 0000 - 0x0001 FFFF		Dedicated
		0x0000 0000 - 0x0000 FFFF		

1.3.1. On-chip SRAM memory

The devices of GD32H77x_78x series contain up to 512KB of on-chip SRAM (AXI SRAM), and up to 256KB of RAM shared by ITCM or DTCM, 256KB of RAM shared by ITCM or AXISRAM, 128KB of dedicated ITCM, 16KB of SRAM0 and 16KB of SRAM1, 4KB of backup SRAM. All of AHB SRAM support byte, half-word (16 bits), and word (32 bits) accesses. The on-chip SRAM (AXI SRAM) support byte, half-word (16 bits), word (32 bits) and double words (64 bits) accesses. SRAM0 and SRAM1 can be accessed by almost all AHB masters. The backup SRAM (BKPSRAM) is implemented in the backup domain, which can keep its content even when the V_{DD} power supply is down.

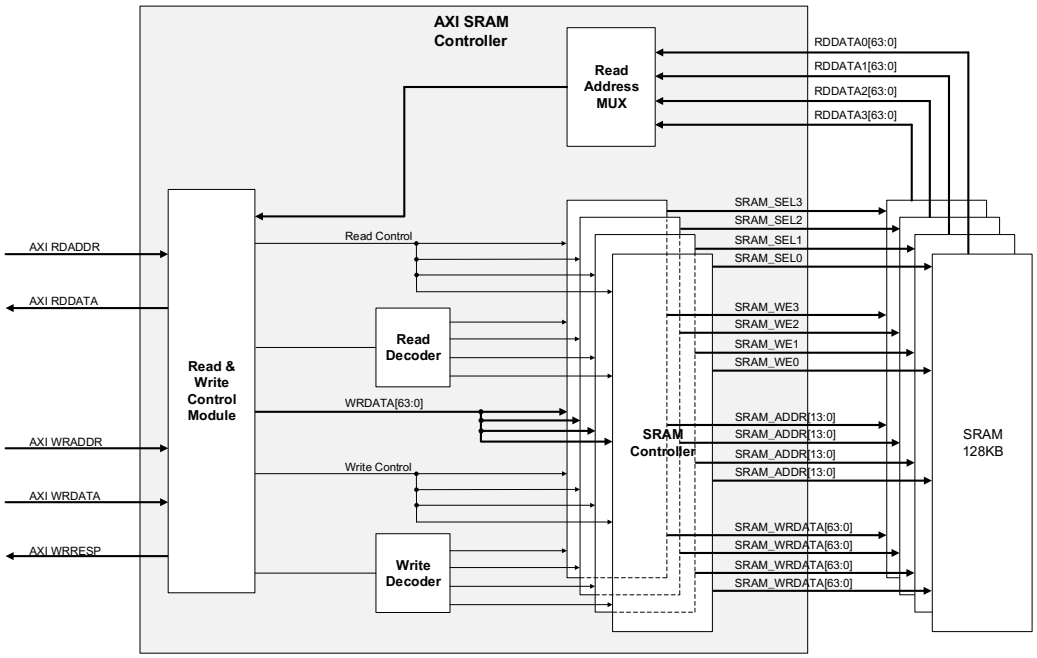
ITCM/DTCM SRAM ECC function

The TCM ECC function (ITCMECCEN / DTCM0ECCEN / DTCM1ECCEN) can be enabled to increase the robustness.

AXI SRAM

The on-chip SRAM (AXI SRAM) controller is consisted of read and write control module, MUX, decoder and SRAM controllers. There is an arbiter with round-robin scheme for each SRAM controller. The block digram is shown in *Figure 1-8. Block diagram of AXI SRAM controller*.

Figure 1-8. Block diagram of AXI SRAM controller



RAM shared by ITCM/DTCM/AXI SRAM

256KB of RAM can be used by ITCM or DTCM, 256KB of RAM can be used by ITCM or AXISRAM, which can be configured through ITCM_DTCM_SHARED and ITCM_AXI_SHARED bits in Option byte status register 1 register, as described in ***Table 1-3. Configuration of ITCM/DTCM/AXI SRAM***. And the block digram is shown in ***Figure 1-9. Block diagram of RAM shared by ITCM/DTCM/AXI SRAM***.

Figure 1-9. Block diagram of RAM shared by ITCM/DTCM/AXI SRAM

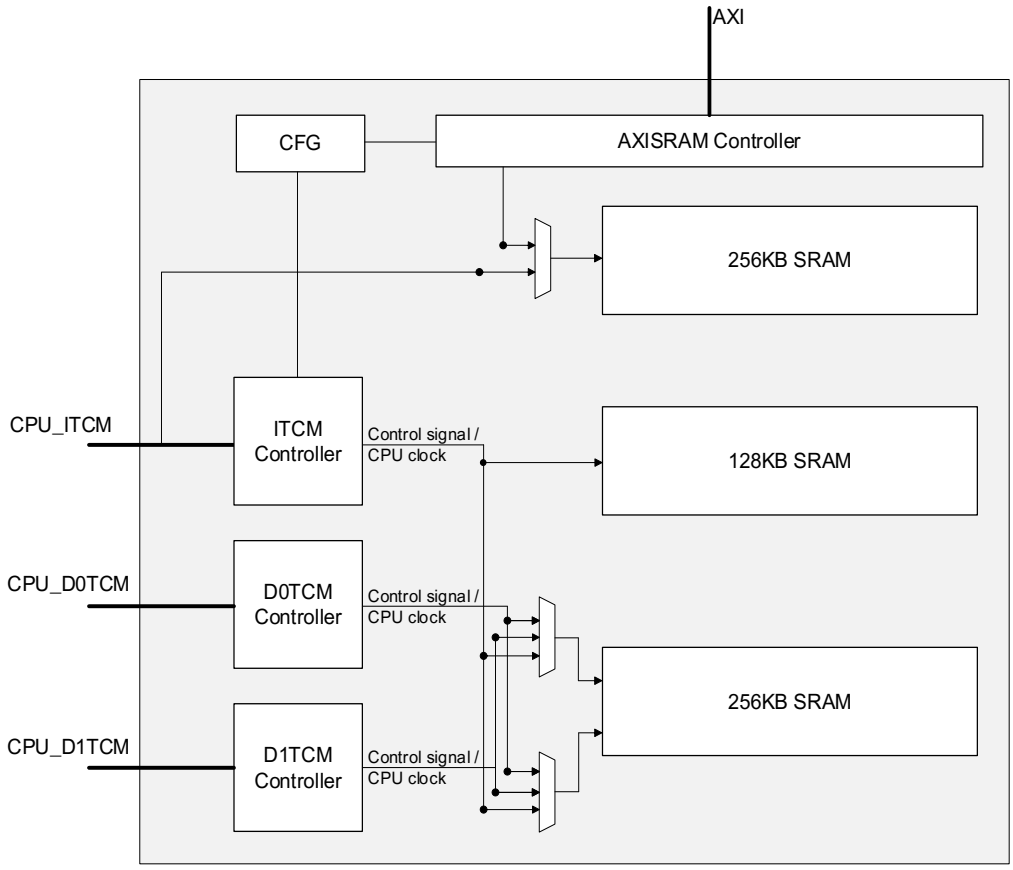


Table 1-3. Configuration of ITCM/DTCM/AXI SRAM

ITCM_DTCM_SHARED [2:0]	ITCM_AXI_SHARED[2:0]	ITCM size(KB)	DTCM size(KB)	AXI SRAM size(KB)
000	000	128	256	256
001	000	192	192	256
010	000	256	128	256
011	000	320	64	256
100	000	384	0	256
000	001	192	256	192
001	001	256	192	192
010	001	320	128	192
…	…	…	…	…
100	100	640	0	0

ECC Function

When reading and writing SRAM, it supports 7/8-bit ECC function. It can correct 1 bit error and detect multiple bits error. For details on the register section, refer to the RAMECCMU.

1.3.2. Non-volatile memory overview

The devices provide high density Non-volatile memory, which is organized as follows:

Up to 2 MB of code non-volatile memory CNVM (one bank)
Up to 7.5 MB of Extended Code non-volatile memory ECNVM for instruction and data
Option bytes to configure the device

Refer to ***Non-volatile memory controller (NVMC)*** Chapter for more details.

1.4. Boot configuration

The GD32H77x_78x devices provide different boot sources which can be selected by the BOOT pin and BOOT_ADDR0/1[15:0] in Boot address for Arm® Cortex®-M7 core register (NVMC_BTADDR_MDF). The details are shown in ***Table 1-4. Boot mode selection*** and ***Table 1-5. Details of Boot mode***.

The value on the BOOT pin is latched on the 4th rising edge of CK_SYS after a reset. It is up to the user to set the BOOT pin after a power-on reset or a system reset to select the required boot source. Once the BOOT pin has been sampled, it is free and can be used for other purposes. The BOOT_ADDR0[15:0] and BOOT_ADDR1[15:0] address allows to configure the boot memory address to any address from 0x0000 0000 to 0x9000 0000. The boot mode can be obtained from the BOOT_MODE[2:0] bits in the SYSCFG_USERCFG register.

Table 1-4. Boot mode selection

Boot source address	Boot mode selection pin
	BOOT
MSB of the boot address: defined by BOOT_ADDR0[15:0] LSB of the boot address: 0x0000	0
MSB of the boot address: defined by BOOT_ADDR1[15:0] LSB of the boot address: 0x0000	1

Table 1-5. Details of Boot mode

SCR	SPC[7:0]	OTA(FW_UPDATE)	BOOT_ADDRESS (configured in BOOT_ADDRx(x = 0,1))	BOOT MODE	Boot From
1	x	x	XXXX	SECURITY BOOT	ROM
0	Protection level high	x	0x0800_0000~max ECNVM	USER BOOT	BOOT_ADDRESS
			0x7000_0000	USER BOOT	OSPI1
			0x9000_0000	USER BOOT	OSPI0
			other	USER BOOT	0x0800_0000

SCR	SPC[7:0]	OTA(FW_UPDATE)	BOOT_ADDRESS (configured in BOOT_ADDRx(x = 0,1))	BOOT MODE	Boot From
0	No protection / Protection level low	1	x	USER BOOT	Check_bootloader(1)
		0	0x1FF1_0000	USER BOOT	Check_bootloader(1)
		0	0x1FEF_0000	SYSTEM BOOT	BootLoader
		0	0x0800_0000~max ECNVM	USER BOOT	BOOT_ADDRESS
		0	0x7000_0000	USER BOOT	OSPI1
		0	0x9000_0000	USER BOOT	OSPI0
		0	0x0000_0000	SRAM BOOT(ITCM)	0x0000_0000
		0	0x2000_0000	SRAM BOOT(DTCM)	0x2000_0000
		0	0x2400_0000~ max AXI SRAM(ITCM/DTCM/AXI)	SRAM BOOT(AXI SRAM)	BOOT_ADDRESS
		0	0x2408_0000~ max RAM shared	SRAM BOOT(RAM shared)	BOOT_ADDRESS
		0	Other	USER BOOT	0x0800_0000
		0		SYSTEM BOOT	0x1FEF_0000

(1) The GD32H77xxI/GD32H78xxI series does not have Check_bootloader.

The embedded bootloader supports multi interfaces to update the NVMC memory. There will be USART ports, USBHS ports, SDIO ports can be used on GD32H77x_78x line products. The details are shown in the datasheet.

Note: To ensure system reliability and stability, refer to Application Note ***《AN298 GD32H77x_H78x BootLoader Operation Precautions》*** during the hardware design stage. Chapter 2 of this application note describes the process by which the BootLoader detects firmware upgrade interfaces, the pins used at each stage, the timing for releasing unused pins, and hardware design recommendations for these pins. Please pay particular attention to two points: 1. Before an interface is selected, the USART RX pin must maintain a stable logic level. Signals generated by peripherals such as sensor outputs, watchdog ICs, and buttons may cause the BootLoader to select an incorrect interface; 2. During SD card detection, the BootLoader outputs clock and command signals on the SDIO pins. Peripherals connected to these pins may produce abnormal outputs.

The embedded Check_bootloader is stored in the Check Information Block and is used to complete OTA upgrade.

1.5. System configuration controller (SYSCFG)

The system configuration controller main functions are the following:

- Analog switch configuration management
- I2C Fm+ configuration
- Selection of the Ethernet PHY interface.
- Management of the external interrupt line connection to the GPIOs
- Management of the I/O compensation cell
- Management BOR reset level
- Management timer break input lock.

1.6. Timer break input lock

When internal SRAM/NVMC ECC error, LVD or CPU core lockup occurs, this function allows to disable TIMER output. Refer to ***Lockup control register (SYSCFG_LKCTL)*** for details.

1.7. AXI interconnect matrix (AXIIM)

The AXI interconnect is based on the Arm® CoreLink™ NIC-400 Network Interconnect. It has 6 initiator ports or ASIBs (AMBA slave interface blocks), and 9 target ports or AMIBs (AMBA master interface blocks).

1.7.1. Characteristics

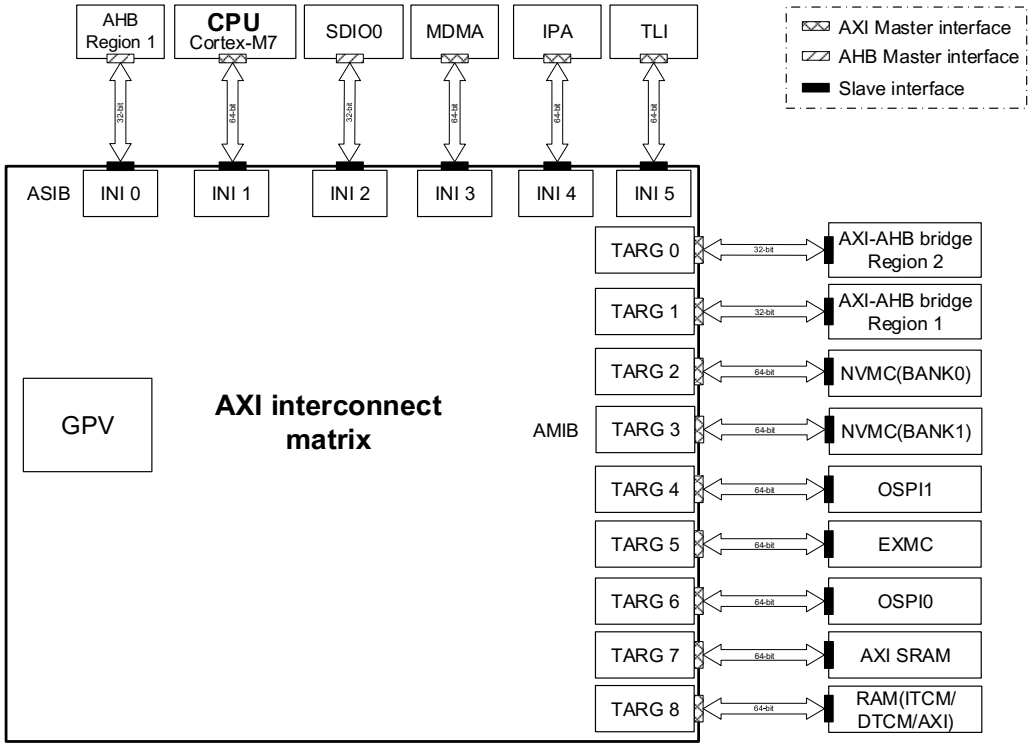
The main functions of AXI interconnect matrix are the following:

- 64-bit AXI bus switch matrix with 6 ASIBs and 9 AMIBs
- Distributed Global Programmers View (GPV)
- Programmable Quality of Service (QoS)

1.7.2. Function overview

The block diagram of AXI interconnect matrix is shown in ***Figure 1-10. Block diagram of AXI interconnect matrix***.

Figure 1-10. Block diagram of AXI interconnect matrix



The configurations of ASIB and AMIB are shown in *Table 1-6. Configuration of ASIBs* and *Table 1-7. Configuration of AMIBs*.

Table 1-6. Configuration of ASIBs

ASIB	Protocol	Bus width	Read issuing	Write issuing	Master interface
INI 0	AHB-lite	32	1	1	AHB Region 1
INI 1	AXI4	64	7	32	CPU Cortex-M7
INI 2	AHB-lite	32	1	1	SDIO0
INI 3	AXI4	64	4	1	MDMA
INI 4	AXI4	64	3	1	IPA
INI 5	AXI4	64	1	1	TLI

Table 1-7. Configuration of AMIBs

AMIB	Protocol	Bus width	Read acceptance	Write acceptance	Total acceptance	Slave interface
TARG 0	AXI4	32	1	1	1	AHB3 Peripherals and Region 2
TARG 1	AXI4	32	1	1	1	Region 1
TARG 2	AXI4	64	3	2	5	NVMC(BANK0)
TARG 3	AXI4	64	3	2	5	NVMC(BANK1)
TARG 4	AXI4	64	2	1	3	OSPI1
TARG 5	AXI4	64	3	3	6	EXMC
TARG 6	AXI4	64	2	1	3	OSPI0
TARG 7	AXI4	64	2	2	2	AXI SRAM
TARG 8	AXI4	64	2	2	2	RAM shared (ITCM/AXI SRAM)

Quality of Service (QoS)

Using QoS provides configurable QoS options for ASIB and AMIB, regulation of read and write requests and programmable QoS facilities for attached AMBA masters. The AXI interconnect matrix uses priority based arbitration when different ASIBs try to access a AMIB. The ASIB has configurable read and write channel priority. The priority range is 0x0 ~ 0xF and 0 is the lowest priority. Refer to register ***AXI Slave Port x read QOS control register (AXI_SPx_RDQOS_CTL)*** and ***AXI Slave Port x write QOS control register (AXI_SPx_WRQOS_CTL)*** for details. Least recently used (LRU) priority scheme is used when the priorities of different ASIBs are the same.

Global Programmers View (GPV)

Global Programmers View (GPV) for the entire interconnect that is configurable so that any master, or a discrete configuration slave interface, can access it. For more information, see the Arm® CoreLink™ QoS-400 Network Interconnect Advanced Quality of Service, Supplement to Arm® CoreLink™ NIC-400 Network Interconnect Technical Reference Manual.

1.8. System configuration registers

SYSCFG base address: 0x5800 0400

1.8.1. Peripheral mode configuration register 0 (SYSCFG_PMCFG0)

Address offset: 0x04
Reset value: 0x0F00 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	IO_SEL_G0[2:0]			PC3SWON	PC2SWON	PA1SWON	PA0SWON	ENET0_PHY_SEL	ENET1_PHY_SEL	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								PB9FMPEN	PB8FMPEN	PB7FMPEN	PB6FMPEN	I2C3FMPEN	I2C2FMPEN	I2C1FMPEN	I2C0FMPEN
								rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30:28	IO_SEL_G0[2:0]	Coder group 0 IO select 000: Connect to AFMT IO 001: Connect to TFMT IO 010: Connect to Endat IO 011: Connect to BISS-C IO 100: Not connected 101: Connect to HDSL Others: Not connected
27	PC3SWON	PC3 switch open This bit controls the analog switch between PC3 and PC3_C, determining whether the pin enables the analog function. 0: PC3_C enables input/output functionality, while PC3 disables input/output functionality (only the analog function is available). The analog functions of PC3 and PC3_C will be connected together, and the analog functions of the corresponding peripherals cannot be used simultaneously. 1: PC3 enables input/output functionality, while PC3_C disables input/output functionality (only the analog function is available). The analog functions of PC3 and PC3_C are independent of each other.
26	PC2SWON	PC2 switch open This bit controls the analog switch between PC2 and PC2_C, determining whether the pin enables the analog function. 0: PC2_C enables input/output functionality, while PC2 disables input/output

		functionality (only the analog function is available). The analog functions of PC2 and PC2_C will be connected together, and the analog functions of the corresponding peripherals cannot be used simultaneously. 1: PC2 enables input/output functionality, while PC2_C disables input/output functionality (only the analog function is available). The analog functions of PC2 and PC2_C are independent of each other.
25	PA1SWON	PA1 switch open This bit controls the analog switch between PA1 and PA1_C, determining whether the pin enables the analog function. 0: PA1_C enables input/output functionality, while PA1 disables input/output functionality (only the analog function is available). The analog functions of PA1 and PA1_C will be connected together, and the analog functions of the corresponding peripherals cannot be used simultaneously. 1: PA1 enables input/output functionality, while PA1_C disables input/output functionality (only the analog function is available). The analog functions of PA1 and PA1_C are independent of each other.
24	PA0SWON	PA0 switch open This bit controls the analog switch between PA0 and PA0_C, determining whether the pin enables the analog function. 0: PA0_C enables input/output functionality, while PA0 disables input/output functionality (only the analog function is available). The analog functions of PA0 and PA0_C will be connected together, and the analog functions of the corresponding peripherals cannot be used simultaneously. 1: PA0 enables input/output functionality, while PA0_C disables input/output functionality (only the analog function is available). The analog functions of PA0 and PA0_C are independent of each other.
23	ENET0_PHY_SEL	Ethernet0 PHY interface selection This bit selects the Ethernet PHY interface. 0: MII is selected 1: RMII is selected
22	ENET1_PHY_SEL	Ethernet1 PHY interface selection This bit selects the Ethernet PHY interface. 0: MII is selected 1: RMII is selected
21:8	Reserved	Must be kept at reset value.
7	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 1: Enable Fm+ mode
6	PB8FMPEN	I2C Fm+ mode on PB8 pin enable

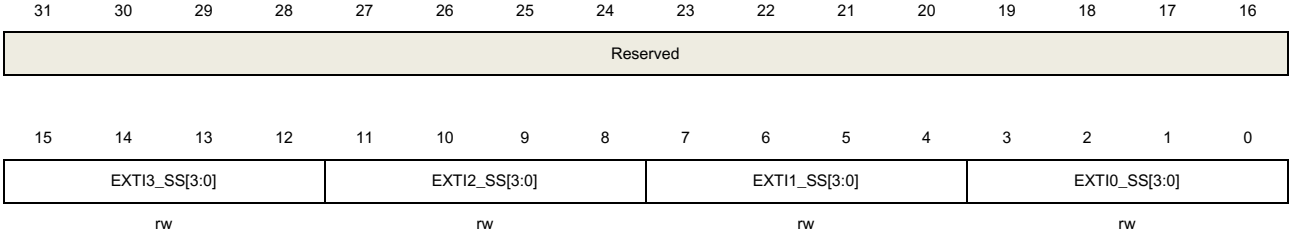
This bit controls I2C Fm+ mode, the speed control of the pin is bypassed.
0: Disable Fm+ mode
1: Enable Fm+ mode

5	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 1: Enable Fm+ mode
4	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 1: Enable Fm+ mode
3	I2C3FMPEN	I2C3 Fm+ mode enable This bit controls I2C3 Fm+ mode. 0: Disable Fm+ mode 1: Enable Fm+ mode
2	I2C2FMPEN	I2C2 Fm+ mode enable This bit controls I2C2 Fm+ mode. 0: Disable Fm+ mode 1: Enable Fm+ mode
1	I2C1FMPEN	I2C1 Fm+ mode enable This bit controls I2C1 Fm+ mode. 0: Disable Fm+ mode 1: Enable Fm+ mode
0	I2C0FMPEN	I2C0 Fm+ mode enable This bit controls I2C0 Fm+ mode. 0: Disable Fm+ mode 1: Enable Fm+ mode

1.8.2. EXTI sources selection register 0 (SYSCFG_EXTISS0)

Address offset: 0x08
Reset value: 0x0000 0000

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



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 0111: PH3 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 0111: PH2 pin 1010: PK2 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 0111: PH1 pin 1010: PK1 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 0111: PH0 pin 1010: PK0 pin

1.8.3. EXTI sources selection register 1 (SYSCFG_EXTISS1)

Address offset: 0x0C

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	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 0111: PH7 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 0111: PH6 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 0111: PH5 pin

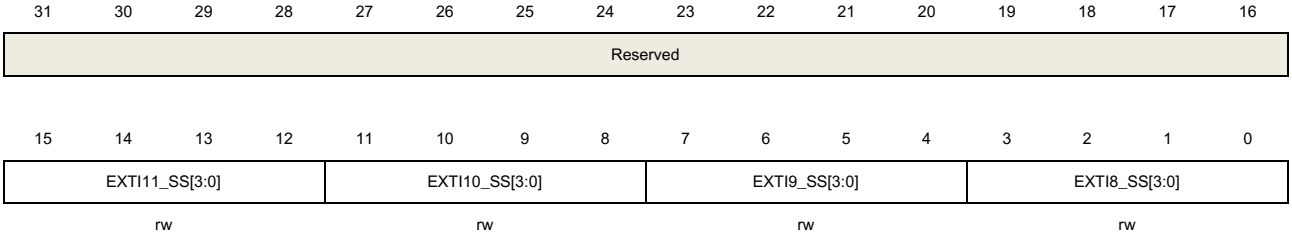
3:0	EXTI4_SS[3:0]	EXTI 4 sources selection 0000: PA4 pin 0001: PB4 pin 0010: PC4 pin 0011: PD4 pin 0100: PE4 pin 0101: PF4 pin 0110: PG4 pin 0111: PH4 pin

1.8.4. EXTI sources selection register 2 (SYSCFG_EXTISS2)

Address offset: 0x10

Reset value: 0x0000 0000

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



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 0110: PG11 pin 0111: PH11 pin 1001: PJ11 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

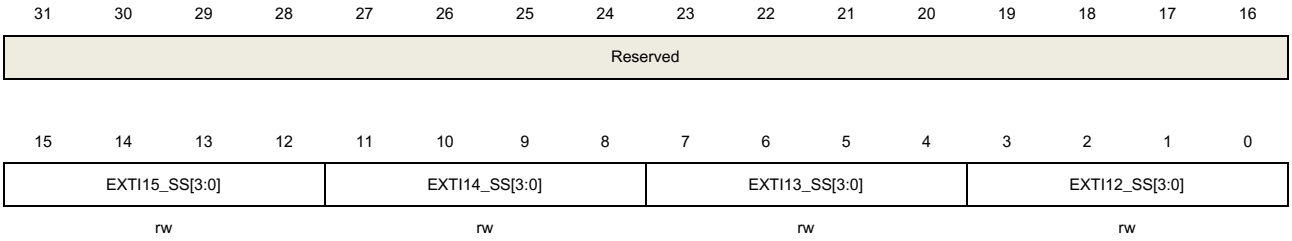
		0111: PH10 pin 1001: PJ10 pin
7:4	EXTI9_SS[3:0]	EXTI 9 sources selection 0000: PA9 pin 0001: PB9 pin 0010: PC9 pin 0011: PD9 pin 0100: PE9 pin 0101: PF9 pin 0110: PG9 pin 0111: PH9 pin 1001: PJ9 pin
3:0	EXTI8_SS[3:0]	EXTI 8 sources selection 0000: PA8 pin 0001: PB8 pin 0010: PC8 pin 0011: PD8 pin 0100: PE8 pin 0101: PF8 pin 0110: PG8 pin 0111: PH8 pin 1001: PJ8 pin

1.8.5. EXTI sources selection register 3 (SYSCFG_EXTISS3)

Address offset: 0x14

Reset value: 0x0000 0000

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



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

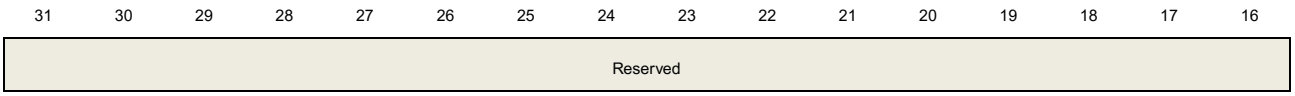
		0011: PD15 pin 0100: PE15 pin 0101: PF15 pin 0110: PG15 pin 0111: PH15 pin
11:8	EXTI14_SS[3:0]	EXTI 14 sources selection 0000: PA14 pin 0001: PB14 pin 0010: PC14 pin 0011: PD14 pin 0100: PE14 pin 0101: PF14 pin 0110: PG14 pin 0111: PH14 pin
7:4	EXTI13_SS[3:0]	EXTI 13 sources selection 0000: PA13 pin 0001: PB13 pin 0010: PC13 pin 0011: PD13 pin 0100: PE13 pin 0101: PF13 pin 0110: PG13 pin 0111: PH13 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 0110: PG12 pin 0111: PH12 pin

1.8.6. Lockup control register (SYSCFG_LKCTL)

Address offset: 0x18

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
AXIRAM_LOCK	ITCM_LOCK	DTCM_LOCK	SRAM0_LOCK	SRAM1_LOCK	Reserved			BKPRAM_LOCK	CPU_LOCK	Reserved			LVD_LOCK	Reserved	
rw	rw	rw	rw	rw				rw	rw				rw		

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	AXIRAM_LOCK	Region 0 AXI-SRAM ECC double error lock bit This bit is set by software and cleared by a system reset only. 0: Region 0 AXI-SRAM ECC double error signal is disconnected from the break input of TIMER0/7/14/15/16/40~44 1: Region 0 AXI-SRAM ECC double error signal is connected from the break input of TIMER0/7/14/15/16/40~44
14	ITCM_LOCK	Region 0 ITCM-RAM ECC double error lock bit This bit is set by software and cleared by a system reset. 0: Region 0 ITCM-RAM ECC double error signal is disconnected from the break input of TIMER0/7/14/15/16/40~44 1: Region 0 ITCM-RAM ECC double error signal is connected from the break input of TIMER0/7/14/15/16/40~44
13	DTCM_LOCK	Region 0 DTCM ECC double error lock bit This bit is set by software and cleared by a system reset. 0: Region 0 DTCM ECC double error signal is disconnected from the break input of TIMER0/7/14/15/16/40~44 1: Region 0 DTCM ECC double error signal is connected from the break input of TIMER0/7/14/15/16/40~44
12	SRAM0_LOCK	Region 1 SRAM0 ECC double error lockup bit This bit is set by software and cleared by a system reset. 0: Region 1 SRAM0 ECC double error signal is disconnected from the break input of TIMER0/7/14/15/16/40~44 1: Region 1 SRAM0 ECC double error signal is connected from the break input of TIMER0/7/14/15/16/40~44
11	SRAM1_LOCK	Region 1 SRAM1 ECC double error lockup bit This bit is set by software and cleared by a system reset. 0: Region 1 SRAM1 ECC double error signal is disconnected from the break input of TIMER0/7/14/15/16/40~44 1: Region 1 SRAM1 ECC double error signal is connected from the break input of TIMER0/7/14/15/16/40~44
10:8	Reserved	Must be kept at reset value.
7	BKPRAM_LOCK	Region 2 backup SRAM ECC double error lockup bit This bit is set by software and cleared by a system reset.

		0: Region 2 backup SRAM ECC double error signal is disconnected from the break input of TIMER0/7/14/15/16/40~44 1: Region 2 backup SRAM ECC double error signal is connected from the break input of TIMER0/7/14/15/16/40~44
6	CPU_LOCK	CPU lockup bit This bit is set by software and cleared by a system reset. 0: CPU lockup signal is disconnected from the break input of TIMER0/7/14/15/16/40~44 1: CPU lockup signal is connected from the break input of TIMER0/7/14/15/16/40~44
5:3	Reserved	Must be kept at reset value.
2	LVD_LOCK	Low voltage detector lockup bit This bit is set by software and cleared by a system reset. 0: LVD signal is disconnected from the break input of TIMER0/7/14/15/16/40~44 1: LVD signal is connected from the break input of TIMER0/7/14/15/16/40~44
1:0	Reserved	Must be kept at reset value.

1.8.7. Bus timeout enable register (SYSCFG_BUSTO)

Address offset: 0x1C

Reset value: 0x0000 0002

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FFTBUSTO	USBHS1BUSTO	USBHS0BUSTO	SDIO1BUSTO	ENET1BUSTO	ENET0BUSTO	DMA1PERIPHBUSTO	DMA1MEMBUSTO	DMA0PERIPHBUSTO	DMA0MEMBUSTO	CPUAHBPBUSTO	IPABUSTO	MDMABUSTO	TLIBUSTO	CPUBUSTO	SDIO0BUSTO
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	ESCBUSTO	ESC bus timeout enable bit. 0: Disable 1: Enable
15	FFTBUSTO	FFT bus timeout enable bit. 0: Disable 1: Enable

14	USBHS1BUSTO	USBHS1 bus timeout enable bit. 0: Disable 1: Enable
13	USBHS0BUSTO	USBHS0 bus timeout enable bit. 0: Disable 1: Enable
12	SDIO1BUSTO	SDIO1 bus timeout enable bit. 0: Disable 1: Enable
11	ENET1BUSTO	ENET1 bus timeout enable bit. 0: Disable 1: Enable
10	ENET0BUSTO	ENET0 bus timeout enable bit. 0: Disable 1: Enable
9	DMA1PERIPHBUSTO	DMA1_PERIPH bus timeout enable bit. 0: Disable 1: Enable
8	DMA1MEMBUSTO	DMA1_MEM bus timeout enable bit. 0: Disable 1: Enable
7	DMA0PERIPHBUSTO	DMA0_PERIPH bus timeout enable bit. 0: Disable 1: Enable
6	DMA0MEMBUSTO	DMA0_MEM bus timeout enable bit. 0: Disable 1: Enable
5	CPUAHBPBUSTO	CPU-AHBP bus timeout enable bit. 0: Disable 1: Enable
4	IPABUSTO	IPA bus timeout enable bit. 0: Disable 1: Enable
3	MDMABUSTO	MDMA bus timeout enable bit. 0: Disable 1: Enable
2	TLIBUSTO	TLI bus timeout enable bit. 0: Disable

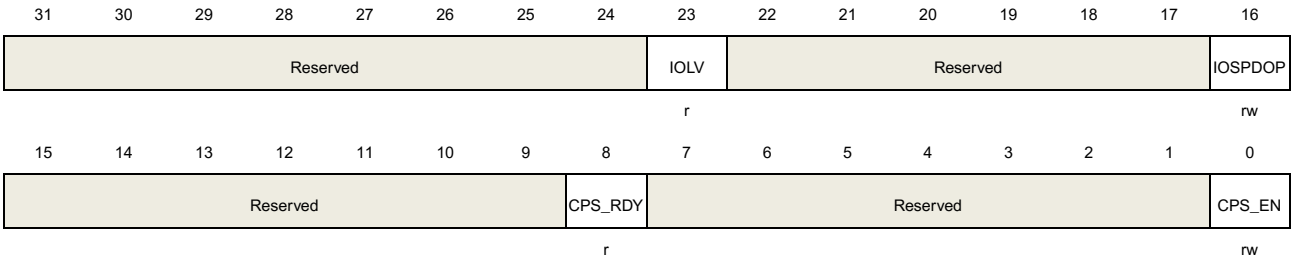
		1: Enable
1	CPUBUSTO	CPU bus timeout enable bit. **Note:** The reset value of CPUBUSTO depends on the configuration of the CPUBUSTO bit in the option byte NVMC_OBSTAT1_EFT. 0: Disable 1: Enable
0	SDIO0BUSTO	SDIO0 bus timeout enable bit. 0: Disable 1: Enable

1.8.8. I/O compensation control register (SYSCFG_CPSCTL)

Address offset: 0x20

Reset value: 0x00X0 0000

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



Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23	IOLV	I/O in low voltage state This bit is set to 1 by setting the IOSPDOPEN bit in the option bytes. 0: Product supply voltage is working higher than 2.7V 1: Product supply voltage is working below 2.7V
22:17	Reserved	Must be kept at reset value.
16	IOSPDOP	I/O speed optimization, High-speed at low-voltage This bit is written by software to optimize the I/O speed when the product voltage is low. It must be used only if the product supply voltage is below 2.7V (IOLV bit is '1'). Setting this bit when V_{DD} is higher than 2.7V might be destructive. 0: No I/O speed optimization 1: I/O speed optimization
15:9	Reserved	Must be kept at reset value.
8	CPS_RDY	Compensation cell ready flag This bit provides the status of the compensation cell.

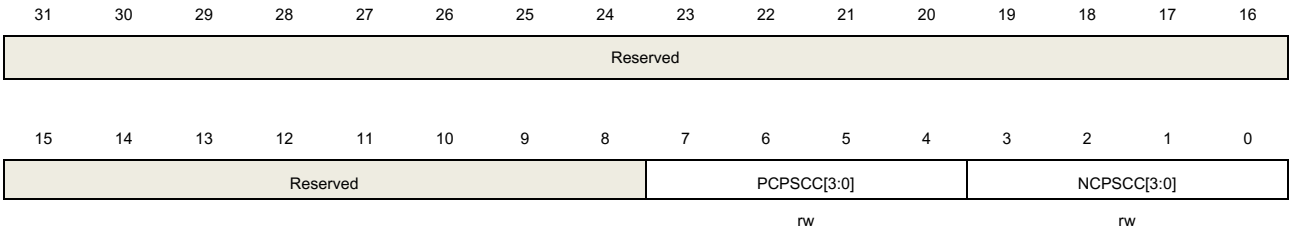
		0: I/O compensation cell not ready 1: I/O compensation cell ready
7:1	Reserved	Must be kept at reset value.
0	CPS_EN	I/O compensation cell enable This bit enables the I/O compensation cell. 0: I/O compensation cell disabled 1: I/O compensation cell enabled

1.8.9. I/O compensation cell code configuration register (SYSCFG_CPSCCCFG)

Address offset: 0x28

Reset value: 0x0000 0088

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



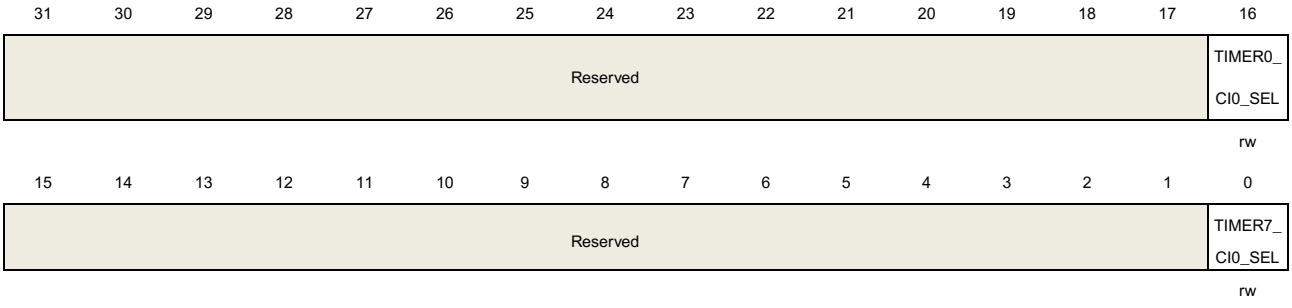
Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:4	PCPSCC[3:0]	PMOS compensation cell code These bits define I/O compensation cell code for PMOS transistors.
3:0	NCPSCC[3:0]	NMOS compensation cell code These bits define I/O compensation cell code for NMOS transistors.

1.8.10. Timer input selection register 0 (SYSCFG_TIMERCISEL0)

Address offset: 0x34

Reset value: 0x0000 0000

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



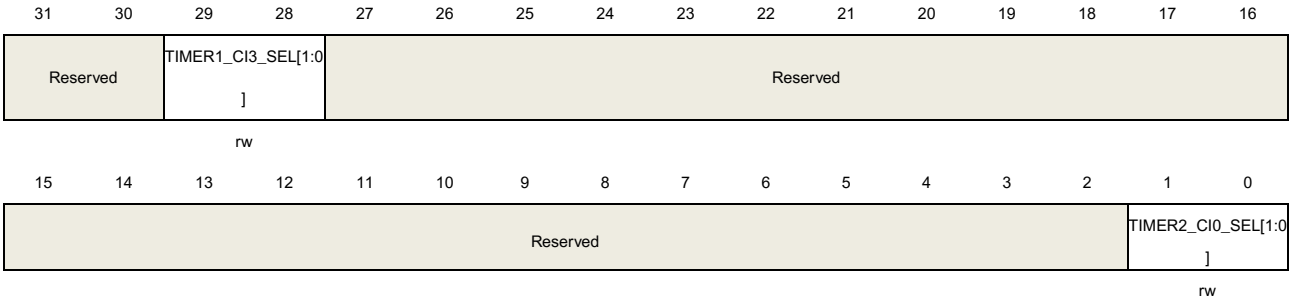
Bits	Fields	Descriptions
31:17	Reserved	Must be kept at reset value.
16	TIMER0_CI0_SEL	Selects TIMER0_CI0 input selection This bit selects the TIMER input source. 0: TIMER0_CH0 input 1: CMP0 output
15:1	Reserved	Must be kept at reset value.
0	TIMER7_CI0_SEL	Selects TIMER7_CI0 input selection This bit selects the TIMER input source. 0: TIMER7_CH0 input 1: CMP1 output

1.8.11. Timer input selection register 1 (SYSCFG_TIMERCISEL1)

Address offset: 0x38

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.
29:28	TIMER1_CI3_SEL[1:0]	TIMER1_CI3 input selection These bits select the TIMER input source. 00: TIMER1_CH3 input 01: CMP0 output 10: CMP1 output 11: CMP0 output or CMP1 output
27:2	Reserved	Must be kept at reset value.
1:0	TIMER2_CI0_SEL[1:0]	TIMER2_CI0 input selection These bits select the TIMER input source. 00: TIMER2_CH0 input 01: CMP0 output 10: CMP1 output

11: CMP0 output or CMP1 output

1.8.12. Timer input selection register 3 (SYSCFG_TIMERCISEL3)

Address offset: 0x40

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		TIMER22_CI3_SEL[1:0]		Reserved											
		rw													

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

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29:28	TIMER22_CI3_SEL[1:0]	TIMER22_CI3 input selection These bits select the TIMER input source. 00: TIMER22_CH3 input 01: CMP0 output 10: CMP1 output 11: CMP0 output or CMP1 output
27:0	Reserved	Must be kept at reset value.

1.8.13. Timer input selection register 4 (SYSCFG_TIMERCISEL4)

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														TIMER30_CI0_SEL[1:0]	
														rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														TIMER31_CI0_SEL[1:0]	
														rw	

Bits	Fields	Descriptions
31:18	Reserved	Must be kept at reset value.

17:16	TIMER30_CI0_SEL[1:0]	TIMER30_CI0 input selection These bits select the TIMER input source. 00: TIMER30_CH0 input 01: CMP0 output 10: CMP1 output 11: CMP0 output or CMP1 output
15:2	Reserved	Must be kept at reset value.
1:0	TIMER31_CI0_SEL[1:0]	TIMER31_CI0 input selection These bits select the TIMER input source. 00: TIMER31_CH0 input 01: CMP0 output 10: CMP1 output 11: CMP0 output or CMP1 output

1.8.14. Timer input selection register 5 (SYSCFG_TIMERCISEL5)

Address offset: 0x48

Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29:28	TIMER42_CI1_SEL[1:0]	Selects TIMER42_CI1 input These bits select the TIMER input source. 00: TIMER42_CH1 input 01: TIMER4_CH1 input 10: TIMER22_CH1 input 11: TIMER23_CH1 input
27	Reserved	Must be kept at reset value.
26:24	TIMER42_CI0_SEL[2:0]	Selects TIMER42_CI0 input These bits select the TIMER input source. 000: TIMER42_CH0 input 001: TIMER4_CH0 input

		010: TIMER22_CH0 input 011: TIMER23_CH0 input 100: LXTAL 101: LPIRC4M 110: CKOUT1
23:22	Reserved	Must be kept at reset value.
21:20	TIMER41_CI1_SEL[1:0]	Selects TIMER41_CI1 input These bits select the TIMER input source. 00: TIMER41_CH1 input 01: TIMER3_CH1 input 10: TIMER4_CH1 input 11: TIMER22_CH1 input
19	Reserved	Must be kept at reset value.
18:16	TIMER41_CI0_SEL[2:0]	Selects TIMER41_CI0 input These bits select the TIMER input source. 000: TIMER41_CH0 input 001: TIMER3_CH0 input 010: TIMER4_CH0 input 011: TIMER22_CH0 input 100: LXTAL 101: LPIRC4M 110: CKOUT1
15:14	Reserved	Must be kept at reset value.
13:12	TIMER40_CI1_SEL[1:0]	Selects TIMER40_CI1 input These bits select the TIMER input source. 00: TIMER40_CH1 input 01: TIMER2_CH1 input 10: TIMER3_CH1 input 11: TIMER4_CH1 input
11	Reserved	Must be kept at reset value.
10:8	TIMER40_CI0_SEL[2:0]	Selects TIMER40_CI0 input These bits select the TIMER input source. 000: TIMER40_CH0 input 001: TIMER2_CH0 input 010: TIMER3_CH0 input 011: TIMER4_CH0 input 100: LXTAL 101: LPIRC4M 110: CKOUT1

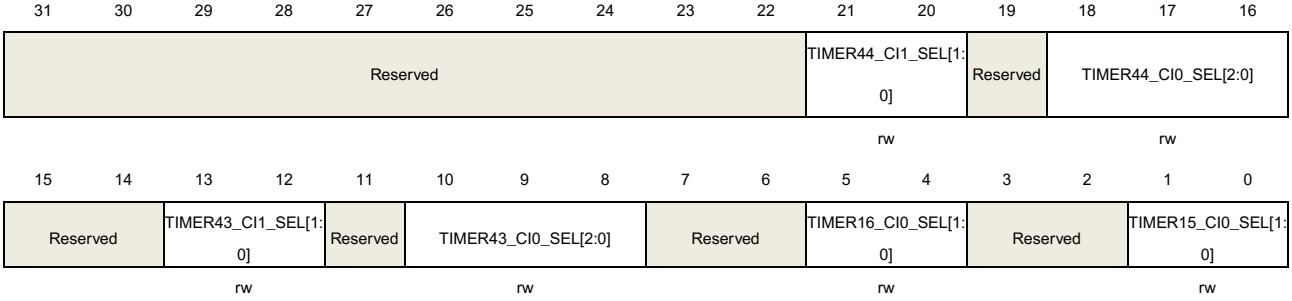
7:6	Reserved	Must be kept at reset value.
5:4	TIMER14_CI1_SEL[1:0]	Selects TIMER14_CI1 input These bits select the TIMER input source. 00: TIMER14_CH1 input 01: TIMER1_CH1 input 10: TIMER2_CH1 input 11: TIMER3_CH1 input
3	Reserved	Must be kept at reset value.
2:0	TIMER14_CI0_SEL[2:0]	Selects TIMER14_CI0 input These bits select the TIMER input source. 000: TIMER14_CH0 input 001: TIMER1_CH0 input 010: TIMER2_CH0 input 011: TIMER3_CH0 input 100: LXTAL 101: LPIRC4M 110: CKOUT1

1.8.15. Timer input selection register 6 (SYSCFG_TIMERCISEL6)

Address offset: 0x4C

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:22	Reserved	Must be kept at reset value.
21:20	TIMER44_CI1_SEL[1:0]	Selects TIMER44_CI1 input These bits select the TIMER input source. 00: TIMER44_CH1 input 01: TIMER23_CH1 input 10: TIMER30_CH1 input 11: TIMER31_CH1 input

19	Reserved	Must be kept at reset value.
18:16	TIMER44_CI0_SEL[2:0]	Selects TIMER44_CI0 input These bits select the TIMER input source. 000: TIMER44_CH0 input 001: TIMER23_CH0 input 010: TIMER30_CH0 input 011: TIMER31_CH0 input 100: LXTAL 101: LPIRC4M 110: CKOUT1
15:14	Reserved	Must be kept at reset value.
13:12	TIMER43_CI1_SEL[1:0]	Selects TIMER43_CI1 input These bits select the TIMER input source. 00: TIMER43_CH1 input 01: TIMER22_CH1 input 10: TIMER23_CH1 input 11: TIMER30_CH1 input
11	Reserved	Must be kept at reset value.
10:8	TIMER43_CI0_SEL[2:0]	Selects TIMER43_CI0 input These bits select the TIMER input source. 000: TIMER43_CH0 input 001: TIMER22_CH0 input 010: TIMER23_CH0 input 011: TIMER30_CH0 input 100: LXTAL 101: LPIRC4M 110: CKOUT1
7:6	Reserved	Must be kept at reset value.
5:4	TIMER16_CI0_SEL[1:0]	Selects TIMER16_CI0 input These bits select the TIMER input source. 00: TIMER16_CH0 input 01: RSPDIF FS 10: CK_HXTAL / RTCDIV 11: CKOUT0
3:2	Reserved	Must be kept at reset value.
1:0	TIMER15_CI0_SEL[1:0]	Selects TIMER15_CI0 input These bits select the TIMER input source. 00: TIMER15_CH0 input 01: IRC32K

10: LXTAL

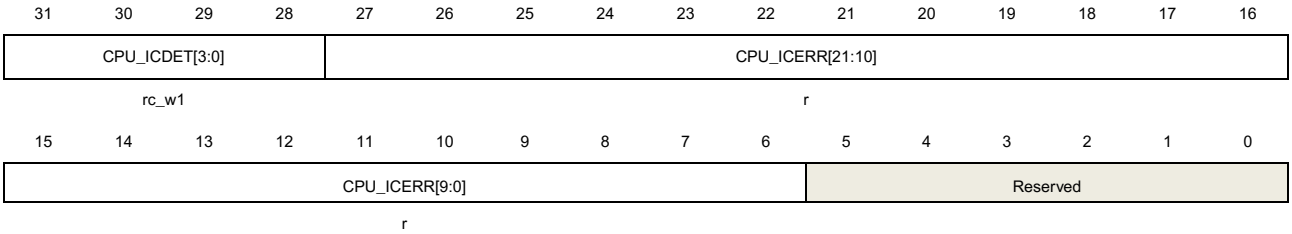
11: WKUP_IT

1.8.16. CPU ICACHE error status register(SYSCFG_CPUICAC)

Address offset: 0x54

Reset value: 0x0000 0000

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



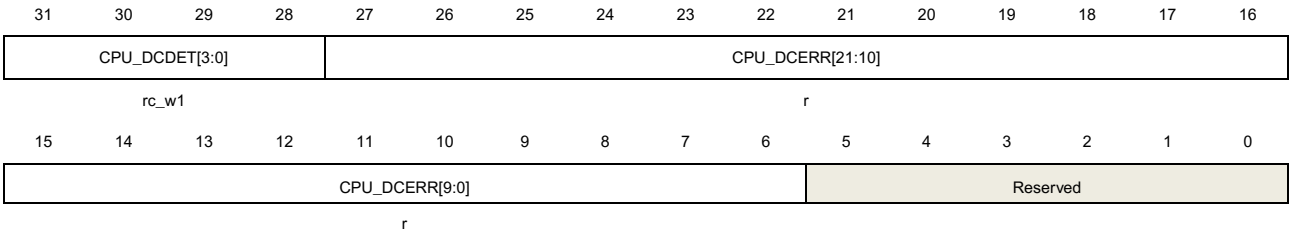
Bits	Fields	Descriptions
31:28	CPU_ICDET[3:0]	The ICACHE error detection information These bits are provided by the CPU to indicate the ICACHE error detection information. Cleared by writing 1.
27:6	CPU_ICERR[21:0]	The ICACHE error bank information These bits are provided by the CPU to indicate the ICACHE error bank information.
5:0	Reserved	Must be kept at reset value.

1.8.17. CPU DCACHE error status register (SYSCFG_CPUDCAC)

Address offset: 0x58

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:28	CPU_DCDET[3:0]	The DCACHE error detection information These bits are provided by the CPU to indicate the DCACHE error detection information. Cleared by writing 1.
27:6	CPU_DCERR[21:0]	The DCACHE error bank information

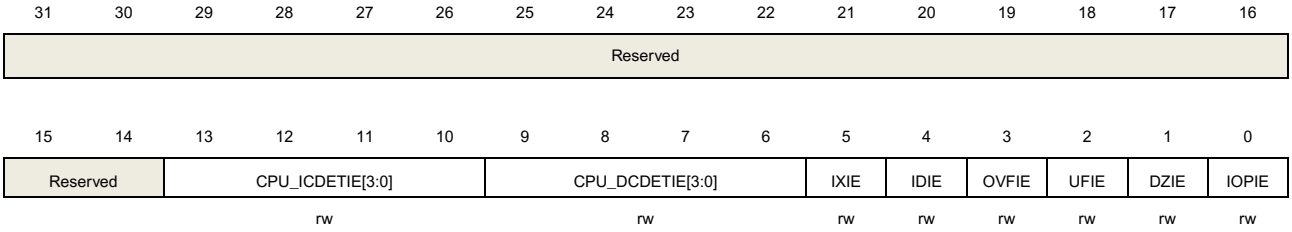
		These bits are provided by the CPU to indicate the DCACHE error bank information.
5:0	Reserved	Must be kept at reset value.

1.8.18. CACHE and FPU interrupt enable register (SYSCFG_CFPUINTEN)

Address offset: 0x5C

Reset value: 0x0000 001F

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



Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value.
13:10	CPU_ICDETIE[3:0]	icache error detection information int enable
9:6	CPU_DCDETIE[3:0]	dcache error bank information int enable
5	IXIE	Inexact interrupt enable bit 0: Inexact interrupt disable 1: Inexact interrupt enable
4	IDIE	Input denormal interrupt enable bit 0: Input denormal interrupt disable 1: Input denormal interrupt enable
3	OVFIE	Overflow interrupt enable bit 0: Overflow interrupt disable 1: Overflow interrupt enable
2	UFIE	Underflow interrupt enable bit 0: Underflow interrupt disable 1: Underflow interrupt enable
1	DZIE	Divide by 0 interrupt enable bit 0: Divide by 0 interrupt disable 1: Divide by 0 interrupt enable
0	IOPIE	Invalid operation interrupt enable bit 0: Invalid operation interrupt disable 1: Invalid operation interrupt enable

1.8.19. SRAM configuration register 0 (SYSCFG_SRAMCFG0)

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
Reserved														SECURE_SRAM_SIZE[1:0]	
														rw	

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1:0	SECURE_SRAM_SIZE[1:0]	Size of secure SRAM The size of SRAM (start address: 0x24000000) initialized to 0 after a system reset (non-power-on reset). These bits are set and cleared by software. 00: 0 Kbytes 01: 32 Kbytes 10: 64 Kbytes 11: 128 Kbytes

1.8.20. Bus timeout states register (SYSCFG_BUSTOSTAT)

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															ESCBUSTOF
															rc_w1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FFTBUSTOF	USBHS1BUSTOF	USBHS0BUSTOF	SDIO1BUSTOF	ENET1BUSTOF	ENET0BUSTOF	DMA1PERIPHBUSTOF	DMA1MEMBUSTOF	DMA0PERIPHBUSTOF	DMA0MEMBUSTOF	CPUAHBPBUSTOF	IPABUSTOF	MDMABUSTOF	TLIBUSTOF	CPUBUSTOF	SDIO0BUSTOF
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
31:17	Reserved	Must be kept at reset value.
16	ESCBUSTOF	ESC bus timeout flag bit. 0: Timeout is not occurred

		1: Timeout occurred
15	FFTBUSTOF	FFT bus timeout flag bit. 0: Timeout is not occurred 1: Timeout occurred
14	USBHS1BUSTOF	USBHS1 bus timeout flag bit. 0: Timeout is not occurred 1: Timeout occurred
13	USBHS0BUSTOF	USBHS0 bus timeout flag bit. 0: Timeout is not occurred 1: Timeout occurred
12	SDIO1BUSTOF	SDIO1 bus timeout flag bit. 0: Timeout is not occurred 1: Timeout occurred
11	ENET1BUSTOF	ENET1 bus timeout flag bit. 0: Timeout is not occurred 1: Timeout occurred
10	ENET0BUSTOF	ENET0 bus timeout enable flag bit. 0: Timeout is not occurred 1: Timeout occurred
9	DMA1PERIPHBUSTOF	DMA1_PERIPH bus timeout flag bit. 0: Timeout is not occurred 1: Timeout occurred
8	DMA1MEMBUSTOF	DMA1_MEM bus timeout flag bit. 0: Timeout is not occurred 1: Timeout occurred
7	DMA0PERIPHBUSTOF	DMA0_PERIPH bus timeout flag bit. 0: Timeout is not occurred 1: Timeout occurred
6	DMA0MEMBUSTOF	DMA0_MEM bus timeout flag bit. 0: Timeout is not occurred 1: Timeout occurred
5	CPUAHBPBUSTOF	CPU-AHBP bus timeout flag bit. 0: Timeout is not occurred 1: Timeout occurred
4	IPABUSTOF	IPA bus timeout flag bit. 0: Timeout is not occurred 1: Timeout occurred

3	MDMABUSTOF	MDMA bus timeout flag bit. 0: Timeout is not occurred 1: Timeout occurred
2	TLIBUSTOF	TLI bus timeout flag bit. 0: Timeout is not occurred 1: Timeout occurred
1	CPUBUSTOF	CPU bus timeout flag bit. 0: Timeout is not occurred 1: Timeout occurred
0	SDIO0BUSTOF	SDIO0 bus timeout flag bit. 0: Timeout is not occurred 1: Timeout occurred

1.8.21. ESC configuration register 0 (SYSCFG_ESCCFG0)

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
SMPS_DRIVE	RMIIEN	RMII_S0[2:0]			PHY_OFFSET[4:0]					PHY_OFFSETSEL	EEPROMSZ	SHIFTSTEN	NLINK_POLSEL[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
RSTOUT_POLSEL	RMII_S1[2:0]			LEDERR_POLSEL	LEDRUN_POLSEL	LINKACT_POLSEL[2:0]			Reserved		RMIITX_S1_VAL[1:0]		RMIITX_S0_VAL[1:0]		ESC_EEPROM
rw	rw			rw	rw	rw					rw		rw		rw

Bits	Fields	Descriptions
31	SMPS_DRIVE	SMPS_DRIVE's drive capability 0: Drive capability is normal 1: Drive capability is reduced
30	RMIIEN	RMII port enable bit. 0: using MII 1: using RMII
29:27	RMII_S0[2:0]	Clock compensation phase setting in RMII mode.
26:22	PHY_OFFSET[4:0]	PHY address offset.
21	PHY_OFFSETSEL	Whether to enable the PHY address offset function of ESC. 0: Offset starting from 0 1: Offset starting from 16
20	EEPROMSZ	Internal EEPROM capacity of ESC.

		0: 1Kbit~15Kbit 1: 16Kbit~4Mbit
19	SHIFTSTEN	Whether to enable the latch function of tx_shift. 0: Disable 1: Enable
18:16	NLINK_POLSEL[2:0]	NLINK signal input level polarity selection bits Three bits correspond to the NLINK signals of three phys. They are independent of each other and do not interfere with one another. 0: Select direct connection 1: Select inversion
15	RSTOUT_POLSEL	Reset_out signal output active polarity selection 0: Active high 1: Active low
14:12	RMII_S1[2:0]	Clock compensation phase setting in RMII mode.
11	LEDERR_POLSEL	led_err signal output active polarity selection 0: Do not invert polarity 1: Invert polarity
10	LEDRUN_POLSEL	led_run signal output active polarity selection 0: Do not invert polarity 1: Invert polarity
9:7	LINKACT_POLSEL[2:0]	link_act signal output active polarity selection Three bits correspond to the LINKACT signals of three phys. They are independent of each other and do not interfere with one another. 0: Do not invert polarity 1: Invert polarity
6:5	Reserved	Must be kept at reset value.
4:3	RMIITX_S1_VAL[1:0]	tx_shift value of RMII SHIFT1
2:1	RMIITX_S0_VAL[1:0]	tx_shift value of RMII SHIFT0
0	ESC_EEPROM	ESC EEPROM configuration 0: Utilize an external EEPROM by default 1: Utilize on-chip SRAM or Non-volatile memory as a substitute for EEPROM

1.8.22. ESC configuration register 1 (SYSCFG_ESCCFG1)

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

Bits	Fields	Descriptions
31:0	ESC_SRAM[31:0]	ESC SRAM configuration bits When the ESC_EEPROM bit is set to 1, this bit field is used to configure the base address of the corresponding SRAM/Non-volatile memory.

1.8.23. ESC configuration register 2 (SYSCFG_ESCCFG2)

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

Bits	Fields	Descriptions
31:0	RST_OUT_CNT[31:0]	The number of clock cycles for which reset_out is held. Clock cycle is 10ns.

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

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

TSCFG0[4:0]..TSCFG7[4:0], TSCFG9[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: The filtered output of channel 2 input (CI2FE2) 01010: The filtered output of channel 3 input (CI3FE3) 01011: The filtered output of multi mode channel 0 input (MCI0FEM0) 01100: The filtered output of multi mode channel 1 input (MCI1FEM1) 01101: The filtered output of multi mode channel 2 input (MCI2FEM2) 01110: The filtered output of multi mode channel 3 input (MCI3FEM3) 01111: Reserved 10000: Reserved 10001: Internal trigger input 12 (ITI12) 10010: Internal trigger input 13 (ITI13) 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: The filtered output of channel 2 input (CI2FE2) 01010: The filtered output of channel 3 input (CI3FE3) 01011: The filtered output of multi mode channel 0 input (MCI0FEM0) 01100: The filtered output of multi mode channel 1 input (MCI1FEM1)

		01101: The filtered output of multi mode channel 2 input (MCI2FEM2) 01110: The filtered output of multi mode channel 3 input (MCI3FEM3) 01111: Reserved 10000: Reserved 10001: Internal trigger input 12 (ITI12) 10010: Internal trigger input 13 (ITI13) 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: The filtered output of channel 2 input (CI2FE2) 01010: The filtered output of channel 3 input (CI3FE3) 01011: The filtered output of multi mode channel 0 input (MCI0FEM0) 01100: The filtered output of multi mode channel 1 input (MCI1FEM1) 01101: The filtered output of multi mode channel 2 input (MCI2FEM2) 01110: The filtered output of multi mode channel 3 input (MCI3FEM3) 01111: Reserved 10000: Reserved 10001: Internal trigger input 12 (ITI12) 10010: Internal trigger input 13 (ITI13) 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.

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

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

TSCFG0[4:0]..TSCFG7[4:0], TSCFG9[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						TSCFG7[4:0]					TSCFG6[4:0]				
						rw					rw				

Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30:26	TSCFG11[4:0]	Non-quadrature decoder mode 2 configuration 00000: disable Others: The counter will count on both rising and falling edges of CI0FE0 and CI1FE1 signals. And the counting direction determined by the CH0P and CH1P bits.
25:21	TSCFG10[4:0]	Non-quadrature decoder mode 1 configuration 00000: disable Others: The CI0FE0 is used as the count direction signal and the CI1FE1 signal is used as the count pulse. And the counter counts on the edge of CI1FE1, which is set by the CH1P.
20:16	TSCFG9[4:0]	Non-quadrature decoder mode 0 configuration 00000: Disable Others: The CI0FE0 is used as the count direction signal and the CI1FE1 signal is used as the count pulse. The counter will count on both rising and falling edges of the CI1FE1 signal.
15:10	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: The filtered output of external trigger input (ETIFP)
01001: The filtered output of channel 2 input (CI2FE2)
01010: The filtered output of channel 3 input (CI3FE3)
01011: The filtered output of multi mode channel 0 input (MCI0FEM0)
01100: The filtered output of multi mode channel 1 input (MCI1FEM1)
01101: The filtered output of multi mode channel 2 input (MCI2FEM2)
01110: The filtered output of multi mode channel 3 input (MCI3FEM3)
01111: Reserved
10000: Reserved
10001: Internal trigger input 12 (ITI12)
10010: Internal trigger input 13 (ITI13)
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: The filtered output of channel 2 input (CI2FE2) 01010: The filtered output of channel 3 input (CI3FE3) 01011: The filtered output of multi mode channel 0 input (MCI0FEM0) 01100: The filtered output of multi mode channel 1 input (MCI1FEM1) 01101: The filtered output of multi mode channel 2 input (MCI2FEM2) 01110: The filtered output of multi mode channel 3 input (MCI3FEM3) 01111: Reserved 10000: Reserved 10001: Internal trigger input 12 (ITI12)

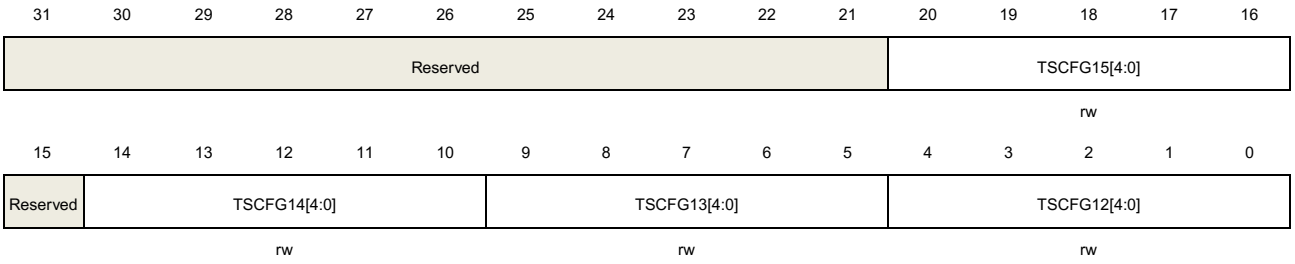
10010: Internal trigger input 13 (ITI13)
10011: Internal trigger input 14 (ITI14)
Others: Reserved

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

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

TSCFG0[4:0]..TSCFG7[4:0], TSCFG9[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).



Bits	Fields	Descriptions
31:21	Reserved	Must be kept at reset value.
20:16	TSCFG15[4:0]	Internal trigger input source configuration 00000: Internal trigger input 0 (ITI0) 00001: Internal trigger input 0 (ITI0) 00010: Internal trigger input 1 (ITI1) 00011: Internal trigger input 2 (ITI2) 00100: Internal trigger input 3 (ITI3) 00101: CI0 edge flag (CI0F_ED) 00110: Reserved 00111: Reserved 01000: Reserved 01001: Reserved 01010: Reserved 01011: Reserved 01100: Reserved 01101: Reserved 01110: Reserved 01111: Reserved 10000: Reserved 10001: Internal trigger input 12 (ITI12) 10010: Internal trigger input 13 (ITI13)

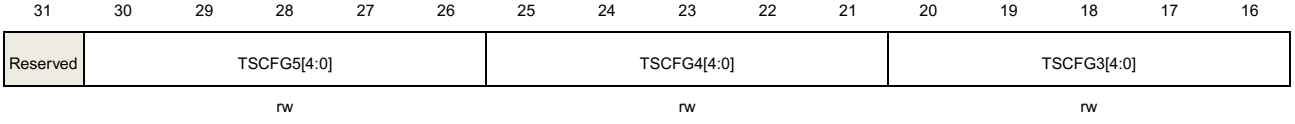
		10011: Internal trigger input 14 (ITI14) Others: Reserved **Note:** When TSCFG15[4:0] is used, TSCFGy[4:0](y=0..7, 9..14) should be zero, otherwise, the ITS trigger cooperate with TSCFGy[4:0], and input source depend on TSCFGy[4:0].
15	Reserved	Must be kept at reset value.
14:10	TSCFG14[4:0]	Quadrature decoder mode 4 configuration 00000: Disable Others: The counter counts on CI1FE1 edge only, while the direction depends on CI0FE0 level.
9:5	TSCFG13[4:0]	Quadrature decoder mode 3 configuration 00000: Disable Others: The counter counts on CI0FE0 edge only, while the direction depends on CI1FE1 level.
4:0	TSCFG12[4:0]	Non-quadrature decoder mode 3 configuration 00000: disable Others: the counter will count on rising or falling edge of CI0FE0 and CI1FE1 signals. When CHxP=0, the counter will counter on the high level or the falling edge or the CixFEx signal; When CHxP=1, the counter will counter on the low level or the rising edge or the CixFEx signal.

1.8.27. TIMERx configuration register 0 (SYSCFG_TIMERxCFG0, x=1, 2, 3, 4, 22, 23, 30, 31)

Address offset: 0x10C for TIMER1
Address offset: 0x118 for TIMER2
Address offset: 0x124 for TIMER3
Address offset: 0x130 for TIMER4
Address offset: 0x154 for TIMER22
Address offset: 0x160 for TIMER23
Address offset: 0x16C for TIMER30
Address offset: 0x178 for TIMER31
Reset value: 0x0000 0000

TSCFG0[4:0]..TSCFG7[4:0], TSCFG9[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).



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: Reserved 01100: Internal trigger input 7 (ITI7) 01101: Reserved 01110: Internal trigger input 9 (ITI9) 01111: Internal trigger input 10 (ITI10) 10000: Internal trigger input 11 (ITI11) 10001: Internal trigger input 12 (ITI12) 10010: Internal trigger input 13 (ITI13) 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: Reserved 01100: Internal trigger input 7 (ITI7) 01101: Reserved 01110: Internal trigger input 9 (ITI9) 01111: Internal trigger input 10 (ITI10) 10000: Internal trigger input 11 (ITI11) 10001: Internal trigger input 12 (ITI12) 10010: Internal trigger input 13 (ITI13) 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: Reserved 01100: Internal trigger input 7 (ITI7) 01101: Reserved 01110: Internal trigger input 9 (ITI9) 01111: Internal trigger input 10 (ITI10) 10000: Internal trigger input 11 (ITI11) 10001: Internal trigger input 12 (ITI12) 10010: Internal trigger input 13 (ITI13) 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.

1.8.28. TIMERx configuration register 1 (SYSCFG_TIMERxCFG1, x=1, 2, 3, 4, 22, 23, 30, 31)

Address offset: 0x110 for TIMER1
Address offset: 0x11C for TIMER2
Address offset: 0x128 for TIMER3
Address offset: 0x134 for TIMER4
Address offset: 0x158 for TIMER22
Address offset: 0x164 for TIMER23
Address offset: 0x170 for TIMER30
Address offset: 0x17C for TIMER31
Reset value: 0x0000 0000

TSCFG0[4:0]..TSCFG7[4:0], TSCFG9[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).



Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30:26	TSCFG11[4:0]	Non-quadrature decoder mode 2 configuration 00000: disable Others: The counter will count on both rising and falling edges of CI0FE0 and CI1FE1 signals. And the counting direction determined by the CH0P and CH1P bits.
25:21	TSCFG10[4:0]	Non-quadrature decoder mode 1 configuration 00000: disable Others: The CI0FE0 is used as the count direction signal and the CI1FE1 signal is used as the count pulse. And the counter counts on the edge of CI1FE1, which is

		set by the CH1P.
20:16	TSCFG9[4:0]	Non-quadrature decoder mode 0 configuration 00000: Disable Others: The CI0FE0 is used as the count direction signal and the CI1FE1 signal is used as the count pulse. The counter will count on both rising and falling edges of the CI1FE1 signal.
15:10	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: The filtered output of external trigger input (ETIFP) 01001: Internal trigger input 4 (ITI4) 01010: Internal trigger input 5 (ITI5) 01011: Reserved 01100: Internal trigger input 7 (ITI7) 01101: Reserved 01110: Internal trigger input 9 (ITI9) 01111: Internal trigger input 10 (ITI10) 10000: Internal trigger input 11 (ITI11) 10001: Internal trigger input 12 (ITI12) 10010: Internal trigger input 13 (ITI13) 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: Reserved
01100: Internal trigger input 7 (ITI7)
01101: Reserved
01110: Internal trigger input 9 (ITI9)
01111: Internal trigger input 10 (ITI10)
10000: Internal trigger input 11 (ITI11)
10001: Internal trigger input 12 (ITI12)
10010: Internal trigger input 13 (ITI13)
10011: Internal trigger input 14 (ITI14)
Others: Reserved

1.8.29. TIMERx configuration register 2 (SYSCFG_TIMERxCFG2, x=1, 2, 3, 4, 22, 23, 30, 31)

Address offset: 0x114 for TIMER1
Address offset: 0x120 for TIMER2
Address offset: 0x12C for TIMER3
Address offset: 0x138 for TIMER4
Address offset: 0x15C for TIMER22
Address offset: 0x168 for TIMER23
Address offset: 0x174 for TIMER30
Address offset: 0x180 for TIMER31
Reset value: 0x0000 0000

TSCFG0[4:0]..TSCFG7[4:0], TSCFG9[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]				
	rw					rw					rw				

Bits	Fields	Descriptions
31:21	Reserved	Must be kept at reset value.
20:16	TSCFG15[4:0]	Internal trigger input source configuration 00000: Internal trigger input 0 (ITI0) 00001: Internal trigger input 0 (ITI0)

		00010: Internal trigger input 1 (ITI1) 00011: Internal trigger input 2 (ITI2) 00100: Internal trigger input 3 (ITI3) 00101: CI0 edge flag (CI0F_ED) 00110: Reserved 00111: Reserved 01000: Reserved 01001: Internal trigger input 4 (ITI4) 01010: Internal trigger input 5 (ITI5) 01011: Reserved 01100: Internal trigger input 7 (ITI7) 01101: Reserved 01110: Internal trigger input 9 (ITI9) 01111: Internal trigger input 10 (ITI10) 10000: Internal trigger input 11 (ITI11) 10001: Internal trigger input 12 (ITI12) 10010: Internal trigger input 13 (ITI13) 10011: Internal trigger input 14 (ITI14) Others: Reserved **Note:** When TSCFG15[4:0] is used, TSCFGy[4:0](y=0..7, 9..14) should be zero, otherwise, the ITS trigger cooperate with TSCFGy[4:0], and input source depend on TSCFGy[4:0].
15	Reserved	Must be kept at reset value.
14:10	TSCFG14[4:0]	Quadrature decoder mode 4 configuration 00000: Disable Others: The counter counts on CI1FE1 edge only, while the direction depends on CI0FE0 level.
9:5	TSCFG13[4:0]	Quadrature decoder mode 3 configuration 00000: Disable Others: The counter counts on CI0FE0 edge only, while the direction depends on CI1FE1 level.
4:0	TSCFG12[4:0]	Non-quadrature decoder mode 3 configuration 00000: disable Others: the counter will count on rising or falling edge of CI0FE0 and CI1FE1 signals. When CHxP=0, the counter will counter on the high level or the falling edge or the CixFEx signal; When CHxP=1, the counter will counter on the low level or the rising edge or the CixFEx signal.

1.8.30. TIMERx configuration register 0 (SYSCFG_TIMERxCFG0, x=14, 40, 41, 42, 43, 44)

Address offset: 0x148 for TIMER14
Address offset: 0x184 for TIMER40
Address offset: 0x190 for TIMER41
Address offset: 0x19C for TIMER42
Address offset: 0x1A8 for TIMER43
Address offset: 0x1B4 for TIMER44
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	TSCFG5[4:0]					TSCFG4[4:0]					TSCFG3[4:0]				
	rw					rw					rw				

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

Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30:26	TSCFG5[4:0]	Event mode configuration A rising edge of the trigger input enables the counter. 00000: Event mode disable 00001: Internal trigger input 0 (ITI0) 00010: Internal trigger input 1 (ITI1) 00011: Internal trigger input 2 (ITI2) 00100: Internal trigger input 3 (ITI3) 00101: CI0 edge flag (CI0F_ED) 00110: The filtered output of channel 0 input (CI0FE0) 00111: The filtered output of channel 1 input (CI1FE1) 01000: Reserved 01001: Reserved 01010: Reserved 01011: The filtered output of multi mode channel 0 input (MCI0FEM0) 10011: Internal trigger input 14 (ITI14) Others: Reserved
25:21	TSCFG4[4:0]	Pause mode configuration The trigger input enables the counter clock when it is high and disables the counter

		when it is low when these bits are not 0. 00000: Pause mode disable 00001: Internal trigger input 0 (ITI0) 00010: Internal trigger input 1 (ITI1) 00011: Internal trigger input 2 (ITI2) 00100: Internal trigger input 3 (ITI3) 00101: Reserved 00110: The filtered output of channel 0 input (CI0FE0) 00111: The filtered output of channel 1 input (CI1FE1) 01000: Reserved 01001: Reserved 01010: Reserved 01011: The filtered output of multi mode channel 0 input (MCI0FEM0) 10011: Internal trigger input 14 (ITI14) Others: Reserved
20:16	TSCFG3[4:0]	Restart mode configuration The counter is reinitialized and the shadow registers are updated on the rising edge of the selected trigger input when these bits are not 0. 00000: Restart mode disable 00001: Internal trigger input 0 (ITI0) 00010: Internal trigger input 1 (ITI1) 00011: Internal trigger input 2 (ITI2) 00100: Internal trigger input 3 (ITI3) 00101: CI0 edge flag (CI0F_ED) 00110: The filtered output of channel 0 input (CI0FE0) 00111: The filtered output of channel 1 input (CI1FE1) 01000: Reserved 01001: Reserved 01010: Reserved 01011: The filtered output of multi mode channel 0 input (MCI0FEM0) 10011: Internal trigger input 14 (ITI14) Others: Reserved
15:0	Reserved	Must be kept at reset value.

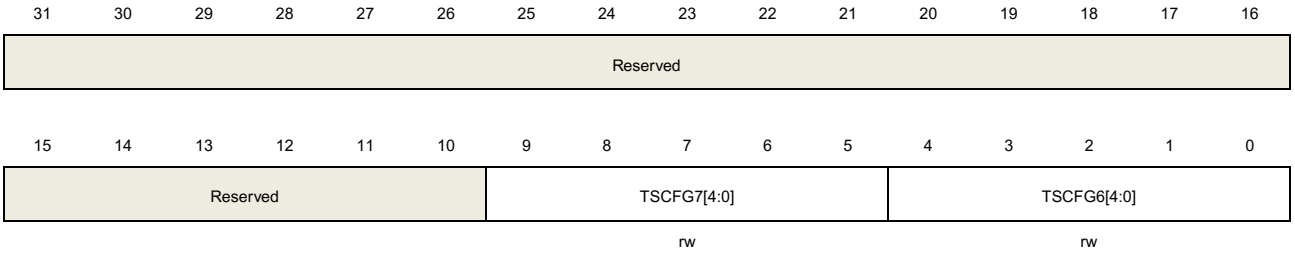
1.8.31. TIMERx configuration register 1 (SYSCFG_TIMERxCFG1, x=14, 40, 41, 42, 43, 44)

Address offset: 0x14C for TIMER14
Address offset: 0x188 for TIMER40
Address offset: 0x194 for TIMER41
Address offset: 0x1A0 for TIMER42
Address offset: 0x1AC for TIMER43

Address offset: 0x1B8 for TIMER44
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: Reserved 01010: Reserved 01011: The filtered output of multi mode channel 0 input (MCI0FEM0) 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: Reserved
01010: Reserved
01011: The filtered output of multi mode channel 0 input (MCI0FEM0)
10011: Internal trigger input 14 (ITI14)
Others: Reserved

1.8.32. TIMERx configuration register 2 (SYSCFG_TIMERxCFG2, x=14, 40, 41, 42, 43, 44)

Address offset: 0x150 for TIMER14
Address offset: 0x18C for TIMER40
Address offset: 0x198 for TIMER41
Address offset: 0x1A4 for TIMER42
Address offset: 0x1B0 for TIMER43
Address offset: 0x1BC for TIMER44
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: Internal trigger input 0 (ITI0) 00001: Internal trigger input 0 (ITI0) 00010: Internal trigger input 1 (ITI1) 00011: Internal trigger input 2 (ITI2) 00100: Internal trigger input 3 (ITI3) 00101: CI0 edge flag (CI0F_ED) 00110: Reserved 00111: Reserved 01000: Reserved 01001: Reserved

01010: Reserved
01011: Reserved
10011: Internal trigger input 14 (ITI14)
Others: Reserved
Note: When TSCFG15[4:0] is used, TSCFGy[4:0](y = 3..7) should be zero, otherwise, the ITS trigger cooperate with TSCFGy[4:0], and input source depend on TSCFGy[4:0].

15:0	Reserved	Must be kept at reset value.

1.8.33. Peripheral mode configuration register 1 (SYSCFG_PMCFG1)

Address offset: 0x204
Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										CK_EXMC_DIS	IO_SEL_G1[2:0]			VOSCFGEN	STBOEN
										rw	rw			rw	rw

Bits	Fields	Descriptions
31:6	Reserved	Must be kept at reset value.
5	CK_EXMC_DIS	Disable CK_EXMC 0: Enable CK_EXMC 1: Disable CK_EXMC
4:2	IO_SEL_G1[2:0]	Coder group 1 IO select 000: Connect to AFMT IO 001: Connect to TFMT IO 010: Connect to Endat IO 011: Connect to BISS-C IO 100: Not connected 101: Connect to HDSL Others: Not connected
1	VOSCFGEN	VOS configuration enable 0: VOS not configurable 1: VOS configurable
0	STBOEN	Standby flag output from PC13 enable **Note:** This bit is mainly used in SMPS bypass mode for external detection of whether the chip is in standby mode. When this bit is set to 1, the PMU

COREPOROFFP bit must also be set to 1.

0: Disable standby flag output from PC13

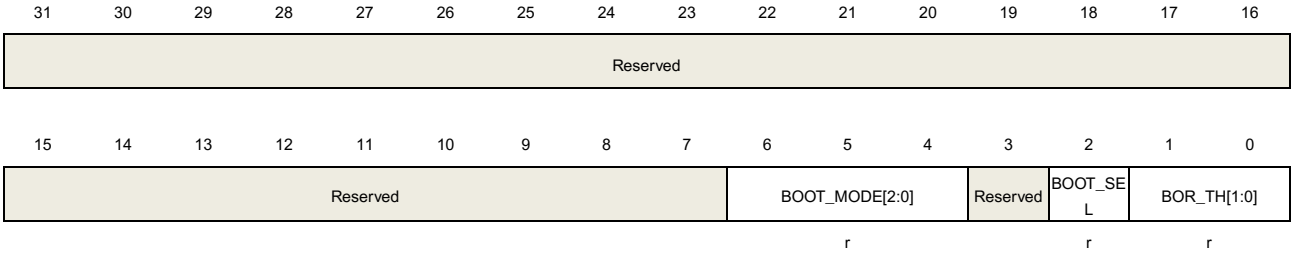
1: Enable standby flag output from PC13

1.8.34. User configuration register (SYSCFG_USERCFG)

Address offset: 0x300

Reset value: 0x0000 00XX

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



Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value.
6:4	BOOT_MODE[2:0]	Boot mode These bits indicate the mode of BOOT. 000: BOOT from SRAM (ITCM/DTCM/RAM shared/AXI SRAM) 001: BOOT from Security 010: BOOT_SYS (BootLoader) 011: BOOT_USER (NVMC/ OSPI0/1) Others: Reserved
3	Reserved	Must be kept at reset value.
2	BOOT_SEL	BOOT selection 0: Option byte selects BOOT_ADDR0 1: Option byte selects BOOT_ADDR1
1:0	BOR_TH[1:0]	BOR threshold status bits 00: No BOR function 01: BOR threshold value 1 10: BOR threshold value 2 11: BOR threshold value 3

1.9. AXI interconnect registers

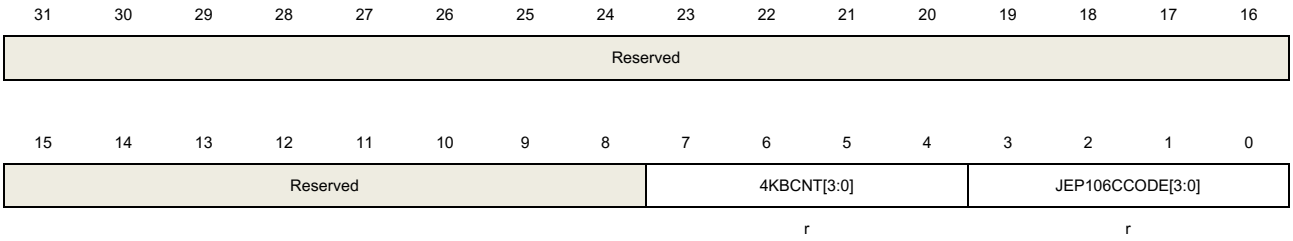
AXI base address: 0x5100 0000

1.9.1. AXI peripheral ID4 register (AXI_PERIPH_ID4)

Address offset: 0x1FD0

Reset value: 0x0000 0004

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



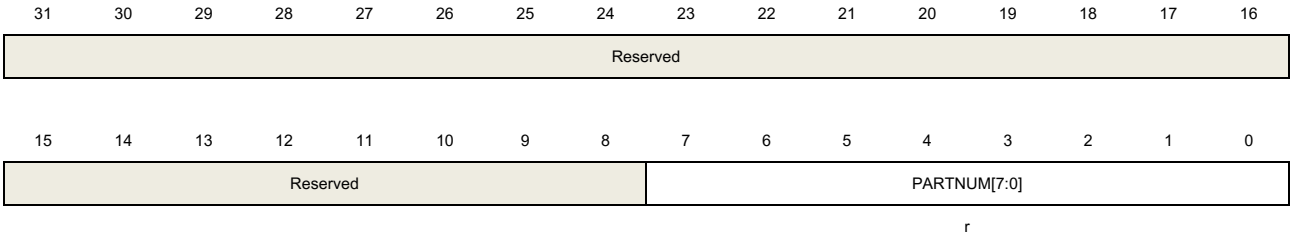
Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:4	4KBCNT[3:0]	4KB count
3:0	JEP106CCODE[3:0]	JEP106 continuation code

1.9.2. AXI peripheral ID0 register (AXI_PERIPH_ID0)

Address offset: 0x1FE0

Reset value: 0x0000 0000

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



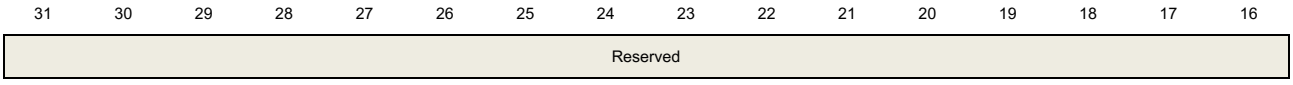
Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	PARTNUM[7:0]	Part number[7:0]

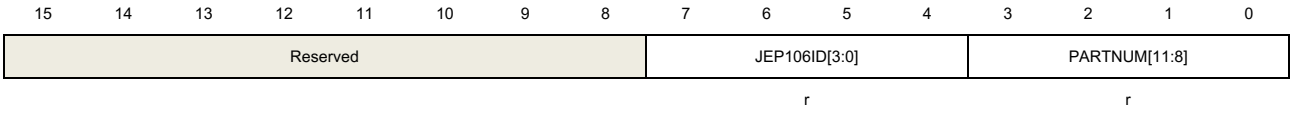
1.9.3. AXI peripheral ID1 register (AXI_PERIPH_ID1)

Address offset: 0x1FE4

Reset value: 0x0000 00B4

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





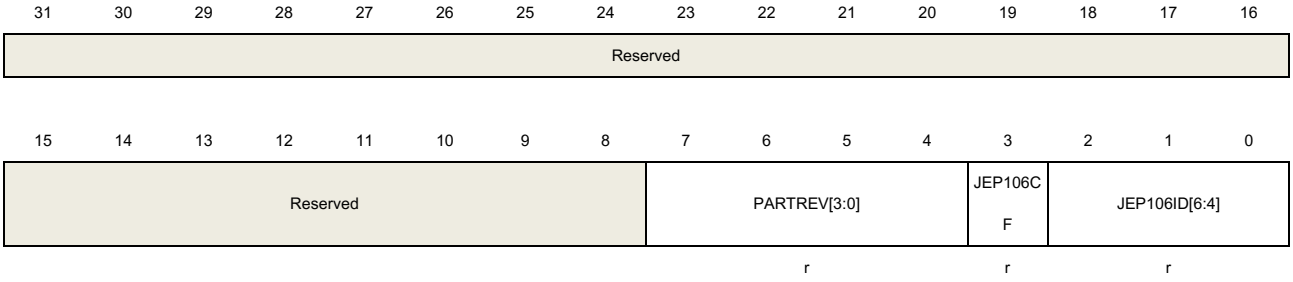
Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:4	JEP106ID[3:0]	JEP106 Identity[3:0]
3:0	PARTNUM[11:8]	Part number[11:8]

1.9.4. AXI peripheral ID2 register (AXI_PERIPH_ID2)

Address offset: 0x1FE8

Reset value: 0x0000 006B

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



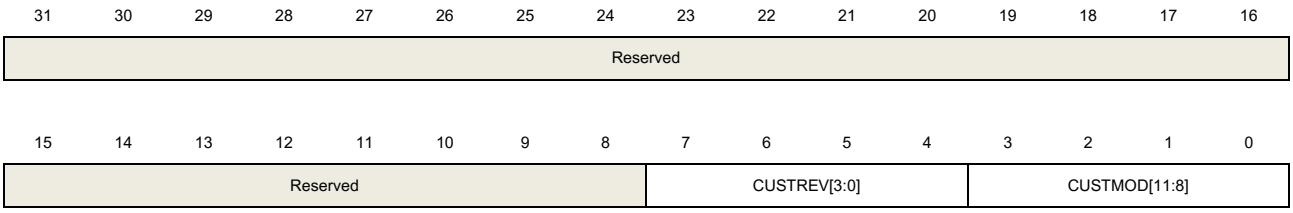
Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:4	PARTREV[3:0]	Part revision
3	JEP106CF	JEP106 code flag
2:0	JEP106ID[6:4]	Part number[6:4]

1.9.5. AXI peripheral ID3 register (AXI_PERIPH_ID3)

Address offset: 0x1FEC

Reset value: 0x0000 0000

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



r r

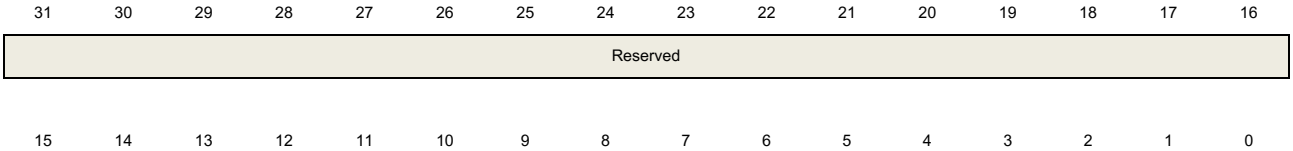
Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:4	CUSTREV[3:0]	Customer version
3:0	CUSTMOD[11:8]	Customer modification

1.9.6. AXI component ID0 register (AXI_COMP_ID0)

Address offset: 0x1FF0

Reset value: 0x0000 000D

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



r

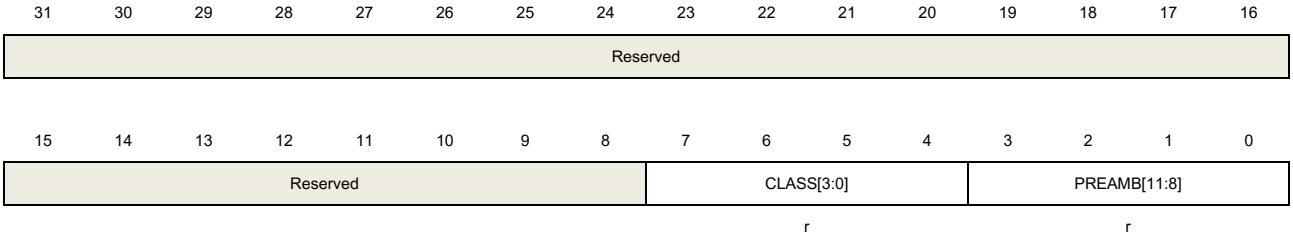
Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	PREAMB[7:0]	Preamble bits

1.9.7. AXI component ID1 register (AXI_COMP_ID1)

Address offset: 0x1FF4

Reset value: 0x0000 00F0

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



r r

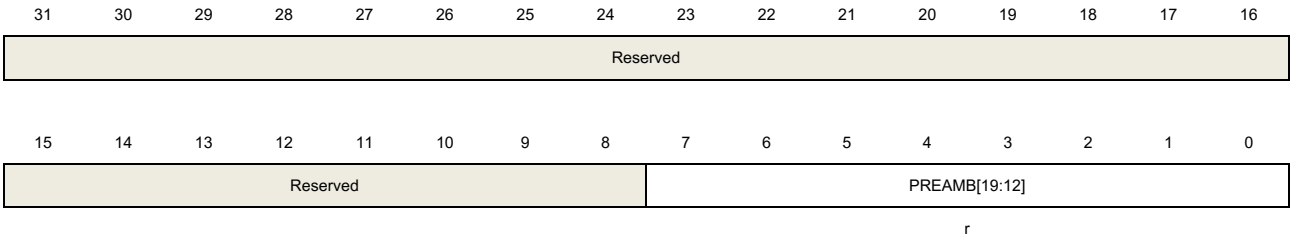
Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:4	CLASS[3:0]	Component class
3:0	PREAMB[11:8]	Preamble bits

1.9.8. AXI component ID2 register (AXI_COMP_ID2)

Address offset: 0x1FF8

Reset value: 0x0000 0005

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



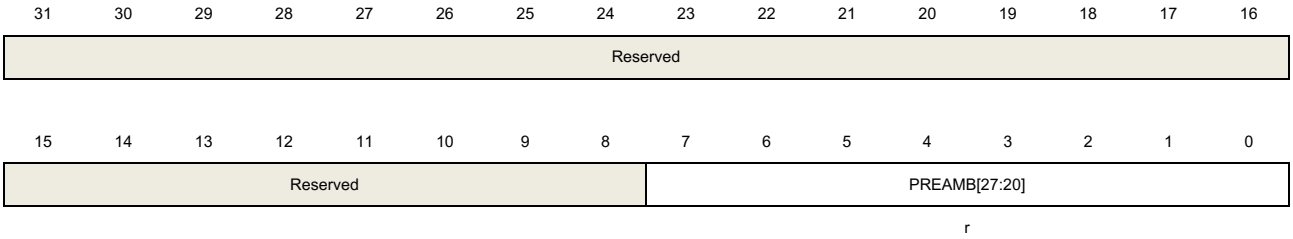
Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	PREAMB[19:12]	Preamble bits

1.9.9. AXI component ID3 register (AXI_COMP_ID3)

Address offset: 0x1FFC

Reset value: 0x0000 00B1

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



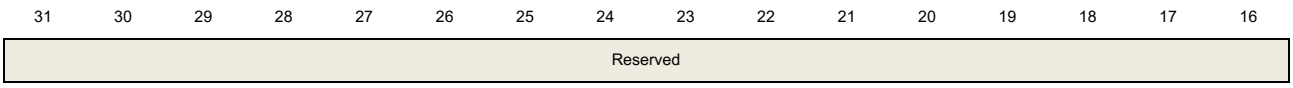
Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	PREAMB[27:20]	Preamble bits

1.9.10. AXI Master Port x bus matrix issuing functionality control register (AXI_MPxBM_ISS_CTL)

Address offset: 0x2008 + 0x1000 * x, where x = 0 to 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
Reserved														WR_ISSOV	RD_ISSOV
														rw	rw

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	WR_ISSOV	Override target write issuing function 0: Normal issuing function 1: The bus matrix write issuing capability is set to 1
0	RD_ISSOV	Override target read issuing function 0: Normal issuing function 1: The bus matrix read issuing capability is set to 1

1.9.11. AXI Master Port x bus matrix functionality control register (AXI_MPxBM_CTL)

Address offset: 0x2024 + 0x1000 * x, where x = 0, 1
Reset value: 0x0000 0000

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

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

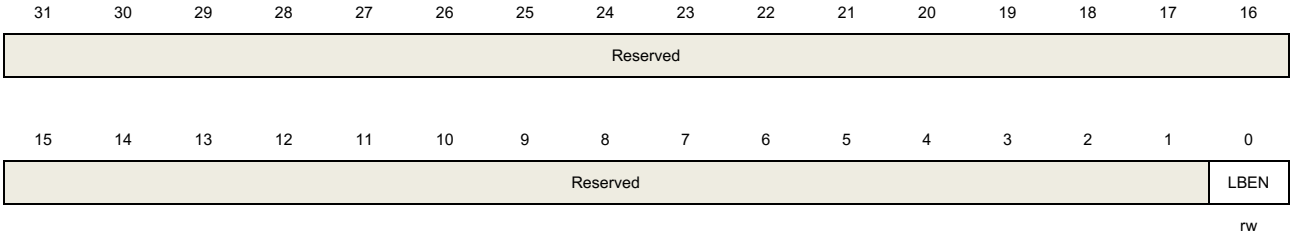
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															BPDIS
															rw

Bits	Fields	Descriptions
31:1	Reserved	Must be kept at reset value.
0	BPDIS	Beats packing function disable configure 0: Normal operation 1: Disable beats packing function

1.9.12. AXI Master Port x long burst functionality control register (AXI_MPx_LB_CTL)

Address offset: 0x202C + 0x1000 * x, where x = 0 and 1
Reset value: 0x0000 0000

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

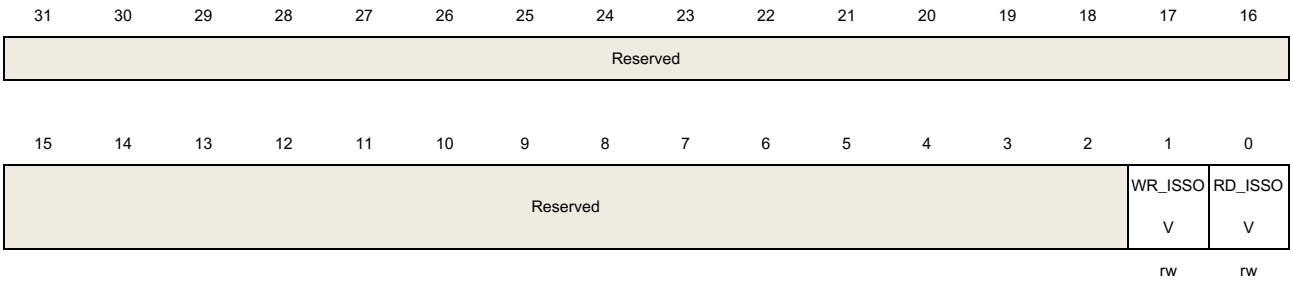


Bits	Fields	Descriptions
31:1	Reserved	Must be kept at reset value.
0	LBEN	Control long burst function 0: Disable long burst generated at the output of the ASIB 1: Enable long burst generated at the output of the ASIB

1.9.13. AXI Master Port x issuing functionality control register (AXI_MPx_ISS_CTL)

Address offset: 0x2108 + 0x1000 * x, where x = 0 to 7
Reset value: 0x0000 0000

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



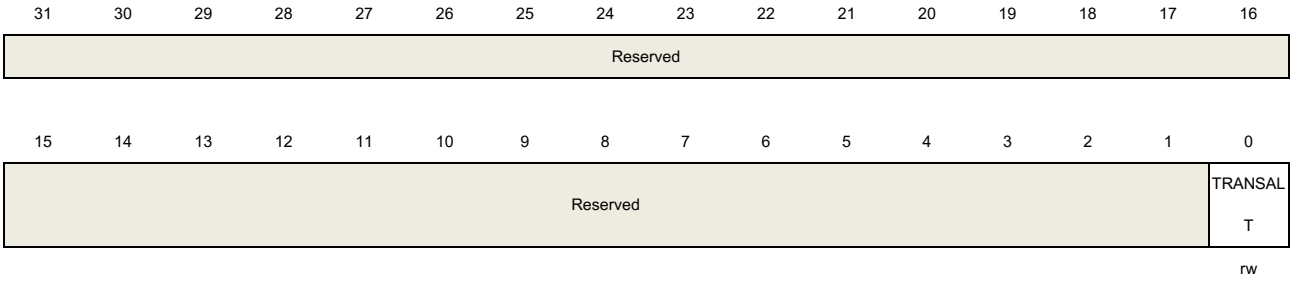
Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	WR_ISSOV	Override AMIB write issuing function 0: Normal issuing function 1: The AMIB write issuing capability is forced set to 1
0	RD_ISSOV	Override AMIB read issuing function 0: Normal issuing function 1: The AMIB read issuing capability is forced set to 1

1.9.14. AXI Slave Port x functionality control register (AXI_SPx_CTL)

Address offset: 0x42024 + 0x1000 * x, where x = 0 and 2

Reset value: 0x0000 0000

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

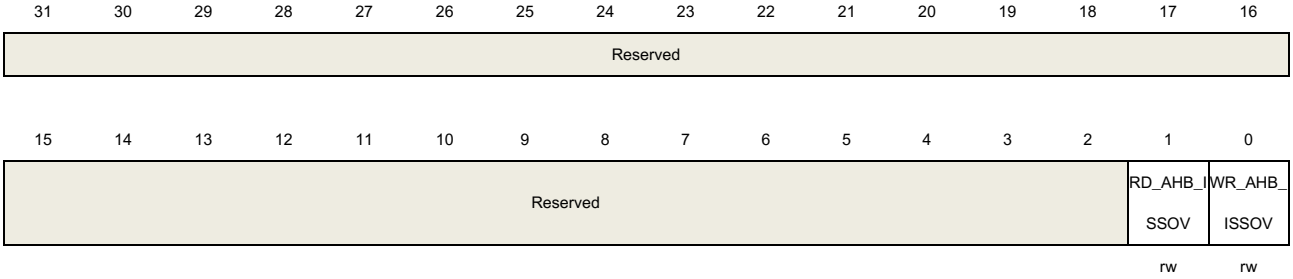


Bits	Fields	Descriptions
31:1	Reserved	Must be kept at reset value.
0	TRANSALT	Transaction alteration configure 0: Normal operation 1: Transaction unaltered where allowed

1.9.15. AXI Slave Port x AHB issuing functionality control register (AXI_SPx_AHBISS_CTL)

Address offset: 0x42028 + 0x1000 * x, where x = 0 and 2
Reset value: 0x0000 0000

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



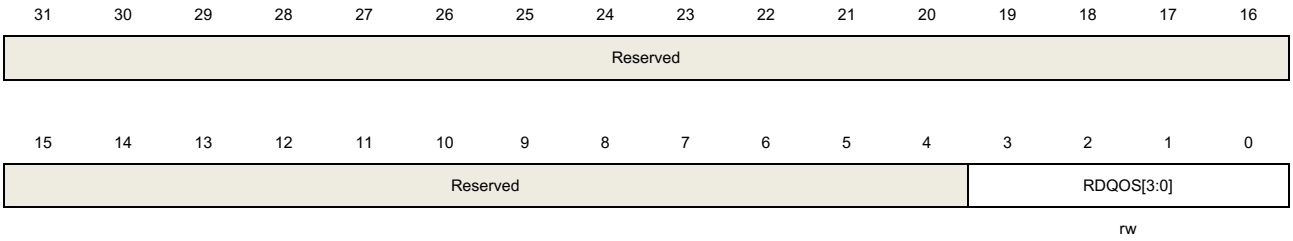
Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	RD_AHB_ISSOV	Converts AHB-Lite read transaction to single beat AXI transaction function 0: Override disabled 1: Override enabled
0	WR_AHB_ISSOV	Converts AHB-Lite write transaction to single beat AXI transaction function 0: Override disabled 1: Override enabled

1.9.16. AXI Slave Port x read QOS control register (AXI_SPx_RDQOS_CTL)

Address offset: 0x42100 + 0x1000 * x, where x = 0 to 5

Reset value: 0x0000 0000

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



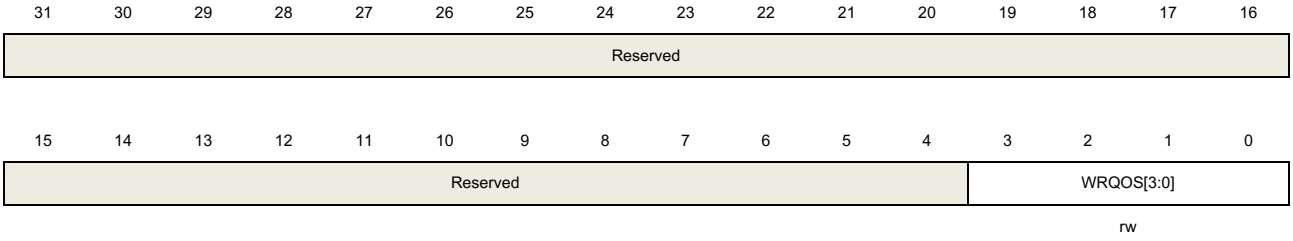
Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value.
3:0	RDQOS[3:0]	Read channel QoS configure 0000: Lowest priority … 1111: Highest priority

1.9.17. AXI Slave Port x write QOS control register (AXI_SPx_WRQOS_CTL)

Address offset: 0x42104 + 0x1000 * x, where x = 0 to 5

Reset value: 0x0000 0000

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



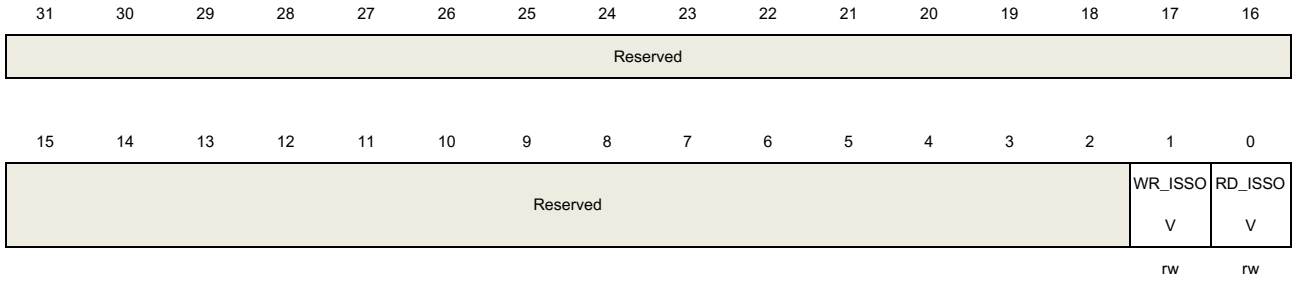
Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value.
3:0	WRQOS[3:0]	Write channel QoS configure 0000: Lowest priority … 1111: Highest priority

1.9.18. AXI Slave Port x issuing functionality control register (AXI_SPx_ISS_CTL)

Address offset: 0x42108 + 0x1000 * x, where x = 0 to 5

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	WR_ISSOV	Override ASIB write issuing function 0: Normal issuing function 1: The ASIB write issuing capability is forced set to 1
0	RD_ISSOV	Override ASIB read issuing function 0: Normal issuing function 1: The ASIB read issuing capability is forced set to 1

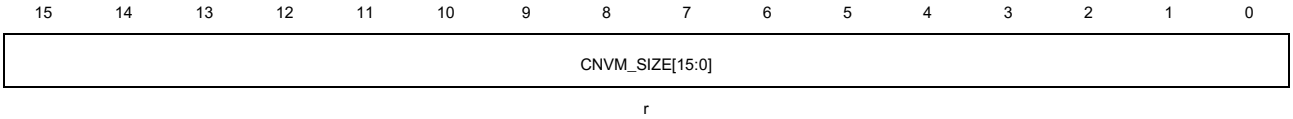
1.10. Device electronic signature

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

1.10.1. CNVM size information

Base address: 0x1FF0 F7DE

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

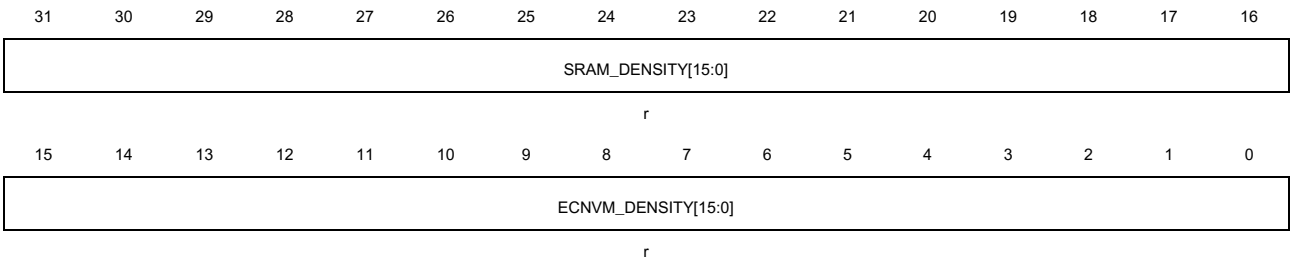


Bits	Fields	Descriptions
15:0	CNVM_SIZE[15:0]	CNVM size The value indicates the CNVM size of the device in Kbytes. Example: 0x0008 indicates 8 Kbytes.

1.10.2. Memory density information

Base address: 0x1FF0 F7E0

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



Bits	Fields	Descriptions
31:16	SRAM_DENSITY [15:0]	SRAM density The value indicates the on-chip SRAM density of the device in Kbytes. Example: 0x0008 indicates 8 Kbytes.
15:0	ECNVM_DENSITY[15:0]	ECNVM density The value indicates the Extended Code non-volatile memory density of the device in Kbytes. Example: 0x0020 indicates 32 Kbytes.

1.10.3. Unique device ID (96 bits)

Base address: 0x1FF0 F7E8

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

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

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

Base address: 0x1FF0 F7EC

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UNIQUE_ID[63:48]															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNIQUE_ID[47:32]															
r															

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

Base address: 0x1FF0 F7F0

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UNIQUE_ID[95:80]															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNIQUE_ID[79:64]															
r															

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

2. Hardware semaphore (HWSEM)

2.1. Overview

Hardware semaphore (HWSEM) provides a non-blocking mechanism to ensure the synchronization of processes. HWSEM realizes 32 semaphores in an atomic way, supporting semaphore write lock and read lock, and semaphore can only be unlocked when bus master and process are matched.

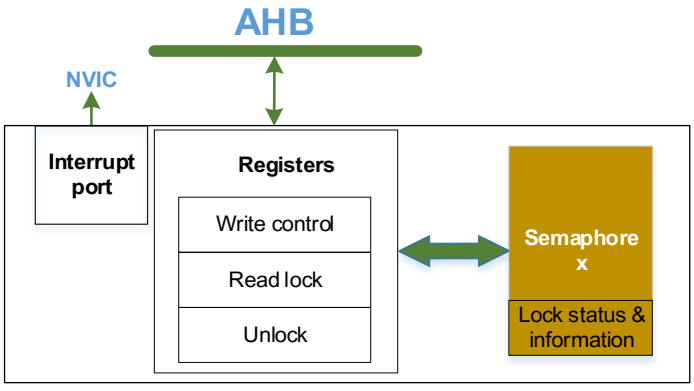
2.2. Characteristics

- 32 semaphores.
- An interrupt is generated when a semaphore is unlocked.
- Semaphore is unlocked only when MID[3:0] and PID[7:0] are matched.
- Indirect mode supports DMA operation locking indication.

2.3. Function overview

2.3.1. Block diagram

Figure 2-1. HWSEM diagram



As shown in *Figure 2-1. HWSEM diagram*, HWSEM includes three sub-blocks:

- Semaphore x sub-block, including semaphore lock status and lock informations.
- Registers, used for accessing the semaphores.
- Interrupt port for interrupt control and status.

2.3.2. Semaphore x

Lock status

LK bit in HWSEM_CTLx (x = 0…31) register shows the lock status of semaphore x.

When LK bit is 0, the semaphore is unlocked.

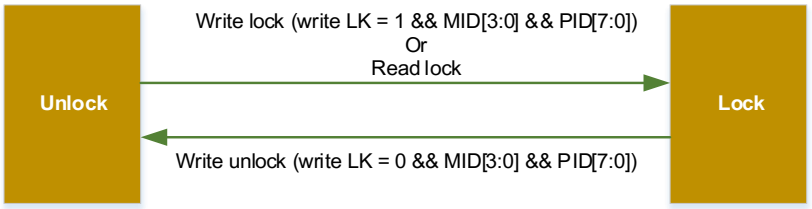
When LK bit is 1, the semaphore is locked.

Lock information

When the semaphore is unlocked, MID[3:0] and PID[7:0] bit fields in HWSEM_CTLx (x = 0…31) register are 0.

When the semaphore is locked, MID[3:0] bit field in HWSEM_CTLx (x = 0…31) / HWSEM_RLKx (x = 0…31) register shows which AHB bus master locked it, and PID[7:0] bit field in HWSEM_CTLx (x = 0…31) / HWSEM_RLKx (x = 0…31) register shows which process of that AHB bus master locked it.

Figure 2-2. HWSEM state machine



2.3.3. Lock a semaphore

HWSEM supports two ways for semaphore locking:

- Write lock.
- Read lock.

Note: Only MID[3:0] which is matched the ID can lock and unlock a semaphore. The bus master ID of CPU/MDMA, DMA0, DMA1 are respectively 4'b1011, 4'b0010, and 4'b0100.

Write lock

When a semaphore is locked by write lock, MID[3:0] bit field in HWSEM_CTLx (x = 0…31) / HWSEM_RLKx (x = 0…31) register is the AHB bus master which locked it, and PID[7:0] bit field in HWSEM_CTLx (x = 0…31) / HWSEM_RLKx (x = 0…31) register is the configured process identification.

Each process of AHB bus master must owns an unique PID[7:0] value.

Write lock operation follows the following steps:

1. Write LK = 1, and write MID[3:0] and PID[7:0] bit fields in HWSEM_CTLx (x = 0…31) register to lock the semaphore.
2. Read the written HWSEM_CTLx (x = 0…31) register, to confirm the MID[3:0] and PID[7:0] bit fields are the written value or not.
3. If the MID[3:0] and PID[7:0] bit fields in HWSEM_CTLx (x = 0…31) register match with the written value, the semaphore is locked by this process. Otherwise, the semaphore is locked by other AHB bus master or process, go to step 1 to try again.

Read lock

When a semaphore is locked by read lock, MID[3:0] bit field in HWSEM_CTLx (x = 0…31) / HWSEM_RLKx (x = 0…31) register is the AHB bus master which locked it, while PID[7:0] bit field in HWSEM_CTLx (x = 0…31) / HWSEM_RLKx (x = 0…31) register is always 0.

If more than one processes use read lock to lock the same semaphore, when the semaphore is read locked by one process, other processes will find that the semaphore is read as locked by themselves, and cannot recognize which process locks it.

Read lock operation follows the following steps:

1. Read HWSEM_RLKx (x = 0…31) register to lock the semaphore.
2. If the MID[3:0] matches with the AHB bus master, and PID[7:0] is read as 0, the semaphore is read locked. While if the semaphore is locked by other AHB bus master (the MID[3:0] does not match with the AHB bus master) or write locked by other processes (PID[7:0] is not 0), go to step 1 to try again.

2.3.4. Unlock a semaphore

When a semaphore is locked, use write unlock to release a semaphore. The operation is protected from being unlocked by the process which does not have the unlocking priority.

Only the process which has the matching MID[3:0] and PID[7:0] can unlock the semaphore. After a valid write unlock operation, LK bit, MID[3:0] bits and PID[7:0] bits in HWSEM_CTLx (x = 0…31) / HWSEM_RLKx (x = 0…31) register are cleared to 0, an interrupt will be generated if SIEx bit in HWSEM_INTEN register is set to 1.

Write unlock operation follows the following steps:

1. Write LK = 0, and write MID[3:0] and PID[7:0] bit fields in HWSEM_CTLx (x = 0…31) register to unlock the semaphore.
2. Read the written HWSEM_CTLx (x = 0…31) register, to confirm the LK bit is 0.
3. If the MID[3:0] and PID[7:0] bit fields in HWSEM_CTLx (x = 0…31) register match with the semaphore lock information, LK bit is 0 and the semaphore is unlocked. Otherwise, the unlock operation will be ignored and LK bit remains 1.

2.3.5. Unlock all semaphores

When the AHB bus master works incorrectly, all unlock operation can be used to clear lock status and lock information of all semaphores.

All unlock operation follows the following steps:

1. Write KEY[15:0] and MID[3:0] bit field of HWSEM_UNLK register to unlock all semaphores.
2. Read the HWSEM_CTLx (x = 0…31) registers, to confirm all the LK bits are 0.
3. If the MID[3:0] bit field in HWSEM_CTLx (x = 0…31) register matches with the semaphore lock information, LK bit is 0 and the semaphore is unlocked, an interrupt will be generated if SIEx bit in HWSEM_INTEN register is set to 1. Otherwise, the unlock operation will be ignored and LK bit remains 1.

2.3.6. Interrupts

The semaphore should be locked and unlocked in user process, it cannot be locked or unlocked in the interrupt service. It is suggested to try to lock a semaphore twice, and only when failed on the second try, enable the semaphore interrupt and wait the interrupt.

Figure 2-3. First try lock success shows when target process first try to lock a semaphore, and succeed to lock it, then the clear flag status operation should be done, and no need to enable the semaphore interrupt.

Figure 2-4. Second try lock success shows when target process fails on the first try of locking a semaphore, then before the second try, the clear flag status operation should be done, then target process succeed to lock it, in this situation, there is also no need to enable the semaphore interrupt.

Figure 2-5. Second try lock fail shows when target process fails on the first try of locking a semaphore, then before the second try, the clear flag status operation should be done, however, target process failed to lock it, in this situation, enable the semaphore interrupt and wait the interrupt, then perform the next try.

Figure 2-3. First try lock success

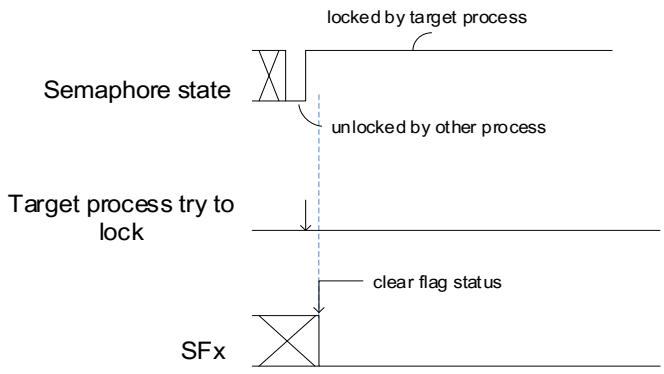


Figure 2-4. Second try lock success

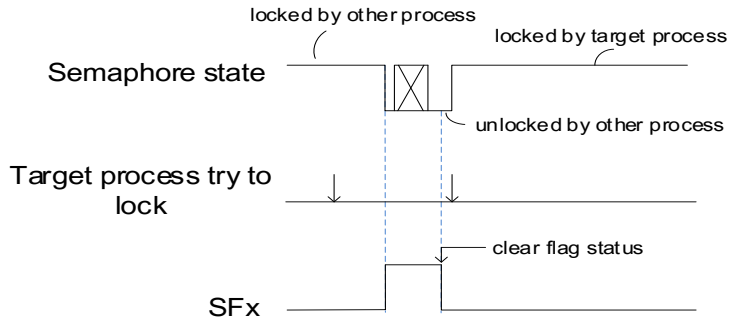
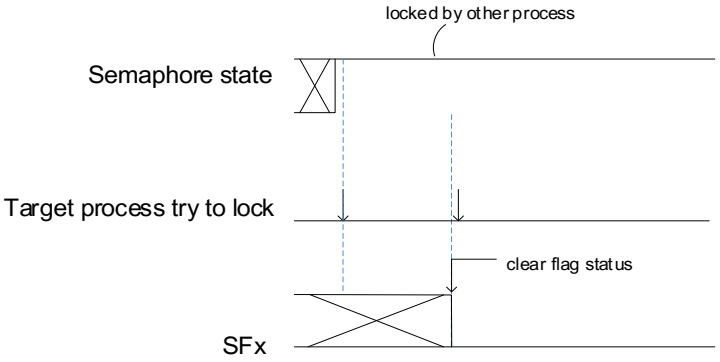


Figure 2-5. Second try lock fail



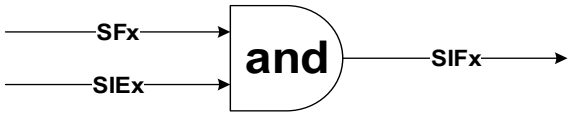
Each semaphore can generate an unlock semaphore interrupt. Each interrupt event has a interrupt flag bit is in HWSEM_INTF register, a clear bit in HWSEM_INTC register, and an enable bit in HWSEM_INTEN register. The relationship is described in the following ***Table 2-1. interrupt events***.

Table 2-1. interrupt events

Interrupt event	Interrupt flag bit	Clear bit	Enable bit
	HWSEM_INTF	HWSEM _INTC	HWSEM _INTEN
Semaphore x unlocked	SIFx	SIFCx	SIEx

The HWSEM interrupt logic is shown in the ***Figure 2-6. HWSEM interrupt logic***, an interrupt can be produced when an interrupt event occurs and enabled the event.

Figure 2-6. HWSEM interrupt logic



Note: "x" indicates semaphore number (x=0…31).

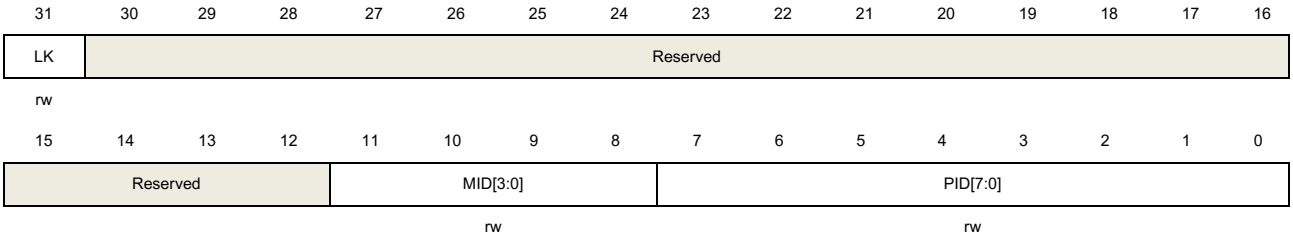
2.4. Register definition

HWSEM base address: 0x5802 6400

2.4.1. Control register (HWSEM_CTLx)(x = 0…31)

Address offset: 0x00 + 0x4 * x
Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31	LK	Lock semaphore 0: Unlock the semaphore when MID[3:0] and PID[7:0] are matched with the lock information 1: Try to lock the semaphore
30:12	Reserved	Must be kept at reset value.
11:8	MID[3:0]	AHB bus master identification Write by software, only when the semaphore is unlocked, and the writing value of MID[3:0] is matched with the AHB bus master in a write lock operation, the value will be written to this bit field. When the writing value of MID[3:0] is matched with the AHB bus master in a write unlock operation, this bit field will be cleared to 0.
7:0	PID[7:0]	Process identification Write by software, only when the semaphore is unlocked, the writing value of PID[7:0] will be written to this bit field in a write lock operation. In a write unlock operation, this bit field will be cleared to 0.

2.4.2. Read lock register (HWSEM_RLKx)(x = 0...31)

Address offset: 0x80 + 0x4 * x
Reset value: 0x0000 0000

HWSEM_RLKx (x = 0…31) register access the same physical addresses as the HWSEM_CTLx (x = 0…31) 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
LK	Reserved														
r															

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

Bits	Fields	Descriptions
31	LK	Lock semaphore by reading When read this bit by a matched AHB bus master, it always returns 1, if the semaphore is unlocked before read, the semaphore will be read locked by hardware, while if the semaphore is already locked before read, this bit will not change. 0: Semaphore is unlocked. 1: Semaphore is locked.
30:12	Reserved	Must be kept at reset value.
11:8	MID[3:0]	Bus master identification When read this bit field by a matched AHB bus master, if the semaphore is unlocked before read, this bit field will be written with the AHB bus master identification, if the semaphore is locked before read, the MID[3:0] of which locks the semaphore will be returned.
7:0	PID[7:0]	Process identification When read this bit field by a matched AHB bus master, if the semaphore is unlocked before read, this bit field will be written with 0 and also be read as 0, if the semaphore is locked before read, the PID[7:0] of which locks the semaphore will be returned.

2.4.3. Interrupt enable register (HWSEM_INTEN)

Address offset: 0x100

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SIE31	SIE30	SIE29	SIE28	SIE27	SIE26	SIE25	SIE24	SIE23	SIE22	SIE21	SIE20	SIE19	SIE18	SIE17	SIE16
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
SIE15	SIE14	SIE13	SIE12	SIE11	SIE10	SIE9	SIE8	SIE7	SIE6	SIE5	SIE4	SIE3	SIE2	SIE1	SIE0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:0	SIEx	Semaphore interrupt enable bit

0: Disable semaphore interrupt

1: Enable semaphore interrupt

2.4.4. Interrupt flag clear register (HWSEM_INTC)

Address offset: 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
SIFC31	SIFC30	SIFC29	SIFC28	SIFC27	SIFC26	SIFC25	SIFC24	SIFC23	SIFC22	SIFC21	SIFC20	SIFC19	SIFC18	SIFC17	SIFC16
w1	w1	w1	w1	w1	w1	w1	w1	w1	w1	w1	w1	w1	w1	w1	w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIFC15	SIFC14	SIFC13	SIFC12	SIFC11	SIFC10	SIFC9	SIFC8	SIFC7	SIFC6	SIFC5	SIFC4	SIFC3	SIFC2	SIFC1	SIFC0
w1	w1	w1	w1	w1	w1	w1	w1	w1	w1	w1	w1	w1	w1	w1	w1

Bits	Fields	Descriptions
31:0	SIFCx	Semaphore interrupt flag clear bit Written by software, and be read as 0. 0: No effect 1: Clear the semaphore flag and interrupt flag

2.4.5. Status register (HWSEM_STAT)

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
SF31	SF30	SF29	SF28	SF27	SF26	SF25	SF24	SF23	SF22	SF21	SF20	SF19	SF18	SF17	SF16
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
SF15	SF14	SF13	SF12	SF11	SF10	SF9	SF8	SF7	SF6	SF5	SF4	SF3	SF2	SF1	SF0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

Bits	Fields	Descriptions
31:0	SFx	Semaphore flag Set by hardware, and clear by writing 1 to SIFCx bit in HWSEM_INTC register. 0: No semaphore unlock event occurs 1: A semaphore unlock event occurs

2.4.6. Interrupt flag register (HWSEM_INTF)

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
SIF31	SIF30	SIF29	SIF28	SIF27	SIF26	SIF25	SIF24	SIF23	SIF22	SIF21	SIF20	SIF19	SIF18	SIF17	SIF16
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
SIF15	SIF14	SIF13	SIF12	SIF11	SIF10	SIF9	SIF8	SIF7	SIF6	SIF5	SIF4	SIF3	SIF2	SIF1	SIF0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

Bits	Fields	Descriptions
31:0	SIFx	Semaphore interrupt flag status Set by hardware, and clear by writing 1 to SIFCx bit in HWSEM_INTC register. 0: No pending interrupt 1: An interrupt is pending

2.4.7. Unlock register (HWSEM_UNLK)

Address offset: 0x140

Reset value: 0x0000 0000

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

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

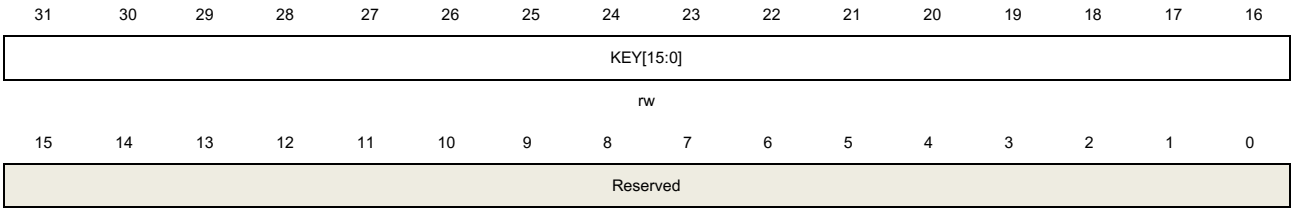
Bits	Fields	Descriptions
31:16	KEY[15:0]	Key value Write only, and read as 0. Only when the value of KEY[15:0] is matched with the KEY[15:0] bits in HWSEM_KEY register, all semaphores of the MID[3:0] which is controlled in this register will be unlocked.
15:12	Reserved	Must be kept at reset value.
11:8	MID[3:0]	Bus master identification to clear Write only, and read as 0. The semaphores of the AHB bus master which has the written value identification is to be cleared.
7:0	Reserved	Must be kept at reset value.

2.4.8. Key register (HWSEM_KEY)

Address offset: 0x144

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:16	KEY[15:0]	Key for unlocking all semaphores of a bus master The key for unlocking all semaphores.
15:0	Reserved	Must be kept at reset value.

3. Interrupt / event controller (EXTI)

3.1. Overview

Cortex®-M7 integrates the Nested Vectored Interrupt Controller (NVIC) for efficient exception and interrupts processing. NVIC facilitates low-latency exception and interrupt handling and power management controls. It is tightly coupled to the processor core. For more details about the NVIC, refer to the Cortex®-M7 technical reference manual.

EXTI (interrupt / event controller) contains up to 38 independent edge detectors and generates interrupt requests or events to the processor. The EXTI has three trigger types: rising edge, falling edge and both edges. Each edge detector in the EXTI can be configured and masked independently.

3.2. Characteristics

- Cortex®-M7 system exception.
- Up to 227 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 38 independent edge detectors in EXTI.
- Three trigger types: rising, falling and both edges.
- Software interrupt or event trigger.
- Trigger sources configurable.

3.3. Interrupt function overview

The Arm® Cortex®-M7 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 3-1. NVIC exception types in Cortex®-M7

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 fixed to 1000 and SysTick clock source is from CK_SYS or CK_SYS/8. The SYST_RVR register is configured to provide a 1ms time-base for the system. When SysTick clock source is from CK_SYS (CK_SYS=a MHz), the SYST_RVR register value is set to (a*1000-1). When the SysTick clock source is from CK_SYS/8, the SYST_RVR register value is set to (a/8*1000-1).

Table 3-2. Interrupt vector table

Interrupt number	Vector number	Peripheral interrupt description	Vector address
IRQ 0	16	WWDGT interrupt	0x0000_0040
IRQ 1	17	VAVD / LVD / VOVD / VUVD through EXTI line detection interrupt	0x0000_0044
IRQ 2	18	RTC tamper and timestamp from EXTI interrupt, LXTAL clock stuck interrupt	0x0000_0048
IRQ 3	19	RTC wakeup from EXTI interrupt	0x0000_004C
IRQ 4	20	NVMC 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

Interrupt number	Vector number	Peripheral interrupt description	Vector address
IRQ 11	27	DMA0 channel 0 global interrupt	0x0000_006C
IRQ 12	28	DMA0 channel 1 global interrupt	0x0000_0070
IRQ 13	29	DMA0 channel 2 global interrupt	0x0000_0074
IRQ 14	30	DMA0 channel 3 global interrupt	0x0000_0078
IRQ 15	31	DMA0 channel 4 global interrupt	0x0000_007C
IRQ 16	32	DMA0 channel 5 global interrupt	0x0000_0080
IRQ 17	33	DMA0 channel 6 global interrupt	0x0000_0084
IRQ 18	34	ADC0 and ADC1 global interrupt, RDCM 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	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	0x0000_00BC
IRQ 32	48	I2C0 error interrupt	0x0000_00C0
IRQ 33	49	I2C1 event interrupt	0x0000_00C4
IRQ 34	50	I2C1 error interrupt	0x0000_00C8
IRQ 35	51	SPI0 global interrupt	0x0000_00CC
IRQ 36	52	SPI1 global interrupt	0x0000_00D0
IRQ 37	53	USART0 global and wakeup interrupt	0x0000_00D4
IRQ 38	54	USART1 global and wakeup interrupt	0x0000_00D8
IRQ 39	55	USART2 global and wakeup interrupt	0x0000_00DC
IRQ 40	56	EXTI line10-15 interrupt	0x0000_00E0
IRQ 41	57	RTC alarm from EXTI interrupt	0x0000_00E4
IRQ 42	58	Reserved	0x0000_00E8
IRQ 43	59	TIMER7 break interrupt	0x0000_00EC
IRQ 44	60	TIMER7 update interrupt	0x0000_00F0
IRQ 45	61	TIMER7 trigger and commutation interrupt	0x0000_00F4
IRQ 46	62	TIMER7 capture compare interrupt	0x0000_00F8
IRQ 47	63	DMA0 channel 7 global interrupt	0x0000_00FC
IRQ 48	64	EXMC global interrupt	0x0000_0100
IRQ 49	65	SDIO0 global interrupt	0x0000_0104
IRQ 50	66	TIMER4 global interrupt	0x0000_0108
IRQ 51	67	SPI2 global interrupt	0x0000_010C

Interrupt number	Vector number	Peripheral interrupt description	Vector address
IRQ 52	68	UART3 global interrupt	0x0000_0110
IRQ 53	69	UART4 global interrupt	0x0000_0114
IRQ 54	70	TIMER5 global interrupt and DAC underrun error interrupt	0x0000_0118
IRQ 55	71	TIMER6 global 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	Ethernet0 global interrupt	0x0000_0134
IRQ 62	78	Ethernet0 wakeup from EXTI interrupt	0x0000_0138
IRQ 63-67	79-83	Reserved	0x0000_013C-0x0000_014C
IRQ 68	84	DMA1 channel 5 global interrupt	0x0000_0150
IRQ 69	85	DMA1 channel 6 global interrupt	0x0000_0154
IRQ 70	86	DMA1 channel 7 global interrupt	0x0000_0158
IRQ 71	87	USART5 global and wakeup interrupt	0x0000_015C
IRQ 72	88	I2C2 event interrupt	0x0000_0160
IRQ 73	89	I2C2 error interrupt	0x0000_0164
IRQ 74	90	USBHS0 endpoint 1 out interrupt	0x0000_0168
IRQ 75	91	USBHS0 endpoint 1 in interrupt	0x0000_016C
IRQ 76	92	USBHS0 wakeup from EXTI interrupt	0x0000_0170
IRQ 77	93	USBHS0 global interrupt	0x0000_0174
IRQ 78	94	DCI global interrupt	0x0000_0178
IRQ 79	95	CAU global interrupt	0x0000_017C
IRQ 80	96	HAU and TRNG global interrupt	0x0000_0180
IRQ 81	97	FPU global interrupt	0x0000_0184
IRQ 82	98	UART6 global interrupt	0x0000_0188
IRQ 83	99	UART7 global interrupt	0x0000_018C
IRQ 84	100	SPI3 global interrupt	0x0000_0190
IRQ 85	101	SPI4 global interrupt	0x0000_0194
IRQ 86	102	SPI5 global interrupt	0x0000_0198
IRQ 87	103	SAI0 global interrupt	0x0000_019C
IRQ 88	104	TLI global interrupt	0x0000_01A0
IRQ 89	105	TLI error interrupt	0x0000_01A4
IRQ 90	106	IPA global interrupt	0x0000_01A8
IRQ 91	107	SAI1 global interrupt	0x0000_01AC
IRQ 92	108	OSPI0 global interrupt	0x0000_01B0

Interrupt number	Vector number	Peripheral interrupt description	Vector address
IRQ 93-94	109-110	Reserved	0x0000_01B4-0x0000_01B8
IRQ 95	111	I2C3 event interrupt	0x0000_01BC
IRQ 96	112	I2C3 error interrupt	0x0000_01C0
IRQ 97	113	RSPDIF global interrupt	0x0000_01C4
IRQ 98-101	114-117	Reserved	0x0000_01C8-0x0000_01D4
IRQ 102	118	DMAMUX overrun interrupt	0x0000_01D8
IRQ 103-108	119-124	Reserved	0x0000_01DC-0x0000_01F0
IRQ 109	125	HPDF global interrupt 4	0x0000_01F4
IRQ 110	126	HPDF global interrupt 0	0x0000_01F8
IRQ 111	127	HPDF global interrupt 1	0x0000_01FC
IRQ 112	128	HPDF global interrupt 2	0x0000_0200
IRQ 113	129	HPDF global interrupt 3	0x0000_0204
IRQ 114	130	SAI2 global interrupt	0x0000_0208
IRQ 115	131	Reserved	0x0000_020C
IRQ 116	132	TIMER14 global interrupt	0x0000_0210
IRQ 117	133	TIMER15 global interrupt	0x0000_0214
IRQ 118	134	TIMER16 global interrupt	0x0000_0218
IRQ 119	135	Reserved	0x0000_021C
IRQ 120	136	MDIO global interrupt	0x0000_0220
IRQ 121	137	Reserved	0x0000_0224
IRQ 122	138	MDMA global interrupt	0x0000_0228
IRQ 123	139	Reserved	0x0000_022C
IRQ 124	140	SDIO1 global interrupt	0x0000_0230
IRQ 125	141	HWSEM global interrupt	0x0000_0234
IRQ 126	142	Reserved	0x0000_0238
IRQ 127	143	ADC2 global interrupt	0x0000_023C
IRQ 128-136	144-152	Reserved	0x0000_0240-0x0000_0260
IRQ 137	153	CMP0 and CMP1 global interrupt, CMP0 and CMP1 through EXTI line detection interrupt	0x0000_0264
IRQ 138-143	154-159	Reserved	0x0000_0268-0x0000_027C
IRQ 144	160	CTC interrupt	0x0000_0280
IRQ 145	161	RAMECCMU global interrupt	0x0000_0284
IRQ 146	162	SAI3 global interrupt	0x0000_0288
IRQ 147-149	163-165	Reserved	0x0000_028C-0x0000_0294

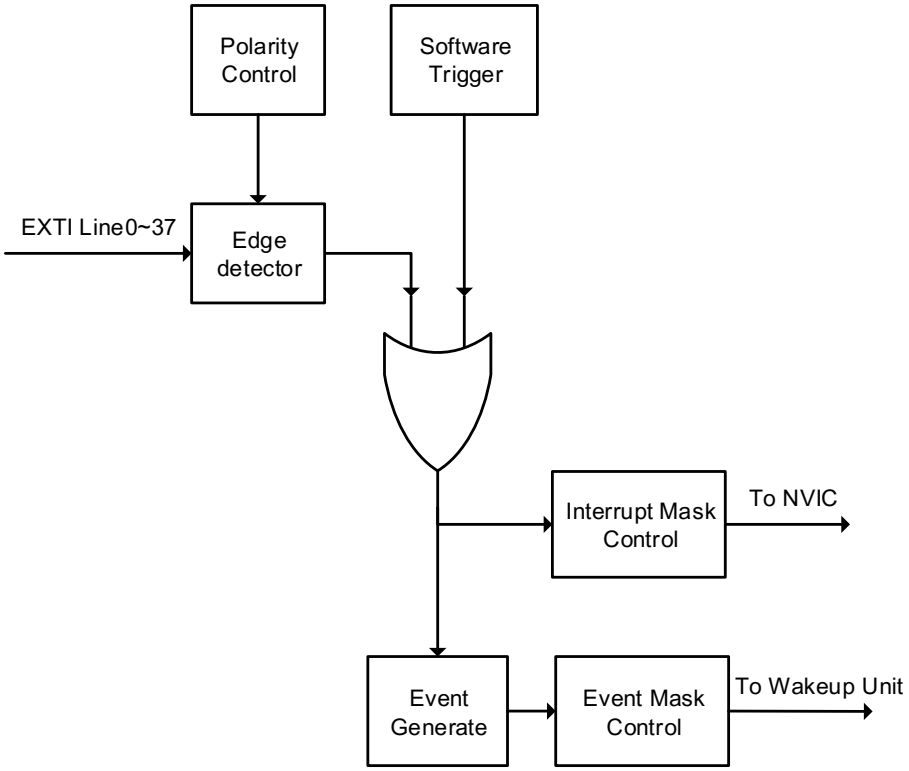
Interrupt number	Vector number	Peripheral interrupt description	Vector address
IRQ 150	166	OSPI1 global interrupt	0x0000_0298
IRQ 151	167	RTDEC0 global interrupt	0x0000_029C
IRQ 152	168	RTDEC1 global interrupt	0x0000_02A0
IRQ 153	169	FAC global interrupt	0x0000_02A4
IRQ 154	170	TMU global interrupt	0x0000_02A8
IRQ 155	171	FFT global interrupt	0x0000_02AC
IRQ 156	172	ESC SYNC0 interrupt	0x0000_02B0
IRQ 157	173	ESC SYNC1 interrupt	0x0000_02B4
IRQ 158	174	ESC IRQ interrupt	0x0000_02B8
IRQ 159-160	175-176	Reserved	0x0000_02BC-0x0000_02C0
IRQ 161	177	TIMER22 global interrupt	0x0000_02C4
IRQ 162	178	TIMER23 global interrupt	0x0000_02C8
IRQ 163	179	TIMER30 global interrupt	0x0000_02CC
IRQ 164	180	TIMER31 global interrupt	0x0000_02D0
IRQ 165	181	TIMER40 global interrupt	0x0000_02D4
IRQ 166	182	TIMER41 global interrupt	0x0000_02D8
IRQ 167	183	TIMER42 global interrupt	0x0000_02DC
IRQ 168	184	TIMER43 global interrupt	0x0000_02E0
IRQ 169	185	TIMER44 global interrupt	0x0000_02E4
IRQ 170	186	TIMER50 global interrupt	0x0000_02E8
IRQ 171	187	TIMER51 global interrupt	0x0000_02EC
IRQ 172	188	USBHS1 endpoint 1 out interrupt	0x0000_02F0
IRQ 173	189	USBHS1 endpoint 1 in interrupt	0x0000_02F4
IRQ 174	190	USBHS1 wakeup from EXTI interrupt	0x0000_02F8
IRQ 175	191	USBHS1 global interrupt	0x0000_02FC
IRQ 176	192	Ethernet1 global interrupt	0x0000_0300
IRQ 177	193	Ethernet1 wakeup from EXTI interrupt	0x0000_0304
IRQ 178	194	Reserved	0x0000_0308
IRQ 179	195	CAN0 wakeup through EXTI line detection interrupt	0x0000_030C
IRQ 180	196	CAN0 interrupt for message buffer	0x0000_0310
IRQ 181	197	CAN0 interrupt for Bus off / Bus off done	0x0000_0314
IRQ 182	198	CAN0 interrupt for Error	0x0000_0318
IRQ 183	199	CAN0 interrupt for Error in fast transmission	0x0000_031C
IRQ 184	200	CAN0 interrupt for Transmit warning	0x0000_0320
IRQ 185	201	CAN0 interrupt for Receive warning	0x0000_0324
IRQ 186	202	CAN1 wakeup through EXTI line detection interrupt	0x0000_0328
IRQ 187	203	CAN1 interrupt for message buffer	0x0000_032C

Interrupt number	Vector number	Peripheral interrupt description	Vector address
IRQ 188	204	CAN1 interrupt for Bus off / Bus off done	0x0000_0330
IRQ 189	205	CAN1 interrupt for Error	0x0000_0334
IRQ 190	206	CAN1 interrupt for Error in fast transmission	0x0000_0338
IRQ 191	207	CAN1 interrupt for Transmit warning	0x0000_033C
IRQ 192	208	CAN1 interrupt for Receive warning	0x0000_0340
IRQ 193	209	CAN2 wakeup through EXTI line detection interrupt	0x0000_0344
IRQ 194	210	CAN2 interrupt for message buffer	0x0000_0348
IRQ 195	211	CAN2 interrupt for Bus off / Bus off done	0x0000_034C
IRQ 196	212	CAN2 interrupt for Error	0x0000_0350
IRQ 197	213	CAN2 interrupt for Error in fast transmission	0x0000_0354
IRQ 198	214	CAN2 interrupt for Transmit warning	0x0000_0358
IRQ 199	215	CAN2 interrupt for Receive warning	0x0000_035C
IRQ 200	216	OTP global interrupt	0x0000_0360
IRQ 201	217	I2C0 wakeup through EXTI line detection interrupt	0x0000_0364
IRQ 202	218	I2C1 wakeup through EXTI line detection interrupt	0x0000_0368
IRQ 203	219	I2C2 wakeup through EXTI line detection interrupt	0x0000_036C
IRQ 204	220	I2C3 wakeup through EXTI line detection interrupt	0x0000_0370
IRQ 205	221	LPDTS interrupt	0x0000_0374
IRQ 206	222	LPDTS wakeup through EXTI line detection interrupt	0x0000_0378
IRQ 207	223	TIMER0 DEC interrupt	0x0000_037C
IRQ 208	224	TIMER7 DEC interrupt	0x0000_0380
IRQ 209	225	TIMER1 DEC interrupt	0x0000_0384
IRQ 210	226	TIMER2 DEC interrupt	0x0000_0388
IRQ 211	227	TIMER3 DEC interrupt	0x0000_038C
IRQ 212	228	TIMER4 DEC interrupt	0x0000_0390
IRQ 213	229	TIMER22 DEC interrupt	0x0000_0394
IRQ 214	230	TIMER23 DEC interrupt	0x0000_0398
IRQ 215	231	TIMER30 DEC interrupt	0x0000_039C
IRQ 216	232	TIMER31 DEC interrupt	0x0000_03A0
IRQ 217	233	CPU D-Cache error detection interrupt	0x0000_03A4
IRQ 218	234	CPU I-Cache error detection interrupt	0x0000_03A8
IRQ 219	235	VOVD / VUVD interrupt after digital filtering	0x0000_03AC
IRQ 220	236	Reserved	0x0000_03B0
IRQ 221	237	TIMER0 channel break interrupt	0x0000_03B4

Interrupt number	Vector number	Peripheral interrupt description	Vector address
IRQ 222	238	TIMER7 channel break interrupt	0x0000_03B8
IRQ 223	239	DSI global interrupt	0x0000_03BC
IRQ 224	240	PKCAU global interrupt	0x0000_03C0
IRQ 225	241	ESC internal interrupt	0x0000_03C4
IRQ 226	242	EDIM global interrupt	0x0000_03C8

3.4. External interrupt and event (EXTI) block diagram

Figure 3-1. Block diagram of EXTI



3.5. External interrupt and event function overview

The EXTI contains up to 38 independent edge detectors and generates interrupt requests or events to the processor. The EXTI has three trigger types: rising edge, falling edge and both edges. Each edge detector in the EXTI can be configured and masked independently.

The EXTI trigger source includes 16 external lines from GPIO pins and 22 lines from internal modules which refers to ***Table 3-3. EXTI source***. 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*** for detail).

EXTI can provide not only interrupts but also event signals to the processor. The Cortex®-M7 processor fully implements the Wait For Interrupt (WFI), Wait For Event (WFE) and the Send Event (SEV) instructions. The Wake-up Interrupt Controller (WIC) enables the processor and NVIC to be put into a very low-power sleep mode leaving the WIC to identify and prioritize interrupts and events. EXTI can be used to wake up processor and the whole system when some expected event occurs, such as a special GPIO pin toggling or RTC alarm.

Hardware trigger

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

1. Configure EXTI sources in SYSCFG module based on application requirement.

2. Configure EXTI_RTENx and EXTI_FTENx to enable the rising or falling detection on related pins. (Software may set both RTENx and FTENx for a pin at the same time to detect both rising and falling changes on this pin).

3. Enable interrupts or events by setting related EXTI_INTENx or EXTI_EVENx bits.

4. EXTI starts to detect changes on the configured pins. The related interrupt or event will be triggered when desired change is detected on these pins. If the interrupt is triggered, the related PDx is set; if the event is triggered, the related PDx is not set. The software should respond to the interrupts or events and clear these PDx bits.

Software trigger

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

1. Enable interrupts or events by setting related EXTI_INTENx or EXTI_EVENx bits.

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

Table 3-3. EXTI source

EXTI line number	Source
0	PA0 / PB0 / PC0 / PD0 / PE0 / PF0 / PG0 / PH0 / PK0
1	PA1 / PB1 / PC1 / PD1 / PE1 / PF1 / PG1 / PH1 / PK1
2	PA2 / PB2 / PC2 / PD2 / PE2 / PF2 / PG2 / PH2 / PK2
3	PA3 / PB3 / PC3 / PD3 / PE3 / PF3 / PG3 / PH3
4	PA4 / PB4 / PC4 / PD4 / PE4 / PF4 / PG4 / PH4
5	PA5 / PB5 / PC5 / PD5 / PE5 / PF5 / PG5 / PH5
6	PA6 / PB6 / PC6 / PD6 / PE6 / PF6 / PG6 / PH6
7	PA7 / PB7 / PC7 / PD7 / PE7 / PF7 / PG7 / PH7
8	PA8 / PB8 / PC8 / PD8 / PE8 / PF8 / PG8 / PH8 / PJ8

EXTI line number	Source
9	PA9 / PB9 / PC9 / PD9 / PE9 / PF9 / PG9 / PH9 / PJ9
10	PA10 / PB10 / PC10 / PD10 / PE10 / PF10 / PG10 / PH10 / PJ10
11	PA11 / PB11 / PC11 / PD11 / PE11 / PF11 / PG11 / PH11 / PJ11
12	PA12 / PB12 / PC12 / PD12 / PE12 / PF12 / PG12 / PH12
13	PA13 / PB13 / PC13 / PD13 / PE13 / PF13 / PG13 / PH13
14	PA14 / PB14 / PC14 / PD14 / PE14 / PF14 / PG14 / PH14
15	PA15 / PB15 / PC15 / PD15 / PE15 / PF15 / PG15 / PH15
16	VAVD / LVD / VOVD / VUVD
17	RTC alarm
18	RTC tamper and timestamp event
19	RTC wakeup
20	CMP0 output
21	CMP1 output
22	Ethernet1 wakeup
23	Ethernet0 wakeup
24	CAN0 wakeup
25	CAN1 wakeup
26	CAN2 wakeup
27	USART0 wakeup
28	USART1 wakeup
29	USART2 wakeup
30	USART5 wakeup
31	USBHS0 wakeup
32	USBHS1 wakeup
33	I2C0 wakeup
34	I2C1 wakeup
35	I2C2 wakeup
36	I2C3 wakeup
37	LPDTS wakeup

3.6. Register definition

EXTI base address: 0x5800 0000

3.6.1. 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 line x is disabled 1: Interrupt from line x is enabled

3.6.2. 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 line x is disabled 1: Event from line x is enabled

3.6.3. 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 line x is invalid 1: Rising edge of line x is valid as an interrupt / event request

3.6.4. 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 line x is invalid 1: Falling edge of line x is valid as an interrupt / event request

3.6.5. 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

3.6.6. Pending register 0 (EXTI_PD0)

Address offset: 0x14

Reset value: 0xXXXX XXXX where X is undefined.

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
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 line x is not triggered 1: EXTI line x is triggered. This bit is cleared to 0 by writing 1 to it.

3.6.7. Interrupt enable register 1 (EXTI_INTEN1)

Address offset: 0x18

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										INTEN37	INTEN36	INTEN35	INTEN34	INTEN33	INTEN32
										rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:6	Reserved	Must be kept at reset value.
5:0	INTENx	Interrupt enable bit x (x = 32…37) 0: Interrupt from line x is disabled 1: Interrupt from line x is enabled

3.6.8. Event enable register 1 (EXTI_EVEN1)

Address offset: 0x1C

Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										EVEN37	EVEN36	EVEN35	EVEN34	EVEN33	EVEN32
										rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:6	Reserved	Must be kept at reset value.
5:0	EVENx	Event enable bit x (x = 32…37) 0: Event from line x is disabled 1: Event from line x is enabled

3.6.9. 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										RTEN37	RTEN36	RTEN35	RTEN34	RTEN33	RTEN32
										rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:6	Reserved	Must be kept at reset value.
5:0	RTENx	Rising edge trigger enable bit x (x = 32…37) 0: Rising edge of line x is invalid 1: Rising edge of line x is valid as an interrupt / event request

3.6.10. 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										FTEN37	FTEN36	FTEN35	FTEN34	FTEN33	FTEN32
										rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:6	Reserved	Must be kept at reset value.
5:0	FTENx	Falling edge trigger enable bit x (x = 32…37) 0: Falling edge of line x is invalid 1: Falling edge of line x is valid as an interrupt / event request

3.6.11. 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										SWIEV37	SWIEV36	SWIEV35	SWIEV34	SWIEV33	SWIEV32
										rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:6	Reserved	Must be kept at reset value.
5:0	SWIEVx	Interrupt / event software trigger bit x (x = 32…37) 0: Deactivate the EXTIx software interrupt / event request 1: Activate the EXTIx software interrupt / event request

3.6.12. Pending register 1 (EXTI_PD1)

Address offset: 0x2C

Reset value: 0x0000 00XX where X is undefined.

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

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

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

rc_w1 rc_w1 rc_w1 rc_w1 rc_w1 rc_w1

Bits	Fields	Descriptions
31:6	Reserved	Must be kept at reset value.
5:0	PDx	Interrupt pending status bit x (x = 32…37) 0: EXTI line x is not triggered 1: EXTI line x is triggered. This bit is cleared to 0 by writing 1 to it.

4. Direct memory access controller (DMA)

4.1. Overview

The direct memory access (DMA) controller provides a hardware method of transferring data between peripherals and/or memory without intervention from the MCU, thereby increasing system performance by off-loading the MCU from copying large amounts of data and avoiding frequent interrupts to serve peripherals needing more data or having available data.

Two AHB master interfaces and eight four-word depth 32-bit width FIFOs are presented in each DMA controller, which achieves a high DMA transmission performance. There are 16 independent channels in the DMA controller (8 for DMA0 and 8 for DMA1). Each channel is assigned a specific or multiple target peripheral devices for memory access request management. Two arbiters respectively for memory and peripheral are implemented inside to handle the priority among DMA requests.

Both the DMA controller and the Cortex®-M7 core implement data access through the system bus. An arbitration mechanism is implemented to solve the competition between these two masters. When the same peripheral is targeted, the MCU access will be suspended for some specific bus cycles. A round-robin scheduling algorithm is utilized in the bus matrix to guaranty at least half the bandwidth to the MCU.

4.2. Characteristics

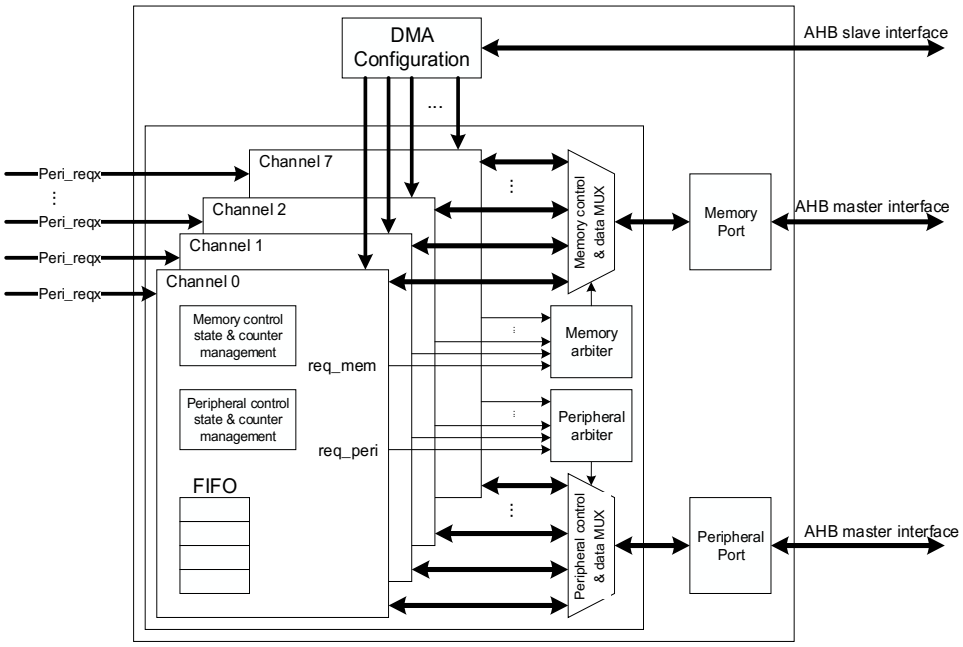
- Two AHB master interface for transferring data, and one AHB slave interface for programming DMA.
- 16 channels (8 for DMA0 and 8 for DMA1) and each channel are configurable.
- Support independent single, 4, 8, 16-beat incrementing burst memory and peripheral transfer.
- Support switch-buffer transmission between peripheral and memory.
- Software DMA channel priority (low, medium, high, ultra high) and hardware DMA channel priority (DMA channel 0 has the highest priority and DMA channel 7 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 three transfer modes:
- Read from memory and write to peripheral.
- Read from peripheral and write to memory.
- Read from memory and write to memory.
- Support two data processing modes by use of the four-word depth 32-bit width FIFOs:

- Multi-data mode: Pack/Unpack data when memory transfer width are different from peripheral transfer width.
- Single-data mode: Read data from source when FIFO is empty and write data to destination when one data has been pushed into FIFO.
- One separate interrupt per channel with five types of event flags.
- Support interrupt enable and clear.

4.3. Function overview

4.3.1. Block diagram

Figure 4-1. Block diagram of DMA



As shown in ***Figure 4-1. Block diagram of DMA***, a DMA controller consists of four main parts:

- DMA configuration through AHB slave interface.
- Data access through two AHB master interfaces respectively for memory access and peripheral access.
- Two arbiters inside to manage multiple peripheral requests coming at the same time.
- Channel data management to control data packing/unpacking and counting.

The DMA controller transfers data from one address to another without CPU intervention. It supports multiple data sizes, burst types, address generation algorithm, priority levels and several transfer modes to allow for flexible application by configuring the corresponding bits in DMA registers. All the DMA registers can be 32-bit configured through AHB slave interface.

Three transfer modes are supported, including peripheral-to-memory, memory-to-peripheral

and memory-to-memory, which is determined by the TM bits in the DMA_CHxCTL register, as listed in *Table 4-1. Transfer mode*.

Table 4-1. Transfer mode

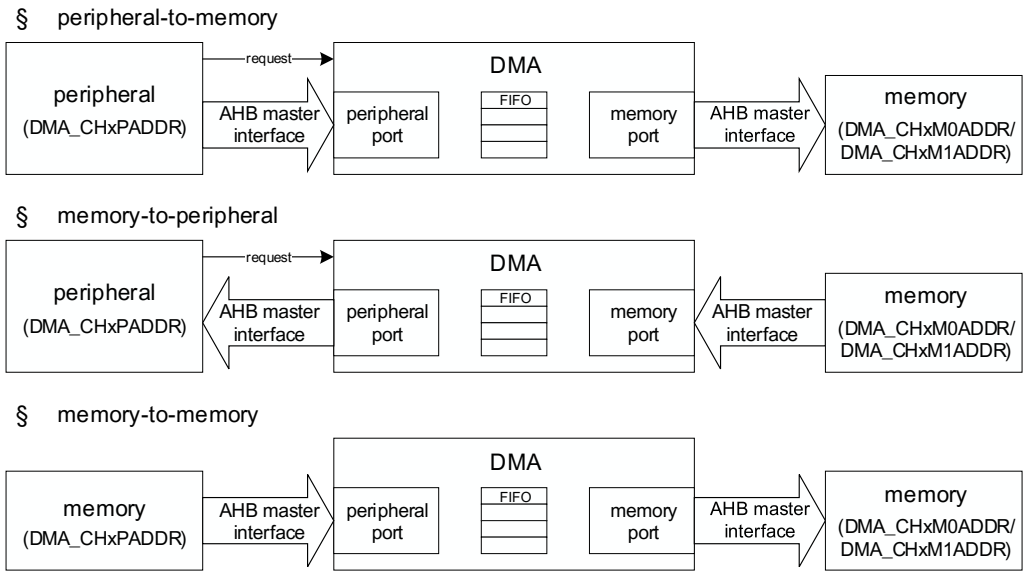
Transfer mode	TM[1:0]	Source	Destination
Peripheral to memory	00	DMA_CHxPADDR	DMA_CHxM0ADDR/ DMA_CHxM1ADDR
Memory to peripheral	01	DMA_CHxM0ADDR/ DMA_CHxM1ADDR	DMA_CHxPADDR
Memory to memory	10	DMA_CHxPADDR	DMA_CHxM0ADDR/ DMA_CHxM1ADDR

Note:

1. The MBS bit in DMA_CHxCTL register determines which is selected as the memory buffer address in DMA_CHxM0ADDR and DMA_CHxM1ADDR register. For more information, refer to section *Switch-buffer mode*.

2. The TM bits in DMA_CHxCTL register are forbidden to configure to 0b11, or the channel will be automatically disabled.

Figure 4-2. Data stream for three transfer modes



As shown in *Figure 4-2. Data stream for three transfer modes*, two AHB master interfaces are implemented in each DMA respectively for memory and peripheral.

- Memory to peripheral: read data from memory through AHB master interface for memory, and write data to peripheral through AHB master interface for peripheral.
- Peripheral to memory: read data from peripheral through AHB master interface for peripheral, and write data to memory through AHB master interface for memory.
- Memory to memory: read data from memory through AHB master interface for peripheral, and write data to another memory through AHB master interface for memory.

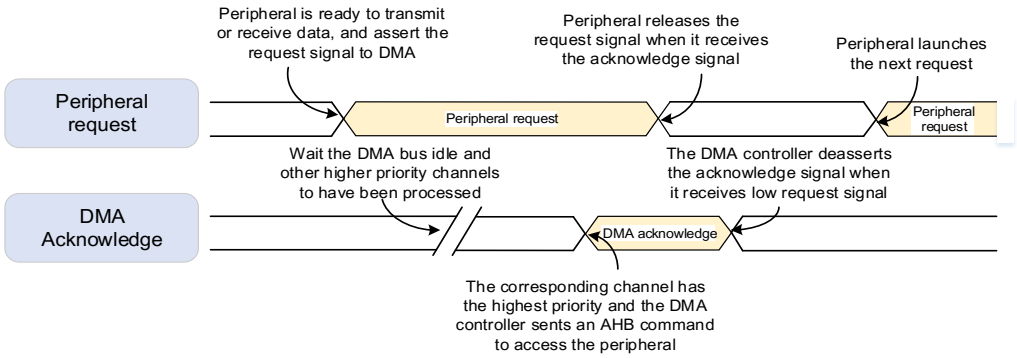
4.3.2. Peripheral handshake

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

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

Figure 4-3. Handshake mechanism shows how the handshake mechanism works between the DMA controller and peripherals.

Figure 4-3. Handshake mechanism



4.3.3. Data process

Arbitration

Two arbiters are implemented in each DMA respectively for memory and peripheral port. When two or more requests are received at the same time, 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.
- Hardware priority: For channels with equal software priority level, priority is given to the channel with lower channel number. For example, when channel 0 and channel 2 are configured with the same software priority, the priority of channel 0 is higher than that of channel 2.

If the RREN bit in the DMA_CH0CTL register is enabled, the arbiter will perform round-robin arbitration for channels configured as ultra-high priority. When there are no requests from channels with ultra-high priority, other requesting channels will be arbitrated based on their configured software priority and channel number.

Transfer width, burst and counter

Transfer width

PWIDTH and MWIDTH in the DMA_CHxCTL register indicate the data width of a peripheral and memory transfer. The DMA supports 8-bit, 16-bit and 32-bit transfer width. In multi-data mode, if PWIDTH is not equal to MWIDTH, the DMA can automatically packs/unpacks data to achieve an integrated and correct data transfer operation. In single-data mode, MWIDTH is automatically locked as PWIDTH by hardware immediately after enable the DMA channel.

Transfer burst type

PBURST and MBURST in the DMA_CHxCTL register indicate the burst type of a peripheral and memory transfer. The DMA supports single burst, 4-beat, 8-beat and 16-beat incrementing burst for peripheral port and memory port. In single-data mode, only single burst type is supported and PBURST and MBURST are automatically locked as '00' by hardware immediately after enable the DMA channel.

In peripheral-to-memory or memory-to-peripheral mode, if PBURST is different from '00', the DMA responds with an incrementing burst transfer of 4, 8, 16-beat based on the PBURST bits for each peripheral request. If the remaining bytes number of data item to be transferred is less than the bytes number needed for a burst transfer, the remaining data items are transferred in single transaction.

AMBA protocol specifies that bursts must not cross a 1KB address boundary, or a transfer error will be responded to the master. In each DMA, the peripheral burst transfer crossing a 1KB address boundary is decomposed to 4, 8 or 16 single transactions depend on the PBURST bits, as the same as the memory burst transfer.

Transfer counter

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

The CNT bits are related to peripheral transfer width, the number of data bytes to be transferred is the CNT bits multiplied by the byte number of the peripheral transfer width. For example, if the PWIDTH bits are equal to '10', and the number of data bytes to be transferred is CNT×4. The CNT bits is decreased by 1 when a single or a beat of the burst peripheral transfer (the source memory transfer in the memory-to-memory mode) has been completed even if the transfer mode is peripheral-to-memory or memory-to-memory.

When configuring the CNT bits, the following rules must be respected to guarantee a good DMA operation:

1. If the circular mode is disabled by clearing the CMEN bit in the DMA_CHxCTL register, the rules to configure the CNT bits in the DMA_CHxCNT register based on the transfer width are listed in the ***Table 4-2. CNT configuration***. The number of data bytes must be an integer

multiple of the memory transfer width to guarantee an integrated single memory transfer.

Note: The number of data bytes does not need to be an integer multiple of the bytes number of a memory burst transfer or a peripheral burst number if the PBURST or/and MBURST bits are not equal to '00'. The remaining data that not enough for a burst transfer can be divided into single transfer automatically.

Table 4-2. CNT configuration

PWIDTH	MWIDTH	CNT
8-bit	16-bit	Multiple of 2
8-bit	32-bit	Multiple of 4
16-bit	32-bit	Multiple of 2
Others		Any value

2. If the circular mode is enabled by setting the CMEN bit in the DMA_CHxCTL register. The number of data bytes must be an integer multiple of the byte number of a peripheral burst transfer and a memory burst transfer to guarantee an integrated memory and peripheral burst transfer:

- $CNT / PBURST_beats$ must be an integer;
- $(CNT \times PWIDTH_bytes) / (MBURST_beats \times MWIDTH_bytes)$ must be an integer.
  - PWIDTH_bytes is the byte number of the peripheral transfer width, 1 for 8-bit, 2 for 16-bit and 4 for 32-bit.
  - PBURST_beats is the beat number of a peripheral burst transfer, 1 for single burst, 4 for INCR4 (4-beat incrementing burst), 8 for INCR8 (8-beat incrementing burst) and 16 for INCR16 (16-beat incrementing burst).
  - MWIDTH_bytes is the byte number of the memory transfer width, 1 for 8-bit, 2 for 16-bit and 4 for 32-bit.
  - MBURST_beats is the beat number of a memory burst transfer, 1 for single burst, 4 for INCR4, 8 for INCR8 and 16 for INCR16.

For example:

1. If PWIDTH is 16-bit, PBURST is INCR4, MWIDTH is 8-bit and MBURST is INCR16, $CNT/4$ and $(CNT\times2)/(1\times16)$ must be an integer, so the CNT bits must be configured to the multiple of 8.
2. If the If PWIDTH is 8-bit, PBURST is INCR16, MWIDTH is 16-bit and MBURST is INCR4, $CNT/16$ and $(CNT\times1)/(2\times4)$ must be an integer, so the CNT bits must be configured to the multiple of 16.

Note: When the switch-buffer mode is enabled by setting the SBMEN bit in the DMA_CHxCTL register, the circular mode is enabled automatically by hardware, and the above rules must also be respected.

FIFO

A four-word depth FIFO is implemented as a data buffer for each DMA channel. Data reading from the source address is stored in the FIFO temporarily and transmitted to the destination through the destination port. Two data processing modes are supported depend on the FIFO configuration, including single-data mode and multi-data mode. When the transfer mode is memory-to-memory, only multi-data mode is supported to implement the DMA data processing.

Multi-data mode

The multi-data mode is selected by configuring the MDMEN bit in the DMA_CHxFCTL register to '1'.

In this mode, the DMA responds the source request when there is enough FIFO space for a source transfer, pushing the data reading from the source address into the FIFO. If the destination is a peripheral, the DMA responds the peripheral request when there is enough FIFO data for a peripheral burst transfer. If the memory is configured as the destination, the FIFO counter critical value configured in the FCCV bits of the DMA_CHxFCTL register controls the memory data processing. Only when the FIFO counter is reached the critical value, the data in the FIFO are entirely popped and written into the memory address.

To guarantee a good DMA behavior, the FIFO counter critical value (FCCV bits in the DMA_CHxFCTL register) must be an integer multiple of a memory burst transfer to ensure there is enough data for memory burst transfers. The configuration rules of the FIFO counter critical value depending on memory transfer width and memory burst types are listed in ***Table 4-3. FIFO counter critical value configuration rules***.

Table 4-3. FIFO counter critical value configuration rules

MWIDTH	MBURST	FIFO counter critical value			
		1-word	2-word	3-word	4-word
8-bit	single	4 single transactions	8 single transactions	12 single transactions	16 single transactions
	INCR4	1 burst transaction	2 burst transactions	3 burst transactions	4 burst transactions
	INCR8	ERROR	1 burst transaction	ERROR	2 burst transactions
	INCR16	ERROR	ERROR	ERROR	1 burst transaction
16-bit	single	2 single transactions	4 single transactions	6 single transactions	8 single transactions
	INCR4	ERROR	1 burst transaction	ERROR	2 burst transactions
	INCR8	ERROR	ERROR	ERROR	1 burst transaction
	INCR16	ERROR	ERROR	ERROR	ERROR
32-bit	single	1 single transaction	2 single transactions	3 single transactions	4 single transactions
	INCR4	ERROR	ERROR	ERROR	1 burst transactions
	INCR8	ERROR	ERROR	ERROR	ERROR

MWIDTH	MBURST	FIFO counter critical value			
		1-word	2-word	3-word	4-word
	INCR16	ERROR	ERROR	ERROR	ERROR

Note: When the transfer mode is peripheral-to-memory, if the PBURST_beats×PWIDTH_bytes = 16, the FIFO counter critical value must not be equal to '2b10'. When receiving a peripheral request, DMA initiates a peripheral burst transfer to entirely fill the FIFO. Then DMA launches memory burst transfers to pop three words from the FIFO depending on the FIFO counter critical value and a word is still remained in the FIFO. There is no enough space for a peripheral burst transfer and the FIFO counter critical value is not reached, which makes DMA transfer frozen.

Single-data mode

The single-data mode is selected by configuring the MDMEN bit in the DMA_CHxFCTL register to '0'. In this mode, only single transfer is supported to implement the DMA data access, and the FIFO counter critical value configured in the FCCV bits of the DMA_CHxFCTL register has no meaning.

In single-data mode, DMA responds the source request only when the FIFO is empty, pushing the data reading from the source address into the FIFO whatever the source transfer width is. When the FIFO is not empty, DMA responds the destination request, popping the data from the FIFO and writing it to the destination address.

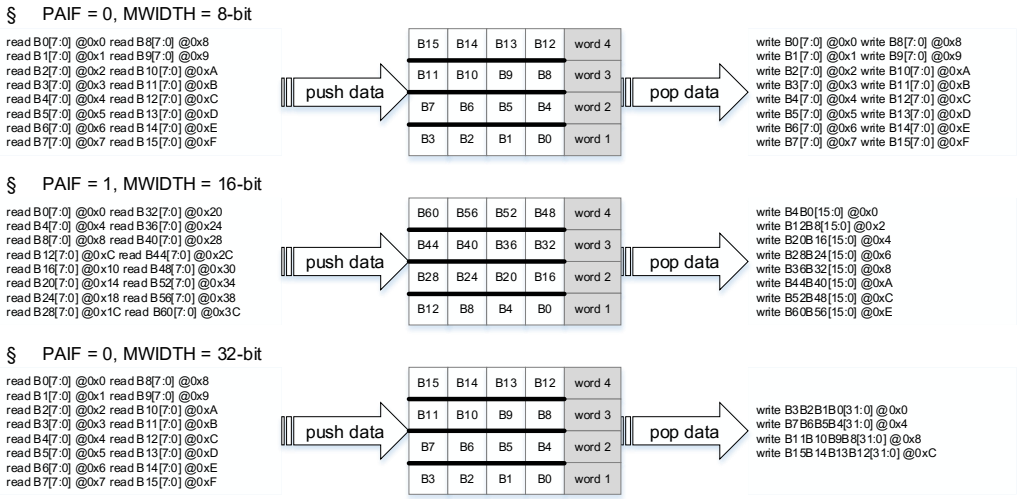
Pack/Unpack

In single-data mode, the MWIDTH bits are equal to the PWIDTH bits by force, data packing/unpacking is not needed.

In multi-data mode, the independent PWIDTH and MWIDTH bits configuration are supported for flexible DMA transfer. When the PWIDTH bits and the MWIDTH bits are not equal, DMA reading access and writing access are executed in different transfer width, and DMA packs/unpacks the data automatically. In DMA transfer operation, only little-endian addressing for both memory and peripheral is supported.

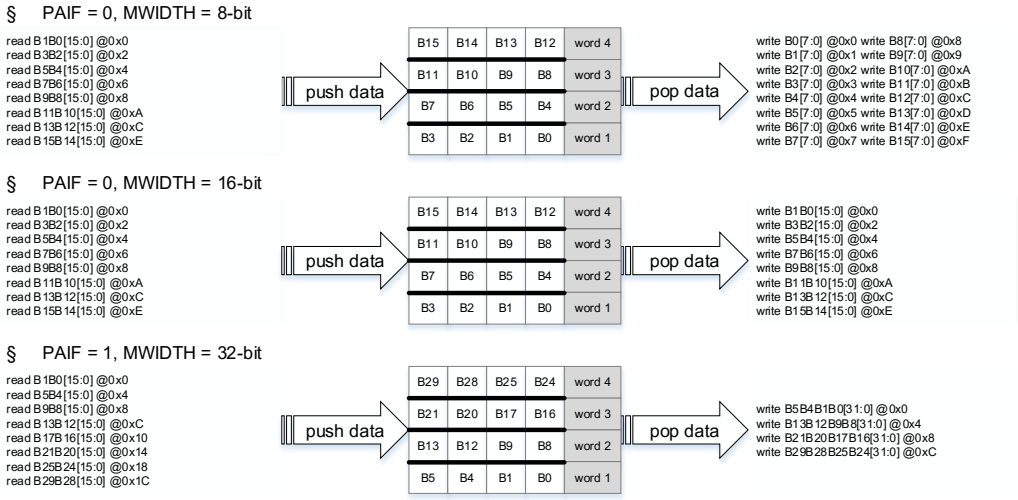
Suppose the CNT bits are 16, the PWIDTH bits are equal to '00', and both PNAGA and MNAGA are set. The DMA transfer operations for different MWIDTH are shown in the ***Figure 4-4. Data packing/unpacking when PWIDTH = '00'***.

Figure 4-4. Data packing/unpacking when PWIDTH = '00'



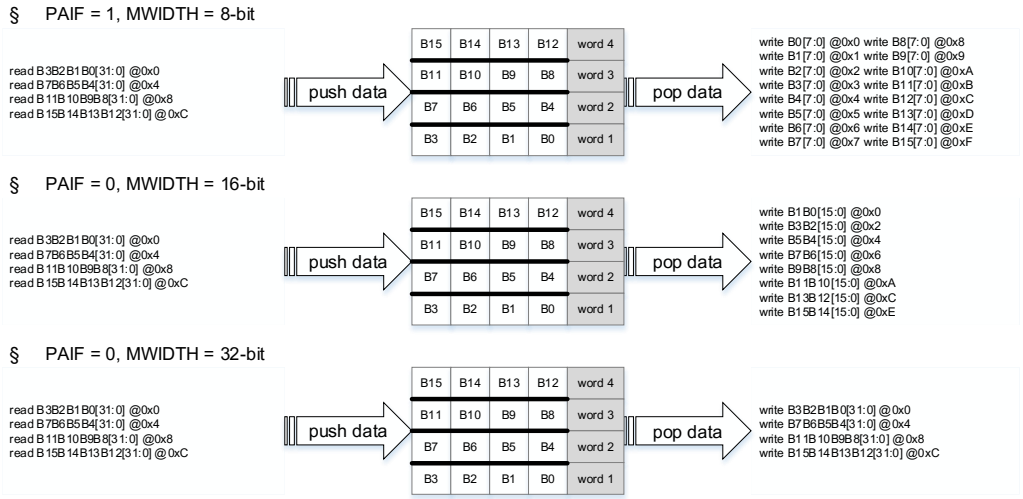
Suppose the CNT bits are 8, the PWIDTH bits are equal to '01', and both PNAGA and MNAGA are set. The DMA transfer operations for different MWIDTH are shown in the ***Figure 4-5. Data packing/unpacking when PWIDTH = '01'***.

Figure 4-5. Data packing/unpacking when PWIDTH = '01'



Suppose DMA_CHxCNT is 4, the PWIDTH bits are equal to '10', and both PNAGA and MNAGA are set. The DMA transfer operations for different MWIDTH are shown in the ***Figure 4-6. Data packing/unpacking when PWIDTH = '10'***.

Figure 4-6. Data packing/unpacking when PWIDTH = '10'



4.3.4. 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_CHxM0ADDR, and DMA_CHxM1ADDR).

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. In Multi-data mode with PBURST in the DMA_CHxCTL register is '00', if PAIF in the DMA_CHxCTL register is enabled, the next peripheral address increment is fixed to 4, and has nothing to do with the peripheral transfer data width. The PAIF has no meaning to the memory address generation.

Note: If PAIF in the DMA_CHxCTL register is enable, the peripheral base address configured in the DMA_CHxPADDR register must be 32-bit alignment.

4.3.5. Circular mode

Circular mode is implemented to handle continue peripheral requests. The CMEN bit in DMA_CHxCTL register is used to enable/disable the circular mode. Circular mode is available only when DMA controls the transfer flow.

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 respond the peripheral request until a transfer error is detected or the CHEN bit in the DMA_CHxCTL register is cleared.

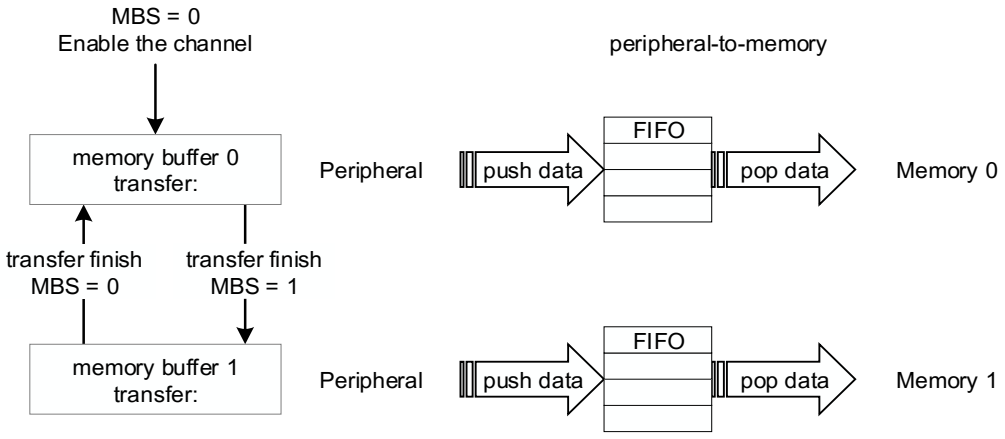
4.3.6. Switch-buffer mode

Similar to circular mode, switch-buffer mode is also implemented to handle continues peripheral requests. The SBMEN bit in the DMA_CHxCTL register is used to enable/disable the switch-buffer mode. When the switch-buffer mode is enabled, the circular mode is automatically enabled immediately after the channel is enabled. Switch-buffer mode is only available when the transfer mode is peripheral-to-memory or memory-to-peripheral. When the transfer mode is memory-to-memory, the switch-buffer mode is automatically disabled immediately after the channel is enabled.

Switch-buffer mode is supported with two memory buffers and the base address of the two memory buffers are separately configured in the DMA_CHxM0ADDR and DMA_CHxM1ADDR register. In switch-buffer mode, the DMA memory pointer switches from the current memory buffer to another at the end of every DMA transfer. During the DMA transmission, the memory buffer not being processed by DMA can be accessed by other AHB masters. In switch-buffer mode, the base address of the memory buffer not accessed by DMA can be updated even if the channel is enabled.

The MBS bit in the DMA_CHxCTL register is configured to select which memory buffer is accessed by DMA at the first DMA transfer before the channel is enabled. In switch-buffer mode, this bit switches automatically between '0' and '1' at the end of every DMA transfer, and can be used as a flag indicating the current memory buffer accessed by DMA during the transmission. The DMA operation of switch-buffer mode are shown in *Figure 4-7. DMA operation of switch-buffer mode*.

Figure 4-7. DMA operation of switch-buffer mode



4.3.7. Transfer operation

Three transfer modes are supported to implement the data transfer, including peripheral-to-memory, memory-to-peripheral and memory-to-memory. Memory and peripheral can be configured as source and destination relatively.

Memory transfer

- Peripheral-to-memory mode:
  - In single-data mode, when the FIFO is not empty, DMA initiates a single memory transfer and writes data into the corresponding memory address.
  - In multi-data mode, when the FIFO counter reaches the critical value, DMA starts single or burst memory transfers to entirely fetch the FIFO data and write to the memory.
- Memory-to-peripheral mode:
  - In single-data mode, when the channel is enabled, DMA starts a single memory transfer and pushes the reading data into the FIFO immediately. During the transmission, the memory transfer is initiated only when the FIFO is empty.
  - In multi-data mode, when the channel is enabled, DMA starts several single or burst transfers to fill up the FIFO whether the peripheral request is asserted or not. During the transmission, the memory transfer is initiated once when there is enough space for it in the FIFO.
- Memory-to-memory mode: When the FIFO counter reaches the critical value, DMA starts single or burst memory transfers to entirely fetch the FIFO data and write to the memory.

Peripheral transfer

- Peripheral-to-memory mode: When receiving a peripheral request and there is enough space in the FIFO for a peripheral transfer, DMA starts a peripheral transfer and pushes the reading data into the FIFO.
- Memory-to-peripheral mode: When receiving a peripheral request and there is enough data in the FIFO for a peripheral transfer, DMA starts a peripheral transfers to fetch the FIFO data and write to the peripheral.

4.3.8. Transfer finish

The DMA transfer is finished automatically and the FTFIFx bit in the DMA_INTF0 or DMA_INTF1 register is set when one of the following situations occurs:

- Transfer completion.
- Software clear.
- Error detection.

Transfer completion

When enabled, the DMA begins to transfer data between peripheral and memory. After the pre-programmed number of data items has been transferred successfully, the DMA transfer is completed and the CHEN bit is automatically cleared in the DMA_CHxCTL register.

- Peripheral-to-memory mode: When the CNT bits reach to zero and the contents of the FIFO have been entirely transferred into the memory, an end of transfer is generated.
- Memory-to-peripheral mode: When the CNT bits reach to zero, an end of transfer is

achieved.

- Memory-to-memory: When the CNT bits reach to zero and the contents of the FIFO have been entirely transferred into the memory, an end of transfer is generated.

Software clear

The DMA transfer can be stopped by clearing the CHEN bit in the DMA_CHxCTL register by software. After the software cleared operation, the CHEN bit is still read as 1 to indicate that there are memory or peripheral transfers still active or the remaining data in the FIFO need to be transferred.

- Peripheral-to-memory: After the software cleared operation, the peripheral transfer is stopped when the current single or burst transfer is completed. To ensure that the data had been read from peripheral can be entirely transferred into the memory, the memory transfer continues to be active until the FIFO is empty. If the remaining byte number in the FIFO is not enough for a burst memory transfer, these data items are transferred in single transaction. If the remaining byte number is less than the memory transfer width, these data items are still written in memory transfer width with MSBs filled with zero. The software can read the CNT bits to calculate the number of valid data items in the memory. After the contents of the FIFO has been entirely transferred into the memory, the CHEN bit is cleared automatically by hardware and the FTFIFx bit in the DMA_INTF0 or DMA_INTF1 register is set.
- Memory-to-peripheral: After the software cleared operation, the DMA transfer is stopped when the current memory and peripheral transfer are completed. Then the CHEN bit is cleared automatically by hardware and the FTFIFx bit in the DMA_INTF0 or DMA_INTF1 register is set.
- Memory-to-memory: The same as the peripheral-to-memory mode with the source memory transfer is implemented through the peripheral port.

Error detection

Three types error can disable the DMA transfer:

- FIFO error: When a wrong FIFO configuration is detected, the DMA channel is disabled immediately without starting any transfers. In this situation, the FTFIFx is not asserted. For more information about the FIFO error, refer to section ***Error***.
- Bus error: When the memory or peripheral port attempts to access an address beyond the access scope, a bus error is detected and the DMA transfer is stopped immediately without setting the FTFIFx. If this error is aroused by the peripheral port, the CNT bits are still decreased by 1. For more information about the bus error, refer to section ***Error***.
- Register access error: In switch-buffer mode, an access error is detected when a write command is active on the memory base address register which is being accessed by DMA. When this error occurs, the DMA operation is the same as it after the CHEN bit software cleared. For more information about the register access error, refer to section ***Error***.

4.3.9. 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 or wait the current DMA transfer finished. When the CHEN bit is read as '0', configuring and starting a new DMA transfer is allowed.
2. Clear the FTFIFx bit in the DMA_INTF0 or DMA_INTF1 register, or a new DMA transfer can not be re-enabled.
3. Configure the TM bits in the DMA_CHxCTL register to set the transfer mode.
4. Configure the memory and peripheral burst types, the target memory buffer, switch-buffer mode, priority of the channel, memory and peripheral transfer width, memory and peripheral address generation algorithm, circular mode in the DMA_CHxCTL register.
5. Configure multi-data mode, and the FCCV bits to set the FIFO counter critical value if multi-data mode is enabled in the DMA_CHxFCTL register.
6. Configure the enable bit for full transfer finish interrupt, half transfer finish interrupt, transfer access error interrupt, single-data mode exception interrupt in the DMA_CHxCTL register and the enable bit for FIFO error and exception interrupt in the DMA_CHxFCTL register.
7. Configure the DMA_CHxPADDR register for setting the peripheral base address.
8. If the switch-buffer mode is enabled, configure the memory base address in DMA_CHxM0ADDR and DMA_CHxM1ADDR register . If only one memory buffer is to be used, configure the DMA_CHxM0ADDR or DMA_CHxM1ADDR corresponding with the MBS bit in the DMA_CHxCTL register.
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.

When restarting the suspended DMA transfer, it is recommended to respect the following steps:

1. Read the CHEN bit and ensure the DMA suspend operation has been completed. When the CHEN bit is read as '0', DMA is idle, restarting the DMA transfer is allowed.
2. Clear the FTFIFx bit in the DMA_INTF0 or DMA_INTF1 register, or the DMA transfer can not be re-enabled.
3. Read the DMA_CHxCNT register to obtain the number of the remaining data items and calculate the number of the data items had already been transferred.
4. Configure the DMA_CHxPADDR register to update the peripheral address pointer.
5. Configure the DMA_CHxM0ADDR or the DMA_CHxM1ADDR register to update the memory address pointer.

6. Configure the DMA_CHxCNT with the number of the remaining data items.
7. Configure the CHEN bit with '1' in the DMA_CHxCTL register to restart the channel.

4.4. Interrupts

Each DMA channel has a dedicated interrupt. There are five interrupt events connected to each interrupt, including full transfer finish interrupt, half transfer finish interrupt, transfer access error interrupt, single-data mode exception interrupt, and FIFO error and exception interrupt. A DMA channel interrupt may be produced when any interrupt event occurs on the channel.

Each interrupt event has a dedicated flag bit in the DMA_INTF0 or DMA_INTF1 register, a dedicated clear bit in the DMA_INTC0 and DMA_INTC1 register, and a dedicated enable bit in the DMA_CHxCTL and CHxFCTL register, as described in the ***Table 4-4. DMA interrupt events***.

Table 4-4. DMA interrupt events

Interrupt event	Flag bit	Enable bit	Clear bit
	DMA_INTF0 or DMA_INTF1	DMA_CHxCTL or DMA_CHxFCTL	DMA_INTC0 or DMA_INTC1
Full transfer finish	FTFIF	FTFIE	FTFIFC
Half transfer finish	HTFIF	HTFIE	HTFIFC
Transfer access error	TAEIF	TAEIE	TAEIFC
Single-data mode exception	SDEIF	SDEIE	SDEIFC
FIFO error and exception	FEEIF	FEEIE	FEEIFC

These five events can be divided into three types:

- Flag: Full transfer finish flag and half transfer finish flag.
- Exception: Single-data mode exception and FIFO exception.
- Error: Transfer access error and FIFO error.

When the exception events occur, the DMA transmission is not affected and continues transferring normally. When the error events are detected, the DMA transmission is stopped. These three types of event are described in detail in the following sections.

4.4.1. Flag

Two flag events are supported, including full transfer finish flag and half transfer finish flag.

The full transfer finish flag is asserted, when one of the following situations occurs:

- The CNT bits reach to zero.
- When the channel is disabled by software before the end of the transfer, the current memory and peripheral is completed and the contents of the FIFO are entirely written

into the memory in peripheral-to-memory or memory-to-memory mode.

- When the channel is disabled because of register access error before the end of the transfer, the current memory and peripheral is completed and the contents of the FIFO are entirely written into the memory in peripheral-to-memory or memory-to-memory mode.

When the full transfer finish flag is asserted and the enabled bit for the full transfer finish interrupt is set, an interrupt is generated.

The half transfer finish flag is asserted, only when DMA is the transfer flow controller and half of the CNT bits are transferred. If peripheral is the transfer flow controller, DMA does not know when half of data items has been transferred and the half transfer finish flag will stay zero.

When the half transfer finish flag is asserted and the enabled bit for the half transfer finish interrupt is set, an interrupt is generated.

4.4.2. Exception

Two exception events are supported, including single-data mode exception and FIFO exception. These exceptions have no effect on the DMA transmission.

Single-data mode exception

This exception can be detected only when the single-data mode is enabled and the transfer mode is peripheral-to-memory. When a peripheral request is valid and the FIFO is not empty, there are two or more data items stored in the FIFO after responding the peripheral request, which could be a problem for the subsequent processing of the data and the single-data mode exception bit SDEIFx will be set.

When the single-data mode exception is asserted and the enabled bit for the single-data mode exception interrupt is set, an interrupt is generated.

FIFO exception

When a FIFO underrun or a FIFO overrun condition occurs, the FIFO exception is asserted. This exception can be detected only when the transmission is between peripheral and memory.

In peripheral-to-memory mode, when a peripheral request is valid and there is not enough space in the FIFO for the single or burst peripheral transfer, a FIFO overrun condition is detected. This peripheral request is not responded until the FIFO space is enough, and the accuracy of the data transmission will not be destroyed.

In memory-to-peripheral mode, when a peripheral request is valid and there is not enough data in the FIFO for the single or burst peripheral, a FIFO underrun condition is detected. This peripheral request is not responded until the data number in the FIFO is enough, and the accuracy of the data transmission will not be destroyed.

When the FIFO exception is asserted and the enabled bit for the FIFO error and exception interrupt is set, an interrupt is generated.

4.4.3. Error

FIFO error and transfer access error (including the register access error and bus error) can be detected during the DMA transmission, and the transmission can be stopped when one of the errors occurs.

FIFO error

For a DMA operation, when the multi-data mode is enabled, the right and wrong configurations of the FIFO counter critical value corresponding with the memory transfer width and memory burst types are listed in ***Table 4-3. FIFO counter critical value configuration rules***.

If a wrong configuration is detected after enable the channel, a FIFO error is generated and the channel is disabled immediately without starting any transfers.

When the FIFO error is asserted and the enabled bit for the FIFO error and exception interrupt is set, an interrupt is generated.

Register access error

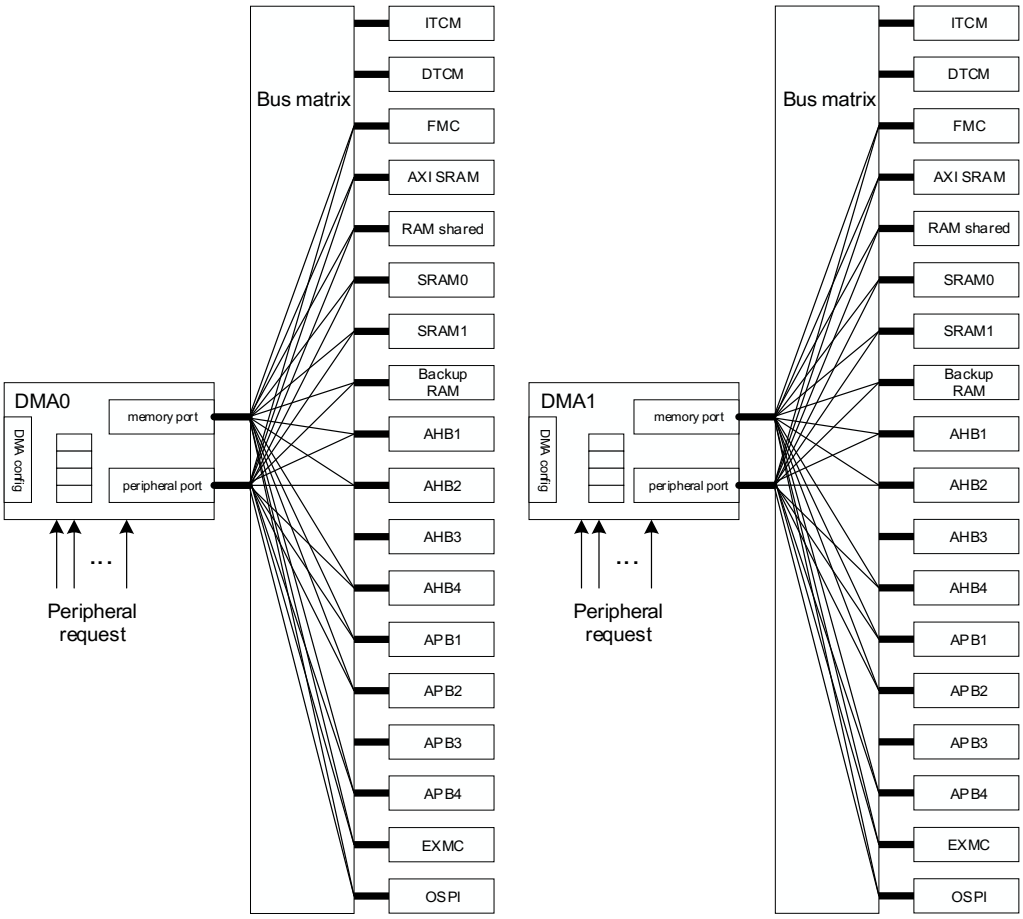
The register access error is detected only when the switch-buffer is enabled. If the software attempts to update a memory address register currently accessed by the DMA controller, a register access error is detected. For example, when the memory 0 buffer is the current source or destination, a write access on the DMA_CHxM0ADDR register could produce a register access error. When a register access error occurs, the DMA transmission is stopped when the current memory and peripheral transfer are completed and the valid FIFO data are entirely drained into the memory if needed.

When the register access error is asserted and the enabled bit for the transfer access error and exception interrupt is set, an interrupt is generated.

Bus error

When the address accessed by the DMA controller is beyond the allowed area, a response error will be received and the channel is disabled immediately. The allowed and forbidden access region for DMA0 and DMA1 are shown in ***Figure 4-8. System connection of DMA0 and DMA1***. When the bus error is asserted and the enabled bit for the transfer access error and exception interrupt is set, an interrupt is generated.

Figure 4-8. System connection of DMA0 and DMA1



4.4.4. 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, refers to ***Table 6-2. Request multiplexer input mapping***.

4.5. Register definition

DMA0 base address: 0x4002 0000

DMA1 base address: 0x4002 0400

4.5.1. Interrupt flag register 0 (DMA_INTF0)

Address offset: 0x00

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				FTFIF3	HTFIF3	TAEIF3	SDEIF3	Reserved	FEEIF3	FTFIF2	HTFIF2	TAEIF2	SDEIF2	Reserved	FEEIF2
				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
Reserved				FTFIF1	HTFIF1	TAEIF1	SDEIF1	Reserved	FEEIF1	FTFIF0	HTFIF0	TAEIF0	SDEIF0	Reserved	FEEIF0
				r	r	r	r		r	r	r	r	r		r

Bits	Fields	Descriptions
31:28 15:12	Reserved	Must be kept at reset value.
27/21/11/5	FTFIFx	Full Transfer finish flag of channel x (x=0…3) Hardware set and software cleared by writing 1 to the corresponding bit in DMA_INTC0 register. 0: Transfer has not finished on channel x 1: Transfer has finished on channel x
26/20/10/4	HTFIFx	Half transfer finish flag of channel x (x=0…3) Hardware set and software cleared by writing 1 to the corresponding bit in DMA_INTC0 register. 0: Half number of transfer has not finished on channel x 1: Half number of transfer has finished on channel x
25/19/9/3	TAEIFx	Transfer access error flag of channel x (x=0…3) Hardware set and software cleared by writing 1 to the corresponding bit in DMA_INTC0 register. 0: Transfer access error has not occurred on channel x 1: Transfer access error has occurred on channel x
24/18/8/2	SDEIFx	Single data mode exception of channel x (x=0…3) Hardware set and software cleared by writing 1 to the corresponding bit in DMA_INTC0 register. 0: Single data mode exception has not occurred on channel x 1: Single data mode exception has occurred on channel x

23/17/7/1	Reserved	Must be kept at reset value.
22/16/6/0	FEEIFx	FIFO error and exception of channel x (x=0…3) Hardware set and software cleared by writing 1 to the corresponding bit in DMA_INTC0 register. 0: FIFO error or exception has not occurred on channel x 1: FIFO error or exception has occurred on channel x

4.5.2. Interrupt flag register 1 (DMA_INTF1)

Address offset: 0x04

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				FTFIF7	HTFIF7	TAEIF7	SDEIF7	Reserved	FEEIF7	FTFIF6	HTFIF6	TAEIF6	SDEIF6	Reserved	FEEIF6
				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
Reserved				FTFIF5	HTFIF5	TAEIF5	SDEIF5	Reserved	FEEIF5	FTFIF4	HTFIF4	TAEIF4	SDEIF4	Reserved	FEEIF4
				r	r	r	r		r	r	r	r	r		r

Bits	Fields	Descriptions
31:28 15:12	Reserved	Must be kept at reset value.
27/21/11/5	FTFIFx	Full Transfer finish flag of channel x (x=4…7) Hardware set and software cleared by writing 1 to the corresponding bit in DMA_INTC1 register. 0: Transfer has not finished on channel x 1: Transfer has finished on channel x
26/20/10/4	HTFIFx	Half transfer finish flag of channel x (x=4…7) Hardware set and software cleared by writing 1 to the corresponding bit in DMA_INTC1 register. 0: Half number of transfer has not finished on channel x 1: Half number of transfer has finished on channel x
25/19/9/3	TAEIFx	Transfer access error flag of channel x (x=4…7) Hardware set and software cleared by writing 1 to the corresponding bit in DMA_INTC1 register. 0: Transfer access error has not occurred on channel x 1: Transfer access error has occurred on channel x
24/18/8/2	SDEIFx	Single data mode exception of channel x (x=4…7) Hardware set and software cleared by writing 1 to the corresponding bit in DMA_INTC1 register. 0: Single data mode exception has not occurred on channel x

		1: Single data mode exception has occurred on channel x
23/17/7/1	Reserved	Must be kept at reset value.
22/16/6/0	FEEIFx	FIFO error and exception of channel x (x=4…7) Hardware set and software cleared by writing 1 to the corresponding bit in DMA_INTC1 register. 0: FIFO error or exception has not occurred on channel x 1: FIFO error or exception has occurred on channel x

4.5.3. Interrupt flag clear register 0 (DMA_INTC0)

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				FTFIFC3	HTFIFC3	TAEIFC3	SDEIFC3	Reserved	FEEIFC3	FTFIFC2	HTFIFC2	TAEIFC2	SDEIFC2	Reserved	FEEIFC2
				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
Reserved				FTFIFC1	HTFIFC1	TAEIFC1	SDEIFC1	Reserved	FEEIFC1	FTFIFC0	HTFIFC0	TAEIFC0	SDEIFC0	Reserved	FEEIFC0
				w	w	w	w		w	w	w	w	w		w

Bits	Fields	Descriptions
31:28 15:12	Reserved	Must be kept at reset value.
27/21/11/5	FTFIFCx	Clear bit for Full transfer finish flag of channel x (x=0…3) 0: No effect 1: Clear full transfer finish flag
26/20/10/4	HTFIFCx	Clear bit for half transfer finish flag of channel x (x=0…3) 0: No effect 1: Clear half transfer finish flag
25/19/9/3	TAEIFCx	Clear bit for transfer access error flag of channel x (x=0…3) 0: No effect 1: Clear transfer access error flag
24/18/8/2	SDEIFCx	Clear bit for single data mode exception of channel x (x=0…3) 0: No effect 1: Clear single data mode exception flag
23/17/7/1	Reserved	Must be kept at reset value.
22/16/6/0	FEEIFCx	Clear bit for FIFO error and exception of channel x (x=0…3) 0: No effect 1: Clear FIFO error and exception flag

4.5.4. Interrupt flag clear register 1 (DMA_INTC1)

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				FTFIFC7	HTFIFC7	TAEIFC7	SDEIFC7	Reserved	FEEIFC7	FTFIFC6	HTFIFC6	TAEIFC6	SDEIFC6	Reserved	FEEIFC6
				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
Reserved				FTFIFC5	HTFIFC5	TAEIFC5	SDEIFC5	Reserved	FEEIFC5	FTFIFC4	HTFIFC4	TAEIFC4	SDEIFC4	Reserved	FEEIFC4
				w	w	w	w		w	w	w	w	w		w

Bits	Fields	Descriptions
31:28 15:12	Reserved	Must be kept at reset value.
27/21/11/5	FTFIFCx	Clear bit for full transfer finish flag of channel x (x=4…7) 0: No effect 1: Clear full transfer finish flag
26/20/10/4	HTFIFCx	Clear bit for half transfer finish flag of channel x (x=4…7) 0: No effect 1: Clear half transfer finish flag
25/19/9/3	TAEIFCx	Clear bit for transfer access error flag of channel x (x=4…7) 0: No effect 1: Clear transfer access error flag
24/18/8/2	SDEIFCx	Clear bit for single data mode exception of channel x (x=4…7) 0: No effect 1: Clear single data mode exception flag
23/17/7/1	Reserved	Must be kept at reset value.
22/16/6/0	FEEIFCx	Clear bit for FIFO error and exception of channel x (x=4…7) 0: No effect 1: Clear FIFO error and exception flag

4.5.5. Channel x control register (DMA_CHxCTL)

x = 0...7, where x is a channel number

Address offset: 0x10 + 0x18 × 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						RREN	MBURST[1:0]		PBURST[1:0]		Reserved	MBS	SBMEN	PRIO[1:0]	
						rw	rw		rw			rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PAIF	MWIDTH[1:0]		PWIDTH[1:0]		MNAGA	PNAGA	CMEN	TM[1:0]		TFCS	FTFIE	HTFIE	TAEIE	SDEIE	CHEN
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	RREN	Round-robin group arbitration for "ultra high" priority level channels enable Software set and clear. 0: Disable round-robin group arbitration for "ultra high" priority level channels. 1: Enable round-robin group arbitration for "ultra high" priority level channels. This bit cannot be written when CHEN is '1'. This bit can only be operated through DMA_CH0CTL. After this bit is enabled, it will apply to all channels.
24:23	MBURST[1:0]	Transfer burst type of memory Software set and clear. 00: single burst 01: INCR4 (4-beat incrementing burst) 10: INCR8 (8-beat incrementing burst) 11: INCR16 (16-beat incrementing burst) These bits can not be written when CHEN is '1'. These bits are automatically locked as '00' by hardware immediately after enable CHEN if MDMEN in the DMA_CHxFCTL register is configured to '0'.
22:21	PBURST[1:0]	Transfer burst type of peripheral Software set and clear 00: single burst 01: INCR4 (4-beat incrementing burst) 10: INCR8 (8-beat incrementing burst) 11: INCR16 (16-beat incrementing burst) These bits can not be written when CHEN is '1'. These bits are automatically locked as '00' by hardware immediately after enable CHEN if MDMEN in the DMA_CHxFCTL register is configured to '0'.
20	Reserved	Must be kept at reset value.
19	MBS	Memory buffer select Hardware and software set, hardware and software clear. 0: Memory 0 is selected as memory transfer area 1: Memory 1 is selected as memory transfer area This bit can not be written when CHEN is '1'.

		During the transmission, this bit can be set and cleared by hardware at the end of transfer to indicate which memory buffer is being accessed by DMA.
18	SBMEN	Switch-buffer mode enable Software set and clear. 0: Disable switch-buffer mode 1: Enable switch-buffer mode This bit can not be written when CHEN is '1'.
17:16	PRIO[1:0]	Priority level Software set and clear. 00: Low 01: Medium 10: High 11: Ultra high These bits can not be written when CHEN is '1'.
15	PAIF	Peripheral address increment fixed Software set and clear. 0: The peripheral address increment is determined by PWIDTH 1: The peripheral address increment is fixed to 4 This bit can not be written when CHEN is '1'. During the transmission, when PNAGA is configured to '0', this bit has no effect. These bits are automatically locked as '0' by hardware immediately after enable CHEN if MDMEN in the DMA_CHxFCTL register is configured to '0' or PBURST are not equal to '00'.
14:13	MWIDTH[1:0]	Transfer width of memory Software set and clear. 00: 8-bit 01: 16-bit 10: 32-bit 11: Reserved These bits can not be written when CHEN is '1'. These bits are automatically locked as PWIDTH by hardware immediately after enable CHEN if MDMEN in the DMA_CHxFCTL register is configured to '0'.
12:11	PWIDTH[1:0]	Transfer width of peripheral Software set and clear. 00: 8-bit 01: 16-bit 10: 32-bit 11: Reserved These bits can not be written when CHEN is '1'.
10	MNAGA	Next address generation algorithm of memory Software set and clear

		0: Fixed address mode 1: Increasing address mode This bit can not be written when CHEN is '1'.
9	PNAGA	Next address generation algorithm of peripheral Software set and clear 0: Fixed address mode 1: Increasing address mode This bit can not be written when CHEN is '1'.
8	CMEN	Circular mode enable Software set and clear. 0: Disable circular mode. 1: Enable circular mode This bit can not be written when CHEN is '1'. This bit is automatically locked as '1' by hardware immediately after enable CHEN if SBMEN is configured to '1'.
7:6	TM[1:0]	Transfer mode Software set and clear. 00: Read from peripheral and write to memory 01: Read from memory and write to peripheral 10: Read from memory and write to memory 11: Reserved These bits can not be written when CHEN is '1'.
5	TFCS	Transfer flow controller select Software set and clear. 0: DMA is selected as the transfer flow controller 1: Peripheral is selected as the transfer flow controller This bit can not be written when CHEN is '1'.
4	FTFIE	Enable bit for full transfer finish interrupt Software set and clear. 0:Disable full transfer finish interrupt 1:Enable full transfer finish interrupt
3	HTFIE	Enable bit for half transfer finish interrupt Software set and clear. 0: Disable half transfer finish interrupt 1: Enable half transfer finish interrupt
2	TAEIE	Enable bit for tranfer access error interrupt Software set and clear. 0: Disable tranfer access error interrupt 1: Enable tranfer access error interrupt
1	SDEIE	Enable bit for single data mode exception interrupt

		Software set and clear. 0: Disable single data mode exception interrupt 1: Enable single data mode exception interrupt
0	CHEN	Channel enable Software set, hardware clear. 0: Disable channel 1: Enable channel When this bit is asserted, the DMA transfer is started. This bit is automaticlly cleared when one of the following situations occurs: When the transfer of channel is fully finished. When a wrong FIFO configuration or a transfer access error is detected. After a software clear operation, this bit is still read as 1 to indicate that there are memory or peripheral transfers still active until hardware has terminated all activity, at which point this bit is read as 0. Software can therefore poll this bit to determine when this channel is free for a new DMA transfer.

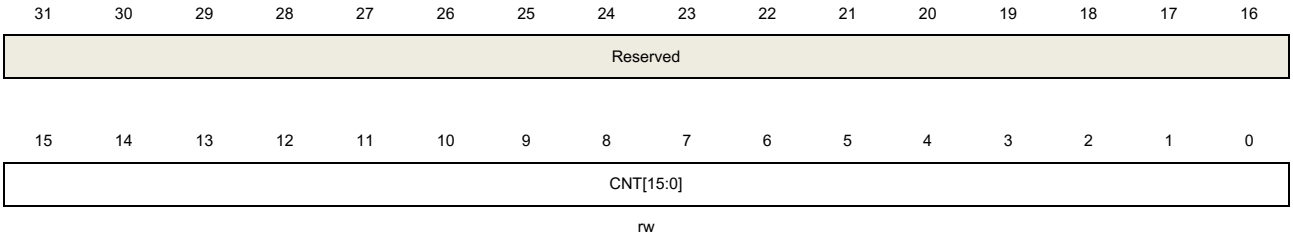
4.5.6. Channel x counter register (DMA_CHxCNT)

x = 0...7, where x is a channel number

Address offset: 0x14 + 0x18 × 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'. These bits are related to PWIDTH. During the transmission, These bits signify the number of remaining data to be transferred. After each DMA peripheral transfer, CNT is decremented by 1. If CMEN or SBMEN in the DMA_CHxCTL register is configured to '1', CNT can be reloaded automatically to the original value at the end of transfer.

4.5.7. Channel x peripheral base address register (DMA_CHxPADDR)

x = 0...7, where x is a channel number

Address offset: 0x18 + 0x18 × 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. Note: If PAIF in the DMA_CHxCTL register is enable, these bits must be configured to 32-bit alignment.

4.5.8. Channel x memory 0 base address register (DMA_CHxM0ADDR)

x = 0...7, where x is a channel number

Address offset: 0x1C + 0x18 × 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
M0ADDR[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
M0ADDR[15:0]															
rw															

Bits	Fields	Descriptions
31:0	M0ADDR[31:0]	Memory 0 base address When MBS in the DMA_CHxCTL register is read as to '0', these bits specify the memory base address accessed by DMA during the transmission. These bits can not be written when CHEN in the DMA_CHxCTL register is '1' and MBS in the DMA_CHxCTL register is read as '0'. When memory 0 is selected as memory transfer area and 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 memory 0 is selected as memory transfer area and MWIDTH in the DMA_CHxCTL register is 10 (32-bit), the two LSBs of these bits are ignored. Access is automatically aligned to a word address.

4.5.9. Channel x memory 1 base address register (DMA_CHxM1ADDR)

x = 0...7, where x is a channel number

Address offset: 0x20 + 0x18 × 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
M1ADDR[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
M1ADDR[15:0]															
rw															

Bits	Fields	Descriptions
31:0	M1ADDR[31:0]	Memory 1 base address When MBS in the DMA_CHxCTL register is read as to '1', these bits specific the memory base address accessed by DMA during the transmission. These bits can not be written when CHEN in the DMA_CHxCTL register is '1' and MBS in the DMA_CHxCTL register is read as '1'. When memory 1 is selected as memory tranfer area and 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 memory 1 is selected as memory tranfer area and MWIDTH in the DMA_CHxCTL register is 10 (32-bit), the two LSBs of these bits are ignored. Access is automatically aligned to a word address.

4.5.10. Channel x FIFO control register (DMA_CHxFCTL)

x = 0...7, where x is a channel number

Address offset: 0x24 + 0x18 × x

Reset value: 0x0000 0021

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								FEEIE	Reserved	FCNT[2:0]			MDMEN	FCCV[1:0]	
								rw		r			rw	rw	

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7	FEEIE	Enable bit for FIFO error and exception interrupt Software set and clear. 0: Disable FIFO error and exception interrupt 1: Enable FIFO error and exception interrupt
6	Reserved	Must be kept at reset value.
5:3	FCNT[2:0]	FIFO counter Hardware set and clear. 000: FIFO is not empty and contains less than 1 word. 001: FIFO contains more than 1 word but less than 2 words. 010: FIFO contains more than 2 words but less than 3 words. 011: FIFO contains more than 3 words but less than 4 words. 100: Empty 101: Full 110~111: Reserved These bits specific the number of data stored in FIFO during the transmission. When MDMEN is configured to '0', these bits has no meaning.
2	MDMEN	Multi-data mode enable Software set and clear. 0: Disable Multi-data mode 1: Enable Multi-data mode These bits can not be written when CHEN in the DMA_CHxCTL register is '1'. These bits are automatically locked as '1' by hardware immediately after enable CHEN in the DMA_CHxCTL register if TM in the DMA_CHxCTL register is configured to '10'.
1:0	FCCV[1:0]	FIFO counter critical value Software set and clear 00: One word 01: Two Words 10: Three Words 11: Four Words These bits can not be written when CHEN in the DMA_CHxCTL register is '1'. When MDMEN is configured to '0', these bits has no meaning.

5. Master direct memory access controller (MDMA)

5.1. Overview

The master direct memory access (MDMA) controller provides a hardware method of transferring data between peripherals and/or memory without intervention from the CPU, thereby increasing system performance by off-loading the CPU from copying large amounts of data and avoiding frequent interrupts to serve peripherals needing more data or having available data.

An AXI master interface, an AHB master interface and two 16 depth 64-bit width FIFOs are presented in MDMA controller, which achieves a high MDMA transmission performance. The AXI interface is used for main memory and peripheral register access (system access port), and the AHB interface is used for Cortex®-M7 TCM memory access (TCM access port). MDMA can be used in combination with a DMA controller (DMA0 or DMA1). The MDMA can provide up to 16 channels. Each channel request can be selected among any request source. The built-in arbiter is used to handle priority among MDMA requests.

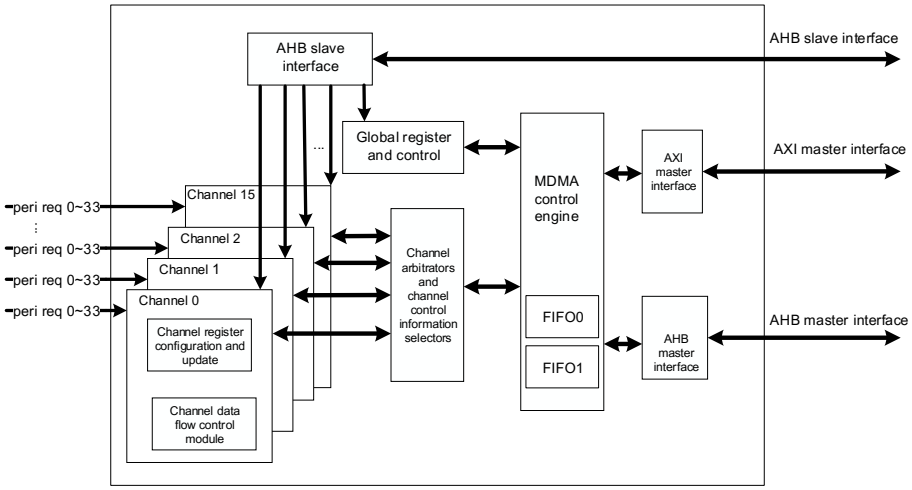
5.2. Characteristics

- AXI / AHB master interfaces, the AXI bus interface is used for transferring data between peripherals and memory, and the AHB bus interface is used for accessing Cortex®-M7 TCM memory.
- 16 channels and each channel supports software triggering and requests can be selected among any request source.
- Support independent single, 2, 4, 8, 16, 32, 64, 128-beat incrementing burst source and destination transfer.
- Software MDMA channel priority (low, medium, high, ultra high) and hardware MDMA channel priority (the lower the channel number, the higher the priority).
- Support independent 8, 16, 32, 64-bit width source and destination transfer.
- Support independent fixed, increasing and decreasing address generation algorithm of source and destination.
- The data length and address increment of the source and destination can be configured.
- Support three transfer modes:
  - Read from memory and write to memory (software triggered).
  - Read from peripheral and write to memory (or memory mapped peripherals).
  - Read from memory (or memory mapped peripherals) and write to peripheral.
- Automatic pack / unpack of data to optimize bandwidth when the data width of the source and destination are different.
- 27 hardware trigger sources, all channels can be connected to any hardware trigger source
- Two FIFOs of 16 double word depth to maximize data bandwidth and bus utilization.

- The AHB bus interface is used to access Cortex®-M7 TCM memory. And only when the increment and data size are identical and lower than or equal to 32-bit, burst access is allowed. When the increment and data size is larger than 32 bits, burst access is prohibited.
- Five types of event flags and independent interrupts for each channel that can be enabled and cleared.

5.3. Function overview

Figure 5-1. Block diagram of MDMA



As is shown in *Figure 5-1. Block diagram of MDMA*, MDMA controller consists of four parts:

- AHB slave interface for MDMA configuration;
- An AXI master interface and an AHB master interface for data transmission;
- Arbiter inside to manage requests coming at the same time;
- Data processing and counting.

The MDMA controller transfers data from one address to another without CPU intervention. It supports multiple data sizes, burst types, address generation algorithm, priority levels and several transfer modes to allow for flexible application by configuring the corresponding bits in MDMA registers. All the MDMA registers can be 32-bit configured through AHB slave interface.

TRIGMOD[1:0] in the MDMA_CHxCFG register determines the data transfer mode of MDMA, as shown in *Table 5-1. Transfer mode*.

Table 5-1. Transfer mode

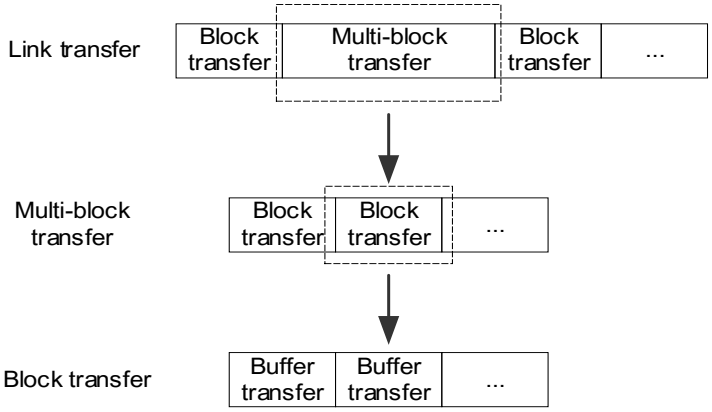
Transfer mode	TRIGMOD[1:0]
Buffer transfer	00
Block transfer	01
Multi-block transfer	10

Transfer mode	TRIGMOD[1:0]
Link transfer	11

- Buffer transfer can transmit up to 128 bytes at a time.
- Block transfer can transmit a maximum of 64KB at a time. The number of bytes to be transferred can be configured by TBNUM[16:0] in the MDMA_CHxBTCFG register. The transfer process is automatically split into multiple buffer transfers by the hardware.
- Multi-block transfer includes multiple block transfers. The number of blocks to be transferred can be configured by BRNUM[11:0] in the MDMA_CHxBTCFG register.
- Link transfer contains multiple multi-block/block transfers and the link address can be configured in the MDMA_CHxLADDR register.

The connections of the four modes is shown in ***Figure 5-2. Connections of the four modes***.

Figure 5-2. Connections of the four modes



The MDMA controller has 16 channels, each channel supports software triggering and can be selected between any of the request sources shown in ***Table 5-2. MDMA hardware request sources***. The channel x hardware trigger source can be selected by configuring the TRIGSEL[5:0] bit field in the MDMA_CHxCTL1 register.

Table 5-2. MDMA hardware request sources

request sources TRIGSEL[5:0]	source
0	DMA0_CH0_TRIG
1	DMA0_CH1_TRIG
2	DMA0_CH2_TRIG
3	DMA0_CH3_TRIG
4	DMA0_CH4_TRIG
5	DMA0_CH5_TRIG
6	DMA0_CH6_TRIG
7	DMA0_CH7_TRIG
8	DMA1_CH0_TRIG
9	DMA1_CH1_TRIG
10	DMA1_CH2_TRIG

request sources TRIGSEL[5:0]	source
11	DMA1_CH3_TRIG
12	DMA1_CH4_TRIG
13	DMA1_CH5_TRIG
14	DMA1_CH6_TRIG
15	DMA1_CH7_TRIG
16	TLI_INT
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	OSPI0_FT
23	OSPI0_TC
24	IPA_CLUT_TRIG
25	IPA_TC_TRIG
26	IPA_TWM_TRIG
27	Reserved
28	Reserved
29	SDIO0_DATA_END
30	SDIO0_BUF_END
31	SDIO0_CMD_END
32	OSPI1_FT
33	OSPI1_TC

5.3.1. Data process

Arbitration

MDMA manages requests based on channel request priority through an arbiter. When more than one requests are received at the same time, the arbiter determines which channel is selected to respond according to the following priority rules:

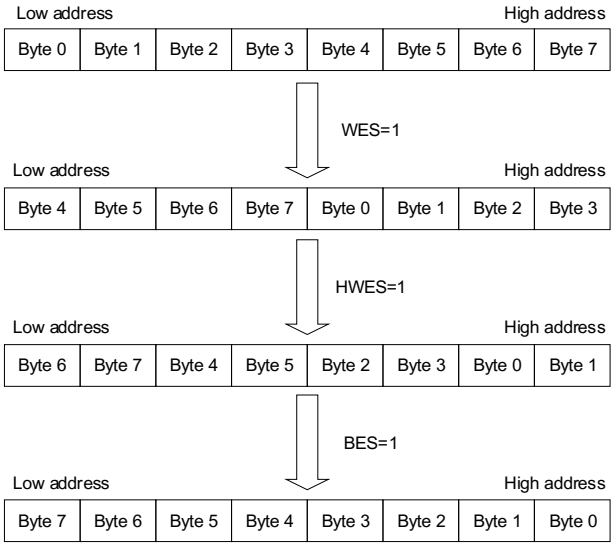
- Software priority: Four levels, including low, medium, high and ultra high by configuring the PRIO[1:0] bits in the MDMA_CHxCTL0 register.
- Hardware priority: For channels with equal software priority level, priority is given to the channel with lower channel number. For example, if channel 0 and channel 2 are configured with the same software priority, the priority of channel 0 is higher than that of channel 2.

Data type

The word, halfword, and byte exchange operations on the target data can be configured by

the WES/HWES/BES bits in the MDMA_CHxCTL0 register. The data exchange process is shown in *Figure 5-3. Word, halfword, byte order exchange*.

Figure 5-3. Word, halfword, byte order exchange

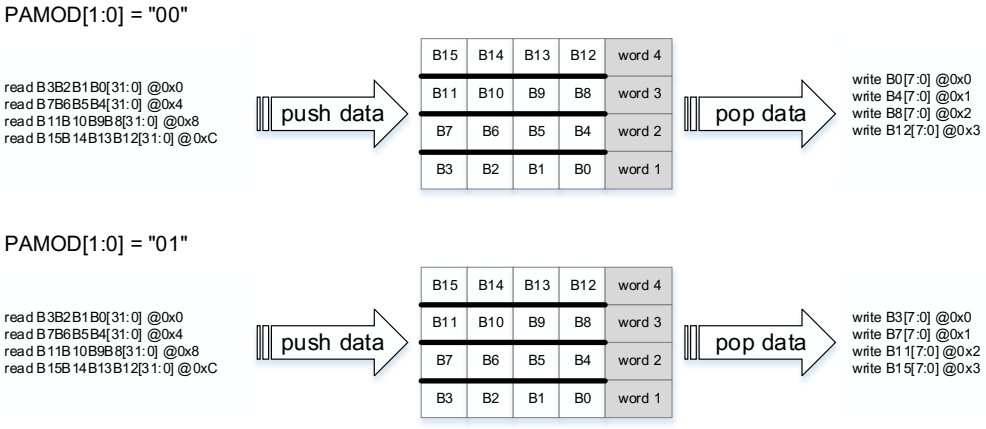


Transfer width

The SWIDTH[1:0] and DWIDTH[1:0] bit fields of MDMA_CHxCFG register determine the data width of source and destination respectively. MDMA controller supports data widths of 8, 16, 32 and 64 bits. When PKEN is enabled and the data width SWIDTH[1:0] and DWIDTH[1:0] are not equal, MDMA packs/unpacks data automatically for data transmission in order to optimize the bandwidth. When PKEN is disabled and the data width SWIDTH[1:0] and DWIDTH[1:0] are not equal, the padding and alignment mode can be selected by configuring PAMOD[1:0] bits in the MDMA_CHxCFG register.

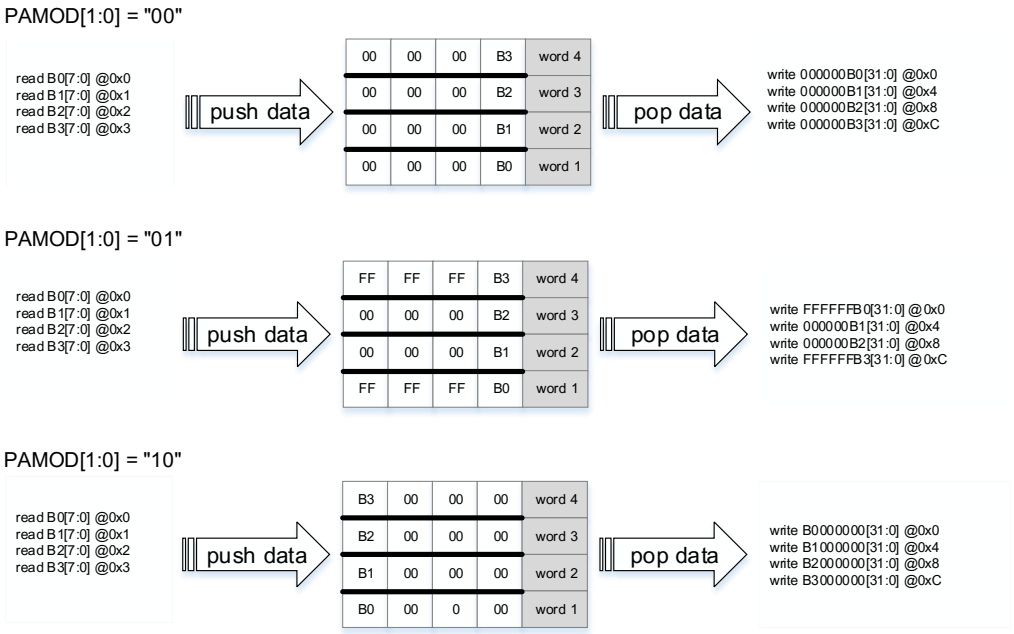
For example, when SWIDTH[1:0] = 10 (32 bits) and DWIDTH[1:0] = 00 (8 bits), the padding and alignment methods are shown in *Figure 5-4. Data padding and alignment (source greater than destination)*.

Figure 5-4. Data padding and alignment (source greater than destination)



Suppose the MSB of B0 and B3 is 1, and the MSB of B1 and B2 is 0, when SWIDTH[1:0] = 00 (8 bits) and DWIDTH[1:0] = 10 (32 bits), the padding and alignment methods are shown in *Figure 5-5. Data padding and alignment (source less than destination)*.

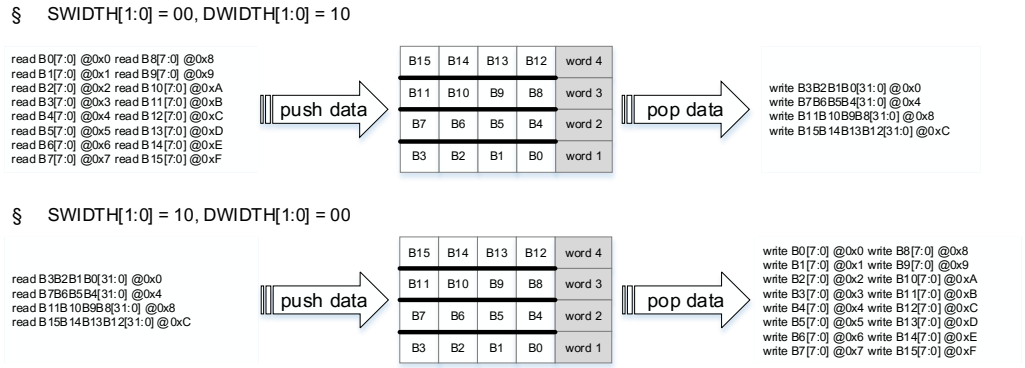
Figure 5-5. Data padding and alignment (source less than destination)



Pack / unpack

In MDMA transmission, the data size of source SWIDTH and the data size of destination DWIDTH are independent of each other, and the configuration is more flexible. When SWIDTH and DWIDTH are not equal, read/write transmission width of MDMA is different, MDMA will pack / unpack the data automatically. If the PKEN bit in MDMA_CHxCFG register is configured as 1, the source data will be packed / unpacked to match the data size of destination. When packing/unpacking data, the little endian mode is used. For example, the packing process when SWIDTH[1:0] = 00 and DWIDTH[1:0] = 10, and the unpacking process when SWIDTH[1:0] = 10 and DWIDTH[1:0] = 00, are shown in *Figure 5-6. Data packing / unpacking*.

Figure 5-6. Data packing / unpacking



Burst transmission

The SBURST[2:0] and DBURST[2:0] bits in MDMA_CHxCFG register determine the burst transmission mode of the source and destination. Both the source and destination of the MDMA supports single burst, 2-beat, 4-beat, 8-beat, 16-beat, 32-beat, 64-beat and 128-beat incrementing burst. For the single data transmission mode, SBURST[2:0] and DBURST[2:0] automatically locked as '00' by hardware immediately after the MDMA channel is enabled.

Note: The values of SBURST[2:0] and DBURST[2:0] must be programmed to ensure that the burst size is less than the transmission length, otherwise the results will be unpredictable.

FIFO

The MDMA controller provides a 256 bytes buffer that is divided into two FIFOs with a depth of 16 doulbe word, and shared by all channels. The FIFOs are used to store the source data before writing it to the destination temporarily.

FIFO0 is used to store the data to be transferred of the current buffer. When the amount of data in FIFO0 meets the destination burst, MDMA will start the write operation immediately. When all the data to be transferred in the buffer has been read to FIFO0, the arbitrator begins to arbitrate the channel priority and writes the data to be transferred of the next buffer to FIFO1.

If an error occurred during the buffer transfer, the channel will be disabled and the data in FIFO0 and FIFO1 will be discarded.

5.3.2. Address generation

Both source and destination support three address generation algorithms: fixed mode, increasing mode, and decreasing mode independently. The DIMOD[1:0] and SIMOD[1:0] bits in MDMA_CHxCFG register are used to configure the address generation algorithm for the destination and source respectively, as is shown in ***Table 5-3. Source and destination address generation configuration***.

Table 5-3. Source and destination address generation configuration

SIMOD[1:0]		DIMOD[1:0]	
00	No increment	00	No increment
10	Increment of the source is SIOS	10	Increment of the destination is DIOS
11	decrement of the source is SIOS	11	decrement of the destination is DIOS

In fixed mode, SIMOD[1:0] or DIMOD[1:0] is configured as "00", the address of the source and destination is always the original base address (MDMA_CHxSADDR and MDMA_CHxDADDR).

In increasing mode or decreasing mode, the source and destination address of the next transfer is the current address plus/minus 1 (or 2,4,8), depending on the configuration of SIOS[1:0] or DIOS[1:0] in MDMA_CHxCFG.

The increment and data size can be programmed independently to optimize pack operations.

5.3.3. Transfer modes

Buffer transfer mode

MDMA supports single, 2-beat, 4-beat, 8-beat, 16-beat, 32-beat, 64-beat and 128-beat incrementing burst transmission. The burst transmission mode of the source and destination can be configured by SBURST[2:0] and DBURST[2:0] bits in MDMA_CHxCFG register. A buffer transfer is a single or burst transfer of data. SWIDTH[1:0] and DWIDTH[1:0] in MDMA_CHxCFG register are used for configuring the data width of source and destination.

When MDMA receives a request, the arbitrator manages it based on the MDMA channel request priority. If the MDMA_CHxMADDR register is not 0, the request is acknowledged when the mask data is written to the address specified by MADDR[31:0]. Or else, writing data to or reading data from the requesting peripheral will reset the request. If the request is completed by the target peripheral, BWMOD in the MDMA_CHxCFG register must be cleared to avoid wrong new MDMA requests.

If TRIGMOD[1:0] in the MDMA_CHxCFG register is 00 and the buffer transfer has completed, MDMA will wait for another request on the same channel (such as channel A).

- If another channel (such as channel B) is requested before the next request has occurred on channel A, the request from channel B will be responded regardless of whether channel B has a higher priority than channel A.
- If the next request is detected after the completion of a buffer transfer on channel A, and at the same time a request occurs on another channel (channel C), the arbitrator will manage the request event based on MDMA channel request priority.

When the buffer transfer is completed, the TCF bit in MDMA_CHxSTAT0 register will be set. The TCF bit can be cleared by writing 1 to the TCFC bit in MDMA_CHxSTATC register

If TRIGMOD[1:0] is not 00 and the total number of data to be transferred is greater than 128 bytes, then the arbitrator manages the request event based on the MDMA channel request priority after each buffer transfer. If there is no other request of higher priority, the next buffer transfer continues. If there are other requests with higher priorities, MDMA will process the higher priority requests first.

Block transfer mode

In block transfer mode, the block size is configured by the TBNUM[16:0] in MDMA_CHxBTCFG register, and the maximum number of bytes to be transferred in the block is 64KB. When TBNUM[16:0] counts to 0, the block transfer is completed, and TCF bit, BTCF bit and CHTCF bit in MDMA_CHxSTAT0 register will be set. The CHEN bit in the MDMA_CHxCTL0 register will be cleared by the hardware and the channel will not continue to accept MDMA requests.

In the multi-block transfer mode, if the current block is not the last block, the hardware will reload the length of the first block transfer automatically after the completion of the current block transfer, and calculate the new source address and destination address according to the values of DADDRUV and SADDRUV in MDMA_CHxMBADDRU register and SADDRUM bit and DADDRUM bit in MDMA_CHxBTCFG register, then MDMA will start the next block transfer. If the current block is the last block, when TBNUM[16:0] counts to 0 and the block transfer is complete, the TCF bit, BTCF bit, MBTCF bit and CHTCF bit in MDMA_CHxSTAT0 register will be set. The CHEN bit in the MDMA_CHxCTL0 register will be cleared by the hardware and the channel will not continue to accept MDMA requests.

In link mode, if the current block is a block transfer or the last block of multi-block transfer and MDMA_CHxLADDR is not 0, the new block configuration information is loaded according to the address specified by LADDR in the MDMA_CHxLADDR register after the current block transfer is completed, and the new block / multi-block transfer starts. If the current block is a block transfer or the last block of multi-block transfer and MDMA_CHxLADDR is 0, the TCF bit, MBTCF / BTCF bit and CHTCF bit in the MDMA_CHxSTAT0 register will be set. The CHEN bit in the MDMA_CHxCTL0 register will be cleared by the hardware and the channel will not continue to accept MDMA requests.

If the block size is not an integer multiple of the source or destination data size, the BZERR bit in MDMA_CHxSTAT1 register will be set by hardware. The BZERR bit can be cleared by writing 1 to the ERRC bit in MDMA_CHxSTATC register.

The TCF bit, BTCF bit, MBTCF bit and CHTCF bit can be cleared by writing 1 to the TCFC bit, the BTCFC bit, MBTCFC and CHTCFC bit in MDMA_CHxSTATC register respectively.

Multi-block transfer mode

The number of blocks to be transferred can be configured by BRNUM[11:0] in the MDMA_CHxBTCFG register. When BRNUM[11:0] is not 0, the multi-block transfer mode is enabled. BRNUM[11:0] can be configured from 0 to 4095. When a block transfer is completed, the BRNUM value is reduced by 1. The source address and the destination address of the

next block transfer will be update in MDMA_CHxSADDR register and MDMA_CHxDADDR register according to the update mode configured by the SADDRUM bit and DADDRUM bit in the MDMA_CHxBTCFG register. The update mode of source and destination address is shown in *Table 5-4. Update mode of source and destination address*. TBNUM[16:0] in MDMA_CHxBTCFG register will reload the value of the first block transfer. When the last block transfer is completed, the TCF bit, BTCF bit, MBTCF bit, and CHTCF bit in the MDMA_CHxSTAT0 register will be set, the CHEN bit in MDMA_CHxCTL0 register will be will be cleared by the hardware and the channel will not continue to accept MDMA requests. The TCF bit, BTCF bit, MBTCF bit and CHTCF bit can be cleared by writing 1 to the TCFC bit, the BTCFC bit, MBTCFC and CHTCFC bit in MDMA_CHxSTATC register respectively.

Table 5-4. Update mode of source and destination address

Source / destination address	Update mode configuration	Updated source and destination address
SADDR	SADDRUM = 0	SADDR = SADDR + SADDRUV
	SADDRUM = 1	SADDR = SADDR - SADDRUV
DADDR	DADDRUM = 0	DADDR = DADDR + DADDRUV
	DADDRUM = 1	DADDR = DADDR - DADDRUV

Note: When the BRNUM[11:0] is 0, the last block transfer is treated as a single block transfer.

Link transfer mode

In link mode, after multi-block/block transmission, the configuration register of the current channel includes MDMA_CHxCFG, MDMA_CHxBTCFG, MDMA_CHxSADDR, MDMA_CHxDADDR, MDMA_CHxMBADDRU, MDMA_CHxLADDR, MDMA_CHxCTL1, MDMA_CHxMADDR and MDMA_CHxMDATA will be load with the data structure at the address LADDR[31:0] defined in the MDMA_CHxLADDR register. As is shown in *Table 5-5. Register link address*. If TRIGMOD[1:0] in MDMA_CHxCFG register is "11", the channel will accept new requests or continue transmission after the configuration register is loaded.

Table 5-5. Register link address

Register	Link address
MDMA_CHxCFG	LADDR[31:0] + 0x00
MDMA_CHxBTCFG	LADDR[31:0] + 0x04
MDMA_CHxSADDR	LADDR[31:0] + 0x08
MDMA_CHxDADDR	LADDR[31:0] + 0x0C
MDMA_CHxMBADDRU	LADDR[31:0] + 0x10
MDMA_CHxLADDR	LADDR[31:0] + 0x14
MDMA_CHxCTL1	LADDR[31:0] + 0x18
MDMA_CHxMADDR	LADDR[31:0] + 0x20
MDMA_CHxMDATA	LADDR[31:0] + 0x24

If the TRIGSEL[5:0] in the MDMA_CHxCTL1 register changes while loading the channel configuration register, the trigger source will be changed by hardware automatically.

Note: In link transfer mode, the SWREQMOD bit and TRIGMOD[1:0] in the MDMA_CHxCFG register cannot be modified.

5.3.4. Transfer status

Transfer complete

When TBNUM[16:0] in MDMA_CHxBTCFG register, BRNUM[11:0] and LADDR[31:0] in MDMA_CHxLADDR register are all 0, or the channel is disabled (CHEN=0) before the end of transmission, and all the remaining data have been transferred from the FIFO to the destination, the CHTCF bit in the MDMA_CHxSTAT0 register will be set after the channel transmission is completed.

Transfer interrupt

Transfer interrupt means that the CHEN in the MDMA_CHxCTL0 register is disabled (CHEN=0) during transmission, and the last data transmission is not continued when the channel is enabled again. After the channel is disabled, CHTCF bit in MDMA_CHxSTAT0 register will be set when all the remaining data in FIFO are transferred to the destination. The number of bytes or blocks not transferred can be viewed in TBNUM[16:0], BRNUM[11:0] in the MDMA_CHxBTCFG register.

Transfer pause

Before the TBNUM[16:0] in the MDMA_CHxBTCFG register counts to 0, the CHEN in MDMA_CHxCTL0 register can be cleared to pause the channel transfer. When the CHTCF bit in the MDMA_CHxSTAT0 register is set to 1, it indicates that the remaining data in the transfer buffer has been transmitted. If the values of MDMA_CHxBTCFG register, MDMA_CHxSADDR register and MDMA_CHxDADDR register are not modified by software, clear the CHTCF bit in MDMA_CHxSTAT0 register and enable the CHEN bit again, then the transmission continues.

Note: When TRIGMOD[1:0] is 11, it is recommended to configure the LADDR field in data structure of the next node as 0 to pause the channel transfer. If the channel transfer is paused by clearing the CHEN in MDMA_CHxCTL0 register, the result will not be guaranteed.

5.3.5. MDMA interrupts and errors

MDMA error flags are shown in *Table 5-6. MDMA error flags*.

Table 5-6. MDMA error flags

Error name	Description
BZERR	Block size error flag
ASERR	Address and size error flag
MDTERR	Mask data transmission error flag

Error name	Description
LDTERR	Link data error flag
ERR	Transmission error flag

The transmission error flag (ERR) will be set when the following occurs:

- A bus error occurred during MDMA read or write access.
- The location of address alignment does not match the size of the data.
- The block size is not a multiple of the data size (source and/or target).

For each MDMA channel, there are five types of interrupt events: channel transfer complete, buffer transfer complete, block transfer complete, multi-block transfer complete, and transfer error.

MDMA_CHxSTAT0 contains the flag bit for each interrupt event, register MDMA_CHxSTATC contains the flag clear bit for each interrupt event, register MDMA_CHxCTL0 contains the enable bit for each interrupt event. As is shown in *Table 5-7. MDMA interrupt events*.

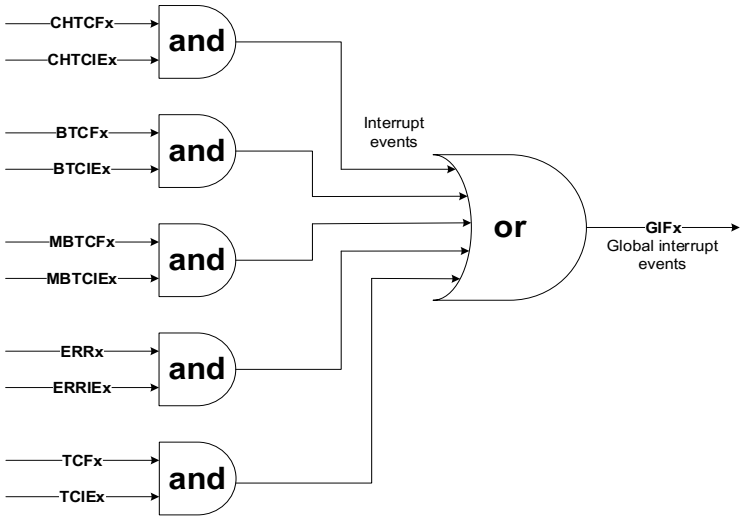
Table 5-7. MDMA interrupt events

Interrupt events	Flag bit	Enable bit	Clear bit
	MDMA_CHxSTAT0	MDMA_CHxCTL0	MDMA_CHxSTATC
channel transfer complete	CHTCF	CHTCIE	CHTCFC
buffer transfer complete	TCF	TCIE	TCFC
block transfer complete	BTCF	BTCIE	BTCFC
multi-block transfer complete	MBTCF	MBTCIE	MBTCFC
transfer error	ERR	ERRIE	ERRC

If at least one of BTCF / MBTCF / CHTCF / ERR / TCF of channel x is set and the corresponding interrupt (BTCIE / MBTCIE / CHTCIE / ERRIE / TCIE) has been enabled, the GIFx in the MDMA_GINTF register will be set. If MDMA interrupt is enabled in the NVIC, an interrupt will be generated.

MDMA interrupt logic is shown in *Figure 5-7. MDMA interrupt logic*. When any of the interrupt is enabled, and the corresponding interrupt event will generate an interrupt.

Figure 5-7. MDMA interrupt logic



Note: "x" represents the number of channels (corresponding to x=0...15).

5.4. Register definition

MDMA base address: 0x5200 0000

5.4.1. Global interrupt flag register (MDMA_GINTF)

Address offset: 0x00
Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GIF15	GIF14	GIF13	GIF12	GIF11	GIF10	GIF9	GIF8	GIF7	GIF6	GIF5	GIF4	GIF3	GIF2	GIF1	GIF0
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	GIFx	Global interrupt flag of channel x (x=0…15) 0: None flag of BTCF / MBTCF / CHTCF / ERR / TCF of the channel x is set, or the flag bit is set but its corresponding interrupt is disabled. 1: At least one of BTCF / MBTCF / CHTCF / ERR / TCF bit of channel x is set and the corresponding interrupt (BTCIE / MBTCIE / CHTCIE / ERRIE / TCIE) has been enabled.

5.4.2. Channel x status register 0 (MDMA_CHxSTAT0)

x = 0...15, where x is a channel number
Address offset: 0x40 + 0x40 × x
Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											TCF	BTCF	MBTCF	CHTCF	ERR
											r	r	r	r	r

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

16	REQAF	Channel x request active flag If the SWREQ bit in MDMA_CHxCTL0 is set, and CHEN is enabled, this bit will be set. When the request of channel x is completed, this bit is cleared by hardware. 0: Transmission of MDMA channel x is not active. 1: Transmission of MDMA channel x is active.
15:5	Reserved	Must be kept at reset value.
4	TCF	Channel x buffer transfer complete flag This bit is set by hardware, and cleared by writing 1 to the corresponding bit in MDMA_CHxSTATC register. 0: Buffer transfer of channel x is not completed. 1: Buffer transfer of channel x is completed.
3	BTCF	Channel x block transfer complete flag This bit is set by hardware, and cleared by writing 1 to the corresponding bit in MDMA_CHxSTATC register. 0: Block transfer of channel x is not completed. 1: Block transfer of channel x is completed.
2	MBTCF	Channel x multi-block transfer complete flag This bit is set by hardware, and cleared by writing 1 to the corresponding bit in MDMA_CHxSTATC register. 0: Multi-block transfer of channel x is not completed. 1: Multi-block transfer of channel x is completed.
1	CHTCF	Channel x channel transfer complete flag This bit is set by hardware, and cleared by writing 1 to the corresponding bit in MDMA_CHxSTATC register. 0: Channel transfer of channel x is not completed. 1: Channel transfer of channel x is completed. **Note**: If the channel is closed during transmission, the CHTCF bit will be set to 1 after the current buffer is completely transferred.
0	ERR	Channel x transfer error flag This bit is set by hardware, and cleared by writing 1 to the corresponding bit in MDMA_CHxSTATC register. 0: No transmission error occurred on channel x. 1: Transmission error occurred on channel x.

5.4.3. Channel x status clear register (MDMA_CHxSTATC)

x = 0...15, where x is a channel number
Address offset: 0x44 + 0x40 × x
Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											TCFC	BTCFC	MBTCFC	CHTCFC	ERRC
											w	w	w	w	w

Bits	Fields	Descriptions
31:5	Reserved	Must be kept at reset value.
4	TCFC	Channel x buffer transfer complete flag clear 0: No effect. 1: Clear the TCF bit in the MDMA_CHxSTAT0 register by writing 1 to this bit.
3	BTCFC	Channel x block transfer complete flag clear 0: No effect. 1: Clear the BTCF bit in the MDMA_CHxSTAT0 register by writing 1 to this bit.
2	MBTCFC	Channel x multi-block transfer complete flag clear 0: No effect. 1: Clear the MBTCF bit in the MDMA_CHxSTAT0 register by writing 1 to this bit.
1	CHTCFC	Channel x channel transfer complete flag clear 0: No effect. 1: Clear the CHTCF bit in the MDMA_CHxSTAT0 register by writing 1 to this bit.
0	ERRC	Channel x transfer error flag clear 0: No effect. 1: Clear the ERR bit in the MDMA_CHxSTAT0 register by writing 1 to this bit.

5.4.4. Channel x status register 1 (MDMA_CHxSTAT1)

x = 0...15, where x is a channel number

Address offset: 0x48 + 0x40 × x

Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				BZERR	ASERR	MDTERR	LDTERR	TERRD	ERRADDR[6:0]						
				r	r	r	r	r	r						

Bits	Fields	Descriptions

31:12	Reserved	Must be kept at reset value.
11	BZERR	Block size error flag When the block size or BTLEN+1 is not an integer multiple of the source or destination data size, this bit will be set by hardware. And this bit is cleared by writing 1 to ERRC bit in MDMA_CHxSTATC register. 0: No block size error is occurred. 1: Block size error is occurred.
10	ASERR	Address and size error flag If the address does not match the data size, the bit will be set by hardware. And this bit is cleared by writing 1 to ERRC bit in MDMA_CHxSTATC register. 0: No address and size error is occurred. 1: Address and size error is occurred.
9	MDTERR	Mask data error flag This bit is set by the hardware when an error occurs while writing the mask data. And this bit is cleared by writing 1 to ERRC bit in MDMA_CHxSTATC register. 0: No mask data error is occurred. 1: Mask data error is occurred.
8	LDTERR	Link data transfer error flag in the last transfer of the channel The bit is set by hardware when an error 224pprox while reading the link data structure. And this bit is cleared by writing 1 to ERRC bit in MDMA_CHxSTATC register. 0: No link data error is occurred. 1: Link data error is occurred.
7	TERRD	Transfer error direction This bit is set by the hardware when a transmission error occurs on the channel due to write access. 0: Read access error. 1: Write access error.
6:0	ERRADDR[6:0]	Transfer error address When a transfer error occurs, these bits store the low 7 bits of the error address. And the absolute error address is ERRADDR + SADDR/DADDR. **Note:** These bits are ignored when link data error is occurred.

5.4.5. Channel x control register 0 (MDMA_CHxCTL0)

x = 0...15, where x is a channel number

Address offset: 0x4C + 0x40 × x

Reset value: 0x0000 0000

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

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

Reserved	SWREQ
	w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	WES	HWES	BES	Reserved			SMODEN	PRIO[1:0]		TCIE	BTCIE	MBTCIE	CHTCIE	ERRIE	CHEN
	rw	rw	rw				rw	rw		rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:17	Reserved	Must be kept at reset value.
16	SWREQ	Software request When the channel is enabled, request for channel x can be activated by setting this bit. And the REQAF bit in the MDMA_CHxSTAT0 register will be set.
15	Reserved	Must be kept at reset value.
14	WES	Word swapping in double word 0: The order of the words is not exchanged in a double word, preserving the little endian order. 1: Exchange the order of the words in a double word. **Note:** If the destination is not a double word, the bit is ignored. When the channel is enabled (CHEN=1), this bit cannot be modified.
13	HWES	Half word swapping in word 0: The order of the half-words is not exchanged in a word, preserving the little endian order. 1: Exchange the order of the half-words in a word. **Note:** If the destination is not a word or double word, the bit is ignored. When the channel is enabled (CHEN=1), this bit cannot be modified.
12	BES	Byte swapping in half word 0: The order of the bytes is not exchanged in a half-word, preserving the little endian order. 1: Exchange the order of the bytes in a half-word. **Note:** If the destination is not a half-word, word or double word, the bit is ignored. When the channel is enabled (CHEN=1), this bit cannot be modified.
11:9	Reserved	Must be kept at reset value.
8	SMODEN	Secure mode enable 0: Secure mode disable 1: Secure mode enable This bit can only be written when the AHB slave port is in secure mode. If SMODEN is 0, all registers of the current channel can be written. If SMODEN is 1, all registers of the current channel are write protected. **Note:** When the channel is enabled (CHEN=1), this bit cannot be modified.
7:6	PRIO[1:0]	Priority level

		Software set and cleared 00: Low 01: Medium 10: High 11: Ultra high **Note:** When the channel is enabled (CHEN=1), these bits cannot be modified.
5	TCIE	Buffer transfer complete interrupt enable This bit is set and cleared by software. 0: Buffer transfer complete interrupt disable. 1: Buffer transfer complete interrupt enable.
4	BTCIE	Block transfer complete interrupt enable This bit is set and cleared by software. 0: Block transfer complete interrupt disable. 1: Block transfer complete interrupt enable.
3	MBTCIE	Multi-block transfer complete interrupt enable This bit is set and cleared by software. 0: Multi-block transfer complete interrupt disable. 1: Multi-block transfer complete interrupt enable.
2	CHTCIE	Channel transfer complete interrupt enable This bit is set and cleared by software. 0: Channel transfer complete interrupt disable. 1: Channel transfer complete interrupt enable.
1	ERRIE	Transfer error interrupt enable This bit is set and cleared by software. 0: Transfer error interrupt disable. 1: Transfer error interrupt enable.
0	CHEN	Channel enable This bit is set and cleared by software. 0: channel disable 1: channel enable **Note:** This bit will be cleared by hardware when the MDMA transfer is completed, or AHB/AXI bus error, BZERR or ASERR is occurred.

5.4.6. Channel x configure register (MDMA_CHxCFG)

x = 0...15, where x is a channel number

Address offset: 0x50 + 0x40 × x

Reset value: 0x0000 0000

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

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

BWMOD	SWREQMOD	TRIGMOD[1:0]	PAMOD[1:0]	PKEN	BTLEN[6:0]	DBURST[2:1]
rw	rw	rw	rw	rw	rw	rw
15	14	13 12	11 10	9 8	7 6 5 4 3 2	1 0

DBURST[0]	SBURST[2:0]	DIOS[1:0]	SIOS[1:0]	DWIDTH[1:0]	SWIDTH[1:0]	DIMOD[1:0]	SIMOD[1:0]
rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31	BWMOD	Bufferable write mode This bit is set and cleared by software. 0: Bufferable write mode disable. 1: Bufferable write mode enable. **Note:** When the channel is enabled (CHEN=1), this bit cannot be modified.
30	SWREQMOD	Software request mode This bit is set and cleared by software. 0: Responds to software requests and hardware requests. 1: Responds to software requests. **Note:** Changing this bit will take effect after the current transmission is completed.
29:28	TRIGMOD[1:0]	Trigger mode This bit is set and cleared by software. 00: A software request or a hardware request triggers a buffer transfer. 01: A software request or a hardware request triggers a block transfer. 10: A software request or a hardware request triggers a multi-block transfer. 11: A software request or a hardware request triggers a complete data transfer (for example, link mode). **Note:** When the channel is enabled (CHEN=1), these bits cannot be modified.
27:26	PAMOD[1:0]	Padding and alignement mode This bit is set and cleared by software.

Size of the source data is larger than that of the destination data		size of the source data is less than that of the destination data	
00	Right aligned, write the low byte of the source to the destination address, and the high byte is discarded.	00	Right aligned, zero for the missing bits.
01	Reserved	01	Right aligned, symbol extension.
10	Reserved Left aligned, the high byte of the source is written to the	10	Left aligned, zero for the missing bits.

	target address, and the low byte is discarded.		
11	Reserved	11	Reserved

Note: When the packet is enabled (PKEN=1) or the source data size is equal to the destination data size, these bits is invalid. When the channel is enabled (CHEN=1), these bits cannot be modified.

25 PKEN

Pack enable

This bit is set and cleared by software.

0: The source data is written to the destination address according to the configuration of PAMOD[1:0].

1: Pack/unpack the source data to match the destination data size.

Note: When the channel is enabled (CHEN=1), this bit cannot be modified.

24:18 BTLEN[6:0]

Buffer transfer length

This bit is set and cleared by software.

The number of bytes to be transferred at a time is BTLEN+1.

Note: BTLEN+1 must be a multiple of DWIDTH and SWIDTH.

17:15 DBURST[2:0]

Transfer burst type of destination

This bit is set and cleared by software.

000: single burst.

001: 2-beat incrementing burst.

010: 4-beat incrementing burst.

011: 8-beat incrementing burst.

100: 16-beat incrementing burst.

101: 32-beat incrementing burst.

110: 64-beat incrementing burst.

111: 128-beat incrementing burst.

Note: When the channel is enabled (CHEN=1), these bits cannot be modified.

14:12 SBURST[2:0]

Transfer burst type of source

These bits are set and cleared by software.

000: single burst.

001: 2-beat incrementing burst.

010: 4-beat incrementing burst.

011: 8-beat incrementing burst.

100: 16-beat incrementing burst.

101: 32-beat incrementing burst.

110: 64-beat incrementing burst.

111: 128-beat incrementing burst.

Note: When the channel is enabled (CHEN=1), these bits cannot be modified.

11:10 DIOS[1:0]

Offset size of destination increment

These bits are set and cleared by software.

00: 8-bit

		01: 16-bit 10: 32-bit 11: 64-bit **Note:** When the channel is enabled (CHEN=1), these bits cannot be modified. If DIOS < DWIDTH and DIMOD is not 00, the result will be unpredictable.
9:8	SIOS[1:0]	Offset size of source increment These bits are set and cleared by software. 00: 8-bit 01: 16-bit 10: 32-bit 11: 64-bit **Note:** When the channel is enabled (CHEN=1), these bits cannot be modified. If SIOS < SWIDTH and SIMOD is not 00, the result will be unpredictable.
7:6	DWIDTH[1:0]	Data size of destination These bits are set and cleared by software. 00: 8-bit 01: 16-bit 10: 32-bit 11: 64-bit **Note:** When the channel is enabled (CHEN=1), these bits cannot be modified.
5:4	SWIDTH[1:0]	Data size of source These bits are set and cleared by software. 00: 8-bit 01: 16-bit 10: 32-bit 11: 64-bit **Note:** When the channel is enabled (CHEN=1), these bits cannot be modified.
3:2	DIMOD[1:0]	Destination increment mode These bits are set and cleared by software. 00: No increment. 01: Reserved. 10: The increment of destination address is DIOS. 11: The decrement of destination address is DIOS. **Note:** When the channel is enabled (CHEN=1), these bits cannot be modified.
1:0	SIMOD[1:0]	Source increment mode These bits are set and cleared by software. 00: No increment. 01: Reserved. 10: The increment of source address is SIOS. 11: The decrement of source address is SIOS. **Note:** When the channel is enabled (CHEN=1), these bits cannot be modified.

5.4.7. Channel x block transfer configure register (MDMA_CHxBTCFG)

x = 0...15, where x is a channel number

Address offset: 0x54 + 0x40 × x

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BRNUM[11:0]												DADDRUM	SADDRUM	Reserved	TBNUM[16]
rw												rw	rw		rw

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

Bits	Fields	Descriptions
31:20	BRNUM[11:0]	Multi-block number **Note:** When the channel is enabled (CHEN=1), these bits cannot be modified.
19	DADDRUM	Multi-block destination address update mode 0: DADDR = DADDR + DADDRUV 1: DADDR = DADDR - DADDRUV **Note:** When the channel is enabled (CHEN=1), this bit cannot be modified.
18	SADDRUM	Multi-block source address update mode 0: SADDR = SADDR + SADDRUV 1: SADDR = SADDR - SADDRUV **Note:** When the channel is enabled (CHEN=1), this bit cannot be modified.
17	Reserved	Must be kept at reset value.
16:0	TBNUM[16:0]	Transfer byte number in block Number of bytes for the current block to be transferred (0-65536). In Multi-block mode, when the block transfer is completed, these bits will be reloaded with the value of the first transfer automatically. **Note:** When the channel is enabled (CHEN=1), these bits cannot be modified. TBNUM must be an integer multiple of the source and target data sizes.

5.4.8. Channel x source address register (MDMA_CHxSADDR)

x = 0...15, where x is a channel number

Address offset: 0x58 + 0x40 × x

Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:0	SADDR[31:0]	Source address

5.4.9. Channel x destination address register (MDMA_CHxDADDR)

x = 0...15, where x is a channel number

Address offset: 0x5C + 0x40 × x

Reset value: 0x0000 0000

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

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

Bits	Fields	Descriptions
31:0	DADDR[31:0]	Destination address

5.4.10. Channel x multi-block address update register (MDMA_CHxMBADDRU)

x = 0...15, where x is a channel number

Address offset: 0x60 + 0x40 × x

Reset value: 0x0000 0000

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

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

Bits	Fields	Descriptions
31:16	DADDRUV[15:0]	Destination address update value These bits are used to configure the increment or decrement of the destination address after the block transfer is completed. To align DADDR with DWIDTH, the

		value of these bits must be an integer multiple of DWIDTH. When BRNUM=0, these bits are invalid. **Note:** When the channel is enabled (CHEN=1), these bits cannot be modified.
15:0	SADDRUV[15:0]	Source address update value These bits are used to configure the increment or decrement of the source address after the block transfer is completed. To align SADDR with SWIDTH, the value of these bits must be an integer multiple of SWIDTH. When BRNUM=0, these bits are invalid. **Note:** When the channel is enabled (CHEN=1), these bits cannot be modified.

5.4.11. Channel x link address register (MDMA_CHxLADDR)

x = 0...15, where x is a channel number

Address offset: 0x64 + 0x40 × x

Reset value: 0x0000 0000

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

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

Bits	Fields	Descriptions
31:0	LADDR[31:0]	Link address If the value of these bits is not 0, after multi-block/block transmission, the configuration register of the current channel includes MDMA_CHxCFG, MDMA_CHxBTCFG, MDMA_CHxSADDR, MDMA_CHxDADDR, MDMA_CHxMBADDRU, MDMA_CHxLADDR, MDMA_CHxCTL1, MDMA_CHxMADDR and MDMA_CHxMDATA will be load with the data structure at the address LADDR[31:0] defined in the MDMA_CHxLADDR register. If the value of these bits is 0, CHTCF bit in MDMA_CHxSTAT0 register will be set, and the CHEN bit will be cleared by hardware. **Note:** 1. When the channel is enabled (CHEN=1), these bits cannot be modified. 2. LADDR[31:0] must be double word aligned.

5.4.12. Channel x control register 1 (MDMA_CHxCTL1)

x = 0...15, where x is a channel number

Address offset: 0x68 + 0x40 × x

Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:18	Reserved	Must be kept at reset value.
17	DBSEL	Destination bus select This bit is used to configure the destination bus for the selected channel x during write operations. 0: The destination bus of channel x is the system bus or AXI bus. 1: The destination bus of channel x is AHB bus or TCM. **Note:** When the channel is enabled (CHEN=1), this bit cannot be modified.
16	SBSEL	Source bus select This bit is used to configure the source bus for the selected channel x during read operations. 0: The source bus of channel x is the system bus or AXI bus. 1: The source bus of channel x is AHB bus or TCM. **Note:** When the channel is enabled (CHEN=1), this bit cannot be modified.
15:6	Reserved	Must be kept at reset value.
5:0	TRIGSEL[5:0]	Trigger select This bit field is used to select the hardware trigger source for channel x. If the SWREQMOD bit is 1, this bit is ignored. **Note:** When the channel is enabled (CHEN=1), this bit cannot be modified.

5.4.13. Channel x mask address register (MDMA_CHxMADDR)

x = 0...15, where x is a channel number

Address offset: 0x70 + 0x40 × x

Reset value: 0x0000 0000

This register has to be accessed by byte (8-bit), half-word (16-bit), 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]	Mask address When the bit field is not 0, the DMA request is acknowledged by writing the MDATA value in the MDMA_CHxMDATA register to the address specified by MADDR.

5.4.14. Channel x mask data register (MDMA_CHxMDATA)

x = 0...15, where x is a channel number

Address offset: 0x74 + 0x40 × x

Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:0	MDATA[31:0]	Mask data

6. DMA request multiplexer (DMAMUX)

6.1. Overview

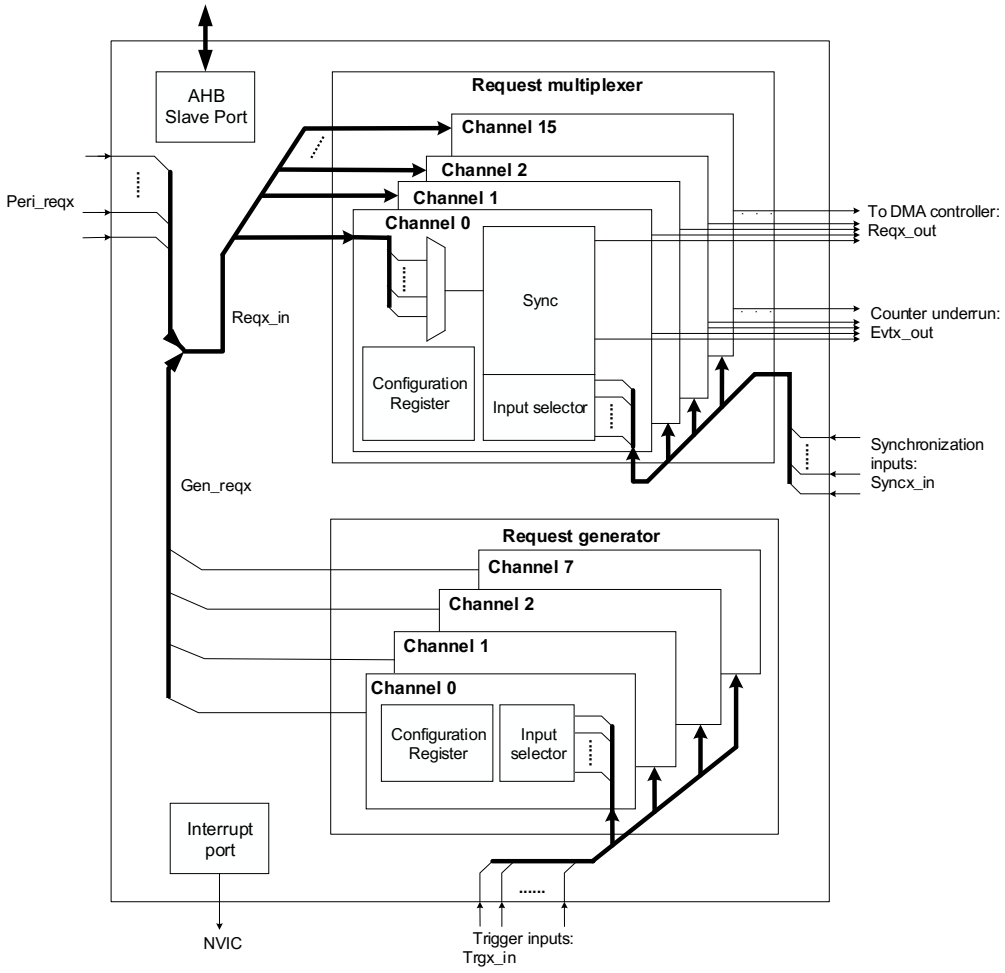
DMAMUX is a transmission scheduler for DMA requests. The DMAMUX request multiplexer is used for routing a DMA request line between the peripherals / generated DMA request (from the DMAMUX request generator) and the DMA controller. Each DMAMUX request multiplexer channel selects a unique DMA request line, 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).

6.2. Characteristics

- 16 channels for DMAMUX request multiplexer.
- 8 channels for DMAMUX request generator.
- Support 59 trigger inputs.
- Support 51 synchronization inputs.
- Each DMAMUX request generator channel has a DMA request trigger input selector, a DMAMUX request generator counter, and the trigger overrun flag.
- Each DMAMUX request multiplexer channel has 190 input DMA request lines from peripherals, a synchronization input selector, one DMA request line output, one channel event output for DMA request chaining, a DMAMUX request multiplexer counter, and the synchronization overrun flag.

6.3. Block diagram

Figure 6-1. Block diagram of DMAMUX



6.4. Signal description

The DMAMUX signals are described as follows:

- 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.

6.5. Function overview

As shown in ***Figure 6-1. 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.

6.5.1. DMAMUX request multiplexer

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

Request multiplexer channel

A DMA request input for the DMAMUX request multiplexer channel x is configured by the MUXID[7:0] bits in the DMAMUX_RM_CHxCFG register, sourced either from the peripherals or from the DMAMUX request generator, the sources can refer to ***Table 6-2. Request multiplexer input mapping***. A DMAMUX request multiplexer channel is connected and dedicated to one single channel of the DMA controller.

Note: The value 0 of MUXID[7:0] bits corresponds to no DMA request line is selected. It is not allowed to configure the same DMA request line (same non-null MUXID[7:0]) to two different request multiplexer channels.

When synchronization mode is disabled

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

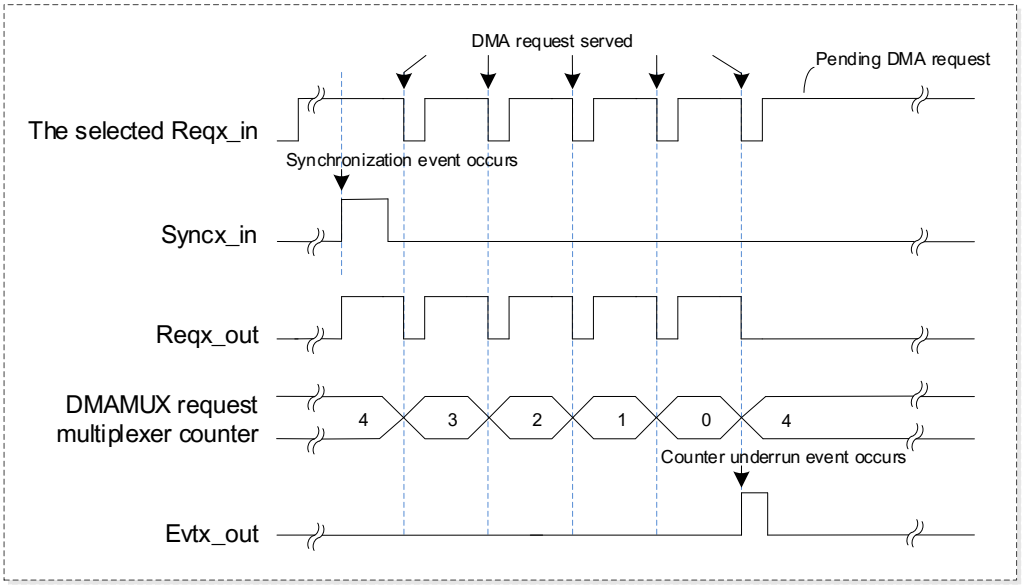
Note: The NBR[4:0] bits value shall only be written by software when both synchronization enable bit SYNCEN and event generation enable EVGEN bit of the corresponding request multiplexer channel x are disabled.

When synchronization mode is enabled

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

Figure 6-2. Synchronization mode shows an example when NBR[4:0]=4, SYNCEN=1, EVGEN=1, SYNCP[1:0]=01.

Figure 6-2. 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[5:0] bits in the DMAMUX_RM_CHxCFG register, the sources can refer to *Table 6-4. 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

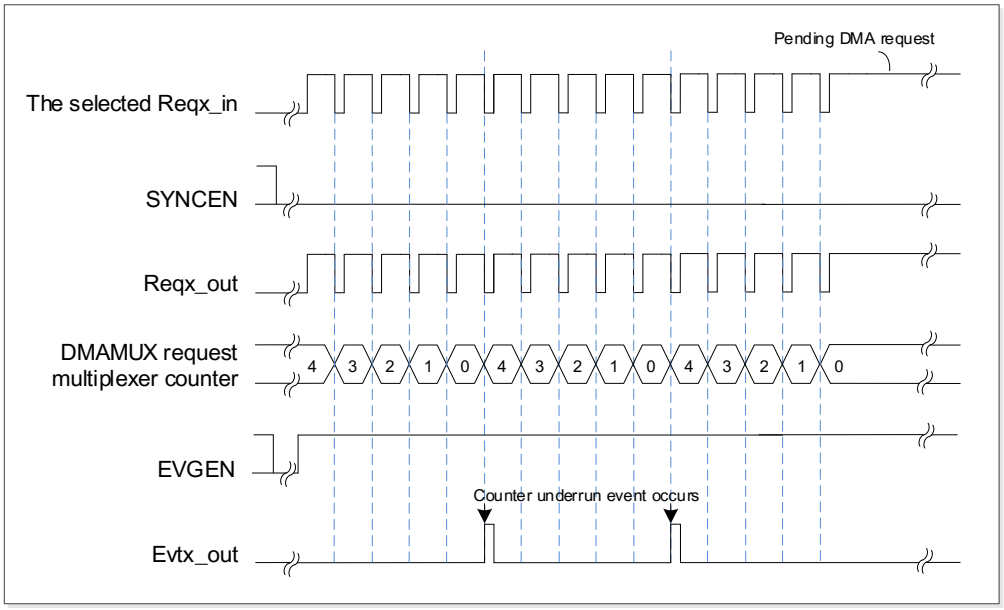
be routed to the DMAMUX multiplexer channel output until a synchronization input event occurs again.

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 Evt0_out ~ Evt3_out 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 6-3. Event generation shows an example when NBR[4:0]=4, SYNCEN=0, EVGEN=1.

Figure 6-3. 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.

6.5.2. 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[5:0] bits in DMAMUX_RG_CHxCFG register, the sources can refer to ***Table 6-3. 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.

6.5.3. 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.

6.5.4. 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 6-1. Interrupt events***.

Table 6-1. Interrupt events

Interrupt event	Flag bit	Clear bit	Enable bit
Synchronization overrun event on DMAMUX request multiplexer channel x	SOIFx in DMAMUX_RM_INTF register	SOIFCx in DMAMUX_RM_INTC register	SOIE in DMAMUX_RM_CHxCFG register
Trigger overrun event on DMAMUX request generator channel y	TOIFy in DMAMUX_RG_INTF register	TOIFCy in DMAMUX_RG_INTC register	TOIE in DMAMUX_RG_CHxCFG register

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.

6.5.5. DMAMUX mapping

DMAMUX is used with DMA0 and DMA1. Channel 0 to 7 of DMAMUX are connected to channel 0 to 7 of DMA0, channel 8 to 15 of DMAMUX are connected to channel 0 to 7 of DMA1.

Request multiplexer input mapping

A DMA request is sourced either from the peripherals or from the DMAMUX request generator, the sources can refer to *Table 6-2. 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 6-2. Request multiplexer input mapping

Request multiplexer channel input identification MUXID[7:0]	Source
1	Gen_req0
2	Gen_req1
3	Gen_req2
4	Gen_req3
5	Gen_req4
6	Gen_req5
7	Gen_req6
8	Gen_req7
9	ADC0_ROUTINE
10	ADC1_ROUTINE
11	TIMER0_CH0
12	TIMER0_CH1
13	TIMER0_CH2
14	TIMER0_CH3
15	TIMER0_MCH0
16	TIMER0_MCH1
17	TIMER0_MCH2
18	TIMER0_MCH3
19	TIMER0_UP
20	TIMER0_TRG
21	TIMER0_CMT
22	TIMER1_CH0
23	TIMER1_CH1
24	TIMER1_CH2
25	TIMER1_CH3
26	TIMER1_UP
27	TIMER1_TRG
28	Reserved
29	TIMER2_CH0
30	TIMER2_CH1
31	TIMER2_CH2
32	TIMER2_CH3

Request multiplexer channel input identification MUXID[7:0]	Source
33	TIMER2_UP
34	Reserved
35	TIMER2_TRG
36	TIMER3_CH0
37	TIMER3_CH1
38	TIMER3_CH2
39	TIMER3_CH3
40	Reserved
41	TIMER3_TRG
42	TIMER3_UP
43	I2C0_RX
44	I2C0_TX
45	I2C1_RX
46	I2C1_TX
47	SPI0_RX
48	SPI0_TX
49	SPI1_RX
50	SPI1_TX
51	USART0_RX
52	USART0_TX
53	USART1_RX
54	USART1_TX
55	USART2_RX
56	USART2_TX
57	TIMER7_CH0
58	TIMER7_CH1
59	TIMER7_CH2
60	TIMER7_CH3
61	TIMER7_MCH0
62	TIMER7_MCH1
63	TIMER7_MCH2
64	TIMER7_MCH3
65	TIMER7_UP
66	TIMER7_TRG
67	TIMER7_CMT
68	TIMER4_CH0
69	TIMER4_CH1
70	TIMER4_CH2
71	TIMER4_CH3

Request multiplexer channel input identification MUXID[7:0]	Source
72	TIMER4_UP
73	Reserved
74	TIMER4_TRG
75	SPI2_RX
76	SPI2_TX
77	UART3_RX
78	UART3_TX
79	UART4_RX
80	UART4_TX
81	DAC0_OUT0
82	DAC0_OUT1
83	TIMER5_UP
84	TIMER6_UP
85	USART5_RX
86	USART5_TX
87	I2C2_RX
88	I2C2_TX
89	DCI
90	CAU_IN
91	CAU_OUT
92	HAU_IN
93	UART6_RX
94	UART6_TX
95	UART7_RX
96	UART7_TX
97	SPI3_RX
98	SPI3_TX
99	SPI4_RX
100	SPI4_TX
101	SAI0_B0
102	SAI0_B1
103	RSPDIF_DATA
104	RSPDIF_CS
105	HPDF_FLT0
106	HPDF_FLT1
107	HPDF_FLT2
108	HPDF_FLT3
109	TIMER14_CH0
110	TIMER14_CH1

Request multiplexer channel input identification MUXID[7:0]	Source
111	TIMER14_MCH0
112	TIMER14_UP
113	TIMER14_TRG
114	TIMER14_CMT
115	TIMER15_CH0
116	TIMER15_MCH0
117	Reserved
118	TIMER15_UP
119	TIMER16_CH0
120	TIMER16_MCH0
121	Reserved
122	TIMER16_UP
123	ADC2_ROUTINE
124	FAC_READ
125	FAC_WRITE
126	TMU_WRITE
127	TMU_READ
128	TIMER22_CH0
129	TIMER22_CH1
130	TIMER22_CH2
131	TIMER22_CH3
132	TIMER22_UP
133	Reserved
134	TIMER22_TRG
135	TIMER23_CH0
136	TIMER23_CH1
137	TIMER23_CH2
138	TIMER23_CH3
139	TIMER23_UP
140	Reserved
141	TIMER23_TRG
142	TIMER30_CH0
143	TIMER30_CH1
144	TIMER30_CH2
145	TIMER30_CH3
146	TIMER30_UP
147	Reserved
148	TIMER30_TRG
149	TIMER31_CH0

Request multiplexer channel input identification MUXID[7:0]	Source
150	TIMER31_CH1
151	TIMER31_CH2
152	TIMER31_CH3
153	Reserved
154	TIMER31_UP
155	TIMER31_TRG
156	TIMER40_CH0
157	TIMER40_MCH0
158	TIMER40_CMT
159	TIMER40_UP
160	TIMER41_CH0
161	TIMER41_MCH0
162	TIMER41_CMT
163	TIMER41_UP
164	TIMER42_CH0
165	TIMER42_MCH0
166	TIMER42_CMT
167	TIMER42_UP
168	TIMER43_CH0
169	TIMER43_MCH0
170	TIMER43_CMT
171	TIMER43_UP
172	TIMER44_CH0
173	TIMER44_MCH0
174	TIMER44_CMT
175	TIMER44_UP
176	TIMER50_UP
177	TIMER51_UP
178	SAI1_B0
179	SAI1_B1
180	SAI2_B0
181	SAI2_B1
182	SPI5_RX
183	SPI5_TX
184	I2C3_RX
185	I2C3_TX
186	CAN0
187	CAN1
188	CAN2

Request multiplexer channel input identification MUXID[7:0]	Source
189	TIMER40_CH1
190	TIMER40_TRG
191	TIMER41_CH1
192	TIMER41_TRG
193	TIMER42_CH1
194	TIMER42_TRG
195	TIMER43_CH1
196	TIMER43_TRG
197	TIMER44_CH1
198	TIMER44_TRG
199	Reserved
200	Reserved
201	RDCM
202	ADC0_INSERTED
203	ADC1_INSERTED
204	ADC2_INSERTED
205	TFMT
206	AFMT
207	ENDAT
208	BISS
209	SAI3_B0
210	SAI3_B1

Trigger input mapping

The DMA request trigger input for the DMAMUX request generator channel x is selected through the TID[5:0] bits in DMAMUX_RG_CHxCFG register, the sources can refer to ***Table 6-3. Trigger input mapping***.

Table 6-3. Trigger input mapping

Trigger input identification TID[5:0]	Source
0	Evt0_out
1	Evt1_out
2	Evt2_out
3	Evt3_out
4	Evt4_out
5	Evt5_out
6	Evt6_out
7	EXTI_0_INT

Trigger input identification TID[5:0]	Source
8	EXTI_1_INT
9	EXTI_2_INT
10	EXTI_3_INT
11	EXTI_4_INT
12	EXTI_5_INT
13	EXTI_6_INT
14	EXTI_7_INT
15	EXTI_8_INT
16	EXTI_9_INT
17	EXTI_10_INT
18	EXTI_11_INT
19	EXTI_12_INT
20	EXTI_13_INT
21	EXTI_14_INT
22	EXTI_15_INT
23	RTC_WAKEUP
24	CMP0_OUTPUT
25	CMP1_OUTPUT
26	I2C0_WAKEUP
27	I2C1_WAKEUP
28	I2C2_WAKEUP
29	I2C3_WAKEUP
30	I2C0_INT_EVENT
31	I2C1_INT_EVENT
32	I2C2_INT_EVENT
33	I2C3_INT_EVENT
34	ADC2_INT
35	ADC2_AWD0_OUT
36	EXTI_0_EVENT
37	EXTI_1_EVENT
38	EXTI_2_EVENT
39	EXTI_3_EVENT
40	EXTI_4_EVENT
41	EXTI_5_EVENT
42	EXTI_6_EVENT
43	EXTI_7_EVENT
44	EXTI_8_EVENT
45	EXTI_9_EVENT
46	EXTI_10_EVENT
47	EXTI_11_EVENT

Trigger input identification TID[5:0]	Source
48	EXTI_12_EVENT
49	EXTI_13_EVENT
50	EXTI_14_EVENT
51	EXTI_15_EVENT
52	EXTI_19_EVENT
53	EXTI_20_EVENT
54	EXTI_21_EVENT
55	EXTI_33_EVENT
56	EXTI_34_EVENT
57	EXTI_35_EVENT
58	EXTI_36_EVENT

Synchronization input mapping

The synchronization input is selected by SYNCID[5:0] bits in the DMAMUX_RM_CHxCFG register, the sources can refer to *Table 6-4. Synchronization input mapping*.

Table 6-4. Synchronization input mapping

Synchronization input identification SYNCID[5:0]	Source
0	Evt0_out
1	Evt1_out
2	Evt2_out
3	Evt3_out
4	Evt4_out
5	Evt5_out
6	Evt6_out
7	EXTI_0_INT
8	EXTI_1_INT
9	EXTI_2_INT
10	EXTI_3_INT
11	EXTI_4_INT
12	EXTI_5_INT
13	EXTI_6_INT
14	EXTI_7_INT
15	EXTI_8_INT
16	EXTI_9_INT
17	EXTI_10_INT
18	EXTI_11_INT
19	EXTI_12_INT
20	EXTI_13_INT

Synchronization input identification SYNCID[5:0]	Source
21	EXTI_14_INT
22	EXTI_15_INT
23	RTC_WAKEUP
24	CMP0_OUTPUT
25	I2C0_WAKEUP
26	I2C1_WAKEUP
27	I2C2_WAKEUP
28	I2C3_WAKEUP
29	EXTI_0_EVENT
30	EXTI_1_EVENT
31	EXTI_2_EVENT
32	EXTI_3_EVENT
33	EXTI_4_EVENT
34	EXTI_5_EVENT
35	EXTI_6_EVENT
36	EXTI_7_EVENT
37	EXTI_8_EVENT
38	EXTI_9_EVENT
39	EXTI_10_EVENT
40	EXTI_11_EVENT
41	EXTI_12_EVENT
42	EXTI_13_EVENT
43	EXTI_14_EVENT
44	EXTI_15_EVENT
45	EXTI_19_EVENT
46	EXTI_20_EVENT
47	EXTI_33_EVENT
48	EXTI_34_EVENT
49	EXTI_35_EVENT
50	EXTI_36_EVENT

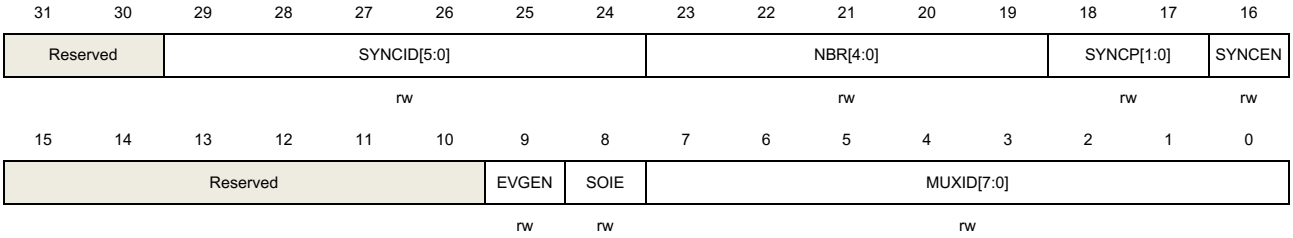
6.6. Register definition

DMAMUX base address: 0x4002 0800

6.6.1. Request multiplexer channel x configuration register (DMAMUX_RM_CHxCFG)

x = 0...15, 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
31:30	Reserved	Must be kept at reset value.
29:24	SYNCID[5:0]	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.

6.6.2. 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
SOIF15	SOIF14	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	r	r

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	SOIFx	Synchronization overrun event flag of request multiplexer channel x If a synchronization event occurs when the DMAMUX request counter value is lower than NBR[4:0], the flag is set. It is cleared by writing 1 to the corresponding SOIFCx bit in DMAMUX_RM_INTC register.

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

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															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOIFC15	SOIFC14	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	w	w

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	SOIFCx	Clear bit for synchronization overrun event flag of request multiplexer channel x. Writing 1 clears the corresponding overrun flag SOIFx in the DMAMUX_RM_INTF register.

6.6.4. Request generator channel x configuration register (DMAMUX_RG_CHxCFG)

x = 0...7, 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]	DMAMUX request generator trigger polarity 00: No event trigger detection 01: Rising edge 10: Falling edge 11: Rising and falling edges
16	RGEN	DMAMUX request generator channel x enable 0: Disable DMAMUX request generator channel x 1: Enable DMAMUX 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

7:6	Reserved	Must be kept at reset value.
5:0	TID[5:0]	Trigger input identification Selects the DMA request trigger input source.

6.6.5. Request generator channel interrupt flag register (DMAMUX_RG_INTF)

Address offset: 0x140

Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								TOIF7	TOIF6	TOIF5	TOIF4	TOIF3	TOIF2	TOIF1	TOIF0
								r	r	r	r	r	r	r	r

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	TOIFx	Trigger overrun event flag of request generator channel x. If a new trigger event occurs before the request generator counter underrun, the flag is set. It is cleared by writing 1 to the corresponding TOIFCx bit in the DMAMUX_RG_INTC register.

6.6.6. Request generator channel interrupt flag clear register (DMAMUX_RG_INTC)

Address offset: 0x144

Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								TOIFC7	TOIFC6	TOIFC5	TOIFC4	TOIFC3	TOIFC2	TOIFC1	TOIFC0
								w	w	w	w	w	w	w	w

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	TOIFCx	Clear bit for trigger overrun event flag of request generator channel x.

Writing 1 clears the corresponding trigger overrun flag TOIFx in the DMAMUX_RG_INTF register.

7. Non-volatile memory controller (NVMC)

7.1. Overview

The non-volatile memory controller, NVMC, provides all the necessary functions for the CNVM (code non-volatile memory) and ECNVM (extend code non-volatile memory) memory. The NVMC provides the read, page erase, sector erase, mass erase, chip erase, program operations, and security protection mechanism. It also handles option byte loading at reset and dynamic option byte changes.

The OTP area can store system parameters. As a non-volatile unit of storage, OTP areas include configuration information and corresponding lock bits. Some bits of OTP cannot be restored to 0 once written to 1, and some bits can be changed from 1 to 0 before being locked. According to the software operation, the OTP controller can program and read 16-bit information at a time.

7.2. Characteristics

- GD32H77xxW/GD32H78xxW/GD32H77xxP/GD32H78xxP/GD32H77xxI/GD32H78xxI support different capacities of CNVM + ECNVM: (MB)
  - GD32H77xxW/GD32H78xxW: 2 + 7.5,
  - GD32H77xxP/GD32H78xxP: 2 + 3.5,
  - GD32H77xxI/GD32H78xxI: 1.625 + 0.
- GD32H77xxW/GD32H78xxW/GD32H77xxP/GD32H78xxP supports two typical modes:
  - Mode 1: Global data integrity enhancement mode, which performs data integrity enhancement configuration for the entire CNVM
  - Mode 2: Flexible mode, where the user selects the size of CNVM to be enhanced according to their own requirements
- The GD32H77xxW/GD32H78xxW support OTA(Over-the-Air) upgrades.
- Up to 2 MB of non-volatile memory CNVM with high-speed read capability.
- Up to 7.5 MB of ECNVM memory for instruction and data, , with support for additional data integrity enhancement features. (Not for GD32H77xxI/GD32H78xxI)
- 256-bit programming operation for CNVM, 256-bit and 64-bit programming operation for ECNVM.
- Page erase (4KB), sector erase, mass erase operation for ECNVM. (Not for GD32H77xxI/GD32H78xxI)
- Sector erase, mass erase operation for CNVM. (Only sector erase in GD32H77xxI/GD32H78xxI)
- Chip erase for CNVM and ECNVM. (Only erase CNVM in GD32H77xxI/GD32H78xxI)
- ECC with intra-double-bit errors corrected and intra-double-bit errors detected per 128-bit CNVM using 16 ECC check bits.
- ECC with single bit error corrected and double bit errors detected per 64-bit ECNVM

using 8 ECC check bits (only for 7.5MB ECNVM).

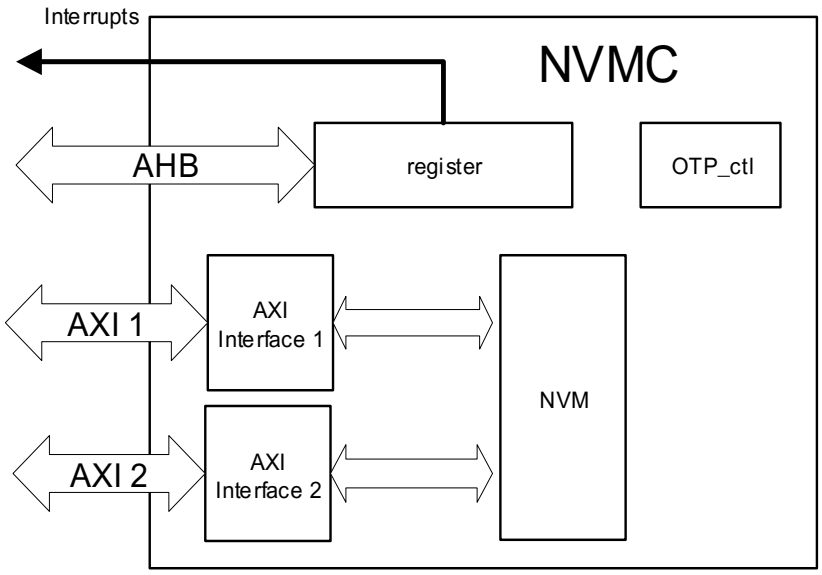
- Cyclic redundancy check (CRC) hardware module for CNVM content check.
- Programable read buffers in each AXI interface module to increase efficiency-The buffer mechanisms of CNVM and ECNVM are different.
- AXI bus matrix support 3 read outstanding and 2 write outstanding for CNVM and ECNVM AXI bus operation.
- Support additional data integrity enhancement features. (Not for GD32H77xxI/GD32H78xxI)
- Option bytes are uploaded to the option byte control registers at every system reset.
- NVM security protection to prevent illegal code / data access.
- Sector erase / program protection to prevent unexpected operation.
- Provides one execute-only dedicated code read protection (DCRP) area.
- Provides one secure user area which is accessible only in secure mode.
- SRAM Boot / System ROM Boot / User Boot / TCM Boot / OSPI0 Boot / OSPI1 Boot.
- One-time programmable nonvolatile OTP storage cells organized as 416B, including configuration information and lock bits.
- Each information is composed of redundant bit for reliability, and information can be repaired when programming to OTP.
- All bits at lock state cannot be modified within OTP.
- The OTP user configuration part can be accessed through the direct AXI bus or OTP register, and the OTP user data area can only be accessed through the AXI bus.

7.3. Function overview

7.3.1. Non-volatile memory architecture

NVMC supports two 64-bit AXI interfaces to access code / data in CNVM or ECNVM, and 32-bit AHB slave interface to access register. ***Figure 7-1. NVMC block diagram*** shows the details.

Figure 7-1. NVMC block diagram



The NVMC serves as an interface to two non-volatile memories, an CNVM and a ECNVM. The NVMC has protection mechanisms to prevent assets stored in NVM at startup time, runtime, and during firmware and configuration updates from being corrupted.

The read / write interface of the nor ECNVM is derived from an asynchronous clock (The asynchronous clock is asynchronous with the AXI clock CK_AXI). At run time, asynchronous clock SCLK_IN is default to 2 division of the AXI clock CK_AXI, and users can also configure faster, but the CK_AXI can not be more than four times the asynchronous clock SCLK_IN, and the minimum value of CK_AXI is a 1/4 division of SCLK_IN. The asynchronous clock SCLK_OUT that accesses nor ECNVM after frequency division should not exceed 100MHz. ***Figure 7-2. ECNVM read and write clock diagram of GD32H77x Devices*** and ***Figure 7-3. ECNVM read and write clock diagram of GD32H78x Devices*** show the details. (GD32H77xxI/GD32H78xxI does not require the configuration SCLK_IN and SCLK_OUT)

Note: When the DTREN bit in the NVMC_PERFCTL register is set to 1, SCLK_OUT is a 1/4 division of SCLK_IN. When the DTREN bit is set to 0, SCLK_OUT is a 1/2 division of SCLK_IN. Once DTREN is determined, the corresponding 1/2 division or 1/4 division is automatically configured by the hardware.

Note: The asynchronous clock SCLK_IN is sourced from the RCU's CK_NVMC_ECNVM. Ensure that the ECNVM clock source is configured properly to avoid overclocking. When the DTREN bit is set to 1, the maximum supported SCLK_OUT frequency is 66 MHz; when the DTREN bit is set to 0, the maximum supported SCLK_OUT frequency is 100 MHz.

Note: For GD32H77xxW/GD32H78xxW, SCLK_OUT supports a maximum of 100MHz over the full VDD voltage range. For GD32H77xxP/GD32H78xxP, SCLK_OUT supports a maximum of 80MHz when VDD is greater than 2.3V, and a maximum of 60MHz when VDD is less than or equal to 2.3V.

Figure 7-2. ECNVM read and write clock diagram of GD32H77x Devices

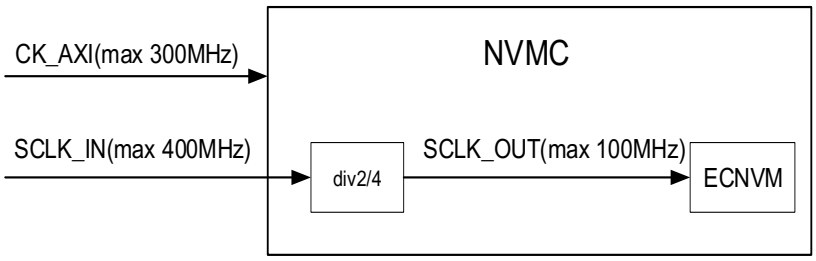
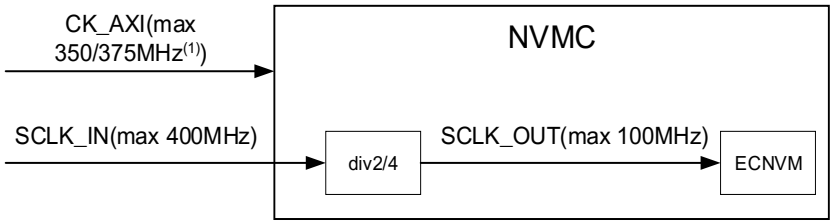


Figure 7-3. ECNVM read and write clock diagram of GD32H78x Devices



Note: (1) Applicable to the high-performance mode of GD32H78x.

The AXI slave interface supports the following access types:

- NVM single read operations supporting 1, 2, 4, 8-byte.
- NVM incrementing burst read operations supporting size 1, 2, 4, 8-byte.
- NVM wrapped burst read operations supporting size 8-byte with 1, 2, 4-beat.
- CNVM burst write accesses, with size 64*4 bits.
- ECNVM burst write accesses, with size 64*4 bits.
- All programming addresses must be aligned with the data width.
- Address can not across 4K boundary in one burst read/write operation.
- All operations exclusive above description will generate a read / write bus error, read operation return zero, write operation is ignored.

The AHB slave interface supports 32-bit word accesses.

The main NVM consists of: An CNVM block of up to 1664 / 2048 KB, it is divided into 26 / 32 sectors; An ECNVM memory block of up to 3584 / 7680 KB, it is divided into 56 / 120 sectors; An information block of 128 KB; A check information block of 32 KB; A special region dedicated to option byte storage, accessible by the application software only through the AHB slave interface.

Applications can access special areas dedicated to option byte storage only through the AHB configuration Register interface.

The following tables show the details of ECNVM organization.

Table 7-1. GD32H77xxW/GD32H78xxW base address and size for 7.5MB ECNVM memory and 2MB CNVM

Block	Name	Address range	Size(bytes)
CNVM Block 2MB	sector 0	0x0800 0000 - 0x0800 FFFF	64KB

Block		Name	Address range	Size(bytes)
		sector 1	0x0801 0000 - 0x0801 FFFF	64KB
		. . .	. . .	. . .
		sector 30	0x081E 0000 - 0x081E FFFF	64KB
		sector 31	0x081F 0000 - 0x081F FFFF	64KB
ECNVM Block	Data ECNVM0 3MB	sector 32	0x0820 0000 - 0x0820 FFFF	64KB
		sector 33	0x0821 0000 - 0x0821 FFFF	64KB
		. . .	. . .	. . .
		sector 79	0x084F 0000 - 0x084F FFFF	64KB
	Bank1 2MB	sector 80	0x0850 0000 - 0x0850 FFFF	64KB
		sector 81	0x0851 0000 - 0x0851 FFFF	64KB
		. . .	. . .	. . .
		sector 110	0x086E 0000 - 0x086E FFFF	64KB
		sector 111	0x086F 0000 - 0x086F FFFF	64KB
	Bank0 2MB	sector 112	0x0870 0000 - 0x0870 FFFF	64KB
		sector 113	0x0871 0000 - 0x0871 FFFF	64KB
		. . .	. . .	
		sector 142	0x088E 0000 - 0x088E FFFF	64KB
		sector 143	0x088F 0000 - 0x088F FFFF	64KB
	Data ECNVM1 512 KB	sector 144	0x0890 0000 - 0x0890 FFFF	64KB
		…	…	…
		sector 151	0x0897 0000 - 0x0897 FFFF	64KB
OTP Block (simulation 260pprox)		OTP area	0x1FFF 0000 - 0x1FFF 19FF	416B
Information Block		Bootloader	0x1FEF 0000 - 0x1FF0 FFFF	128KB
Check Information Block		Check_bootloader	0x1FF1 0000 - 0x1FF1 7FFF	32KB

Note: SWAP = 0, Bank0 indicates the active data integrity enhancement area of CNVM. SWAP = 1, Bank1 indicates the active data integrity enhancement area of CNVM.

Table 7-2. GD32H77xxP/GD32H78xxP base address and size for 3.5MB ECNVM memory

and 2MB CNVM

Block		Name	Address range	Size(bytes)
CNVM Block 2MB		sector 0	0x0800 0000 - 0x0800 FFFF	64KB
		sector 1	0x0801 0000 - 0x0801 FFFF	64KB
		. . .	. . .	. . .
		sector 30	0x081E 0000 - 0x081E FFFF	64KB
		sector 31	0x081F 0000 - 0x081F FFFF	64KB
ECNVM Block	Data ECNVM0 1MB	sector 32	0x0820 0000 - 0x0820 FFFF	64KB
		sector 33	0x0821 0000 - 0x0821 FFFF	64KB
		. . .	. . .	. . .
		sector 47	0x082F 0000 - 0x082F FFFF	64KB
	Bank0 2MB	sector 48	0x0830 0000 - 0x0830 FFFF	64KB
		sector 49	0x0831 0000 - 0x0831 FFFF	64KB
		. . .	. . .	. . .
		sector 78	0x084E 0000 - 0x084E FFFF	64KB
		sector 79	0x084F 0000 - 0x084F FFFF	64KB
	Data ECNVM1 512 KB	sector 80	0x0850 0000 - 0x0850 FFFF	64KB
		…	…	…
		sector 87	0x0857 0000 - 0x0857 FFFF	64KB
OTP Block (simulation 261pprox)		OTP area	0x1FFF 0000 - 0x1FFF 19FF	416B
Information Block		Bootloader	0x1FEF 0000 - 0x1FF0 FFFF	128KB
Check Information Block		Check_bootloader	0x1FF1 0000 - 0x1FF1 7FFF	32KB

Table 7-3. GD32H77xxI/GD32H78xxI base address and size for 0MB ECNVM memory and 1.625MB CNVM

Block	Name	Address range	Size(bytes)
CNVM Block 1.625MB	sector 0	0x0800 0000 - 0x0800 FFFF	64KB
	sector 1	0x0801 0000 - 0x0801 FFFF	64KB
	. . .	. . .	. . .
	sector 24	0x0818 0000 - 0x0818 FFFF	64KB
	sector 25	0x0819 0000 - 0x0819 FFFF	64KB

Block	Name	Address range	Size(bytes)
OTP Block (simulation 262pprox)	OTP area	0x1FFF 0000 - 0x1FFF 19FF	416B
Information Block	Bootloader	0x1FEF 0000 - 0x1FF0 FFFF	128KB

7.3.2. Error Checking and Correcting (ECC)

ECNVM ECC

The ECNVM ECC mechanism supports:

- One error detection and correction
- Two errors detection

When two errors are detected:

The ECCDET bit in NVMC_ECNVM_STAT register will be set. If the ECCDETIE bit in NVMC_ECNVM_CTL register is set, an interrupt is generated. The ECCADDR bits in NVMC_ECNVM_ECCADDR register notice the address where the error occurred. The ECCDET bit in NVMC_ECNVM_STAT register can be cleared by writing 1 to ECCDET bit in NVMC_ECNVM_STAT register.

Note:

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

2. Only the 7.5MB ECNVM has ECC functionality.

3. The ECNVM ECC mechanism supports one error detection and correction, but there is no corresponding one bit error flag.

4. 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.

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

CNVM ECC

The CNVM ECC mechanism supports:

- One error detection and correction
- Two errors detection and correction

When two error is detected:

The AUTOECCDERRDET bit in NVMC_CNVM_STAT register will be set. If the AUTOECCDERRDETIE bit in NVMC_CNVM_CTL register is set, an interrupt is generated. The NVMC_CNVM_AUTOERR2ADDRx register records the address where the error occurred and state of address. The AUTOECCDERRDET bit in NVMC_CNVM_STAT register can be cleared by clearing all AUTOERR2ADDRSTAT of NVMC_CNVM_AUTOERR2ADDRx (x = 0…3).

Note:

1. Data in CNVM memory are 144-bits words: 16 bits are added per 128 bits, but the added 16bits are calculated by hardware and cannot be accessed by user.
2. Support four error detected / corrected entirety for 2*128-bit CNVM word using 2*16 ECC check bits (For 2*128-bit CNVM word, high 128 bits, ECC can detected / corrected double bit errors at most, low 128 bits, ECC can detected / corrected double bit errors at most).
3. The CNVM ECC function can be turned on or off by the CNVM_ECC_OVRD and CNVM_ECC_MODE[1:0]. When CNVM_ECC_OVRD is set to 0 and CNVM_ECC_MODE[1:0] is set to 1 / 2 / 3, CNVM ECC is turned off. Otherwise, CNVM ECC is turned on. This function can be used to test the CNVM ECC feature.
4. When the same address is read twice, the second read is from the buffer. When using CNVM ECC to test related functions, the second read of the same address may not yield the expected value under the current ECC state. Therefore, it is recommended to clear the buffer by setting EMPTY_CNVM_BUF after each ECC switch.
5. When an ECC error is reported, a new read at the error address may not generate an ECC error if the data is still present in the CPU buffer / general buffer, even if AUTOECCDERRDET bit is cleared.

7.3.3. CNVM controller (CNVMC)

Read operations

CNVM block is accessed by the AXI interface. There are two read buffers in CNVM to improve efficiency of consecutive read requests. One is for CPU reading operations while the other is for other masters' reading operation.

These read buffer can prefetch data from CNVM macro according to the AXI read address. If AXI request address of multiple continuous read access hit the read buffer, read data will be directly sent to AXI bus from the read buffer without triggering additional CNVM read operations. The user can utilize this feature to significantly improve the performance of sequentially executed programs. This mechanism occurs every time the read access

permission is granted. When read access is denied for security reasons (e. g., the DCRP protected word), the CNVM issues a corresponding read error response and does not trigger a read operation to the CNVM memory.

Wait state added

The WSCNT bits in the NVMC_PERFCTL register needs to be configured correctly depend on the AHB clock frequency, system clock frequency when reading the CNVM memory. Make sure CK_AHB/(WSCNT+1) <= 50MHz. AHB clock frequency and system clock frequency are usually 1:1 or 1:2 relationship. The recommended configurations are shown in ***Table 7-4. The typical value configuration for WSCNT of GD32H77x Devices.*** And ***Table 7-5. The typical value configuration for WSCNT of GD32H78x Devices.***

Table 7-4. The typical value configuration for WSCNT of GD32H77x Devices.

System clock frequency	AHB clock frequency	WSCNT configured
64MHz	64MHz	1
480MHz	240MHz	4
600MHz	300MHz	5

Table 7-5. The typical value configuration for WSCNT of GD32H78x Devices.

System clock frequency	AHB clock frequency	WSCNT configured
64MHz	64MHz	1
480MHz	240MHz	4
600MHz	300MHz	5
700MHz	350MHz	6
750MHz(1)	375MHz	7

Note: (1) Applicable to the high-performance mode of GD32H78x.

NVM Read-While-Write (RWW) Feature

GD32H77xxW/GD32H78xxW and GD32H77xxP/GD32H78xxP devices support the RWW feature, while GD32H77xxI/GD32H78xxI devices do not support RWW.

The dual-bank structure of NVM is based on CNVM and ECNVM. This structure supports the RWW feature, meaning that when one memory bank is performing a read, program, or erase operation, the other memory bank can simultaneously execute read, program, or erase operations without waiting for the first operation to complete.

Note: When executing code from CNVM or SRAM while performing ECNVM programming, code execution efficiency may be reduced. This issue can be resolved by using MDMA to perform ECNVM programming operations.

Unlock the NVMC_CNVM_CTL and NVMC_OBCTL register

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

The NVMC_OBCTL registers are still protected even the NVMC_ECNVM_CTL / NVMC_CNVM_CTL is unlocked. The unlocking sequence is two write operations, which are writing 0x08192A3B and 0x4C5D6E7F to NVMC_OBKEY register. After the two write operations, the OBLK bit in NVMC_OBCTL register is reset to 0 by hardware. The software can lock the NVMC_OBCTL again by setting the OBLK bit in NVMC_OBCTL register to 1.

Sector erase

The NVMC provides a sector erase function which is used to initialize the contents of an CNVM sector. Each sector can be erased independently without affecting the contents of other sectors. The following steps show the access sequence of the registers for a sector (64 Kbytes) erase operation.

1. Unlock the NVMC_CNVM_CTL register if necessary.
2. Check the BUSY bit in NVMC_CNVM_STAT register to confirm that no CNVM memory operation is in progress (BUSY equal to 0). Otherwise, wait until the operation has finished.
3. Set the SER bit in NVMC_CNVM_CTL register.
4. Write the sector address ADDR to NVMC_CNVM_ADDR register.
5. Send the sector erase command to the NVMC by setting the START bit in NVMC_CNVM_CTL register.
6. Wait until all the operations have finished by checking the value of the BUSY bit in NVMC_CNVM_STAT register.
7. Read and verify the sector if required.

Note: The state of CNVM after erasure is 0..

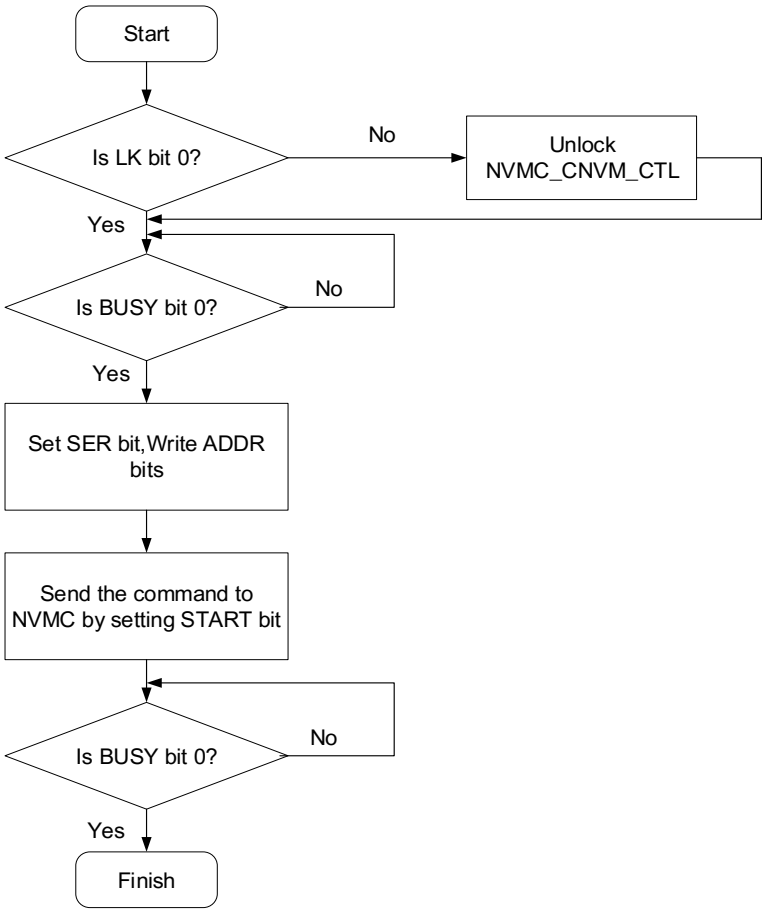
If a CNVM sector contain DCRP, secure user or erase/program protected data, the sector erase operation is stopped and the WPERR bit is set in the NVMC_CNVM_STAT register.

If mass erase and sector erase is request at the same time, the mass erase operation will replace the sector erase operation.

When the operation is executed successfully, the ENDF in NVMC_CNVM_STAT register is set, and an interrupt will be triggered by NVMC if the ENDIE bit in the NVMC_CNVM_CTL register is set. Note that a correct target sector address must be confirmed. Or the software

may run out of control if the target erase sector is being used to fetch codes or to access data. The NVMC will not provide any notification when this occurs. Additionally, the sector erase operation will be ignored on erase / program protected sectors. In this condition, an erase / program protection error interrupt will be triggered by the NVMC if the WPERRIE bit in the NVMC_CNVM_CTL register is set. The software can check the WPERR bit in the NVMC_CNVM_STAT register to detect this condition in the interrupt handler. ***Figure 7-4. Process of sector erase operation*** shows the sector erase operation flow.

Figure 7-4. Process of sector erase operation



Note: When programming, erasing, especially the chip erasing, abnormal power off or reset should be avoided as far as possible, otherwise unpredictable consequences may occur. If there is a risk of power off and reset, avoid using protection-removed chip erasing.

Mass erase

The NVMC provides a mass erase function which is used to initialize the contents of all sectors in CNVM. This erase can affect by setting the MER bit to 1 in the NVMC_CNVM_CTL register. The following steps show the mass erase register access sequence. (GD32H77xxI/GD32H78xxI does not have mass erase functionality.)

1. Unlock the NVMC_CNVM_CTL register if necessary.
2. Check the BUSY bit in NVMC_CNVM_STAT register to confirm that no CNVM memory

operation is in progress (BUSY equals to 0). Otherwise, wait until the operation has finished.

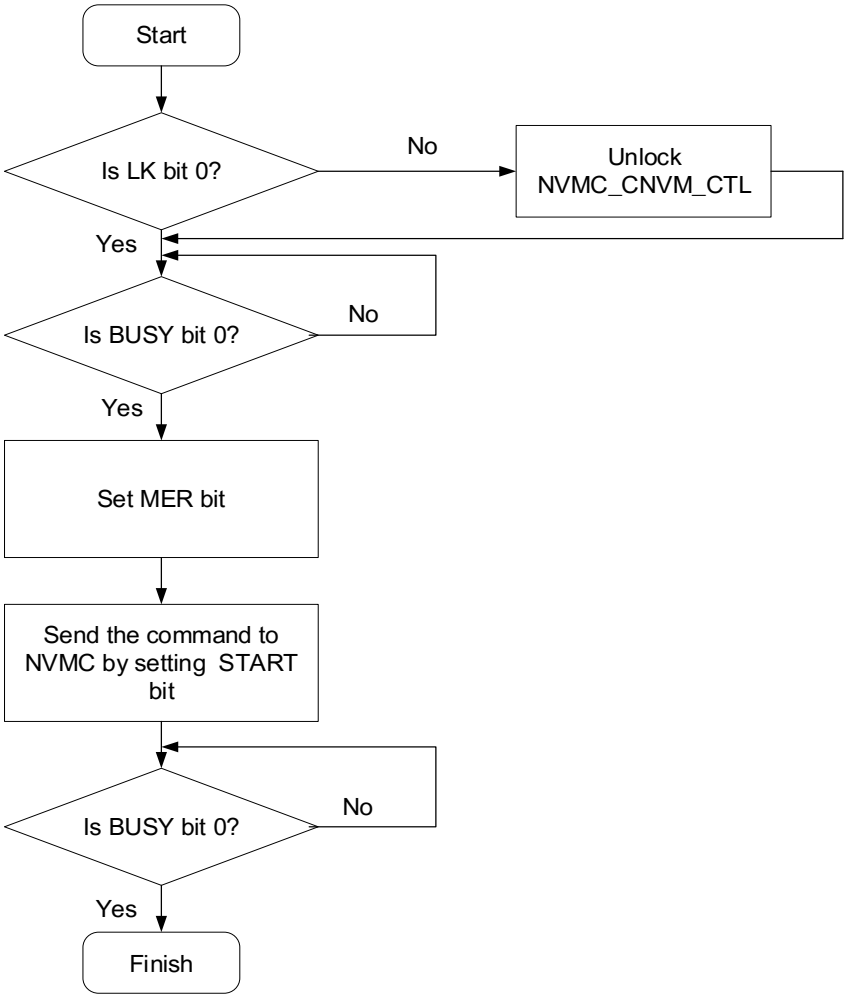
3. Set the MER bit in the NVMC_CNVM_CTL register to enable mass erase operation.
4. Send the mass erase command to the NVMC by setting the START bit in NVMC_CNVM_CTL register.
5. Wait until all the operations have been finished by checking the value of the BUSY bit in NVMC_CNVM_STAT register.
6. Read and verify the CNVM memory if required.

If CNVM contain DCRP, secure user or erase/program protected data, the mass erase operation is stopped and the WPERR bit is set in the NVMC_CNVM_STAT register.

If mass erase and sector erase is request at the same time, the mass erase operation will replace the sector erase operation.

When the operation is executed successfully, the ENDF in NVMC_CNVM_STAT register is set, and an interrupt will be triggered by NVMC if the ENDIE bit in the NVMC_CNVM_CTL register is set. In this condition, an erase/program protection error interrupt will be triggered by the NVMC if the WPERRIE bit in the NVMC_CNVM_CTL register is set. The software can check the WPERR bit in the NVMC_CNVM_STAT register to detect this condition in the interrupt handler. ***Figure 7-5. Process of mass erase operation*** shows the mass erase operation flow.

Figure 7-5. Process of mass erase operation



Note: When programming, erasing, especially the chip erasing, abnormal power off or reset should be avoided as far as possible, otherwise unpredictable consequences may occur. If there is a risk of power off and reset, avoid using protection-removed chip erasing.

Chip erase

GD32H77xxI/GD32H78xxI uses chip erase to erase the full chip CNVM, but does not need to write the ECNVM side register.

Typical chip erase

The NVMC provides a typical chip erase function which is used to erase all sectors in CNVM and ECNVM except sectors contain secure user or protected data. This erase can affect by setting the CER bit to 1 in the NVMC_CNVM_CTL register. The following steps show the typical chip erase register access sequence.

1. Unlock the NVMC_CNVM_CTL register if necessary.
2. Check the BUSY bit in NVMC_CNVM_STAT register to confirm that no CNVM memory operation is in progress (BUSY equals to 0). Otherwise, wait until the operation has

finished. At the same time, check the BUSY bit in NVMC_ECNVM_STAT register to confirm that no ECNVM memory operation is in progress (BUSY equals to 0). Otherwise, wait until the operation has finished.

3. Set the CER bit in the NVMC_CNVM_CTL register to enable chip erase operation.
4. Send the chip erase command to the NVMC by setting the START bit in NVMC_CNVM_CTL register.
5. Wait until all the operations have been finished by checking the value of the BUSY bit in NVMC_CNVM_STAT and NVMC_ECNVM_STAT register.
6. Read and verify the CNVM and ECNVM memory if required.

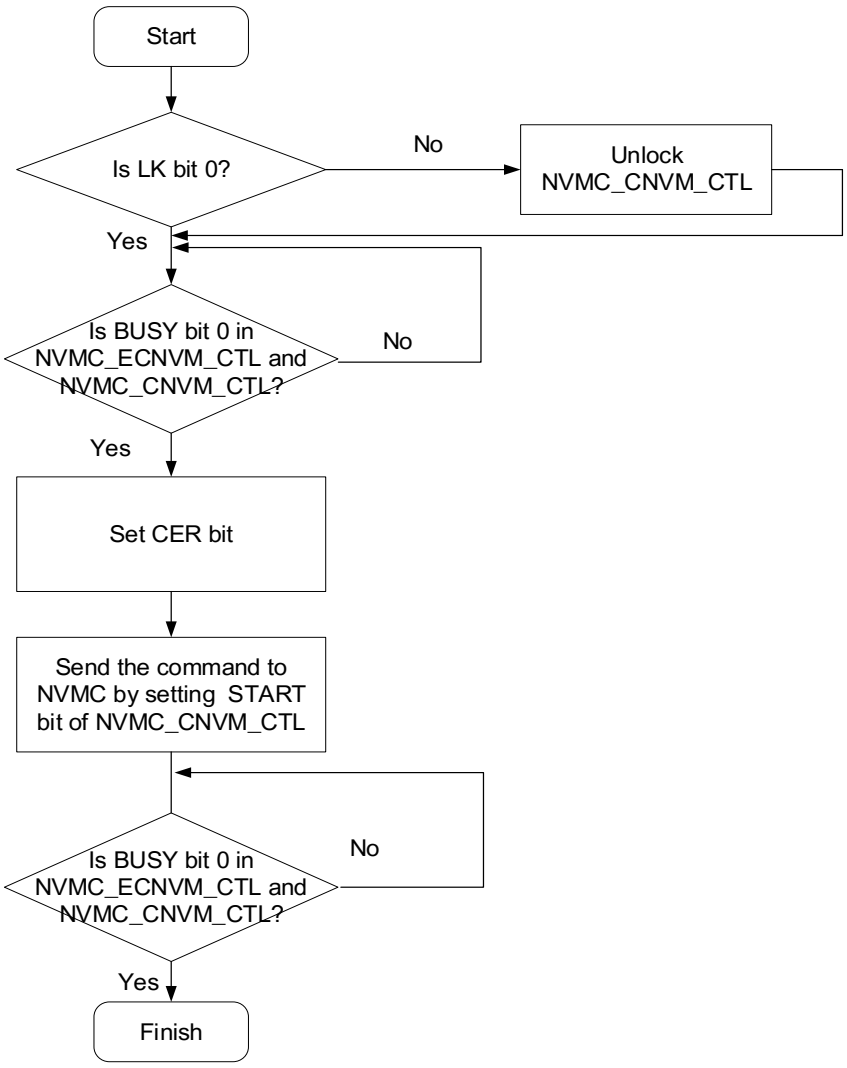
If chip erase and mass erase is request at the same time, the chip erase operation will replace the mass erase operation.

When the operation is executed successfully, the ENDF bit in both NVMC_CNVM_STAT and NVMC_ECNVM_STAT register is set, meanwhile, an interrupt will be triggered by NVMC if the ENDIE bit in the NVMC_CNVM_CTL register is set. Since all CNVM and ECNVM data except sectors contain secure user or protected data will be modified to a value of 0x0000_0000 (CNVM) / 0xFFFF_FFFF (ECNVM), the chip erase operation can be implemented using a program that runs in SRAM or by using the debugging tool that accesses the NVMC registers directly.

Note: The function of ECNVM_ENDF bit in the NVMC_CNVM_STAT is the same as the function of ENDF bit in the NVMC_ECNVM_STAT.

Figure 7-6. Process of typical chip erase operation shows the typical chip erase operation flow.

Figure 7-6. Process of typical chip erase operation



Note: When programming, erasing, especially the chip erasing, abnormal power off or reset should be avoided as far as possible, otherwise unpredictable consequences may occur. If there is a risk of power off and reset, avoid using protection-removed chip erasing.

Protection-removed chip erase

The NVMC provides a protection-removed chip erase function which is used to erase all sectors include sectors contain secure user or protected data. This erase can affect by setting the CER bit to 1 in the NVMC_CNVM_CTL register. The following steps show the protection-removed chip erase register access sequence.

1. Unlock the NVMC_ECNVM_CTL register if necessary.
2. Set CHIP_REMOVE_EN bit in NVMC_ECNVM_CTL register.
3. Unlock the NVMC_OBCTL register if necessary.
4. If a DCRP area exists, set XX_DCRP_EN and XX_DCRP_EREN bit in NVMC_DCRPADDR_MDF register. And set DCRP end address less than DCRP start address by programming DCRP_AREA_END < DCRP_AREA_START to

NVMC_DCRPADDR_MDF register. (XX represents CNVM, ECNVM_BANK0, ECNVM_BANK1, the same below):

5. If a secure user area exists, set XX_SCR_EN and XX_SCR_EREN bit in NVMC_SCRADDR_MDF register. And set secure user area end address less than secure user area start address by programming SCR_AREA_END < SCR_AREA_START to NVMC_SCRADDR_MDF register.
6. Set all WP bits in NVMC_WP0_ECNVM_MDF, NVMC_WP1_ECNVM_MDF and NVMC_WP_CNVM_MDF register if any erase / program protected sector exists.
7. Unlock the NVMC_CNVM_CTL register if necessary.
8. Check the BUSY bit in NVMC_CNVM_STAT register to confirm that no CNVM memory operation is in progress (BUSY equals to 0). Otherwise, wait until the operation has finished. At the same time, check the BUSY bit in NVMC_ECNVM_STAT register to confirm that no ECNVM memory operation is in progress (BUSY equals to 0). Otherwise, wait until the operation has finished.
9. Set the CER bit in the NVMC_CNVM_CTL register to enable chip erase operation.
10. Send the chip erase command to the NVMC by setting the START bit in NVMC_CNVM_CTL register.
11. Wait until all the operations have been finished by checking the value of the BUSY bit in NVMC_CNVM_STAT and NVMC_ECNVM_STAT register. The operation has erased the whole CNVM and ECNVM memory including the sectors containing DCRP-protected and / or secure user data. And the correspond option bytes modified automatically performed for disable all the protections.
12. Read and verify the CNVM and ECNVM memory if required.

Note: (1) In the above sequence, except for the option bytes indicated above, other option bytes cannot be modified. (2) Only if all the conditions in steps 4, 5, and 6 are satisfied, the protection-removed chip erase will be performed. But if any of the steps are not met, only the typical chip erase will be performed, and no error will be raised.

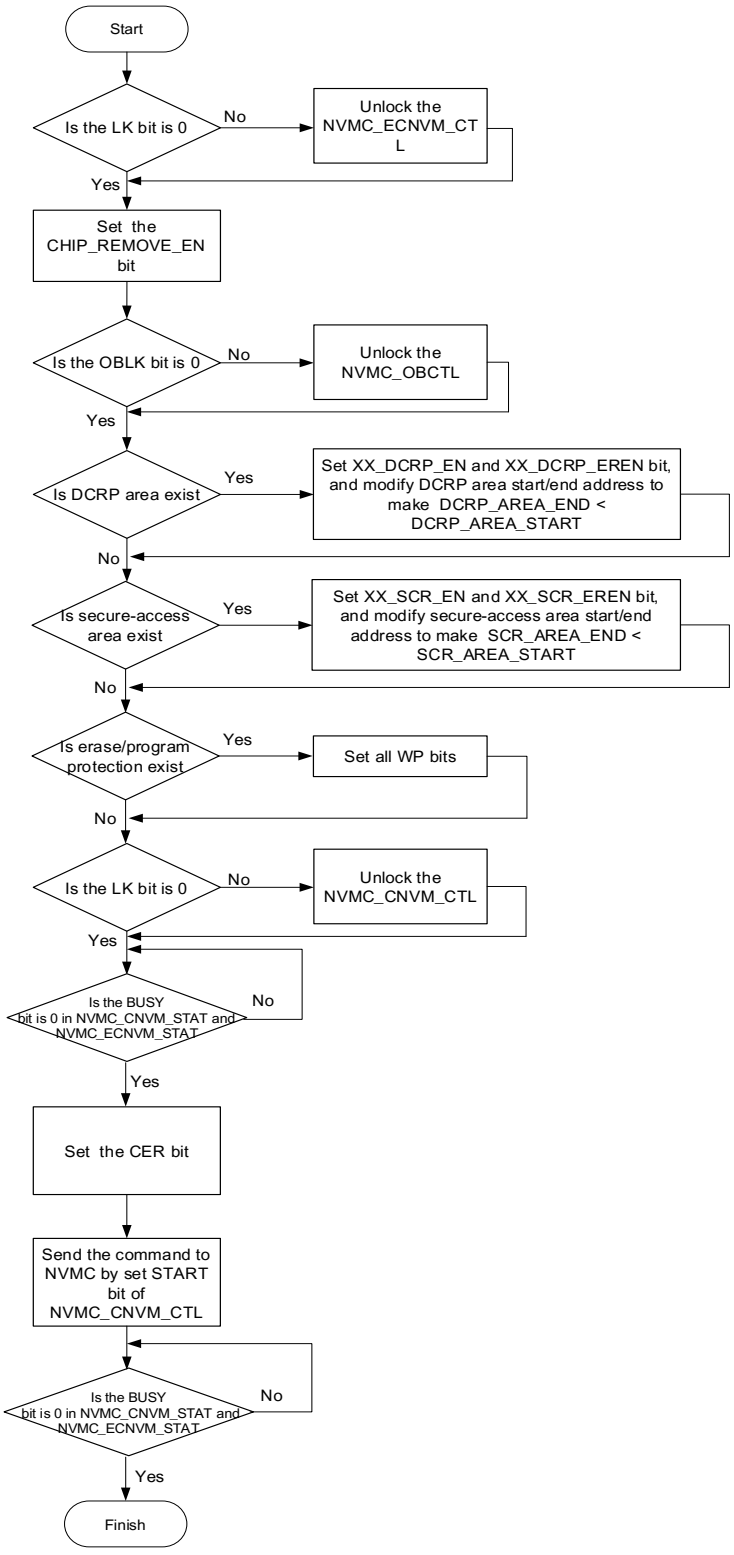
If chip erase and mass erase is request at the same time, the chip erase operation will replace the mass erase operation.

When the operation is executed successfully, the ENDF bit in both NVMC_CNVM_STAT and NVMC_ECNVM_STAT register is set, meanwhile, an interrupt will be triggered by NVMC if the ENDIE bit in the NVMC_CNVM_CTL register is set. Since all CNVM and ECNVM data (including secure user or protected data) will be modified to a value of 0x0000_0000 (CNVM) / 0xFFFF_FFFF (ECNVM), the chip erase operation can be implemented using a program that runs in SRAM or by using the debugging tool that accesses the NVMC registers directly.

Note: The function of ECNVM_ENDF bit in the NVMC_CNVM_STAT is the same as the function of ENDF bit in the NVMC_ECNVM_STAT.

Figure 7-7. Process of protection-removed chip erase operation shows the protection-removed chip erase operation flow.

Figure 7-7. Process of protection-removed chip erase operation



Note: When programming, erasing, especially the chip erasing, abnormal power off or reset should be avoided as far as possible, otherwise unpredictable consequences may occur. If there is a risk of power off and reset, avoid using protection-removed chip erasing.

CNVM Programming

The NVMC provides a 256-bit programming function by AXI interface which is used to modify the CNVM block contents. User must unlock NVMC_CNVM_CTL register and set PG bit before program the CNVM.

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

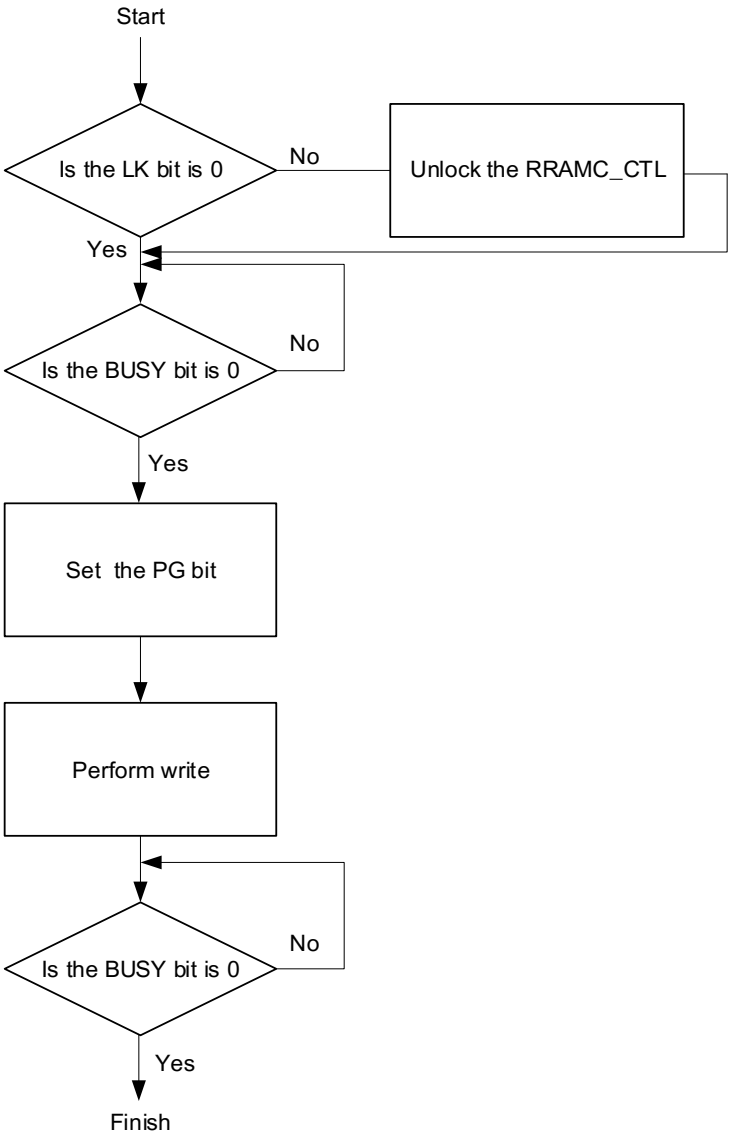
1. Unlock the NVMC_CNVM_CTL register if necessary.
2. Check the BUSY bit in NVMC_CNVM_STAT register to confirm that no CNVM memory operation is in progress (BUSY equals to 0). Otherwise, wait until the operation has finished.
3. Set the PG bit in NVMC_CNVM_CTL register.
4. Write the data to be programed with desired absolute address (0x08XX XXXX).
5. Wait until all the operations have been finished by checking the value of the BUSY bit in NVMC_CNVM_STAT register.
6. Read and verify the CNVM memory if required.

Note: Users must strictly follow the above CNVM programming procedures. Do not set OBSTART, REPSTART, CRCSTART, or any other types of NVM read/write function bits during the programming process.

Since the CPU cache may retain previously read data, it is recommended to either disable the cache before programming the NVM, or perform a cache clean and invalidate operation before reading back the programmed data, in order to prevent stale cached data from being read.

When the operation is executed successfully, the ENDF in NVMC_CNVM_STAT register is set, and an interrupt will be triggered by NVMC if the ENDIE bit in the NVMC_CNVM_CTL register is set. Note that the PG bit must be set before the 256-bit programming operation, or else PGSERR bit in the NVMC_CNVM_STAT register will be set, and an interrupt will be triggered by NVMC if the PGSERRIE bit in the NVMC_CNVM_CTL register is set. Additionally, the program operation will be ignored on erase / program protected sectors and WPERR bit in NVMC_CNVM_STAT is set, and an interrupt will be triggered by NVMC if the WPERRIE bit in the NVMC_CNVM_CTL register is set. The software can check the PGSERR or WPERR bit in the NVMC_CNVM_STAT register to detect which condition occurred in the interrupt handler. ***Figure 7-8. Process of program operation*** displays the programming operation flow.

Figure 7-8. Process of program operation



Note: When programming, erasing, especially the mass erasing, abnormal power off or reset should be avoided as far as possible, otherwise unpredictable consequences may occur. If there is a risk of power off and reset, avoid using protection-removed chip erasing.

CRC

CNVM has a cyclic redundancy check (CRC) hardware module. The integrity of the contents of the CNVM region can be verified by this module. The region can be defined by NVMC_OBREPSTADDR_MDF and the NVMC_OBREPENDADDR_MDF.

The CRC32 algorithm is used to calculate each 256bit of data read out, calculating the lower 32 bits first, and then increasing. And so on.

Fixed polynomial: 0x04C11DB7

$X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$

This 32-bit CRC polynomial is a common polynomial used in Ethernet.

The following steps show the register access sequence of the CRC configuring operation:

1. Unlock the NVMC_CNVM_CTL register if necessary.
2. Program the area address size in the NVMC_OBREPSTADDR_MDF and the NVMC_OBREPENDADDR_MDF.
3. Set the CRCSTART bit.
4. Wait until all the operations have finished by checking the value of the CRCBUSY bit in NVMC_CNVM_STAT register.
5. Gets the CRC result from the NVMC_CNVM_CRCRESULT register.

Note: DCRP and secure user areas should avert calculating CRC since it may cause errors in the results.

CRC startup error condition:

1. The DCRP / secure area does not allow CRC calculation, otherwise, it will report a CRCRERR.
2. When protection level low, in debug mode, boot from SRAM or boot from BootLoader mode, CNVM CRC is forbidden.
3. The NVMC_OBREPSTADDR_MDF cannot be greater than the NVMC_OBREPENDADDR_MDF.
4. The CRC area contains valid CNVM bad block (Storage cells in CNVM with hard defects that cannot correctly write 1 or 0) addresses.

Note: In each of these cases, an error is reported: CRCRERR.

Error description

Erase / program protection error (WPERR)

The following erase operations will be rejected and the WPERR bit in the NVMC_CNVM_STAT register will be set:

- Erase sectors in the valid DCRP area;
- Erase sectors in the valid secure user area (unless the application is executed from a valid secure user area);
- Erase the valid erase / program protection sectors;
- Erase the area not belong to the CNVM and ECNVM block.

The following program operations will be ignored and the WPERR bit in the NVMC_CNVM_STAT register will be set:

- When SPC is protection level low, in debug mode, boot from SRAM or boot from bootloader mode, program the CNVM block.
- Program sectors in the valid DCRP area unless executing GD secure library;
- Program sectors in the valid secure user area unless executing user security code or GD

secure library;

- Program the valid erase / program protection sectors.
- Program the area not belong to the CNVM and ECNVM block.

If WPERRIE bit in NVMC_CNVM_CTL register is set to 1, an interrupt is generated when WPERR flag is set. The software can clear it by writing 1.

Program sequence error (PGSERR)

The following operation will set the PGSERR in the NVMC_CNVM_STAT register, and the current program operation is aborted:

- When a program operation is requested but the program enable bit PG has not been set in NVMC_CNVM_CTL register
- Programming CNVM area without writing in 64bit.
- Programming CNVM area addresses are not in the same 256bit range.

Note: Data written in 256-bit should be sent out in sequential order.

If PGSERRIE bit in NVMC_CNVM_CTL register is set to 1, an interrupt is generated when PGSERR flag is set. The software can clear it by writing 1.

Read protection error (RPERR)

When attempt a read operation to a DCRP, a SPC protected area, NVMC will set the RPERR in the NVMC_CNVM_STAT register, and the current read operation is aborted, but user can request a new read operation.

If RPERRIE bit in NVMC_CNVM_CTL register is set to 1, an interrupt is generated when RPERR flag is set. The software can clear it by writing 1.

Read secure error (RSERR)

When attempt a read operation to a secure user area, NVMC will set the RSERR in the NVMC_CNVM_STAT register, and the current read operation is aborted.

If RSERRIE bit in NVMC_CNVM_CTL register is set to 1, an interrupt is generated when RSERR flag is set. The software can clear it by writing 1.

CNVM ECC error (AUTOECCDERRDET)

When ECC function is enable and two errors are detected, the AUTOECCDERRDET bit in NVMC_CNVM_STAT register will be set. If the AUTOECCDERRDETIE bit in NVMC_CNVM_CTL register is set, an interrupt is generated. The software can clear it by clearing all AUTOERR2ADDRSTAT of NVMC_CNVM_AUTOERR2ADDRx (x = 0…3).

Hard fault errors

The following operation will generate a bus error:

- When SPC protection level low, in debug mode, boot from SRAM or boot from boot loader mode, access CNVM block.
- Access secure user area without correct access rights.
- Access address out of range.
- Any wrong sequence to unlock the NVMC_CNVM_CTL and NVMC_OBCTL register.

CNVM interrupts

The NVMC interrupt events and flags are listed in *Table 7-6. CNVM interrupt requests*

Table 7-6. CNVM interrupt requests

Interrupt flag	Descriptions	Clear method	Interrupt enable bit
ENDF	Operation complete	Write 1 to the corresponding bit of the NVMC_CNVM_STAT register.	ENDIE
WPERR	Erase/Program protection error		WPERRIE
PGSERR	Programming sequence error		PGSERRIE
RPERR	Read protection error		RPERRIE
RSERR	Read security error		RSERRIE
AUTOECCDERRDET	Double error detection flag.		AUTOECCDERRDETIE
CRCENDCAL	CRC calculation complete flag.		CRCENDCALIE
CRCRERR	CRC read error flag.		CRCRERRIE
ECNVM_BLANK	ECNVM data integrity enhanced area content empty.	Write 1 to the corresponding bit of NVMC_CNVM_REPCS register	ECNVM_BLANKIE
REPEND	Self-check repair complete.		REPDONEIE
REPSTRERR	Self-check repair configuration error		

7.3.4. ECNVM controller (ECNVMC)

In this section, GD32H77xxI/GD32H78xxI has no functionality other than option bytes.

Read operations

Main ECNVM block and Information block are accessed by the AXI interface. And option bytes only can be accessed by AHB register interface. There is also a read buffer to improve efficiency of consecutive read requests.

The read buffer can prefetch data from ECNVM macro according to the AXI read address base on register setting. If AXI request address of multiple continuous read access hit the read buffer, read data will be directly sent to AXI bus from the read buffer without triggering additional ECNVM memory read operations. The user can utilize this feature to significantly improve the performance of sequentially executed programs. This mechanism occurs every time the read access permission is granted. When read access is denied for security reasons (e. g., the DCRP protected word), the ECNVM issues a corresponding read error response and does not trigger a read operation to the ECNVM memory.

Unlock the NVMC_ECNVM_CTL register

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

Page erase

The NVMC provides a page erase function which is used to initialize the contents of a ECNVM block page to a high state. Each page can be erased independently without affecting the contents of other pages. The following steps show the access sequence of the registers for a page (4 Kbytes) erase operation.

1. Unlock the NVMC_ECNVM_CTL register if necessary.
2. Check the BUSY bit in NVMC_ECNVM_STAT register to confirm that no ECNVM memory operation is in progress (BUSY equals to 0). Otherwise, wait until the operation has finished.
3. Set the PER bit in NVMC_ECNVM_CTL register.
4. Write the page address ADDR to NVMC_ECNVM_ADDR register.
5. Send the page erase command to the ECNVMC by setting the START bit in NVMC_ECNVM_CTL register.
6. Wait until all the operations have finished by checking the value of the BUSY bit in NVMC_ECNVM_STAT register.
7. Read and verify the page if required.

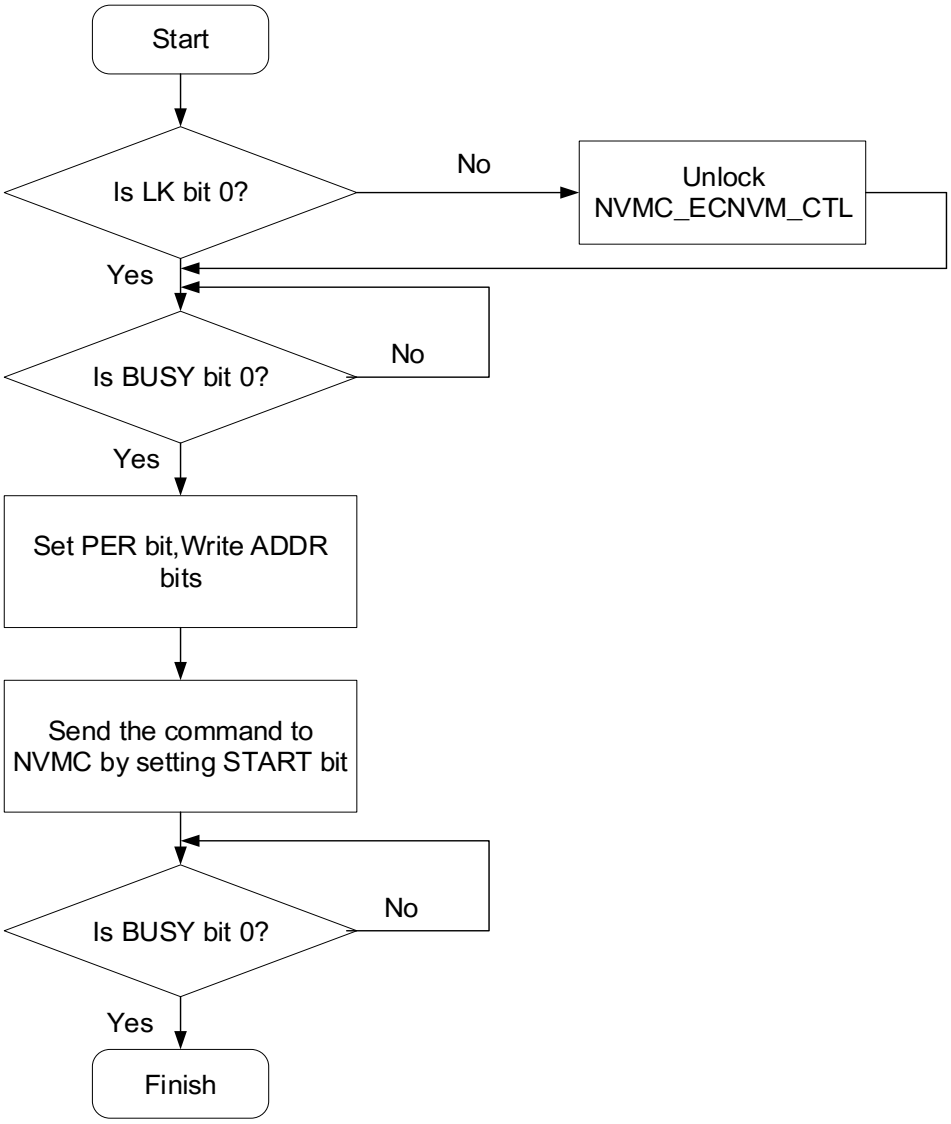
If a ECNVM sector contains DCRP, secure user or erase / program protected data, the page erase operation is stopped and the WPERR bit is set in the NVMC_ECNVM_STAT register.

If sector erase and page erase is request at the same time, the sector erase operation will replace the page erase operation.

When the operation is executed successfully, the ENDF in NVMC_ECNVM_STAT register is

set, and an interrupt will be triggered by NVMC if the ENDIE bit in the NVMC_ECNVM_CTL register is set. Note that a correct target page address must be confirmed. Or the software may run out of control if the target erase page is being used to fetch codes or to access data. The NVMC will not provide any notification when this occurs. Additionally, the page erase operation will be ignored on erase / program protected sectors. In this condition, an erase / program protection error interrupt will be triggered by the NVMC if the WPERRIE bit in the NVMC_ECNVM_CTL register is set. The software can check the WPERR bit in the NVMC_ECNVM_STAT register to detect this condition in the interrupt handler. ***Figure 7-9. Process of page erase operation*** shows the page erase operation flow.

Figure 7-9. Process of page erase operation



Note: When programming, erasing, especially the chip erasing, abnormal power off or reset should be avoided as far as possible, otherwise unpredictable consequences may occur. If there is a risk of power off and reset, avoid using protection-removed chip erasing.

Sector erase

The NVMC provides a sector erase function which is used to initialize the contents of a ECNVM block sector to a high state. Each sector can be erased independently without affecting the contents of other sectors. The following steps show the access sequence of the registers for a sector (64 Kbytes) erase operation.

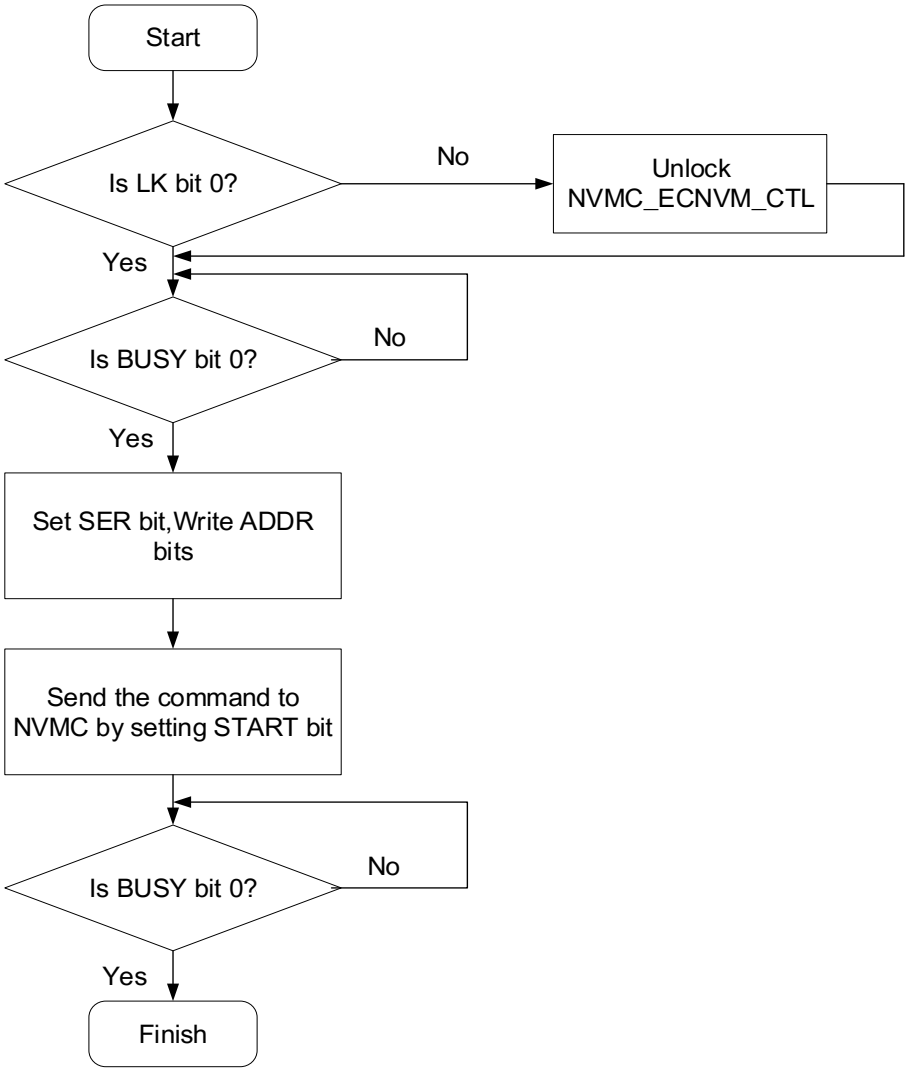
1. Unlock the NVMC_ECNVM_CTL register if necessary.
2. Check the BUSY bit in NVMC_ECNVM_STAT register to confirm that no ECNVM memory operation is in progress (BUSY equal to 0). Otherwise, wait until the operation has finished.
3. Set the SER bit in NVMC_ECNVM_CTL register.
4. Write the sector address ADDR to NVMC_ECNVM_ADDR register.
5. Send the sector erase command to the NVMC by setting the START bit in NVMC_ECNVM_CTL register.
6. Wait until all the operations have finished by checking the value of the BUSY bit in NVMC_ECNVM_STAT register.
7. Read and verify the sector if required.

If a ECNVM sector contains DCRP, secure user or erase / program protected data, the sector erase operation is stopped and the WPERR bit is set in the NVMC_ECNVM_STAT register.

If mass erase and sector erase is request at the same time, the mass erase operation will replace the sector erase operation.

When the operation is executed successfully, the ENDF in NVMC_ECNVM_STAT register is set, and an interrupt will be triggered by NVMC if the ENDIE bit in the NVMC_ECNVM_CTL register is set. Note that a correct target sector address must be confirmed. Or the software may run out of control if the target erase sector is being used to fetch codes or to access data. The NVMC will not provide any notification when this occurs. Additionally, the sector erase operation will be ignored on erase / program protected sectors. In this condition, an erase / program protection error interrupt will be triggered by the NVMC if the WPERRIE bit in the NVMC_ECNVM_CTL register is set. The software can check the WPERR bit in the NVMC_ECNVM_STAT register to detect this condition in the interrupt handler. ***Figure 7-10. Process of sector erase operation*** shows the sector erase operation flow.

Figure 7-10. Process of sector erase operation



Note: When programming, erasing, especially the chip erasing, abnormal power off or reset should be avoided as far as possible, otherwise unpredictable consequences may occur. If there is a risk of power off and reset, avoid using protection-removed chip erasing.

Mass erase

The ECNVMC provides a mass erase function which is used to initialize the contents of ECNVM block to a high state. This erase can affect by setting the MER bit to 1 in the NVMC_ECNVM_CTL register. The following steps show the mass erase register access sequence.

1. Unlock the NVMC_ECNVM_CTL register if necessary.
2. Check the BUSY bit in NVMC_ECNVM_STAT register to confirm that no ECNVM memory operation is in progress (BUSY equals to 0). Otherwise, wait until the operation has finished.
3. Set the MER bit in the NVMC_ECNVM_CTL register to enable mass erase operation.

4. Send the mass erase command to the NVMC by setting the START bit in NVMC_ECNVM_CTL register.
5. Wait until all the operations have been finished by checking the value of the BUSY bit in NVMC_ECNVM_STAT register.
6. Read and verify the ECNVM memory if required.

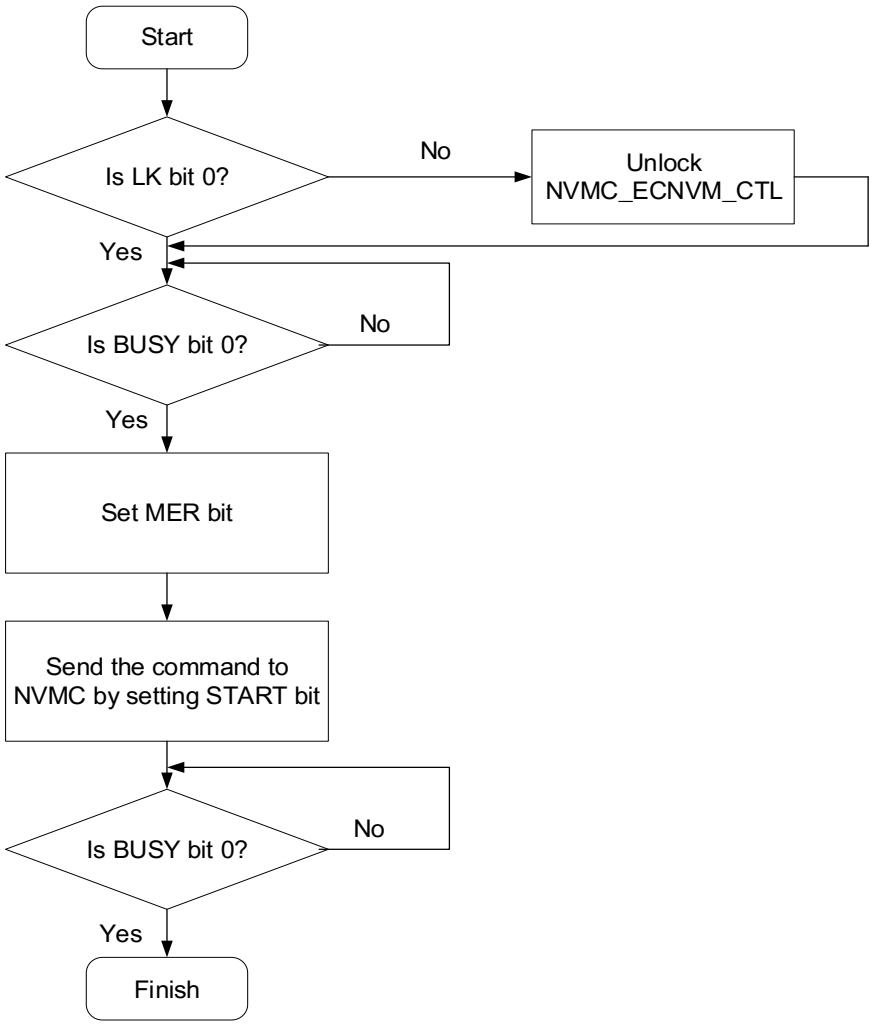
If ECNVM contains DCRP, secure user or erase/program protected data, the mass erase operation is stopped and the WPERR bit is set in the NVMC_ECNVM_STAT register.

If mass erase and sector erase is request at the same time, the mass erase operation will replace the sector erase operation.

When the operation is executed successfully, the ENDF in NVMC_ECNVM_STAT register is set, and an interrupt will be triggered by NVMC if the ENDIE bit in the NVMC_ECNVM_CTL register is set. In this condition, an erase / program protection error interrupt will be triggered by the NVMC if the WPERRIE bit in the NVMC_ECNVM_CTL register is set. The software can check the WPERR bit in the NVMC_ECNVM_STAT register to detect this condition in the interrupt handler.

Figure 7-11. Process of mass erase operation shows the mass erase operation flow.

Figure 7-11. Process of mass erase operation



Note: When programming, erasing, especially the chip erasing, abnormal power off or reset should be avoided as far as possible, otherwise unpredictable consequences may occur. If there is a risk of power off and reset, avoid using protection-removed chip erasing.

ECNVM Programming

The ECNVM provides a 256-bit / 64-bit programming function by AXI interface which is used to modify the ECNVM block contents. User must unlock NVMC_ECNVM_CTL register and set PG bit before program the user ECNVM.

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

1. Unlock the NVMC_ECNVM_CTL register if necessary.
2. Check the BUSY bit in NVMC_ECNVM_STAT register to confirm that no ECNVM memory operation is in progress (BUSY equals to 0). Otherwise, wait until the operation has finished.
3. Set the ECNVM_PG_SZ bit in NVMC_PERFCTL register as required.

4. Set the PG bit in NVMC_ECNVM_CTL register.
5. Write the 256-bit / 64-bit data to be programed with desired absolute address (0x08XX XXXX).
6. Wait until all the operations have been finished by checking the value of the BUSY bit in NVMC_ECNVM_STAT register.
7. Read and verify the ECNVM memory if required.

Note: Users must strictly follow the above ECNVM programming procedures. Do not set OBSTART, REPSTART, CRCSTART, or any other types of NVM read/write function bits during the programming process.

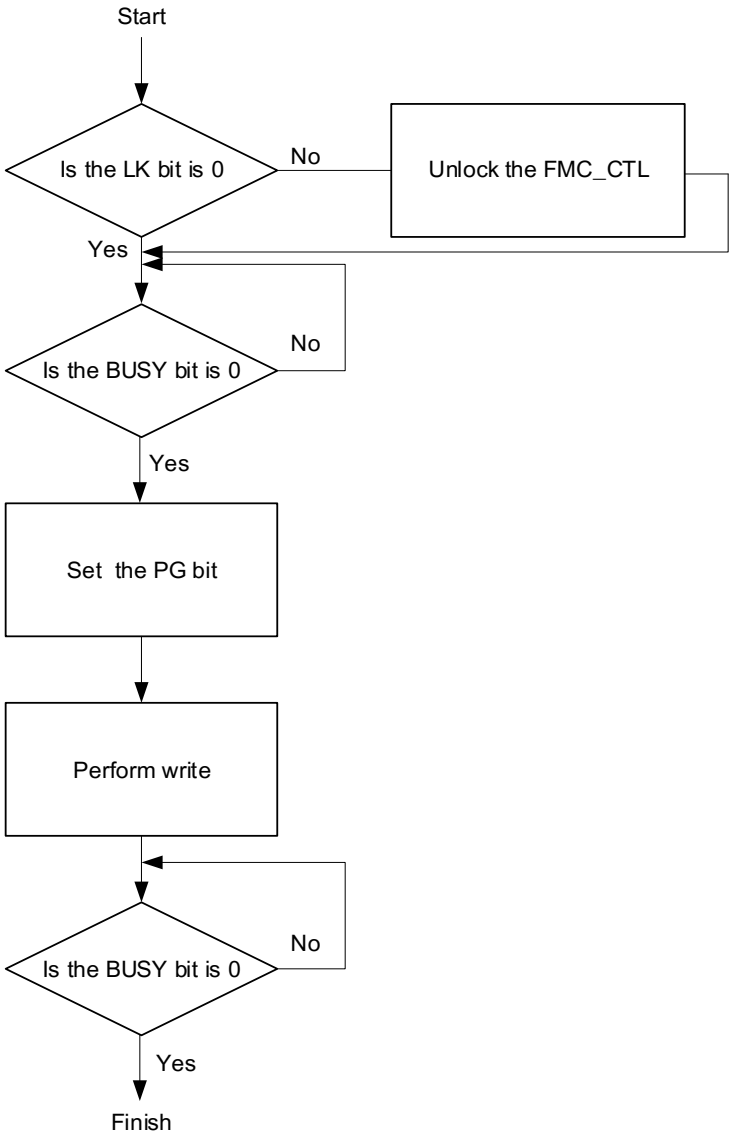
Note: Application software cannot perform programming operations on information block.

When the operation is executed successfully, the ENDF in NVMC_ECNVM_STAT register is set, and an interrupt will be triggered by NVMC if the ENDIE bit in the NVMC_ECNVM_CTL register is set. Note that the PG bit must be set before the 256-bit / 64-bit programming operation, or else PGSERR bit in the NVMC_ECNVM_STAT register will be set, and an interrupt will be triggered by NVMC if the PGSERRIE bit in the NVMC_ECNVM_CTL register is set. Additionally, the program operation will be ignored on erase / program protected sectors and WPERR bit in NVMC_ECNVM_STAT is set, and an interrupt will be triggered by NVMC if the WPERRIE bit in the NVMC_ECNVM_CTL register is set. The software can check the PGSERR or WPERR bit in the NVMC_ECNVM_STAT register to detect which condition occurred in the interrupt handler. ***Figure 7-12. Process of program operation*** displays the word programming operation flow.

User can enable check whether the programming area is all 0xFF before programming by set PGCHEN bit in NVMC_ECNVM_CTL register. If the programming area is not all 0xFF, the PGSERR in the NVMC_ECNVM_STAT register will be set.

If PGCHEN is set and burst program is enable, NVMC will check the ECNVM data according to the burst size. If it is checked that the data is not all 0xFF, this beat program is failed and PGSERR bit in NVMC_ECNVM_STAT register is set. But other beats can be written normally.

Figure 7-12. Process of program operation



Note: When programming, erasing, especially the mass erasing, abnormal power off or reset should be avoided as far as possible, otherwise unpredictable consequences may occur. If there is a risk of power off and reset, avoid using protection-removed chip erasing.

Error description

Erase / program protection error (WPERR)

The following erase operations will be rejected and the WPERR bit in the NVMC_ECNVM_STAT register will be set:

- Erase sectors in the valid DCRP area;
- Erase sectors in the valid secure user area (unless the application is executed from a valid secure user area);

- Erase the valid erase / program protection sectors;
- Erase the area not belong to the ECNVM block.

The following program operations will be ignored and the WPERR bit in the NVMC_ECNVM_STAT register will be set:

- When SPC is protection level low, in debug mode, boot from SRAM or boot from bootloader mode, program the ECNVM block.
- Program sectors in the valid DCRP area unless executing GD secure library;
- Program sectors in the valid secure user area unless executing user security code or GD secure library;
- Program the valid erase / program protection sectors.
- Program the area not belong to the ECNVM block.

If WPERRIE bit in NVMC_ECNVM_CTL register is set to 1, an interrupt is generated when WPERR flag is set. The software can clear it by writing 1.

Program sequence error (PGSERR)

The following operation will set the PGSERR in the NVMC_ECNVM_STAT register, and the current program operation is aborted:

- When a program operation is requested but the program enable bit PG has not been set in NVMC_ECNVM_CTL register
- When PGCHEN bit is enable but the programming area is not all 0xFF before programming.

Note: If PGCHEN is set and burst program is enable, NVMC will check the ECNVM data according to the burst size. If it is checked that the data is not all 0xFF, this beat program is failed and PGSERR is set. But other beats can be programmed normally.

If PGSERRIE bit in NVMC_ECNVM_CTL register is set to 1, an interrupt is generated when PGSERR flag is set. The software can clear it by writing 1.

Read protection error (RPERR)

When attempt a read operation to a DCRP, a SPC protected area, NVMC will set the RPERR in the NVMC_ECNVM_STAT register, and the current read operation is aborted, but user can request a new read operation.

If RPERRIE bit in NVMC_ECNVM_CTL register is set to 1, an interrupt is generated when RPERR flag is set. The software can clear it by writing 1.

Read secure error (RSERR)

When attempt a read operation to a secure user area, NVMC will set the RSERR in the NVMC_ECNVM_STAT register, and the current read operation is aborted.

If RSERRIE bit in NVMC_ECNVM_CTL register is set to 1, an interrupt is generated when RSERR flag is set. The software can clear it by writing 1.

Option byte modify error (OBMERR)

When an error occurs during an option change operation, NVMC will set the OBMERR flag

in the NVMC_OBCTL register, and the current operation is aborted.

If OBMERRIE bit in NVMC_OBCTL register is set to 1, an interrupt is generated when OBMERR flag is set. The software can clear it by writing 1.

ECNVM ECC error (ECCDET)

When ECC function is enable and two errors are detected, the ECCDET bit in NVMC_ECNVM_STAT register will be set. If the ECCDETIE bit in NVMC_ECNVM_CTL register is set, an interrupt is generated. The software can clear it by writing 1.

Hard fault errors

The following operation will generate a bus error:

- When SPC protection level low, in debug mode, boot from SRAM or boot from boot loader mode, access ECNVM block.
- Access secure user area without correct access rights.
- Access address out of range.
- Any wrong sequence to unlock the NVMC_ECNVM_CTL / NVMC_OBCTL register.

ECNVM interrupts

The NVMC interrupt events and flags are listed in *Table 7-7. ECNVM interrupt requests*

Table 7-7. ECNVM interrupt requests

Flag	Description	Clear method	Interrupt enable bit
OBMERR	Option bytes modify error	Write 1 to corresponding bit in NVMC_OBCTL register	OBMERRIE
ENDF	End of operation	Write 1 to corresponding bit in NVMC_ECNVM_STAT register	ENDIE
WPERR	Write protection error		WPERRIE
PGSERR	Program sequence error		PGSERRIE
RPERR	Read protection error		RPERRIE
RSERR	Read secure error		RSERRIE
ECCDET	ECC 2-bit errors detect		ECCDETIE

7.3.5. Data integrity enhancement features

GD provides data integrity enhancement features for CNVM in GD32H77xxW/GD32H78xxW/GD32H77xxP/GD32H78xxP.

Self-repair is a key step in the data integrity enhancement features. This feature can be enabled after a reset or triggered via register settings. The main difference between these two methods lies in the configuration approach: one is based on the value of option bytes (PORRSTREPDIS and SYSRSTREPDIS), while the other can be set via software (REPSTART). For the register-triggered method, the entire NVMC will not accept any new AXI transfers until the action completes. In contrast, the reset-triggered method extends the reset release time. During the self-repair process, any failing CNVM address should be considered a bad block address and recorded in a read-only register

(NVMC_CNVM_ROBBADDRx). Depending on the chip's working state, users may choose to record these bad block addresses in option bytes (NVMC_OBBBADDRx_EFT).

The self-check range (NVMC_OBREPSTADDR_EFT and NVMC_OBREPENDADDR_EFT), initiation frequency, and other parameters have corresponding option bytes or registers. Users can configure them appropriately based on their specific business needs. For example, performing a full-range self-check during idle periods every few months, or conducting partial-area self-checks at each reset to balance power-on time while maintaining the NVM data integrity enhancement features.

7.3.6. Data integrity enhancement mode

Mode 1: Global data integrity enhancement mode, configures data enhancement for the entire CNVM. Suitable for demanding conditions to ensure NVM operates stably under harsh environmental interference. Enabling this mode comes with the following costs:

1. The write speed of CNVM will slow down along with the write speed of ECNVM.
2. The user must first erase the corresponding sector of the target programming address in ECNVM before programming CNVM.
3. Additionally, the user must program all the required data into this sector. These models do not support modifying data at a single address while keeping surrounding data unchanged. Typically, users should not program the active ECNVM data integrity enhancement area.

Mode 1 requires the following option bytes to be configured: PORRSTREPDIS set to 0, HWREPMOD set to 1, DIEEN set to 1, REPUNLK[1:0] set to 0x3, DIESZ[4:0] set to 0x1F, NVMC_OBREPENDADDR_EFT set to 0x081F FFE0, NVMC_OBREPSTADDR_EFT set to 0x0800 0000. The factory default values for the above option bytes are pre-configured, and users can directly use them. Additionally, users can set SYSRSTREPDIS to 0 to trigger self-repair during system reset.

Mode 2: Flexible Mode, users can flexibly configure the size of the CNVM area requiring enhancement based on their needs. For the data integrity enhancement area, the advantages and disadvantages are the same as Mode 1. For the non-data integrity enhancement area, users can achieve higher write speeds and gain more ECNVM storage space. However, the non-data integrity enhancement area lacks additional data integrity enhancement features, making it suitable for storing temporary data requiring fast write operations under complex conditions. Users can configure this flexibly according to their requirements.

Mode 2 requires the following option bytes to be configured: PORRSTREPDIS set to 0, HWREPMOD set to 1, DIEEN set to 1, REPUNLK[1:0] set to 0x3. DIESZ[4:0], NVMC_OBREPENDADDR_EFT, NVMC_OBREPSTADDR_EFT, and SYSRSTREPDIS can be configured based on user requirements.

Note: Only GD32H77xxW/GD32H78xxW/GD32H77xxP/GD32H78xxP support Mode 1 and Mode 2. GD32H77xxI/GD32H78xxI do not support these modes. The CNVM of GD32H77xxI/GD32H78xxI has the same programming and erasing characteristics as the

non-data integrity enhancement area in Mode 2.

Note: When the CPU is executing from ECNVM, do not perform the REPSTART self-check function; otherwise, the self-check cannot be completed.

7.3.7. CNVM bad block feature

The program memory bad block feature is provided in GD32H77xxW/GD32H78xxW/GD32H77xxP/GD32H78xxP, but not in GD32H77xxI/GD32H78xxI. This is another data integrity enhancement feature.

Option bytes can specify four CNVM bad block addresses (NVMC_OBBBADDRx_EFT). After prolonged use, users can configure bad block addresses identified during self-repair into the option bytes. When the AXI bus performs read access to these addresses, the hardware will automatically retrieve the correct redundant area data to ensure proper program execution. However, note that the read speed of bad block addresses will slow down. This impact can be mitigated or eliminated using CPU cache or the NVMC buffer.

Note: The addresses recorded in the registers NVMC_CNVM_ROBBADDRx, NVMC_CNVM_AUTOERR2ADDRx, and NVMC_BLANKADDR will be lost after power-down.

For the addresses recorded in the registers NVMC_CNVM_ROBBADDRx and NVMC_CNVM_AUTOERR2ADDRx, users can manually record them or write them into the option byte register NVMC_OBBBADDRx_MDF and activate them to prevent losing the data after power-down.

For the address recorded by the NVMC_BLANKADDR register, users should record it themselves to prevent loss due to power failure.

7.3.8. OTA

GD32H77xxW/GD32H78xxW support OTA functionality, while GD32H77xxP/GD32H78xxP/GD32H77xxI/GD32H78xxI do not support OTA functionality.

NVMC supports OTA firmware upgrade. ECNVM contains two enhanced CNVM data integrity areas: bank0 and bank1. The corresponding data integrity area is selected based on the option byte SWAP.

The following steps show the register operation sequence during the OTA upgrade process.

1. If the initial state of SWAP is 0, ECNVM bank0 serves as the CNVM data integrity enhanced area. If the SWAP bit is set to 1, ECNVM bank1 becomes the data integrity enhanced area.
2. If SWAP is initially 0, the user programs the new firmware into bank1 and configures the SWAP bit, FW_UPDATE bit, and OTA_CHECK bit in the option bytes.
3. After a system reset or power-on reset, the hardware will load the new firmware into CNVM if the permission configuration is correct.
4. After the new firmware update is completed, since the FW_UPDATE bit is set to 1, the

CPU will be released and jump to check_bootloader.

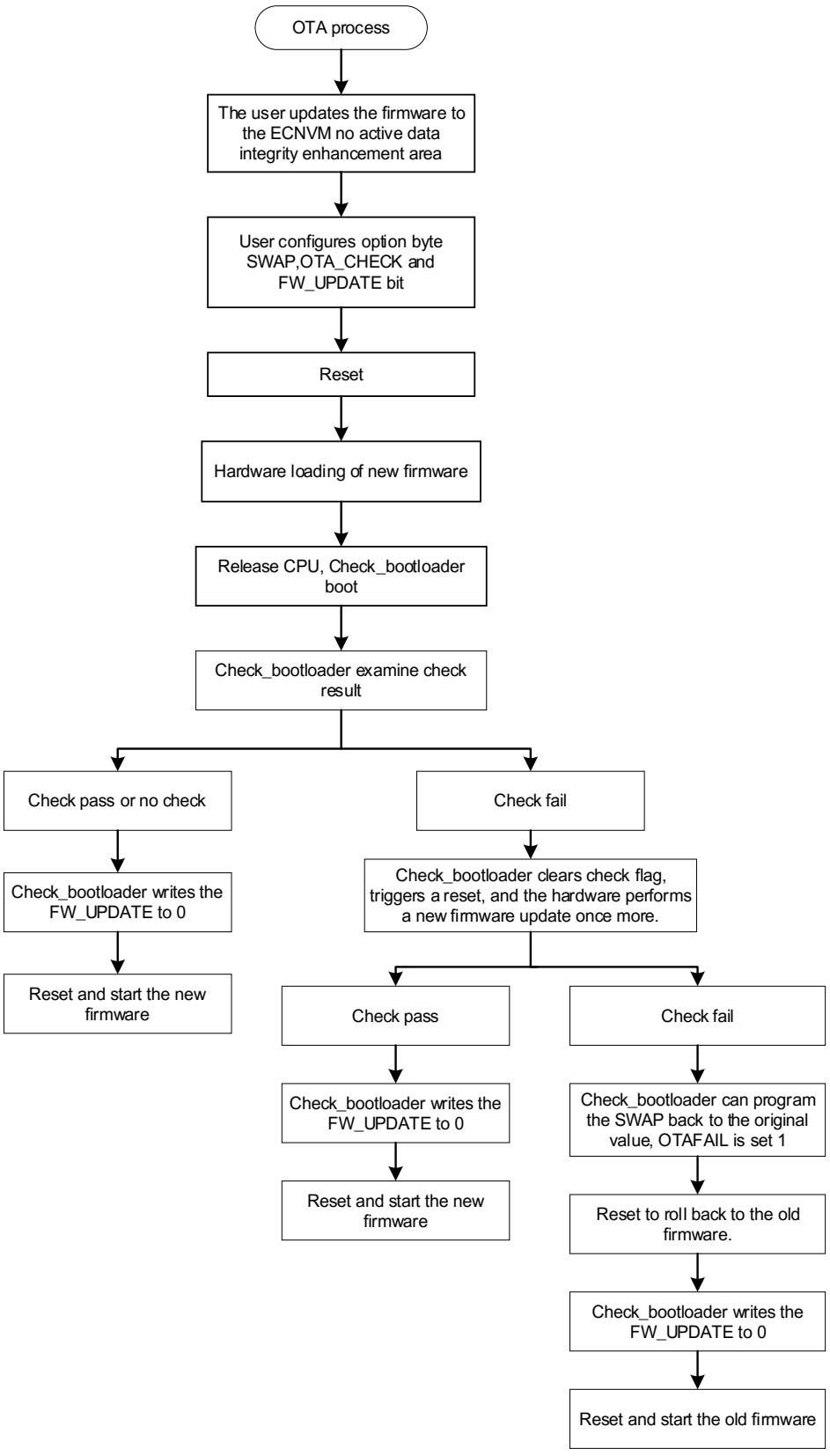
5. Check_bootloader verifies the checksum result. If the verification passes or is not performed, check_bootloader writes FW_UPDATE back to 0. OTA upgrade is successful, and check_bootloader sends a reset command to launch the new firmware.
6. When the system resets and FW_UPDATE is 0, the hardware starts according to the configuration, and the new program begins execution.
7. If the OTA check fails, the upgrade will also fail. Check_bootloader can choose to record the count and perform a reset to trigger hardware reprogramming. If the check still fails, check_bootloader can program SWAP back to its original value, set OTAFAIL to 1, point to the old firmware, and reset to trigger hardware to perform another update of the new firmware. At this point, FW_UPDATE remains set to 1.
8. Since the FW_UPDATE value is still 1, check_bootloader is launched again. Check_bootloader writes FW_UPDATE back to 0. Check_bootloader can also write to the corresponding registers to generate an interrupt upon update failure. Then, check_bootloader performs a system reset to roll back to the old firmware.

Note: In secure mode, if using OTA, CNVM_SCR_EN, ECNVM_BANK0_SCR_EN, and ECNVM_BANK1_SCR_EN must be set to 1.

Figure 7-13. OTA upgrade operation.

Shows the OTA upgrade operation flow.

Figure 7-13. OTA upgrade operation.



7.3.9. Data Integrity Enhanced Area function limitations

GD32H77xxI/GD32H78xxI have no Data Integrity Enhanced Area function limitations.

For security reasons, there are some limitations when using the Data Integrity Enhanced function.

1. When in Data Integrity Enhanced mode, the user should set the same WP and security settings in the CNVM and ECNVM Data Integrity Enhanced Area banks.
2. When programming CNVM in Data Integrity Enhanced mode, the sector erase/program protection bits WP of the corresponding CNVM and ECNVM Data Integrity Enhanced Area should be set to 1, and the security settings must be identical and valid. Otherwise, CNVM and Data Integrity Enhanced Area data may become inconsistent, leading to unintended modification of CNVM.
3. When programming CNVM in Data Integrity Enhanced mode, the hardware will check whether the content of the ECNVM Data Integrity Enhanced Area is empty.
4. The OTA update function will check the following items before starting: (1) The Data Integrity Enhanced Area in CNVM should not have WP protection. (2) The security settings of CNVM and the new ECNVM Data Integrity Enhanced Area should be identical.
5. The self-check repair function will check the following items before starting: (1) Whether the repair area settings are valid. (2) Whether the WP and security settings of CNVM and the Data Integrity Enhanced Area in the repair area are identical.
6. The repair function and OTA update function will bypass valid bad block addresses in CNVM.
7. When the security protection level is high, the OTA function is not supported.

Note: When the Data Integrity Enhanced mode is enabled and ECNVM is used as the Data Integrity Enhanced Area (ECNVM_DIE_DISABLE set to 0), if the swap function is not intended to be used (it is recommended to set SWAP_DISABLE to 1 in the NVMC_OTP_USER_CTL1 register), only the sector erase/program protection settings between CNVM and bank0 of the Data Integrity Enhanced Area need to be consistent. If the swap function is intended to be used, the sector erase/program protection settings of CNVM, bank0, and bank1 must all be consistent.

For example, for GD32H77xxW/GD32H78xxW, when the Data Integrity Enhanced mode is enabled and the swap function is not intended to be used, the SWAP bit in the NVMC_OBSTAT0_EFT register is set to 0. If sector erase/program protection is enabled for sector 0 - sector 1 in CNVM (WP[0] in NVMC_WP_CNVM_EFT is set to 0), the corresponding sector 112 - sector 113 in bank0 of the Data Integrity Enhanced Area must also enable sector erase/program protection (WP[40] in NVMC_WP1_ECNVM_EFT is set to 0).

For GD32H77xxW/GD32H78xxW, when the Data Integrity Enhanced mode is enabled and the swap function is intended to be used, if sector erase/program protection is enabled for sector 0 - sector 1 in CNVM (WP[0] in NVMC_WP_CNVM_EFT is set to 0), in addition to enabling sector erase/program protection for sector 112 - sector 113 in bank0 of the Data Integrity Enhanced Area (WP[40] in NVMC_WP1_ECNVM_EFT is set to 0), sector 80 - sector

81 in bank1 of the Data Integrity Enhanced Area must also enable sector erase/program protection (WP[24] in NVMC_WP0_ECNVM_EFT is set to 0).

7.3.10. Option bytes

Option bytes description

NVMC provides two groups of option byte registers. The option byte register is divided into two categories: opt and opt + reg.

For opt option byte register:

1. "_EFT" registers group (read-only)

These registers store the effective value of the option bytes. After system reset or wakeup from standby mode, the registers values will be automatically loaded from non-volatile memory.

2. "_MDF" registers group (read / write)

These registers store the modification of the option bytes. When user wants to modify the option byte, the desired option byte value needs to be written to the register group.

The option bytes block is reloaded to "_EFT" registers after each system reset, and the option bytes take effect.

For opt + reg option byte register:

3. "_EFT" registers group (read-only)

After system reset or wakeup from standby mode, the registers values will be automatically loaded from non-volatile memory.

4. "_MDF" registers group (read / write)

These registers store the modification of the option bytes. These registers store the effective value. When user wants to modify the option byte, the desired option byte value needs to be written to the register group.

If the option byte function is triggered during the reset phase, the corresponding configuration is the value of the option byte that was previously programmed. After reset, if the software enables the corresponding function of the option byte, the configuration will be the value in the "_MDF" registers. The software can set these "_MDF" registers value before enabling the corresponding function. The option bytes block is reloaded to "_EFT" and "_MDF" registers after each system reset.

Note: For opt + reg option byte register, the values in "_MDF" registers take effect without the need to program them into the NVM.

The option bytes description is shown in the ***Table 7-8. Option byte*** The option bytes are configured according to the requirements of the application.

Table 7-8. Option byte

Name	Factory value	Class	Register mapping
[31]: VFIMG	0x0	opt	NVMC_OBSTAT0_EFT / NVMC_OBSTAT0_MDF
[30]: NOPIPE_MODE	0x1	opt	
[29]: IOSPDOPEN	0x0	opt	
[28]: CNVM_ECC_OVRD	0x1	opt	
[27]: R0_SRAM_ECCEN	0x1	opt	
[26]: R1_SRAM_ECCEN	0x1	opt	
[25]: R2_SRAM_ECCEN	0x1	opt	
[24]: DTCM1ECCEN	0x1	opt	
[23]: DTCM0ECCEN	0x1	opt	
[22]: ITCMECCEN	0x1	opt	
[21]: SCR	0x0	opt	
[20]: SWAP	0x0	opt	
[19]: FW_UPDATE	0x0	opt	
[18]: FWDGSPD_STDBY	0x1	opt	
[17]: FWDGSPD_DPSLP	0x1	opt	
[16]: OTA_CHECK	0x0	opt	
[15:8]: SPC[7:0]	0xAA	opt	
[7]: nRST_STDBY	0x1	opt	
[6]: nRST_DPSLP	0x1	opt	
[5]: VFALL	0x0	opt	
[4]: nWDG_HW	0x1	opt	
[3:2]: BOR_TH[1:0]	0x0	opt	
[1]: NOPRINT	0x0	opt	
[0]: DISLFI	0x0	opt	
[31]: CNVM_DCRP_EREN	0x0	opt	NVMC_DCRPADDR_EFT / NVMC_DCRPADDR_MDF
[30]: ECNVM_BANK0_DCRP_EREN	0x0	opt	
[29]: ECNVM_BANK1_DCRP_EREN	0x0	opt	
[24:16]: DCRP_AREA_END[8:0]	0x0	opt	
[15]: CNVM_DCRP_EN	0x0	opt	
[14]: ECNVM_BANK0_DCRP_EN	0x0	opt	
[13]: ECNVM_BANK1_DCRP_EN	0x0	opt	
[8:0]: DCRP_AREA_START[8:0]	0x1FF	opt	
[31]: CNVM_SCR_EREN	0x0	opt	NVMC_SCRADDR_EFT / NVMC_SCRADDR_MDF
[30]: ECNVM_BANK0_SCR_EREN	0x0	opt	
[29]: ECNVM_BANK1_SCR_EREN	0x0	opt	
[24:16]: SCR_AREA_END[8:0]	0x0	opt	
[15]: CNVM_SCR_EN	0x0	opt	
[14]: ECNVM_BANK0_SCR_EN	0x0	opt	
[13]: ECNVM_BANK1_SCR_EN	0x0	opt	

Name	Factory value	Class	Register mapping
[8:0]: SCR_AREA_START[8:0]	0x1FF	opt	
[15:0]: WP[15:0]	0xFFFF	opt	NVMC_WP_CNVM_EFT / NVMC_WP_CNVM_MDF
[31:16]: BOOT_ADDR1[15:0]	0x1FEF	opt	NVMC_BTADDR_EFT / NVMC_BTADDR_MDF
[15:0]: BOOT_ADDR0[15:0]	0x080X	opt	
[31]: CPUBUSTO	0x1	opt	NVMC_OBSTAT1_EFT / NVMC_OBSTAT1_MDF
[30:16]: DATA[14:0]	0x7FFF	opt	
[6:4]: ITCM_DTCM_SHARED[2:0]	0x0	opt	
[2:0]: ITCM_AXI_SHARED[2:0]	0x0	opt	
[31:0]: WP[31:0]	0xFFFFFFFF	opt	NVMC_WP0_ECNVM_EFT / NVMC_WP0_ECNVM_MDF
[27:0]:WP[59:32]	0x0FFFFFFF	opt	NVMC_WP1_ECNVM_EFT / NVMC_WP1_ECNVM_MDF
[31:26]: UPDATESZ[5:0]	0x3F	opt	NVMC_OBREPCTL_EFT NVMC_OBREPCTL_MDF
[15]: PORRSTREPDIS	0x0	opt	
[14]: SYSRSTREPDIS	0x1	opt	
[13]: BBMODE	0x1	opt + reg	
[12:8]: DIESZ[4:0]	0x1F	opt	
[5]: BBREPLEN	0x1	opt + reg	
[4:3]: REPUNLK	0x3	opt	
[1]: HWREPMOD	0x1	opt	
[0]: DIEEN	0x1	opt	
[20:5]: STARTADDR[15:0]	0x0	opt + reg	NVMC_OBREPSTADDR_EFT NVMC_OBREPSTADDR_MDF
[20:5]: ENDADDR[15:0]	0xFFFF	opt + reg	NVMC_OBREPENDADDR_EFT NVMC_OBREPENDADDR_MDF
[20:5]: OBBBADDR[15:0]	0x0	opt	NVMC_OBBBADDRx_EFT / NVMC_OBBBADDRx_MDF
[0]: OBBBADDRVLD	0x0	opt + reg	

Option bytes details

■ Watchdog management

The registers related to watchdog management are FWDGSPD_DPSLP, FWDGSPD_STDBY and nWDG_HW.

FWDGT can work when FWDGSPD_DPSLP or FWDGSPD_STDBY is set 1. And the device is in deepsleep or standby mode, respectively. IF nWDG_HW is set to 0, the watchdog is automatically enabled when power on.

■ Reset management

The registers related to reset management are BOR_TH[1:0], nRST_STDBY and nRST_DPSLP.

When nRST_STDBY or nRST_DPSLP is cleared to 0, the device resets whenever it enters standby or deepsleep mode. BOR_TH[1:0] to set BOR threshold status bits.

■ I / O speed optimization

IOSPDOPEN is allowed enable status bit for I / O speed optimization at low-voltage. When IOSPDOPEN is set 1, chip operating voltage is less than 2.7V, I / O speed optimization is allowed.

■ Data protection

The registers related to data protection are SPC[7:0], WP[15:0], WP[59:0], DCRP_AREA_START[8:0], DCRP_AREA_END[8:0], ECNVM_BANK0_DCRP_EREN, ECNVM_BANK1_DCRP_EREN, CNVM_DCRP_EREN, SCR_AREA_START[8:0], SCR_AREA_END[8:0], ECNVM_BANK0_SCR_EREN, ECNVM_BANK1_SCR_EREN, CNVM_SCR_EREN.

■ Boot address

The registers related to Boot address are BOOT_ADDR1/0[15:0].

■ CNVM bad block management

When BBREPLEN is set 1, if AXI reads the bad block address in CNVM, the read data is automatically replaced by ECNVM content.

OBBBADDR[15:0] records the bad block address, which is automatically skipped during the self-check process and OTA update process.

■ CNVM data integrity enhancement mode management

DIEEN can be used to configure the CNVM data integrity enhancement mode.

When DIEEN is set to 1, DIESZ[4:0] can be used to configure the size of the data integrity enhancement range. The data integrity enhancement address starts from 0 with a granularity of 64KB. Hence, when set to 5'h1f, the 2MB space of CNVM is within the data integrity enhancement range.

■ CNVM self-check repair management

STARTADDR[15:0] and ENDADDR[15:0] are used to set the start address and end address of the self-check repair. Self-check end address must be same or greater than the start address.

Option bytes modify

The following steps show the option byte modify sequence.

1. Unlock the NVMC_OBCTL register if necessary.

2. Check the BUSY bit in NVMC_ECNVM_STAT register to confirm that no ECNVM memory operation is in progress (BUSY equals to 0). Otherwise, wait until the operation has finished. (For GD32H77xxI/GD32H78xxI, check the BUSY bit in NVMC_CNVM_STAT register to confirm that no CNVM memory operation is in progress (BUSY equals to 0). Otherwise, wait until the operation has finished.)
3. Write the desired option byte value in NVMC_XXX_MDF.
4. Send the option bytes change command to the NVMC by setting the OBSTART bit in NVMC_OBCTL register.
5. Wait until all the operations have been finished by checking the value of the BUSY bit in NVMC_ECNVM_STAT register. (For GD32H77xxI/GD32H78xxI, wait until all the operations have been finished by checking the value of the BUSY bit in NVMC_CNVM_STAT register.)
6. Launch a system reset to start option bytes loading.
7. Read from NVMC_XXX_EFT register and verify the option bytes if required.

Note: "XXX" includes OBSTAT0, DCRPADDR, SCRADDR, WP_CNVM, BTADDR, OBSTAT1, WP0_ECNVM, WP1_ECNVM, OBREPCTL, OBREPSTADDR, OBREPENDADDR or OBBBADDRx.

When the operation is executed successfully, the ENDF in NVMC_ECNVM_STAT register is set, and an interrupt will be triggered by NVMC if the ENDIE bit in the NVMC_ECNVM_CTL register is set. (For GD32H77xxI/GD32H78xxI, when the operation is executed successfully, the ENDF in NVMC_CNVM_STAT register is set, and an interrupt will be triggered by NVMC if the ENDIE bit in the NVMC_CNVM_CTL register is set.)

If OBLK bit in the NVMC_OBCTL register is set, the _MDF register is forbidden to modify. When the OBSTART bit is set to 1, NVMC will compare the effective value (_EFT) with a new value (_MDF) to check whether have option byte needs to be changed.

Option bytes modify rules

Some of the option bytes modification must respect specific rules. If one of the rules below is not met, OBMERR flag is set and modification operation is stopped. If all rules below are met, NVMC starts to modify option byte and updates the _EFT option byte register value.

■ Security protection (SPC)

If SPC is set to level high, any option byte modification operation is forbidden. Therefore, if the user application attempts to demote the SPC level, OBMERR flag is set and all programmed modification will be ignored.

■ Erase / program protection (WP)

These option bytes manage erase / program protection in the NVMC_WP0_ECNVM_EFT, NVMC_WP1_ECNVM_EFT, NVMC_WP_CNVM_EFT register. If the SPC protection level is not level high, they can be modified without any restriction.

■ DCRP area size (DCRP_AREA_START and DCRP_AREA_END)

The size of DCRP area can be increased without any restriction, but it must contain the original DCRP area, which means DCRP_AREA_START cannot be larger and DCRP_AREA_END cannot be smaller.

When removing the DCRP area, the XX_DCRP_EREN bit (in the NVMC_DCRPADDR_EFT or NVMC_DCRPADDR_MDF register) must be set to 1, and a SPC level low to no protection demotion or a protection-removed mass erase must be performed simultaneously (XX represents CNVM, ECNVM_BANK0、ECNVM_BANK1, the same below).

When reducing the size of DCRP area, the XX_DCRP_EREN bit (in the NVMC_DCRPADDR_EFT or NVMC_DCRPADDR_MDF register) must be set to 1, and a SPC level low to no protection demotion must be performed simultaneously.

Note: When removing or reducing the size of DCRP area by SPC level low to no protection demotion, if the XX_DCRP_EREN bits (in the NVMC_DCRPADDR_EFT and NVMC_DCRPADDR_MDF registers) are both 0, an option byte modification error (OBMERR) will be generated.

■ XX_DCRP_EREN

If this option bit is set, the content of the corresponding DCRP area is erased during SPC level low to no protection demotion or a protection-removed mass erase. Otherwise will be preserved.

The XX_DCRP_EREN bit can be set without any restriction. The XX_DCRP_EREN bit can be reset only when the SPC level low to no protection demotion or a protection-removed mass erase is requested simultaneously. Otherwise OBMERR flag is set.

■ Secure mode (SCR)

If there is no valid secure user area and no valid DCRP area, SCR option bit can be reset freely. If a valid DCRP area or a valid secure user area exist, the way to reset SCR option bit is: to perform a SPC level low to no protection demotion or protection-removed chip erase when XX_DCRP_EREN (in NVMC_DCRPADDR_EFT or NVMC_DCRPADDR_MDF register) is set to 1 and XX_SCR_EREN (in NVMC_SCRADDR_EFT or NVMC_SCRADDR_MDF register) is set to 1. Otherwise OBMERR flag is set.

Note: It is recommended to make both SCR_AREA_START > SCR_AREA_END and DCRP_AREA_START > DCRP_AREA_END programmed when reset the SCR option bit during an SPC level low to no protection demotion or protection-removed chip erase.

■ Secure user area size (SCR_AREA_START and SCR_AREA_END)

GD secure libraries run on user secure applications or devices can modify them without any restrictions.

For user non-secure applications, when removing the secure user area, the XX_SCR_EREN bit (in the NVMC_SCRADDR_EFT or NVMC_SCRADDR_MDF register) must be set to 1, and a SPC level low to no protection demotion or a protection-removed mass erase must be performed simultaneously.

Note: When removing or reducing the size of SCR area by SPC level low to no protection demotion, if the XX_SCR_EREN bits (in the NVMC_SCRADDR_EFT and NVMC_SCRADDR_MDF registers) are both 0, an option byte modification error (OBMERR) will be generated.

■ XX_SCR_EREN

If XX_SCR_EREN bit is set, the content of the corresponding secure user area is erased during a SPC level low to no protection demotion or a protection-removed mass erase. Otherwise the content is preserved.

The XX_SCR_EREN bit can be set without any restriction. The XX_SCR_EREN bits can be reset only when the SPC level low to no protection demotion or a protection-removed mass erase is requested simultaneously. Otherwise OBMERR flag is set.

7.3.11. Sector erase / program protection

GD32H77xxI/GD32H78xxI has no ECNVM sector erase / program protection function, but the sector erase / program protection function of CNVM still exists.

The NVMC provides sector erase / program protection functions to prevent inadvertent operations on the ECNVM memory. The sector erase or program will not be accepted by the NVMC on protected sectors. If the sector erase or program command is sent to the NVMC on a protected sector, the WPERR bit in the NVMC_ECNVM_STAT (NVMC_CNVM_STAT) register will then be set by the NVMC. Note that the WPERR also set when sector erase while sector address not valid. The sector protection function can be individually enabled by configuring the WP[15:0] / WP[59:0] bit field to 0 in the option bytes.

Table 7-9. WP bit for sectors protected

Block	WP bit	Sectors protected
CNVM	WP[0]	Sector 0~Sector 1
	WP[1]	Sector 2~Sector 3
	. . .	. . .
	WP[7]	Sector14~Sector 15
	. . .	. . .
	WP[15]	Sector 30~Sector 31
ECNVM	WP[0]	Sector 32~Sector 33
	WP[1]	Sector 34~Sector 35
	. . .	. . .
	WP[29]	Sector 90~Sector 91

Block	WP bit	Sectors protected
	WP[30]	Sector 92~Sector 93
	. . .	. . .
	WP[59]	Sector 150~Sector 151

Note: For WP [x] (x=0 … 15), 1 bit corresponds to 2 sectors. For WP [x] (x=0 … 59), 1 bit corresponds to 2 sectors.

Erase / program protected sectors cannot either be deleted or programmed. Therefore, if a sector is erase / program protected, mass erase cannot be performed unless a protection-removed chip erase or a SPC level low to no protection demotion is performed.

When SPC is set to level high, WP[15:0] / WP[59:0] cannot be modified, else WP[15:0] / WP[59:0] can be modified without any restrictions.

Note: DCRP or secure user areas are erase / program protected.

7.3.12. Security protection

The NVMC provides a security protection function to prevent illegal code / data access on the ECNVM memory and CNVM. This function is useful for protecting the software or firmware from illegal users. Security protection is global. Not only ECNVM memory but also other secured regions are protected. Other secured regions include backup SRAM (BKPSRAM), RTC backup registers. There are 3 levels for protecting.

No protection

When both SPC_H and SPC_L bits in OTP are 0, and SPC[7:0] bits in NVMC_OBSTAT0_EFT is set to 0xAA, no protection performed. The ECNVM, CNVM and option bytes block are accessible by all operations. And access to other secured regions is also allowed.

Protection level low

When SPC_H in OTP is 0, as long as setting SPC_L to 1 or SPC[7:0] option bits to any value except 0xAA or 0xCC, protection level low performed. The ECNVM and CNVM can only be accessed by user code. In debug mode, boot from SRAM or boot from boot loader mode, all operations to ECNVM and CNVM or other secured regions (e.g. backup SRAM) is forbidden. If a read operation is executed to ECNVM / CNVM in debug mode, boot from SRAM or boot from boot loader mode, a read protection error RPERR with bus error will be generated. If a program / erase operation is executed to ECNVM / CNVM in debug mode, boot from SRAM or boot from boot loader mode, the WPERR bit in NVMC_ECNVM_STAT (NVMC_CNVM_STAT) register will be set. At protection level low, option bytes block is accessible by all operations. If program back to no protection level by setting SPC byte to

0xAA, an erase events ECNVM and CNVM will be performed.

The SPC level low to no protection demotion will cause erase events below:

- For the ECNVM and CNVM block sectors not belonging to DCRP / secure user area / WP, these sectors will be mass erased.
- Erase / program protection sectors will be erased.
- For DCRP area, if both XX_DCRP_EREN bits in NVMC_DCRPADDR_EFT and NVMC_DCRPADDR_MDF are 0, the sectors belonging DCRP areas will not be erased. Otherwise DCRP areas will be erased (overlapped or not with secure-only area).
- For secure user area, if both XX_SCR_EREN bits in NVMC_SCRADDR_EFT and NVMC_SCRADDR_MDF are 0, the sectors belonging secure-only areas will not be erased. Otherwise secure-only areas will be erased (overlapped or not with DCRP area).
- If at least one CNVM_DCRP_EREN in NVMC_DCRPADDR_EFT or NVMC_DCRPADDR_MDF is 1, and at least one CNVM_SCR_EREN in NVMC_SCRADDR_EFT or NVMC_SCRADDR_MDF is 1, the whole CNVM block will be erased.
- If at least one ECNVM_BANK0_DCRP_EREN in NVMC_DCRPADDR_EFT or NVMC_DCRPADDR_MDF is 1, and at least one ECNVM_BANK0_SCR_EREN in NVMC_SCRADDR_EFT or NVMC_SCRADDR_MDF is 1, the whole bank0 block will be erased.
- If at least one ECNVM_BANK1_DCRP_EREN in NVMC_DCRPADDR_EFT or NVMC_DCRPADDR_MDF is 1, and at least one ECNVM_BANK1_SCR_EREN in NVMC_SCRADDR_EFT or NVMC_SCRADDR_MDF is 1, the whole bank1 block will be erased.
- The other secured regions (such as backup SRAM) are erased.

Note: The sector protections (DCRP, secure user area) are removed only if the protected sector boundaries are modified by the user application. XX = CNVM, ECNVM_BANK0、ECNVM_BANK1.

Note: Modifying the SPC from no protection to low protection level requires a power reset while disconnecting the debugger and JTAG/SWD devices for the chip to start normally.

Protection level high

When SPC_H in the OTP is 1 or the SPC option byte is set to 0xCC, protection level high performed. When this level is programmed, debug mode, boot from SRAM or boot from boot loader mode are disabled. The ECNVM and CNVM block is accessible by all operations from user code. The user option bits can be read but can not be modified. And accesses to the other secured areas are also allowed. The SPC byte cannot be reprogrammed. So, if protection level high is programmed, it cannot move back to protection level low or no protection level.

The debug port is always disabled when level high is active. Therefore, it is impossible to debug and analyze the defective parts with level high security protection. If security protection level high is set while the debugger is still connected, apply a power on reset.

User can configure the SPC protection level refer to *Table 7-10. SPC protection level configuration*

Table 7-10. SPC protection level configuration

NVMC_OTP_USER_CTL0 register		NVMC_OBSTAT0_MDF register	NVMC_OBSTAT0_EFT register	SPC level
SPC_H	SPC_L	SPC[7:0]	SPC[7:0]	
1	0 or 1	Any value	= 0xCC	Level high
0 or 1	0 or 1	0xCC	= 0xCC	Level high
0	1	Any value except 0xCC	= 0xFF	Level low
0	0	Any value except 0xCC or 0xAA	= SPC[7:0] value in NVMC_OBSTAT0_MDF register	Level low
0	0	0xAA	= 0xAA	No protection

Note: (1) If SPC level setting in OTP macro is different from SPC level setting in option byte, the SPC level depend on the higher level of the two settings. And SPC[7:0] bits in NVMC_OBSTAT0_EFT register reflect the effective SPC level. (2) If SPC_L bit in OTP macro is set, SPC level low to no protection demotion is forbidden. Otherwise OBMERR bit will be set in NVMC_OBCTL register.

7.3.13. Dedicated code read protection area

GD32H77xxI/GD32H78xxI has no dedicated code read protection area of ECNVM.

In the ECNVM and CNVM block, NVMC can define executable-only areas, allowing only instruction transactions from the system, but not data access.

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

One DCRP area can be defined by setting the DCRP_AREA_END[8:0] and DCRP_AREA_START[8:0] option bytes with a granularity of 4 Kbytes. This means that the actual DCRP area size is defined by (XX represents CNVM, ECNVM_BANK0, ECNVM_BANK1, the same below):

- [(DCRP_AREA_END[8:0] - DCRP_AREA_START[8:0]) + 1] x 4Kbytes

As an example, to set a DCRP area on sector 0, the option bytes should be set as follows:

- DCRP_AREA_START[8:0] = 0x000,
- DCRP_AREA_END[8:0] = 0x00F,

So the DCRP size is:

- [(DCRP_AREA_END[8:0] - DCRP_AREA_START[8:0]) + 1] x 4Kbytes = 64 Kbytes.

Note: Only when CNVM_DCRP_EN is set, the DCRP area configured in the CNVM region will take effect. Only when ECNVM_BANK0_DCRP_EN is set, the DCRP area configured in

the bank0 region will take effect. Only when ECNVM_BANK1_DCRP_EN is set, the DCRP area configured in the bank1 region will take effect.

The minimum DCRP area is 8Kbytes (the start address of the DCRP area should be less than the end address). The maximum DCRP area is the entire CNVM / bank0 / bank1 block, configured by setting the same value to the start and end addresses of the DCRP area.

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 in ECNVM block will return 0 and a RPERR flag in NVMC_ECNVM_STAT register is set, and write operations will be ignored and a WPERR in NVMC_ECNVM_STAT register is set.

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

Only CPU can modify the DCRP area definition bits and XX_DCRP_EREN bit. If DCRP area is valid (not empty), during SPC level low to no protection demotion, if in NVMC_DCRPADDR_EFT and NVMC_DCRPADDR_MDF, the XX_DCRP_EREN bits are both set to 0, the contents of the DCRP area will not be erased, otherwise the contents will be erased.

Besides the option bytes, the DCRP area can also be configured by modifying the system parameter in OTP macro with granularity of 4KB bytes.

If the DCRP_EFFCTL bit in NVMC_OTP_USER_CTL0 parameter in OTP macro is 1 and DCRP_AREA_END[8:0] and DCRP_AREA_START[8:0] in OTP macro is not all 0, DCRP area is defined by the DCRP_AREA_END[8:0] and DCRP_AREA_START[8:0] bits in the NVMC_OTP_USER_CTL3 parameter. Otherwise, DCRP area is defined by the XX_DCRP_AREA_END[8:0] and XX_DCRP_AREA_START[8:0] bits in option bytes. For more details, please refer to ***Table 7-11. DCRP area configuration***

Table 7-11. DCRP area configuration

DCRP area configure in option byte	DCRP area configure in OTP	DCRP_EFFCTL configure in OTP	Effective DCRP area
Area1	Start and end addresses are both 0.	0	Area1(2)
	Start and end addresses are both 0.	1	Area1(2)
	Area2(1)	0	Area1(2)
	Area2(1)	1	Area2

Note: (1) The start and end addresses of Area2 are not both 0. (2) If the DCRP area is defined by OTP, the value of DCRP_AREA_START[8:0] in the NVMC_DCRPADDR_MDF and NVMC_DCRPADDR_EFT registers is { DCRP_AREA_START[8:0] in OTP}, and the value of DCRP_AREA_END[8:0] is { DCRP_AREA_END[8:0] in OTP}.

If the DCRP area is decided by OTP. The DCRP_AREA_END[8:0] bits or DCRP_AREA_START[8:0] bits in NVMC_DCRPADDR_MDF register can not be modified. So the DCRP area is fixed. The protection-removed erase will not work.

Note: (1) The DCRP configuration priority of OTP is higher than the ECNVM option byte, so in the product, if configure DCRP area by the option byte, the start and end address of the DCRP area in OTP should be both set to 0, and the DCRP_AREALK bit in OTP should be set to 1, the DCRP_EFFCTL bit in OTP should be set to 0, otherwise there may be vulnerability in the DCRP area. (2) If the user has security considerations, please configure the DCRP area by OTP, otherwise there may be security risks.

7.3.14. Secure user area

GD32H77xxI/GD32H78xxI has no secure user area of ECNVM.

In the ECNVM and CNVM block, NVMC can define secure user areas which can only be accessed when the CPU executes a secure application.

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.

One secure user area can be defined by setting the SCR_AREA_END and SCR_AREA_START option bytes with a granularity of 4 Kbytes. This means that the actual secure user area size is defined by (XX represents CNVM, ECNVM_BANK0, ECNVM_BANK1, the same below):

- [(SCR_AREA_END[8:0] - SCR_AREA_START[8:0]) + 1] x 4Kbytes

As an example, to set a secure user area on sector 0, the option bytes should be set as follows:

- SCR_AREA_START[8:0] = 0x000,
- SCR_AREA_END[8:0] = 0x00F,

So the secure user size is:

[(SCR_AREA_END[8:0] - SCR_AREA_START[8:0]) + 1] x 4Kbytes = 64 Kbytes.

Note: Only when CNVM_SCR_EN is set, the SCR area configured in the CNVM region will take effect. Only when ECNVM_BANK0_SCR_EN is set, the SCR area configured in the bank0 region will take effect. Only when ECNVM_BANK1_SCR_EN is set, the SCR area configured in the bank1 region will take effect.

SCR_AREA_END[8:0] and SCR_AREA_START[8:0] option bytes can be modified by CPUs running secure libraries or application secure code. But for user non-secure applications, these option bytes can be modified only during SPC level low to no protection demotion or a protection-removed mass erase.

The minimum secure user area is 8Kbytes (the start address of the secure user area should be less than the end address). The maximum secure user area is the entire CNVM / bank0 /

bank1 block, configured by setting the same value to the start and end addresses of the secure user area.

When executing code in this area, the debug events will be ignored. Only CPU can access it by executing GD secure library or user secure application. In all other cases, accessing the secure user area is illegal. For example, read operations in ECNVM block will return 0 and a RSERR flag in NVMC_ECNVM_STAT register is set, and write operations will be ignored and a WPERR in NVMC_ECNVM_STAT register is set.

A valid secure user area is erase-protected. Cannot erase sectors located in this area. Unless the application is executed from a valid secure user area, it is impossible to erase the sector located in the area. If a valid secure user area is defined, mass erase cannot be performed unless erase is performed during the SPC level low to no protection demotion or protection-removed mass erase.

Only CPU can modify the secure user area definition bits and XX_SCR_EREN bit. If secure user area is valid (not empty), during the SPC level low to no protection demotion, if the XX_SCR_EREN bit in NVMC_SCRADDR_EFT register and the XX_SCR_EREN bit in NVMC_SCRADDR_MDF register are both set to 0, the contents of the secure user area will not be erased, otherwise the contents will be erased.

Besides the option bytes, the secure user area can also be configured by modifying the User control parameter in OTP macro with granularity of 4KB bytes.

If the SCR_EFFCTL bit in user control parameter in OTP macro is 1 and SCR_AREA_END[8:0] and SCR_AREA_START[8:0] in OTP macro is not all 0, secure user area is defined by the SCR_AREA_END[8:0] and SCR_AREA_START[8:0] bits in the NVMC_OTP_USER_CTL2 parameter. Otherwise, secure user area is defined by the SCR_AREA_END[8:0] and SCR_AREA_START[8:0] bits in option bytes. For more details, please refer to *Table 7-12. Secure user area configuration*

Table 7-12. Secure user area configuration

Secure user area configure in option byte	Secure user area configure in OTP	SCR_EFFCTL configure in OTP	Effective Secure user area
Area1	Start and end addresses are both 0.	0	Area1(2)
	Start and end addresses are both 0.	1	Area1(2)
	Area2(1)	0	Area1(2)
	Area2(1)	1	Area2

Note: (1) The start and end addresses of Area2 are not both 0. (2) If the secure user area is defined by OTP, the value of SCR_AREA_START[8:0] in the NVMC_SCRADDR_MDF and NVMC_SCRADDR_EFT registers is {SCR_AREA_START[8:0] in OTP}, and the value of SCR_AREA_END[8:0] is {SCR_AREA_END[8:0] in OTP}.

If the secure user area is decided by OTP. The SCR_AREA_END[8:0] bits or SCR_AREA_START[8:0] bits in NVMC_SCRADDR_MDF register can not be modified. So

the secure user area is fixed. The protection-removed erase will not work.

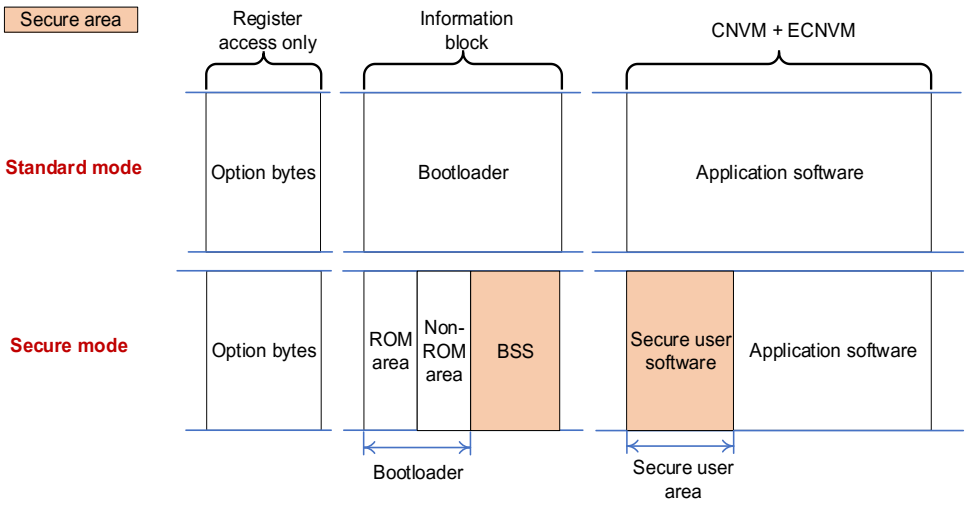
Note: (1) The secure configuration priority of OTP is higher than the ECNVM option byte, so in the product, if configure the secure user area by the option byte, the start and end address of the secure user area in OTP should be both set to 0, SCR_AREALK bit in OTP should be set to 1, SCR_EFFCTL bit in OTP should be set to 0, otherwise there may be vulnerability in the secure user area. (2) If the user has security considerations, please configure the secure user area by OTP, otherwise there may be security risks.

7.3.15. Secure mode

Security should be performed for some sensitive programs to avoid potentially malware attacks. For example, licensed firmware update software requires highly protection because it processes confidential data (such as encryption keys) that cannot get by other processes.

Secure areas with limited access is provided. In this area, secure services can be built that can be executed before any user application. These secure areas and their included software can be accessed only in secure mode. ***Figure 7-14. Memory architecture in standard mode and secure mode*** shows the details of the area.

Figure 7-14. Memory architecture in standard mode and secure mode



Secure user area is accessed once after reset, the area code is hidden after execution. Basic security service is Gigadevice software to configure secure services. Secure user software is located in secure user area and executed once after reset. Secure user software can be used to implement secure boot and licensed firmware update.

Secure mode and secure user area can be configured by option bytes or OTP. User can set the secure user area in the option byte by basic security service to make the secure code and data be configured in the secure user area.

In secure mode, secure firmware store in information block to support boot. Secure user software is secure application code, data, or algorithms stored in ECNVM block.

If secure user area is not set, MCU will jump to the requested boot address which set by the BOOT_ADDR0[15:0] option bits in NVMC_BTADDR_EFT register.

In secure mode, regardless of the startup configuration (BOOT pins and boot addresses), MCU will be forced to boot from secure ROM area.

Secure mode is enabled as long as one of the SCR bits in the Option byte or OTP is set to 1. At this time, the SCR bit of the NVMC_OBSTAT0_EFT register is 1. And after the SCR bit is enabled, a system reset is required to activate secure mode.

Standard mode can be returned when the SCR bit in option byte is 1 and the SCR bit in OTP is 0. To return to standard mode, the secure user area, CNVM_SCR_EN, ECNVM_BANK0_SCR_EN, ECNVM_BANK1_SCR_EN and DCRP area, CNVM_DCRP_EN, ECNVM_BANK0_DCRP_EN, ECNVM_BANK1_DCRP_EN need to be removed before or at the same time as the SCR option bit is cleared. For more details, please refer to the modify rules of the relevant option bytes.

Note: It is strongly recommended that users configure the secure user area / DCRP area of the CNVM and ECNVM Data Integrity Enhancement (DIE) banks consistently when using ECNVM as a Data Integrity Enhancement area (ECNVM_DIE_DISABLE set to 0). If the CNVM Data Integrity Enhancement area is not changed (SWAP_DISABLE in the NVMC_OTP_USER_CTL1 register is set to 1), consistency must only be ensured between the CNVM and DIE banks in terms of secure area / DCRP configuration. If the CNVM Data Integrity Enhancement area needs to be modified, consistency must be ensured across the secure area / DCRP configurations for CNVM and both DIE banks. If users do not use ECNVM as a Data Integrity Enhancement area (ECNVM_DIE_DISABLE set to 1), only the CNVM secure user area / DCRP area needs to be configured without configuring the ECNVM secure user area / DCRP area.

Note: For security reasons, before jumping from the secure user area to the normal user area, it is necessary to clear or disable the cache.

Note: In security mode, when erasing the secure user area and DCRP area via SPC level low to no protection demotion or protection-removed chip erase, it is required to enable the CNVM_SCR_EREN, ECNVM_BANK0_SCR_EREN, ECNVM_BANK1_SCR_EREN bit in NVMC_SCRADDR_EFT register and CNVM_DCRP_EREN, ECNVM_BANK0_DCRP_EREN, ECNVM_BANK1_DCRP_EREN bit in NVMC_DCRPADDR_EFT register, set secure user area end address less than secure user area start address, set DCRP end address less than DCRP start address, set the SCR bit in NVMC_OBSTAT0_EFT register to 0.

7.3.16. OTP

GD32H77xxI/GD32H78xxI has no ECNVM, repair and OTA function in OTP.

OTP macro description

The OTP macro stores 2 system parameters: User configuration parameter and user data parameter. Each parameter can be read and written, where base user configuration parameter can be read and written directly by the bus and read by registers. User data can only be read and written directly by the bus.

Only 16 bits of information are read or written each time. Each 16bit of information of user data consists of 8 bits of lock and 8 bits of configuration information. For example, the configuration information of bit0 corresponds to the lock of bit8. In the user configuration, the configuration information parameter composed of multiple information bits has a corresponding unified lock bit, and the configuration information of a single bit uses an independent lock bit. For details, see the register description. The value read by the bus is the latest value stored in macro, and the register value is updated only after system reset. Among them, safety-related registers (SPC_H, SPC_L, NDBG, JTAGNSW, SCR / DCRP-related software register values and all lock bits) are reset only during a power reset, not during a system reset. After a system and power reset, the data is loaded into registers. The values of other registers are reset during a system reset. Parameters and lock take effect after the reset. After some parameters are written to 1, the corresponding lock value is automatically set to 1 (Note that the value in macro does not change, but the value in the register is automatically changed to 1). Each information data in the user configuration and user data region is stored redundantly. An error flag will be raised when redundant data stored in macro is faulty, or when an invalid address is read or written, or when a parameter already locked / lock bit is modified.

The following *Table 7-13. System parameters* shows the details of each OTP byte.

Table 7-13. System parameters

Parameter		Width/bytes	Logic address	Program-protected	Read-protected	Description
user configuration	base user configuration	2B	0x1FFF_0000-0x1FFF_0001	The higher 8 bits have corresponding lock bits. The parameter value and lock bit cannot be modified after lock, otherwise PGLKERRF is set.	After power or system is reset, the AHB software register configuration information part load value remains unchanged, and the software readable. During the system reset process, the lock bit remains unchanged, while other parameters	Lower 16 bits in *OTP user control register 0 (NVMC_OTP_USER_CTL0)*

Parameter		Width/bytes	Logic address	Program-protected	Read-protected	Description
					are reset during the system reset. AXI bus can read and write.	
		2B	0x1FFF_0020-0x1FFF_0021		The values of SPC_H, SPC_L, NDBG, JTAGNSW, SCR / DCRP related software register, and all the lock bit software registers are reset only during a power reset. After a system and power reset, the data is loaded. The values of other registers are reset during a system reset, and data is loaded after the system reset. Moreover, NDBG and JTAGNSW take effect after the power reset.	Higher 16 bits in ***OTP user control register 0 (NVMC_OTP_USER_CTL0)***
		2B	0x1FFF_0040-0x1FFF_0041			Lower 16 bits in ***OTP User Control Register 1 (NVMC_OTP_USER_CTL1)***
		2B	0x1FFF_0060-0x1FFF_0061	After the lock is locked, the parameter value and lock bit cannot be modified, otherwise PGLKERRF is set.		Lower 16 bits associated with SCR in ***OTP User Control Register 2 (NVMC_OTP_USER_CTL2)***
		2B	0x1FFF_0080-0x1FFF_0081			Higher 16 bits associated with SCR in ***OTP User Control Register 2 (NVMC_OTP_USER_CTL2)***
		2B	0x1FFF_00A0-0x1FFF_00A1			Lower 16 bits associated with DCRP in ***OTP User Control Register 3 (NVMC_OTP_USER_CTL3)***
		2B	0x1FFF_00C0-0x1FFF_00C1			Higher 16 bits associated with DCRP in ***OTP User Control Register 3 (NVMC_OTP_USER_CTL3)***
		2B	0x1FFF_00E0-0x1FFF_00E1		After system or power is reset, the AHB software register load value remains unchanged, and the software readable. During the system reset process, the values of the AHB software	Lower 16 bits in ***OTP User Control Register 4 (NVMC_OTP_USER_CTL4)***

Parameter		Width/bytes	Logic address	Program-protected	Read-protected	Description
					registers remain unchanged. AXI bus can read and write.	
		2B	0x1FFF_0100~0x1FFF_0101	The higher 8 bits have corresponding lock bits. The parameter value and lock bit cannot be modified after lock, otherwise PGLKERRF is set.	After power or system is reset, the AHB software register configuration information part load value remains unchanged, and the software readable. During the system reset process, the lock bit remains unchanged, while other parameters are reset during the system reset process. AXI bus can read and write.	Reserve space for other operations. For more details, refer to lower 16 bits ***OTP User Control Register 5 (NVMC_OTP_USER_CTL5)***
		2B	0x1FFF_0120~0x1FFF_0121			Reserve space for other operations. For more details, refer to higher 16 bits ***OTP User Control Register 5 (NVMC_OTP_USER_CTL5)***
	DBGPSWD_DATA	8B	0x1FFF_0140~0x1FFF_0141	The parameter value and lock bit cannot be modified after lock, otherwise PGLKERRF is set.	Parameters can only be written through hardware registers (AXI access), cannot be read through hardware registers, software cannot read and write registers (AHB inaccessible)	This parameter is used as the debug password for debug services.
			0x1FFF_0160~0x1FFF_0161			
			0x1FFF_0180~0x1FFF_0181			
			0x1FFF_01A0~0x1FFF_01A1			
	OTP_AES_KEY	16B	0x1FFF_01C0~0x1FFF			The AES key used to encrypt the firmware image.

Parameter		Width/bytes	Logic address	Program-protected	Read-protected	Description
			_01C1(x = 14)			
			…			
			0x1FFF_02A0~0x1FFF_02A1(x = 21)			
User data	User data	372B	0x1FFF_02C0~0x1FFF_02C1(x = 22)	The higher 8 bits have corresponding lock bits. The parameter value and lock bit cannot be modified after lock, otherwise PGLKERRF is set.	Users can read and write only through the AXI bus	User defined data. User data have 32B Public key HASH (The corresponding storage is 0x1FFF_02C0-0x1FFF_02C1, …,0x1FFF_04A0-0x1FFF_04A1)
			0x1FFF_02E0~0x1FFF_02E1(x = 23)			
			…			
			0x1FFF_0E60~0x1FFF_0E61(x = 115)			
			…			
			0x1FFF_19C0~0x1FFF_19C1(x = 206)			
			0x1FFF_19E0~0x1FFF_19E1(x = 207)			

Note: The logical address is the lower 2 bits of the 32bit address. That is, for each 256bit of data, the parameter is stored in the lower 16 bits of data, and 0 is added to the higher 240 bits of data. The parameter is stored in 0x1FFF_0000 + 0x20 * x - 0x1FFF_0001 + 0x20 * x

(x = 0 … 207). The 0 is stored in 0x1FFF_0002 + 0x20 * x - 0x1FFF_001F + 0x20 * x (x = 0 … 207).

Program operation: Supports only a single 16bit write, and can only write address 0x1FFF_0000 + 0x20 * x (x=0 ... 207). Otherwise, PGSERR is set in NVMC_OTP_STAT.

Read operation: Supports only a single 16bit read, and can only read address 0x1FFF_0000 + 0x20 * x (x =0 ... 207). Non-aligned read hardware does not do additional restrictions, but it may cause the readout data of the parameter incorrect. Users should avoid this situation.

The user DBGPSWD_DATA word and OTP_AES_KEY parameters in the USER CFG area cannot be read by AXI, and there is no AHB register. However, the correct values written can be verified through a CRC check.

Note: NVMC_OTP_USER_CTL0 ~ NVMC_OTP_USER_CTL5 is a read-only register of AHB software. Users can read and write the corresponding OTP macro of the read-only register through the AXI bus. After a system or power reset, the AHB software read-only register loads the corresponding parameters from the OTP macro.

OTP read operation

If the read address is the area accessible to the OTP, only 16 bits of data can be read each time, including the configuration information and the lock bit of the configuration information. During the reading process, the controller calibrates the redundant information of the 16-bit data. If the information stored in the OTP area is inconsistent, a DATAERR flag is generated.

OTP write and repair operation

The following steps show the register access sequence of the OTP writing operation.

1. Unlock the NVMC_CNVM_CTL register if necessary.
2. Check the BUSY bit in NVMC_OTP_STAT register to confirm that no OTP memory operation is in progress (BUSY equals to 0). Otherwise, wait until the operation has finished.
3. Set the PG bit in NVMC_CNVM_CTL register.
4. Write the data to be programed with desired absolute address (0x1FFF XXXX).
5. Wait until all the operations have been finished by checking the value of the BUSY bit in NVMC_OTP_STAT register.

Note: User must unlock NVMC_CNVM_CTL register and set PG bit before program the OTP. Otherwise, PGSERR is set in NVMC_OTP_STAT.

If the read lock bit is set to 1, the software should not modify the corresponding register (That is, if the original configuration information is 0 and the lock bit corresponding to the information is 1, the configuration information cannot be written to 1). Otherwise, the PGLKERRF will be raised. Similarly, the lock bit can only write 1. After being locked, 0 cannot be written.

When the bus initiates a write operation, the repair operation is carried out at the same time,

and the inconsistent redundant bits are uniformly modified to be consistent. The modification principle is to modify all redundant bits to be consistent according to mostly consistent level values. When exactly half of the data in the redundancy is inconsistent, it is first repaired to 1 during the write operation. (Both information bits and lock bits with inconsistent redundancy can be repaired) In addition, the OTP has the repair function. When the software configuration repair start function is enabled, the entire OTP area will be repaired. Repair operations can only fix less than half of the errors correctly. If equal to half of the redundant bits are faulty, the repair is not guaranteed. If more than half of the redundant bits are faulty, the repair will not succeed.

Note: In the 0x1FFF 0000 - 0x1FFF 6FFF area, all addresses except 0x1FFF 0000-0x1FFF 19FF are illegal addresses and cannot be read or written by AXI. For parameters with read-write restrictions, corresponding read or write operations by AXI are not allowed. Otherwise, it would be considered an illegal read-write operation.

When an illegal read-write operation is performed through the AXI bus, the AXI bus error is reported.

AES key CRC function

The standard CRC-8-CCITT algorithm is used in CRC calculation. The CRC algorithm in this module is used to verify the values in the OTP_AES_KEY AXI registers.

After continuously write 16 bytes AES key to the OTP_AES_KEY AXI registers whose logic address are 0x1FFF_01C0~0x1FFF_01C1…0x1FFF_02A0~0x1FFF_02A1. When AES_KEY_ON is set 0, after the system resets, the OTP_AES_KEY stored in the OTP area will be loaded. The hardware CRC module will automatically calculate the corresponding CRC checking code based on the AES key value in the OTP_AES_KEY AXI registers and store the calculation result to the AES_KEY_CRC bit field in the NVMC_OTP_CTL register. At this time, user can compare the CRC checking code calculated by hardware with the software CRC checking code calculated by the user software. If the checking codes calculated by the software and hardware are the same, it indicates the 16 bytes AES key written to the OTP_AES_KEY AXI register is correct, otherwise the AES key written to the register is wrong.

DP CRC function

The CRC-8 algorithm is used in CRC calculation. The CRC algorithm in this module is used to verify the values in the DBGPSWD_DATA AXI registers.

After continuously write 8 bytes debug password data to the DBGPSWD_DATA AXI registers whose logic address are 0x1FFF_0140~0x1FFF_0141…0x1FFF_01A0~0x1FFF_01A1. When DPLK is set 0, after the system resets, the DBGPSWD_DATA stored in the OTP area will be loaded. The hardware CRC module will automatically calculate the corresponding CRC checking code based on the debug password data in the DBGPSWD_DATA AXI registers and store the calculation result to the DP_CRC bit field in the NVMC_OTP_CTL register. At this time, user can compare the CRC checking code calculated by hardware with the software CRC checking code calculated by the user software. If the checking codes calculated by the

software and hardware are the same, it indicates the 8 bytes debug password data written to the DBGPSWD_DATA AXI register is correct, otherwise the debug password data written to the register is wrong.

Note: When using the DP CRC source code to calculate the CRC checking code of DBGPSWD_DATA, the input parameter 'data' must be written in the reverse order of the write sequence of DBGPSWD_DATA.

The source code for CRC-8 is as follows:

Table 7-14. DP CRC source code

```
uint8_t get_dp_crc(uint8_t *data, uint32_t length)
{
    uint8_t i;
    unsigned char crc = 0x5A;
    unsigned char polynomial = 0x07;
    for (i = 0; i < length; ++i) {
        crc ^= data[i];
        for (unsigned char bit = 0; bit < 8; ++bit) {
            if (crc & 0x80) {
                crc = (crc << 1) ^ polynomial;
            } else {
                crc <<= 1;
            }
        }
    }
    return crc ^ 0x3C;
}
```

OTP interrupts

The following operations will set the error flag in the NVMC_OTP_STAT register:

- During read and write operations, the redundancy bit of the corresponding address is inconsistent, and the DATAERRF is set.
- During read and write operations, the multiple redundancy bits of the corresponding address are inconsistent, and the MULTI_DATAERRF is set.
- If you try to modify a bit that is already locked, or if you try to change a lock that is already 1 to 0, the PGLKERRF is set and the write operation is not performed. In addition, the lower 13 bits of the logical address corresponding to the lock error will be stored in the PGLKERR_ADDR register.
- When AES_KEY_ON and DPLK is set 1, OTP_AES_KEY and DBGPSWD_DATA can not be written, if write, the PGLKERRF is set and the write operation is not performed.
- If the repair or write operation is complete in the OTP area, the ENDF bit in NVMC_OTP_STAT register is set.

- If an incorrect write operation occurs in the OTP area, the PGSERR bit in NVMC_OTP_STAT register is set.

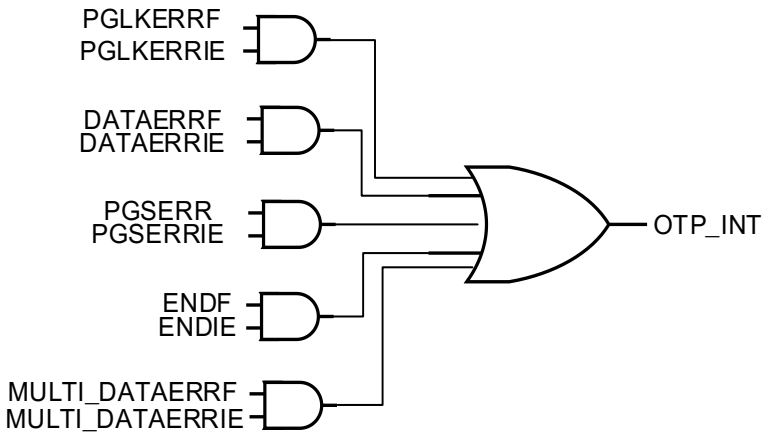
The OTP interrupt events and flags are listed in *Table 7-15. OTP interrupt requests*

Table 7-15. OTP interrupt requests

Interrupt event	Event flag	Enable Control bit
Data error interrupt	DATAERRF	DATAERRIE
Write to locked parameter error interrupt	PGLKERRF	PGLKERRIE
Multi data error interrupt	MULTI_DATAERRF	MULTI_DATAERRIE
OTP operation end interrupt	ENDF	ENDIE
OTP program sequence error interrupt	PGSERR	PGSERRIE

All of the interrupt events are Ored together before being sent to the interrupt controller, so the OTP 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 7-15. OTP interrupt mapping diagram



Note: The register bits in the figure above are all OTP register bits.

7.4. Register definition

NVMC base address: 0x5200 2000

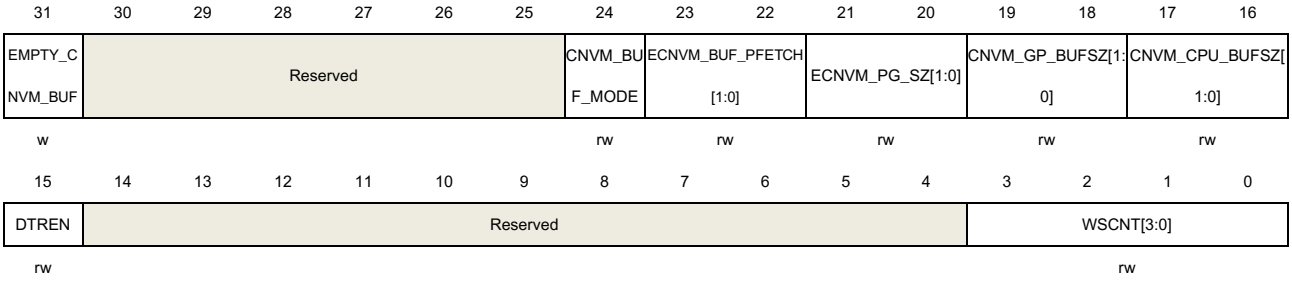
7.4.1. NVMC registers

NVM Performance Control Register (NVMC_PERFCTL)

Address offset: 0x000

Reset value: 0x0000 0001

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



Bits	Fields	Descriptions
31	EMPTY_CNVM_BUF	The function to clear the CNVM buffers (general buffer and CPU buffer) takes effect when the buffers are idle. When this bit is written to 1, the function to clear the CNVM buffers is enabled. After the CNVM buffers are cleared, the hardware automatically clears this bit. **Note:** When this bit is set to 1, the CNVM buffers are not cleared immediately but take effect after CNVM prefetch is complete and the buffers are idle. Therefore, after setting this bit to 1, a delay is required before performing CNVM read operations to avoid reading stale data from the buffers before they are cleared (this can be verified by checking the ECBBUSY bit in the NVMC_CS register to ensure the CNVM buffers have been cleared).
30:25	Reserved	The reset value must be maintained.
24	CNVM_BUF_MODE	CNVM buffer mode 0: CNVM CPU buffer cache stores data corresponding to read commands issued by the CPU. Data corresponding to AXI read commands issued by other masters is cached in the CNVM general buffer. 1: CNVM CPU buffer only caches instruction-type CPU read data. All other types of data are stored in the CNVM general buffer.
23:22	ECNVM_BUF_PFETCH[1:0]	ECNVM buffer prefetch control 00: No prefetch (prefetch 0 line + current buffer 1 line) 01: 64*4*1 (prefetch 1 line + current buffer 1 line) 10: 64*4*3 (prefetch 3 line + current buffer 1 line)

11: No prefetch (prefetch 0 line + current buffer 1 line)

The buffer depth is 64*4*4 (four buffer rows, each containing 256-bit data). Different prefetch strategies are applied according to the configuration. After the current ECNVM data in the buffer is read, if there is no ECNVM read request, prefetch is performed according to the settings (sequentially placing data from ECNVM without read requests into the buffer). For three consecutive prefetches, a check is performed between each prefetch to promptly meet the current bus read request. If the request is not met, the current prefetch may be terminated.

Note: A larger buffer depth does not necessarily mean higher efficiency - efficiency depends on program behavior.

21:20 ECNVM_PG_SZ[1:0] ECNVM programming size

00: 8 bytes

01: 32 bytes

Others: Reserved.

1) The above configuration specifies the programming size for a single ECNVM operation.
2) The size of the AXI bus must be 64 bits.
3) Each programming operation increments by 8 bytes based on the first data address.
4) 00: 8-byte programming, each programming operation writes 64 bits.
5) 01: 32-byte programming, capable of programming 64*4 bits in a single operation or achieving 64*4 bits through multiple operations.
6) The software must ensure that the size of data written in each programming operation matches the configuration and that the write sequence is incremental.
7) When PGCHEN is set, for 8-byte and 32-byte programming, the hardware checks if the programming area is fully 0xFF before programming. If not, PGSERR will be set.

19:18 CNVM_GP_BUFSZ[1:0] CNVM generic buffer size (number of buffer lines: for masters other than the CPU)

00: buf_size = 2^0 = 1

01: buf_size = 2^1 = 2

10: buf_size = 2^2 = 4

11: buf_size = 23 = 8

buf_size is measured in 256-bit units.

Note: A larger number of buffer lines does not necessarily mean higher efficiency - efficiency depends on program behavior.

17:16 CNVM_CPU_BUFSZ[1:0] CNVM CPU buffer size (number of buffer lines)

00: buf_size = 2^0 = 1

01: buf_size = 2^1 = 2

10: buf_size = 2^2 = 4

11: buf_size = 23 = 8

buf_size unit is 256-bit.

Note: A larger number of buffer lines does not necessarily mean higher efficiency -

efficiency depends on program behavior.

15	DTREN	ECNVM DTR (Double Data Rate) mode enable. (Not applicable to GD32H77xxP/GD32H78xxP/GD32H77xxI/GD32H78xxI) When DTREN is set to 1, ECNVM DTR mode is enabled, allowing data to be read on both the rising and falling edges of the ECNVM clock to improve efficiency. Within the full VDD voltage range, the maximum supported asynchronous clock frequency for this mode is 66 MHz. When DTREN is set to 0, ECNVM DTR mode is disabled. When VDD is greater than 2.3V, the maximum supported asynchronous clock frequency is 100 MHz. When VDD is less than or equal to 2.3V, the maximum supported asynchronous clock frequency is 85 MHz. 0: Disable ECNVM DTR mode. 1: Enable ECNVM DTR mode. **Note:** When writing 1 to this bit, DTR mode does not take effect immediately (the bit does not immediately change to 1). DTR mode becomes effective only when this bit is read back as 1. Asynchronous clock frequency can only switch to the maximum value corresponding to DTR mode after DTR mode becomes effective. Regardless of whether this bit is written as 1 or 0, it only takes effect after the corresponding value is read back.
14:4	Reserved	The reset value must be maintained.
3:0	WSCNT[3:0]	Wait State Counter Register According to the frequency and voltage conditions of the embedded CNVM storage interface, the application software configures the wait states for CNVM read operations. Ensure that CK_AHB / (WSCNT + 1) <= 50MHz. For example, if the frequency of CK_AHB is 300MHz, WSCNT should be set to 5. 0000：Reserved 0001: Add 1 wait state 0010: Add 2 wait states ... 1111: Add 15 wait states **Note:** The hardware does not check whether the configuration is correct.

Option Byte Operation Unlock Register (NVMC_OBKEY)

Address offset: 0x008

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

w

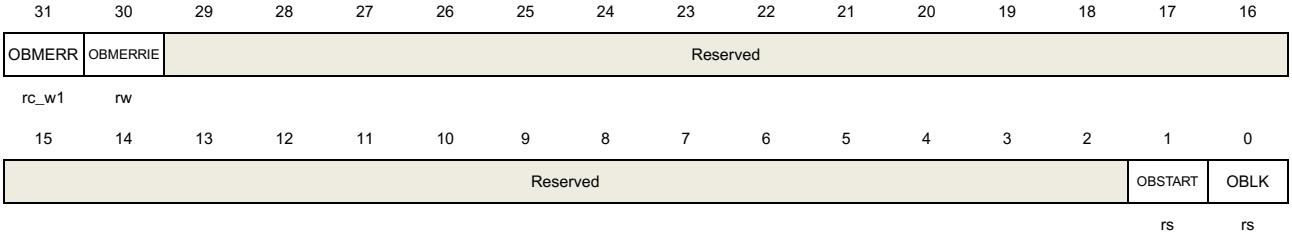
Bits	Fields	Descriptions
31:0	OBKEY[31:0]	These bits can only be written by software. Write the unlock value to OBKEY[31:0] to unlock the NVMC_OBCTL register.

Option Byte Control Register (NVMC_OBCTL)

Address offset: 0x018

Reset value: 0x0000 0001

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



Bits	Fields	Descriptions
31	OBMERR	Option byte modification error flag bit. This bit is set by hardware and cleared by software by writing 1. 0: No option byte modification error occurred. 1: An option byte modification error occurred.
30	OBMERRIE	Option byte modification error interrupt enable bit. This bit can be set to 1 and cleared to 0 by software only when OBLK is set to 0. 0: Disable option byte modification error interrupt. 1: Enable option byte modification error interrupt.
29:2	Reserved	The reset value must be maintained.
1	OBSTART	Send the option byte command to NVMC. This bit can only be set to 1 by software when OBLK is set to 0. It is cleared by hardware when the BUSY bit is cleared to 0.
0	OBLK	NVMC_OBCTL lock bit. When the value is written to the NVMC_OBKEY register in the correct sequence, this bit is cleared to 0 by hardware. Set to 1 by software.

Option byte status register 0 (NVMC_OBSTAT0_EFT)

Offset address: 0x01C

Reset value: 0xXXXX XXXX, factory value: 0x5FC6 AAD0

This register holds the effective value of the corresponding option bits. After reset, it loads the value from the option bytes.

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
VFIMG	NOPIPE_MODE	IOSPDOPEN	CNVM_ECC_OVRD	R0_SRAM_ECCEN	R1_SRAM_ECCEN	R2_SRAM_ECCEN	DTCM1ECCEN	DTCM0ECCEN	ITCMECCEN	SCR	SWAP	FW_UPDATE	FWDGSP D_STDBY	FWDGSP D_DPSLP	OTA_CHECK
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
SPC[7:0]								nRST_STDBY	nRST_DPSLP	VFALL	nWDG_HW	BOR_TH[1:0]		NOPRINT	DISLFI
r								r	r	r	r	r		r	r

Bits	Fields	Descriptions
31	VFIMG	Verify firmware image enable status bit This bit is factory set to 0. 0: Disable firmware image authentication function. 1: Enable firmware image authentication function.
30	NOPIPE_MODE	Pipeline mode status bit When the AXI bus reads CNVM, different modes can be selected based on the system clock frequency. 0: Timing-friendly mode, pipeline between the bus and CNVM enabled, read requests are sent to CNVM after being latched in one stage. 1: Performance-priority mode, pipeline between the bus and CNVM disabled.
29	IOSPDOPEN	Allow enable status bit for I/O speed optimization at low voltage 0: The chip operating voltage is greater than 2.7V, so I/O speed optimization is not allowed. 1: The chip operating voltage is below 2.7V, so I/O speed optimization is allowed.
28	CNVM_ECC_OVRD	CNVM ECC coverage enable status bit 0: CNVM ECC is controlled by CNVM_ECC_MODE. 1: CNVM ECC feature enabled When this bit is set to 0, the CNVM_ECC_MODE bit takes effect. Users can use this feature to test ECC functionality or other correction functions. When this bit is set to 1, the default ECC functionality is enabled. Enable the CNVM ECC function for normal use of CNVM.
27	R0_SRAM_ECCEN	ECC enable status bit for Region 0 SRAM 0: Disable ECC functionality of Region 0 SRAM. 1: Enable ECC functionality of Region 0 SRAM.
26	R1_SRAM_ECCEN	ECC enable status bit for Region 1 SRAM 0: Disable ECC functionality of Region 1 SRAM. 1: Enable ECC functionality of Region 1 SRAM.
25	R2_SRAM_ECCEN	ECC enable status bit for Region 2 SRAM 0: Disable ECC functionality of Region 2 SRAM.

		1: Enable ECC function of region 2 SRAM
24	DTCM1ECCEN	ECC enable status bit for DTCM1 0: Disable ECC function of DTCM1 1: Enable ECC function of DTCM1
23	DTCM0ECCEN	ECC enable status bit for DTCM0 0: Disable ECC function of DTCM0 1: Enable ECC function of DTCM0
22	ITCMECCEN	ECC enable status bit of ITCM 0: Disable ECC function of CPU ITCM 1: Enable ECC function of CPU ITCM
21	SCR	Safety mode enable status bit 0: Disable secure mode 1: Enable secure mode
20	SWAP	SWAP status bit (Not applicable to GD32H77xxP/GD32H78xxP and GD32H77xxI/GD32H78xxI) CNVM corresponds to the ECNVM enhanced data integrity area control to support OTA 0: CNVM content is consistent with the ECNVM Bank0 data integrity enhancement area. 1: CNVM content is consistent with the ECNVM Bank1 data integrity enhancement area.
19	FW_UPDATE	Firmware update status bit (Not applicable to GD32H77xxP/GD32H78xxP and GD32H77xxI/GD32H78xxI) When the firmware update flag is 1, the hardware updates the ECNVM data integrity enhancement area data to CNVM after reset based on the SWAP bit. After the update operation is completed, the CPU jumps to the software program of the GD checksum bootloader. 0: Disable hardware update of ECNVM data integrity enhancement area data to CNVM. 1: Enable hardware update of ECNVM data integrity enhancement area data to CNVM.
18	FWDGSPD_STDBY	Independent watchdog pause option status bit in Standby mode 0: Pause the independent watchdog in Standby mode. 1: Run the independent watchdog in Standby mode.
17	FWDGSPD_DPSLP	Independent watchdog pause option status bit in Deep Sleep mode 0: Pause the independent watchdog in Deep Sleep mode. 1: Run the independent watchdog in Deep Sleep mode.
16	OTA_CHECK	OTA verification status bit (not applicable to GD32H77xxI/GD32H78xxI) Indicates whether hardware verification is performed during the OTA upgrade.

		0: Disable hardware verification. 1: Enable hardware checking
15:8	SPC[7:0]	Security protection level status bit 0xAA: No protection state 0xCC: High security protection level Any value other than 0xAA or 0xCC: Low security protection level.
7	nRST_STDBY	Standby mode entry reset option status bit 0: Generate a reset when entering Standby mode 1: No reset generated when entering Standby mode
6	nRST_DPSLP	Reset option status bits in Deep Sleep mode 0: Generate a reset when entering deep sleep mode 1: No reset generated when entering deep sleep mode
5	VFALL	Firmware image (image must include certificate) verification status bit The factory default value of this bit is 0. 0: Disable firmware image verification 1: Enable firmware image verification
4	nWDG_HW	Watchdog control status bit 0: Hardware control watchdog 1: Software control watchdog
3:2	BOR_TH[1:0]	Brown-Out Reset (BOR) threshold status bit 00: No BOR function 01：BOR threshold value 1 10：BOR threshold value 2 11：BOR threshold value 3
1	NOPRINT	Status bit for disabling message printing during secure boot The factory default value of this bit is 0. 0: Do not print information 1: Print information
0	DISLFI	Firmware installation authorization function setting. The factory default value of this bit is 0. 0: Enable authorized firmware installation. 1: Disable authorized firmware installation.

Option byte status register 0 (NVMC_OBSTAT0_MDF)

Offset address: 0x020
Reset value: 0xXXXX XXXX

This register holds the modification value of the corresponding option bits. After a system reset, the value reflects the effective value of the corresponding option bits.

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
VFIMG	NOPIPE_MODE	IOSPDOPEN	CNVM_ECC_OVRD	R0_SRAM_ECCEN	R1_SRAM_ECCEN	R2_SRAM_ECCEN	DTCM1ECCEN	DTCM0ECCEN	ITCMECCEN	SCR	SWAP	FW_UPDATE	FWDGSP D_STDBY	FWDGSP D_DPSLP	OTA_CHECK
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
SPC[7:0]								nRST_STDBY	nRST_DPSLP	VFALL	nWDG_HW	BOR_TH[1:0]		NOPRINT	DISLFI
rw								rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31	VFIMG	Firmware image verification enable configuration bit. This bit is factory set to 0. 0: Disable firmware image authentication function. 1: Enable firmware image authentication function.
30	NOPIPE_MODE	Pipeline mode. When the AXI bus reads CNVM, different modes can be selected based on the system clock frequency. 0: Timing-friendly mode, pipeline between the bus and CNVM enabled, read requests are sent to CNVM after being latched in one stage. 1: Performance-priority mode, pipeline between the bus and CNVM disabled.
29	IOSPDOPEN	Enable configuration bit for I/O speed optimization function under low voltage 0: The chip operating voltage is greater than 2.7V, so I/O speed optimization is not allowed. 1: The chip operating voltage is below 2.7V, so I/O speed optimization is allowed.
28	CNVM_ECC_OVRD	Enable configuration bit for CNVM ECC coverage 0: CNVM ECC is controlled by CNVM_ECC_MODE. 1: CNVM ECC feature enabled When this bit is set to 0, the CNVM_ECC_MODE bit takes effect. Users can use this feature to test ECC functionality or other correction functions. When this bit is set to 1, the default ECC functionality is enabled. Enable the CNVM ECC function for normal use of CNVM.
27	R0_SRAM_ECCEN	Enable configuration bit for ECC functionality of Region 0 SRAM. 0: Disable ECC functionality of Region 0 SRAM. 1: Enable ECC functionality of Region 0 SRAM.
26	R1_SRAM_ECCEN	Enable configuration bit for ECC functionality of Region 1 SRAM. 0: Disable ECC functionality of Region 1 SRAM. 1: Enable ECC functionality of Region 1 SRAM.
25	R2_SRAM_ECCEN	Enable configuration bit for ECC functionality of Region 2 SRAM. 0: Disable ECC functionality of Region 2 SRAM.

		1: Enable ECC function of region 2 SRAM
24	DTCM1ECCEN	ECC enable configuration bit for DTCM1 0: Disable ECC function of DTCM1 1: Enable ECC function of DTCM1
23	DTCM0ECCEN	ECC enable configuration bit for DTCM0 0: Disable ECC function of DTCM0 1: Enable ECC function of DTCM0
22	ITCMECCEN	ECC enable configuration bit for ITCM 0: Disable ECC function of CPU ITCM 1: Enable ECC function of CPU ITCM
21	SCR	Secure mode enable configuration bit 0: Disable secure mode 1: Enable secure mode
20	SWAP	SWAP configuration bit (Not applicable to GD32H77xxP/GD32H78xxP and GD32H77xxI/GD32H78xxI) CNVM corresponds to the ECNVM enhanced data integrity area control to support OTA 0: CNVM content is consistent with the ECNVM Bank0 data integrity enhancement area. 1: CNVM content is consistent with the ECNVM Bank1 data integrity enhancement area.
19	FW_UPDATE	Firmware update configuration bit (not applicable to GD32H77xxP/GD32H78xxP and GD32H77xxI/GD32H78xxI). When the firmware update flag is 1, the hardware copies the ECNVM data integrity enhancement area data to CNVM after reset based on the SWAP bit. After the update operation is completed, the CPU jumps to the GD checksum bootloader software program. 0: Disable hardware update of ECNVM data integrity enhancement area data to CNVM. 1: Enable hardware update of ECNVM data integrity enhancement area data to CNVM.
18	FWDGSPD_STDBY	Independent watchdog pause option configuration bit in Standby mode. 0: Pause the independent watchdog in Standby mode. 1: Run the independent watchdog in Standby mode.
17	FWDGSPD_DPSLP	Independent watchdog pause option configuration bit in Deep Sleep mode. 0: Pause the independent watchdog in Deep Sleep mode. 1: Run the independent watchdog in Deep Sleep mode.
16	OTA_CHECK	OTA verification configuration bit (not applicable to GD32H77xxI/GD32H78xxI). Indicates whether hardware verification is performed during the OTA upgrade.

		0: Disable hardware verification. 1: Enable hardware checking
15:8	SPC[7:0]	Security protection level configuration bit 0xAA: No protection state 0xCC: High security protection level Any value other than 0xAA or 0xCC: Low security protection level.
7	nRST_STDBY	Standby mode entry reset option configuration bit 0: Generate a reset when entering Standby mode 1: No reset generated when entering Standby mode
6	nRST_DPSLP	Deep sleep mode entry reset option configuration bit 0: Generate a reset when entering deep sleep mode 1: No reset generated when entering deep sleep mode
5	VFALL	Firmware image (image must include certificate) verification configuration bit The factory default value of this bit is 0. 0: Disable firmware image verification 1: Enable firmware image verification
4	nWDG_HW	Watchdog control configuration bit 0: Hardware control watchdog 1: Software control watchdog
3:2	BOR_TH[1:0]	Brown-Out Reset (BOR) threshold configuration bit 00: No BOR function 01：BOR threshold value 1 10：BOR threshold value 2 11：BOR threshold value 3
1	NOPRINT	No information printed during secure boot The factory default value of this bit is 0. 0: Do not print information 1: Print information
0	DISLFI	Authorized firmware installation configuration bit The factory default value of this bit is 0. 0: Enable authorized firmware installation. 1: Disable authorized firmware installation.

DCRP address register (NVMC_DCRPADDR_EFT)

Offset address: 0x028

Reset value: 0xXXXX 0XXX

GD32H77xxW/GD32H78xxW/GD32H77xxP/GD32H78xxP: Factory value is 0x0000 01FF

GD32H77xxI/GD32H78xxI: Factory value is 0x0000 819F

Note: For GD32H77xxI/GD32H78xxI, the written values of DCRP_AREA_START[8:0] and

DCRP_AREA_END[8:0] in NVMC_DCRPADDR_MDF should not exceed 0x19F; otherwise, unpredictable consequences may occur.

This register holds the effective value of the corresponding option bits. After reset, it loads the value from the option bytes.

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
CNVM_DCRP_EREN	ECNVM_BANK0_DCRP_EREN	ECNVM_BANK1_DCRP_EREN	Reserved				DCRP_AREA_END[8:0]								
r	r	r									r				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNVM_DCRP_EN	ECNVM_BANK0_DCRP_EN	ECNVM_BANK1_DCRP_EN	Reserved				DCRP_AREA_START[8:0]								
r	r	r									r				

Bits	Fields	Descriptions
31	CNVM_DCRP_EREN	CNVM DCRP area erase enable status bit 0: DCRP is not erased 1: When SPC is downgraded or a chip erase operation with clear protection is executed, the DCRP area is erased.
30	ECNVM_BANK0_DCRP_EREN	ECNVM bank0 DCRP area erase enable status bit (not applicable to GD32H77xxI/GD32H78xxI). 0: DCRP is not erased 1: When SPC is downgraded or a chip erase operation with clear protection is executed, the DCRP area is erased.
29	ECNVM_BANK1_DCRP_EREN	ECNVM bank1 DCRP area erase enable status bit (not applicable to GD32H77xxI/GD32H78xxI). 0: DCRP is not erased 1: When SPC is downgraded or a chip erase operation with clear protection is executed, the DCRP area is erased.
28:25	Reserved	The reset value must be maintained.
24:16	DCRP_AREA_END[8:0]	DCRP area end address status bit. This bit field contains the last 4KB block of the DCRP region. The last byte address of the region = (DCRP_AREA_END[8:0] + 1) * 4096 - 1 + base address When CNVM_DCRP_EN is set to 1, the base address is the CNVM start address in the CNVM area. When ECNVM_BANK0_DCRP_EN is set to 1, the base address is the bank0 start address in the bank0 area.

When ECNVM_BANK1_DCRP_EN is set to 1, the base address is the bank1 start address in the bank1 area.
If DCRP_AREA_END[8:0] equals DCRP_AREA_START[8:0], the entire CNVM / bank0 / bank1 is designated as the DCRParea.
If DCRP_AREA_END[8:0] is less than DCRP_AREA_START[8:0], the DCRParea is invalid.
Note: The minimum size of the DCRP area is 8K bytes.

15	CNVM_DCRP_EN	CNVM area DCRP area enable status bit (GD32H77xxI/GD32H78xxI: This bit is set to 1 by default and cannot be modified). 0: Disable CNVM region DCRParea 1: Enable CNVM region DCRP region
14	ECNVM_BANK0_DCRP_EN	ECNVM bank0 area DCRP area enable status bit (Not applicable to GD32H77xxI/GD32H78xxI). 0: Disable ECNVM bank0 region DCRP region 1: Enable ECNVM bank0 DCRParea
13	ECNVM_BANK1_DCRP_EN	ECNVM bank1 area DCRP area enable status bit (Not applicable to GD32H77xxI/GD32H78xxI). 0: Disable ECNVM bank1 DCRParea 1: Enable ECNVM bank1 DCRParea
12:9	Reserved	The reset value must be maintained.
8:0	DCRP_AREA_START[8:0]	DCRP area start address status bit. This bit field contains the starting 4K-byte block of the DCRParea. Region first byte address = DCRP_AREA_START[8:0] * 4096 + base address When CNVM_DCRP_EN is set to 1, the base address is the CNVM start address in the CNVM area. When ECNVM_BANK0_DCRP_EN is set to 1, the base address is the bank0 start address in the bank0 area. When ECNVM_BANK1_DCRP_EN is set to 1, the base address is the bank1 start address in the bank1 area. If DCRP_AREA_END[8:0] equals DCRP_AREA_START[8:0], the entire CNVM / bank0 / bank1 is designated as the DCRParea. If DCRP_AREA_END[8:0] is less than DCRP_AREA_START[8:0], the DCRParea is invalid. **Note:** The minimum size of the DCRP area is 8K bytes.

DCRP Address Register (NVMC_DCRPADDR_MDF)

Offset address: 0x02C
Reset value: 0xXXXX 0XXX

This register holds the modification value of the corresponding option bits. After a system reset, the value reflects the effective value of the corresponding option bits.

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
CNVM_DCRP_EREN	ECNVM_BANK0_DCRP_EREN	ECNVM_BANK1_DCRP_EREN	Reserved				DCRP_AREA_END[8:0]								
rw	rw	rw									rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNVM_DCRP_EN	ECNVM_BANK0_DCRP_EN	ECNVM_BANK1_DCRP_EN	Reserved				DCRP_AREA_START[8:0]								
rw	rw	rw									rw				

Bits	Fields	Descriptions
31	CNVM_DCRP_EREN	CNVM DCRP area erase enable configuration bit 0: DCRP is not erased 1: When SPC is downgraded or a chip erase operation with clear protection is executed, the DCRP area is erased.
30	ECNVM_BANK0_DCRP_EREN	ECNVM bank0 DCRP area erase enable configuration bit (not applicable for GD32H77xxI/GD32H78xxI) 0: DCRP is not erased 1: When SPC is downgraded or a chip erase operation with clear protection is executed, the DCRP area is erased.
29	ECNVM_BANK1_DCRP_EREN	ECNVM bank1 DCRP area erase enable configuration bit (not applicable for GD32H77xxI/GD32H78xxI) 0: DCRP is not erased 1: When SPC is downgraded or a chip erase operation with clear protection is executed, the DCRP area is erased.
28:25	Reserved	The reset value must be maintained.
24:16	DCRP_AREA_END[8:0]	DCRP region end address configuration bit This bit field contains the last 4KB block of the DCRP region. The last byte address of the region = (DCRP_AREA_END[8:0] + 1) * 4096 - 1 + base address When CNVM_DCRP_EN is set to 1, the base address is the CNVM start address in the CNVM area. When ECNVM_BANK0_DCRP_EN is set to 1, the base address is the bank0 start address in the bank0 area. When ECNVM_BANK1_DCRP_EN is set to 1, the base address is the bank1 start address in the bank1 area. If DCRP_AREA_END[8:0] equals DCRP_AREA_START[8:0], the entire CNVM / bank0 / bank1 is designated as the DCRParea. If DCRP_AREA_END[8:0] is less than DCRP_AREA_START[8:0], the DCRParea

		is invalid. **Note:** The minimum size of the DCRP area is 8K bytes.
15	CNVM_DCRP_EN	CNVM region DCRP enable configuration bit (GD32H77xxI/GD32H78xxI: This bit is defaulted to 1 and cannot be modified). 0: Disable CNVM region DCRParea 1: Enable CNVM region DCRP region
14	ECNVM_BANK0_DCRP_EN	ECNVM bank0 region DCRP enable configuration bit (Not applicable for GD32H77xxI/GD32H78xxI). 0: Disable ECNVM bank0 region DCRP region 1: Enable ECNVM bank0 DCRParea
13	ECNVM_BANK1_DCRP_EN	Enable configuration bit for ECNVM bank1 DCRParea (not applicable to GD32H77xxI/GD32H78xxI) 0: Disable ECNVM bank1 DCRParea 1: Enable ECNVM bank1 DCRParea
12:9	Reserved	The reset value must be maintained.
8:0	DCRP_AREA_START[8:0]	Start address configuration bit for DCRParea This bit field contains the starting 4K-byte block of the DCRParea. Region first byte address = DCRP_AREA_START[8:0] * 4096 + base address When CNVM_DCRP_EN is set to 1, the base address is the CNVM start address in the CNVM area. When ECNVM_BANK0_DCRP_EN is set to 1, the base address is the bank0 start address in the bank0 area. When ECNVM_BANK1_DCRP_EN is set to 1, the base address is the bank1 start address in the bank1 area. If DCRP_AREA_END[8:0] equals DCRP_AREA_START[8:0], the entire CNVM / bank0 / bank1 is designated as the DCRParea. If DCRP_AREA_END[8:0] is less than DCRP_AREA_START[8:0], the DCRParea is invalid. **Note:** The minimum size of the DCRP area is 8K bytes.

Secure User Area Address Register (NVMC_SCRADDR_EFT)

Offset address: 0x030

Reset value: 0xXXXX 0XXX

GD32H77xxW/GD32H78xxW/GD32H77xxP/GD32H78xxP: Factory value is 0x0000 01FF

GD32H77xxI/GD32H78xxI: Factory value is 0x0000 819F

Note: For GD32H77xxI/GD32H78xxI, ensure that the written values of SCR_AREA_START[8:0] and SCR_AREA_END[8:0] in NVMC_SCRADDR_MDF do not exceed 0x19F; otherwise, unpredictable consequences may occur.

This register holds the effective value of the corresponding option bits. After reset, it loads the value from the option bytes.

After reset, based on the SWAP status, the read-only register updates the active configuration of ECNVM Bank0 or Bank1.

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
CNVM_SCR_EREN	ECNVM_BANK0_SCR_EREN	ECNVM_BANK1_SCR_EREN	Reserved				SCR_AREA_END[8:0]								
r	r	r									r				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNVM_SCR_EN	ECNVM_BANK0_SCR_EN	ECNVM_BANK1_SCR_EN	Reserved				SCR_AREA_START[8:0]								
r	r	r									r				

Bits	Fields	Descriptions
31	CNVM_SCR_EREN	CNVM secure userarea erase enable status bit 0: CNVM secure user area is not erased 1: When SPC is downgraded or a chip erase operation with clearance protection is executed, the CNVM secure user area is erased.
30	ECNVM_BANK0_SCR_EREN	ECNVM Bank0 secure user area erase enable status bit (not applicable to GD32H77xxI/GD32H78xxI) 0: ECNVM bank0 secure user area is not erased 1: When SPC is downgraded or a chip erase operation with clearance protection is executed, the ECNVM bank0 secure user area is erased.
29	ECNVM_BANK1_SCR_EREN	ECNVM Bank1 secure user area erase enable status bit (not applicable to GD32H77xxI/GD32H78xxI) 0: ECNVM bank1 secure user area is not erased 1: When SPC is downgraded or a chip erase operation with clear protection is executed, the ECNVM bank1 secure user area is erased.
28:25	Reserved	The reset value must be maintained.
24:16	SCR_AREA_END[8:0]	Secure user area end address status bit This bitfield contains the last 4K byte block of the secure user area. The last byte address of the area = (SCR_AREA_END[8:0] + 1) * 4096 - 1 + base address. When CNVM_SCR_EN is set to 1, the base address is the starting address of the CNVM region. When ECNVM_BANK0_SCR_EN is set to 1, the base address is the starting address of the bank0 region. When ECNVM_BANK1_SCR_EN is set to 1, the base address is the starting address of bank1 in the bank1 region. If SCR_AREA_END[8:0] is equal to SCR_AREA_START[8:0], the entire

		CNVM/Bank0/Bank1 is the secure user area. If SCR_AREA_END[8:0] is less than SCR_AREA_START[8:0], the secure user area is invalid. **Note:** The minimum size of the secure user area is 8KB.
15	CNVM_SCR_EN	CNVM area secure user area enable status bit (GD32H77xxI/GD32H78xxI: This bit is set to 1 by default and cannot be modified). 0: Disable CNVM area secure user area 1: Enable CNVM area secure user area
14	ECNVM_BANK0_SCR_EN	ECNVM bank0 secure user area enable status bit (Not applicable to GD32H77xxI/GD32H78xxI). 0: Disable ECNVM Bank0 secure user area 1: Enable ECNVM Bank0 secure user area
13	ECNVM_BANK1_SCR_EN	ECNVM bank1 secure user area enable status bit (Not applicable to GD32H77xxI/GD32H78xxI). 0: Disable ECNVM Bank1 secure user area 1: Enable ECNVM Bank1 secure user area
12:9	Reserved	The reset value must be maintained.
8:0	SCR_AREA_START[8:0]	Secure user area start address status bit This bit field contains the starting 4KB block of the secure user area. The address of the first byte in the region = SCR_AREA_START[8:0] * 4096 + base address. When CNVM_SCR_EN is set to 1, the base address is the starting address of the CNVM region. When ECNVM_BANK0_SCR_EN is set to 1, the base address is the starting address of the bank0 region. When ECNVM_BANK1_SCR_EN is set to 1, the base address is the starting address of bank1 in the bank1 region. If SCR_AREA_END[8:0] is equal to SCR_AREA_START[8:0], the entire CNVM/Bank0/Bank1 is the secure user area. If SCR_AREA_END[8:0] is less than SCR_AREA_START[8:0], the secure user area is invalid. **Note:** The minimum secure user area size is 8KB.

Secure User Area Address Register (NVMC_SCRADDR_MDF)

Offset address: 0x34
Reset value: 0xXXXX 0XXX

This register holds the modification value of the corresponding option bits. After a system reset, the value reflects the effective value of the corresponding option bits.

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
CNVM_SCR_EREN	ECNVM_BANK0_SCR_EREN	ECNVM_BANK1_SCR_EREN	Reserved				SCR_AREA_END[8:0]								
rw	rw	rw									rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNVM_SCR_EN	ECNVM_BANK0_SCR_EN	ECNVM_BANK1_SCR_EN	Reserved				SCR_AREA_START[8:0]								
rw	rw	rw									rw				

Bits	Fields	Descriptions
31	CNVM_SCR_EREN	CNVM secure user area erase enable configuration bit 0: CNVM secure user area is not erased 1: When SPC is downgraded or a chip erase operation with clearance protection is executed, the CNVM secure user area is erased.
30	ECNVM_BANK0_SCR_EREN	ECNVM bank0 secure user area erase enable configuration bit (not applicable to GD32H77xxI/GD32H78xxI) 0: ECNVM bank0 secure user area is not erased 1: When SPC is downgraded or a chip erase operation with clearance protection is executed, the ECNVM bank0 secure user area is erased.
29	ECNVM_BANK1_SCR_EREN	ECNVM bank1 secure user area erase enable configuration bit (not applicable to GD32H77xxI/GD32H78xxI) 0: ECNVM bank1 secure user area is not erased 1: When SPC is downgraded or a chip erase operation with clear protection is executed, the ECNVM bank1 secure user area is erased.
28:25	Reserved	The reset value must be maintained.
24:16	SCR_AREA_END[8:0]	Secure user area end address configuration bits This bitfield contains the last 4K byte block of the secure user area. The last byte address of the area = (SCR_AREA_END[8:0] + 1) * 4096 - 1 + base address. When CNVM_SCR_EN is set to 1, the base address is the starting address of the CNVM region. When ECNVM_BANK0_SCR_EN is set to 1, the base address is the starting address of the bank0 region. When ECNVM_BANK1_SCR_EN is set to 1, the base address is the starting address of bank1 in the bank1 region. If SCR_AREA_END[8:0] is equal to SCR_AREA_START[8:0], the entire CNVM/Bank0/Bank1 is the secure user area. If SCR_AREA_END[8:0] is less than SCR_AREA_START[8:0], the secure user area is invalid.

Note: The minimum size of the secure user area is 8KB.

15	CNVM_SCR_EN	CNVM area secure user area enable configuration bit (GD32H77xxI/GD32H78xxI: This bit is set to 1 by default and cannot be modified). 0: Disable CNVM area secure user area 1: Enable CNVM area secure user area
14	ECNVM_BANK0_SCR_EN	ECNVM bank0 secure user area enable configuration bit (not applicable to GD32H77xxI/GD32H78xxI). 0: Disable ECNVM Bank0 secure user area 1: Enable ECNVM Bank0 secure user area
13	ECNVM_BANK1_SCR_EN	ECNVM Bank1 secure user area enable configuration bit (not applicable to GD32H77xxI/GD32H78xxI) 0: Disable ECNVM Bank1 secure user area 1: Enable ECNVM Bank1 secure user area
12:9	Reserved	The reset value must be maintained.
8:0	SCR_AREA_START[8:0]	Secure user area start address configuration bit This bit field contains the starting 4KB block of the secure user area. The address of the first byte in the region = SCR_AREA_START[8:0] * 4096 + base address. When CNVM_SCR_EN is set to 1, the base address is the starting address of the CNVM region. When ECNVM_BANK0_SCR_EN is set to 1, the base address is the starting address of the bank0 region. When ECNVM_BANK1_SCR_EN is set to 1, the base address is the starting address of bank1 in the bank1 region. If SCR_AREA_END[8:0] is equal to SCR_AREA_START[8:0], the entire CNVM/Bank0/Bank1 is the secure user area. If SCR_AREA_END[8:0] is less than SCR_AREA_START[8:0], the secure user area is invalid. **Note:** The minimum secure user area size is 8KB.

CNVM Erase/Programming Protection Register (NVMC_WP_CNVM_EFT)

Offset Address: 0x038

Reset value: 0xXXXX XXXX

The actual factory value of GD32H77xxW/GD32H78xxW/GD32H77xxP/GD32H78xxP/GD32H77xxI/GD32H78xxI is 0x0000 FFFF.

Below are the valid factory values of GD32H77xxW/GD32H78xxW/GD32H77xxP/GD32H78xxP/GD32H77xxI/GD32H78xxI (determined by CNVM capacity and operable by the user). Values outside the valid factory default range remain unchanged as the actual factory default value.

GD32H77xxW/GD32H78xxW and GD32H77xxP/GD32H78xxP: Valid factory value is 0x0000 FFFF.
GD32H77xxI/GD32H78xxI: Valid factory value is 0x0000 1FFF.

This register holds the effective value of the corresponding option bits. After reset, it loads the value from the option bytes.
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
WP[15:0]															
r															

Bits	Fields	Descriptions
31:16	Reserved	The reset value must be maintained.
15:0	WP[15:0]	CNVM sector erase/program protection option status bit For WP[15:0], each bit represents the erase/program protection status of the corresponding 2 sectors. 0: The corresponding 2 sectors are under erase/program protection. 1: The corresponding 2 sectors are not under erase/program protection.

CNVM Erase/Programming Protection Register (NVMC_WP_CNVM_MDF)

Offset Address: 0x03C
Reset value: 0xXXXX XXXX

This register holds the modification value of the corresponding option bits. After a system reset, the value reflects the effective value of the corresponding option bits.

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
WP[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	The reset value must be maintained.
15:0	WP[15:0]	CNVM sector erase/program protection option configuration bit For WP[15:0], each bit represents the erase/program protection status of the corresponding 2 sectors. 0: The corresponding 2 sectors are under erase/program protection.

1: The corresponding 2 sectors are not under erase/program protection.

Boot Loader Address Register (NVMC_BTADDR_EFT)

Offset Address: 0x040
Reset value: 0xXXXX XXXX, factory value is 0x1FEF 080X

Note: The specific meaning of 'X' can be found in the errata sheet. This register reflects the active value of the corresponding option bits. After reset, the value from the option bytes is loaded.

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
BOOT_ADDR1[15:0]															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BOOT_ADDR0[15:0]															
r															

Bits	Fields	Descriptions
31:16	BOOT_ADDR1[15:0]	Boot loader address 1 status bit If the BOOT pin is pulled high, the high 16 bits of the boot address are determined by this field.
15:0	BOOT_ADDR0[15:0]	Boot loader address 0 status bit If the BOOT pin is pulled low, the high 16 bits of the boot address are determined by this field.

Boot Loader Address Register (NVMC_BTADDR_MDF)

Offset address: 0x044
Reset value: 0xXXXX XXXX

This register holds the modification value of the corresponding option bits. After a system reset, the value reflects the effective value of the corresponding option bits.

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
BOOT_ADDR1[15:0]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BOOT_ADDR0[15:0]															
rw															

Bits	Fields	Descriptions
31:16	BOOT_ADDR1[15:0]	Boot loader address 1 configuration bit.

		If the BOOT pin is pulled high, the high 16 bits of the boot address are determined by this field.
15:0	BOOT_ADDR0[15:0]	Boot address 0 configuration bit. If the BOOT pin is pulled low, the high 16 bits of the boot address are determined by this field.

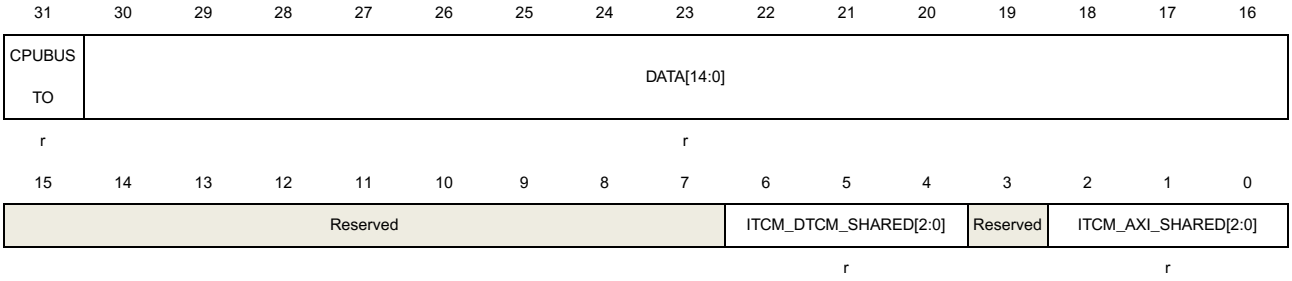
Option Byte Status Register 1 (NVMC_OBSTAT1_EFT)

Offset address: 0x050

Reset value: 0xXXXX 0XXX, factory value is 0xFFFF 0000

This register holds the effective value of the corresponding option bits. After reset, it loads the value from the option bytes.

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



Bits	Fields	Descriptions
31	CPUBUSTO	CPU bus timeout enable status bit. The reset value of the CPUBUSTO bit in the SYSCFG_BUSTO register depends on this bit.
30:16	DATA[14:0]	User-defined option byte status bit.
15:7	Reserved	The reset value must be maintained.
6:4	ITCM_DTCM_SHARED[2:0]	ITCM size status bit in the RAM shared by ITCM and DTCM. DTCM + ITCM size cannot exceed 256KB bytes. 000: 0-byte ITCM 001: 64-KB ITCM 010: 128-KB ITCM 011: 192-KB ITCM 100: 256-KB ITCM 101~111: 0-byte ITCM
3	Reserved	The reset value must be maintained.
2:0	ITCM_AXI_SHARED[2:0]	ITCM size status bits in ITCM and AXI shared RAM. AXI + ITCM size cannot exceed 256 KB. 000: 0-KB ITCM 001: 64-KB ITCM

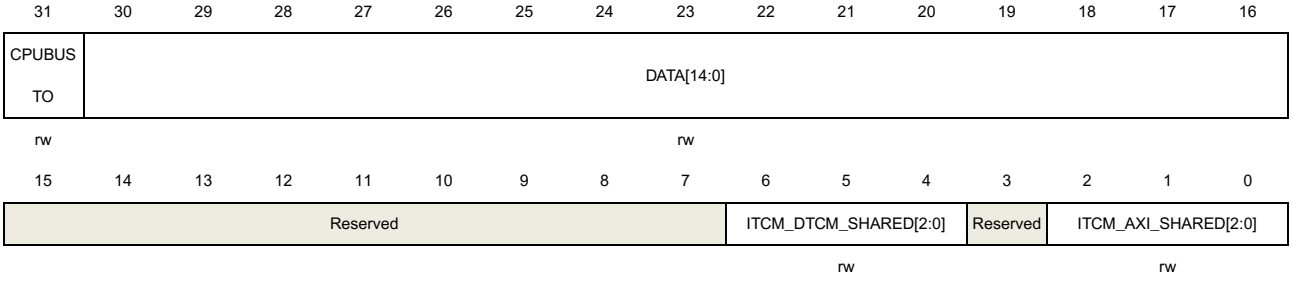
010: 128-KB ITCM
011: 192-KB ITCM
100: 256-KB ITCM
101~111: 0-KB ITCM

Option Byte Status Register 1 (NVMC_OBSTAT1_MDF)

Offset address: 0x054
Reset value: 0xXXXX 0XXX

This register holds the modification value of the corresponding option bits. After a system reset, the value reflects the effective value of the corresponding option bits.

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



Bits	Fields	Descriptions
31	CPUBUSTO	CPU bus timeout enable configuration bit. The reset value of the CPUBUSTO bit in the SYSCFG_BUSTO register depends on this bit.
30:16	DATA[14:0]	User-defined option byte configuration bit.
15:7	Reserved	The reset value must be maintained.
6:4	ITCM_DTCM_SHARED[2:0]	ITCM size configuration bits in ITCM and DTCM shared RAM. DTCM + ITCM size cannot exceed 256KB bytes. 000: 0-byte ITCM 001: 64-KB ITCM 010: 128-KB ITCM 011: 192-KB ITCM 100: 256-KB ITCM 101~111: 0-byte ITCM
3	Reserved	The reset value must be maintained.
2:0	ITCM_AXI_SHARED[2:0]	ITCM size configuration bits in ITCM and AXI shared RAM. AXI + ITCM size cannot exceed 256 KB. 000: 0-KB ITCM 001: 64-KB ITCM 010: 128-KB ITCM

011: 192-KB ITCM
100: 256-KB ITCM
101~111: 0-KB ITCM

ECNVM Erase/Program Protection Register 0 (NVMC_WP0_ECNVM_EFT)

Offset Address: 0x080
Reset value: 0xXXXX XXXX
GD32H77xxW/GD32H78xxW/GD32H77xxP/GD32H78xxP/GD32H77xxI/GD32H78xxI actual factory default value: 0xFFFF FFFF
Below are the valid factory values of GD32H77xxW/GD32H78xxW/GD32H77xxP/GD32H78xxP/GD32H77xxI/GD32H78xxI (determined by CNVM capacity and operable by the user). Values outside the valid factory default range remain unchanged as the actual factory default value.
GD32H77xxW/GD32H78xxW: Valid factory default value is 0xFFFF FFFF
GD32H77xxP/GD32H78xxP: Valid factory default value is 0x0FFF FFFF
GD32H77xxI/GD32H78xxI: This register function is not available, do not configure.

This register holds the effective value of the corresponding option bits. After reset, it loads the value from the option bytes.
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
WP[31:16]															
r															

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

Bits	Fields	Descriptions
31:0	WP[31:0]	ECNVM sector erase/program protection option status bit For WP[31:0], each bit reflects the erase/program protection status of the corresponding 2 sectors. 0: The corresponding 2 sectors are under erase/program protection. 1: The corresponding 2 sectors are not under erase/program protection. WP[0]: Sector 32 to Sector 33 WP[1]: Sector 34 to Sector 35 … **Note:** This register specifically controls the sectors of ECNVM. Starting from Sector 32, each bit corresponds to two sectors (128KB). For different devices, the starting sector of ECNVM bank0 may vary.

ECNVM Erase/Program Protection Register 1 (NVMC_WP1_ECNVM_EFT)

Offset Address: 0x084
Reset value: 0xXXXX XXXX

GD32H77xxW/GD32H78xxW/GD32H77xxP/GD32H78xxP/GD32H77xxI/GD32H78xxI factory default value: 0x0FFF FFFF

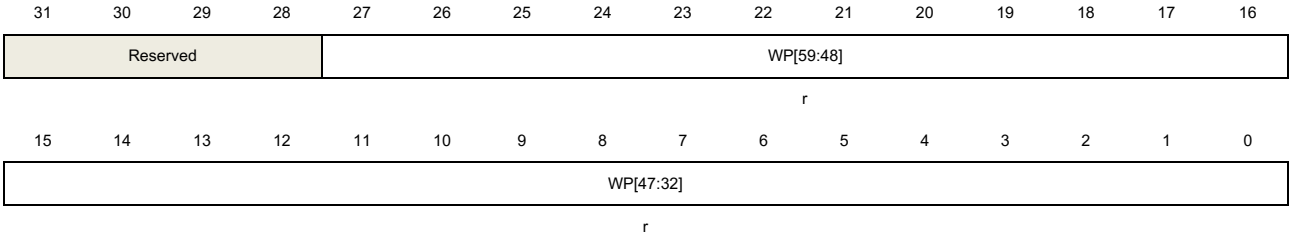
Below are the valid factory default values of GD32H77xxW/GD32H78xxW/GD32H77xxP/GD32H78xxP/GD32H77xxI/GD32H78xxI (determined by CNVM capacity and operable by the user). Values outside the valid factory default range remain unchanged as the actual factory default value.

GD32H77xxW/GD32H78xxW: Valid factory default value is 0x0FFF FFFF

GD32H77xxP/GD32H78xxP and GD32H77xxI/GD32H78xxI: This register function is not available, do not configure.

This register holds the effective value of the corresponding option bits. After reset, it loads the value from the option bytes.

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



Bits	Fields	Descriptions
31:28	Reserved	The reset value must be maintained.
27:0	WP[59:32]	ECNVM sector erase/program protection option status bit For WP[59:32], each bit reflects the erase/program protection status of the corresponding 2 sectors. 0: The corresponding 2 sectors are under erase/program protection. 1: The corresponding 2 sectors are not under erase/program protection. WP[32]: Sector 96 to Sector 97 WP[33]: Sector 98 to Sector 99 … **Note:** This register specifically controls the sectors of ECNVM. Each bit corresponds to two sectors (128KB). For different devices, the starting sector of ECNVM bank0 may vary.

ECNVM Erase/Programming Protection Register 0 (NVMC_WP0_ECNVM_MDF)

GD32H77xxI/GD32H78xxI do not have this register function, do not set.

Offset address: 0x088

Reset value: 0xXXXX XXXX

This register holds the modification value of the corresponding option bits. After a system reset, the value reflects the effective value of the corresponding option bits.

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

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

Bits	Fields	Descriptions
31:0	WP[31:0]	ECNVM sector erase/program protection option configuration bit For WP[31:0], each bit reflects the erase/program protection status of the corresponding 2 sectors. 0: The corresponding 2 sectors are under erase/program protection. 1: The corresponding 2 sectors are not under erase/program protection. WP[0]: Sector 32 to Sector 33 WP[1]: Sector 34 to Sector 35 … **Note:** This register specifically controls the sectors of ECNVM. Starting from Sector 32, each bit corresponds to two sectors (128KB). For different devices, the starting sector of ECNVM bank0 may vary.

ECNVM Erase/Programming Protection Register 1 (NVMC_WP1_ECNVM_MDF)

GD32H77xxP/GD32H78xxP and GD32H77xxI/GD32H78xxI do not have this register function, do not set.

Offset address: 0x08C

Reset value: 0xXXXX XXXX

This register holds the modification value of the corresponding option bits. After a system reset, the value reflects the effective value of the corresponding option bits.

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				WP[59:48]											
				rw											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WP[47:32]															
rw															

Bits	Fields	Descriptions
31:28	Reserved	The reset value must be maintained.
27:0	WP[59:32]	ECNVM sector erase/program protection option configuration bit For WP[59:32], each bit reflects the erase/program protection status of the corresponding 2 sectors. 0: The corresponding 2 sectors are under erase/program protection. 1: The corresponding 2 sectors are not under erase/program protection.

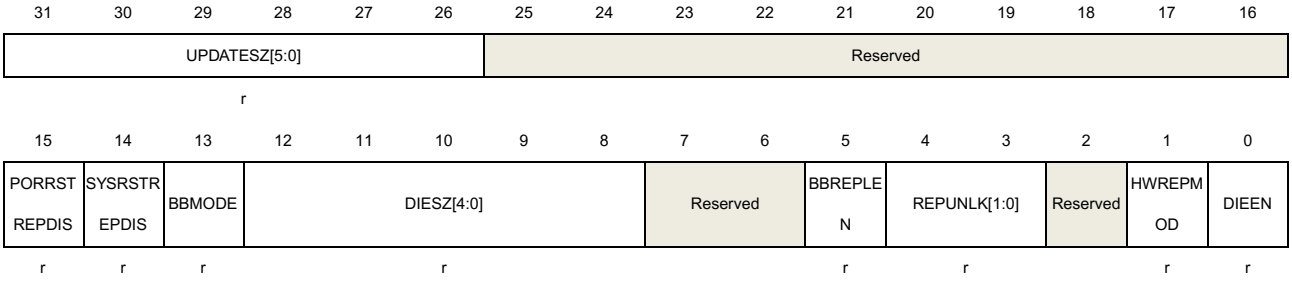
WP[32]: Sector 96 to Sector 97
WP[33]: Sector 98 to Sector 99
…
Note: This register specifically controls the sectors of ECNVM. Each bit corresponds to two sectors (128KB). For different devices, the starting sector of ECNVM bank0 may vary.

Option Byte Repair Control Register (NVMC_OBREPCTL_EFT)

Offset address: 0x090
Reset value: 0xXXXX XXXX
GD32H77xxW/GD32H78xxW/GD32H77xxP/GD32H78xxP: Factory value is 0xFE10 7F3B
GD32H77xxI/GD32H78xxI: Factory value is 0xFE10 DF00
Note: Reserved bits must remain at factory value.

This register holds the effective value of the corresponding option bits. After reset, it loads the value from the option bytes.

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



Bits	Fields	Descriptions
31:26	UPDATESZ[5:0]	During OTA upgrade, the size of the update area is adjusted with a granularity of 32KB. For example, when set to 6'h3F, the update area size is 2MB. **Note:** For Mode 2, the size of the update area configured by this bit must be less than or equal to the data integrity enhancement area size configured by the DIESZ bit.
25:16	Reserved	The reset value must be maintained.
15	PORRSTREPDIS	Power-on reset self-check repair enable status bit 0: Enable power-on reset self-check repair function 1: Disable power-on reset self-check repair function **Note:** In Mode 1 and Mode 2, this bit is set to 0.
14	SYSRSTREPDIS	System reset self-check repair enable status bit 0: Enable system reset self-check repair function 1: Disable system reset self-check repair function
13	BBMODE	Mode status bit to determine whether the repair was successful after completion. 0: After the repair, all errors are logged to ROBBCNT[3:0] and ROBBADDR[15:0].

		1: After the repair, errors exceeding 1 bit are logged to ROBBCNT[3:0] and ROBBADDR[15:0], while 1-bit errors are not logged.
12:8	DIESZ[4:0]	Data integrity enhancement area size status bit When the data integrity enhancement mode is enabled, the address of the data integrity enhancement region starts from 0 with a granularity of 64KB, ensuring consistency between the data integrity enhancement region and CNVM data. **Note:** It is recommended to run the program in SRAM to modify DIESZ[4:0]. **Note**: Mode 1: This bit is configured as 0x1F. Mode 2: Users can configure the size based on requirements, with part of the region enhancing data integrity and the remaining part achieving high-speed write capability.
7:6	Reserved	The reset value must be maintained.
5	BBREPLEN	Bad block address replacement status bit 0: No replacement. 1: When CNVM reads the bad block address (recorded in NVMC_OBBBADDRx_EFT and NVMC_CNVM_ROBBADDRx), hardware automatically retrieves ECNVM data from the data integrity enhancement region.
4:3	REPUNLK[1:0]	Repair unlock status bit 11: Repair unlock. Others: Reserved. **Note:** For Mode 1 and Mode 2, this bit is configured as 0x3.
2	Reserved	The reset value must be maintained.
1	HWREPMOD	Hardware self-check mode status bit 0: No self-check 1: Hardware self-check mode **Note:** In Mode 1 and Mode 2, this bit is set to 1.
0	DIEEN	Data integrity enhancement enable status bit. 0: Disable data integrity enhancement mode 1: Enable data integrity enhancement mode **Note:** In Mode 1 and Mode 2, this bit is set to 1. **Note:** It is recommended to execute code modifying DIEEN in SRAM, and erase all CNVM and ECNVM data integrity enhancement areas before modification to avoid data inconsistency.

Option Byte Repair Control Register (NVMC_OBREPCTL_MDF)

GD32H77xxI/GD32H78xxI do not have this register function, do not set.

Offset address: 0x094

Reset value: 0x0000 00XX

This register holds the modification value of the corresponding option bits. After a system reset, the value reflects the effective value of the corresponding option bits.

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
UPDATESZ[5:0]						Reserved									
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PORRSTREPDIS	SYSRSTREPDIS	BBMODE	DIESZ[4:0]					Reserved		BBREPLEN	REPUNLK[1:0]		Reserved	HWREPMOD	DIEEN
rw	rw	rw	rw							rw	rw			rw	rw

Bits	Fields	Descriptions
31:26	UPDATESZ[5:0]	During OTA upgrade, the size of the update area is adjusted with a granularity of 32KB. For example, when set to 6'h3F, the update area size is 2MB. **Note:** For Mode 2, the size of the update area configured by this bit must be less than or equal to the data integrity enhancement area size configured by the DIESZ bit.
25:16	Reserved	The reset value must be maintained.
15	PORRSTREPDIS	Power-on reset self-check repair enable configuration bit 0: Enable power-on reset self-check repair function 1: Disable power-on reset self-check repair function **Note:** In Mode 1 and Mode 2, this bit is set to 0.
14	SYSRSTREPDIS	System reset self-check repair enable configuration bit 0: Enable system reset self-check repair function 1: Disable system reset self-check repair function
13	BBMODE	Mode configuration bit to determine if the repair is successful after the fix. 0: After the repair, all errors are logged to ROBBCNT[3:0] and ROBBADDR[15:0]. 1: After the repair, errors exceeding 1 bit are logged to ROBBCNT[3:0] and ROBBADDR[15:0], while 1-bit errors are not logged.
12:8	DIESZ[4:0]	Data integrity enhancement region size configuration bit. When the data integrity enhancement mode is enabled, the address of the data integrity enhancement region starts from 0 with a granularity of 64KB, ensuring consistency between the data integrity enhancement region and CNVM data. **Note:** It is recommended to run the program in SRAM to modify DIESZ[4:0]. **Note:** Mode 1: This bit is configured as 0x1F. Mode 2: Users can configure the size based on requirements, with part of the region enhancing data integrity and the remaining part achieving high-speed write capability.
7:6	Reserved	The reset value must be maintained.
5	BBREPLEN	Bad block address replacement configuration bit. 0: No replacement. 1: When CNVM reads the bad block address (recorded in NVMC_OBBBADDRx_EFT and NVMC_CNVM_ROBBADDRx), hardware

		automatically retrieves ECNVM data from the data integrity enhancement region.
4:3	REPUNLK[1:0]	Repair unlock configuration bit. 11: Repair unlock. Others: Reserved. **Note:** For Mode 1 and Mode 2, this bit is configured as 0x3.
2	Reserved	The reset value must be maintained.
1	HWREPMOD	Hardware self-check mode configuration bit. 0: No self-check 1: Hardware self-check mode **Note:** In Mode 1 and Mode 2, this bit is set to 1.
0	DIEEN	Data integrity enhancement enable status bit. 0: Disable data integrity enhancement mode 1: Enable data integrity enhancement mode **Note:** In Mode 1 and Mode 2, this bit is set to 1. **Note:** It is recommended to execute code modifying DIEEN in SRAM, and erase all CNVM and ECNVM data integrity enhancement areas before modification to avoid data inconsistency.

Option Byte Repair Start Address Register (NVMC_OBREPSTADDR_EFT)

Offset address: 0x098

Reset Value: 0x08XX XXXX, Default Value: 0x0800 0000

This register holds the effective value of the corresponding option bits. After reset, it loads the value from the option bytes.

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											STARTADDR[15:11]				
											r				

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

Bits	Fields	Descriptions
31:21	Reserved	The reset value must be maintained.
20:5	STARTADDR[15:0]	Self-check repair start offset address status bit. The address is located in the CNVM address area with a granularity of 256 bits. This bit is used not only for self-repair area configuration but also for CRC calculation area configuration. GD32H77xxW/GD32H78xxW/GD32H77xxP/GD32H78xxP support self-repair functionality, and

		GD32H77xxW/GD32H78xxW/GD32H77xxP/GD32H78xxP/GD32H77xxI/GD32H78xxI support CRC verification functionality. The address of the first byte in the region = STARTADDR[15:0] * 32 + 0x0800 0000. When STARTADDR[15:0] > ENDADDR[15:0], the region is invalid. The minimum region configuration is 32 bytes; in this case, STARTADDR[15:0] = ENDADDR[15:0]. The maximum region configuration is the entire CNVM block; in this case, STARTADDR[15:0] = 0, ENDADDR[15:0] = 0xFFFF (GD32H77xxW/GD32H78xxW/GD32H77xxP/GD32H78xxP) STARTADDR[15:0] = 0, ENDADDR[15:0] = 0xCFFF (GD32H77xxI/GD32H78xxI) **Note:** For Mode 1 and Mode 2, the size of the self-check repair area must be less than or equal to the data integrity enhancement area size configured by the DIESZ bits.
4:0	Reserved	The reset value must be maintained.

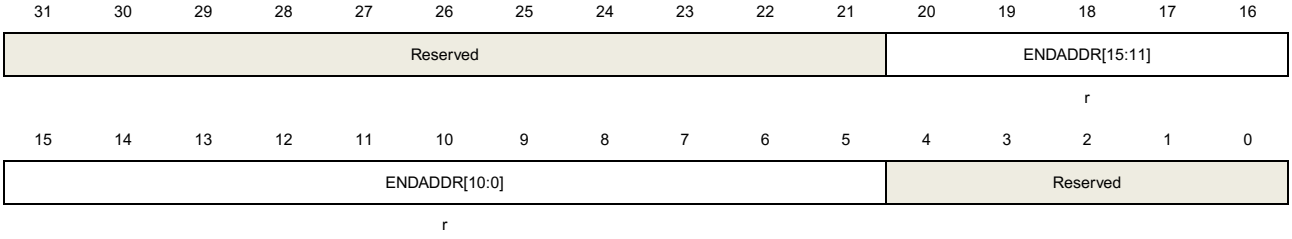
Option Byte Repair End Address Register (NVMC_OBREPENDADDR_EFT)

Offset address: 0x09C

Reset value: 0x08XX XXXX
GD32H77xxW/GD32H78xxW/GD32H77xxP/GD32H78xxP: Default value is 0x081F FFE0
GD32H77xxI/GD32H78xxI: Default value is 0x0800 0000

This register holds the effective value of the corresponding option bits. After reset, it loads the value from the option bytes.

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



Bits	Fields	Descriptions
31:21	Reserved	The reset value must be maintained.
20:5	ENDADDR[15:0]	Self-repair end offset address status bit The address is located in the CNVM address area with a granularity of 256 bits. This bit is used not only for self-repair area configuration but also for CRC calculation area configuration. GD32H77xxW/GD32H78xxW/GD32H77xxP/GD32H78xxP support self-repair functionality, and GD32H77xxW/GD32H78xxW/GD32H77xxP/GD32H78xxP/GD32H77xxI/GD32H78xxI support CRC verification functionality.

		Region last byte address = (ENDADDR[15:0] + 1) * 32 - 1 + 0x0800 0000. When STARTADDR[15:0] > ENDADDR[15:0], the region is invalid. The minimum region configuration is 32 bytes; in this case, STARTADDR[15:0] = ENDADDR[15:0]. The maximum region configuration is the entire CNVM block; in this case, STARTADDR[15:0] = 0, ENDADDR[15:0] = 0xFFFF (GD32H77xxW/GD32H78xxW/GD32H77xxP/GD32H78xxP) STARTADDR[15:0] = 0, ENDADDR[15:0] = 0xCFFF (GD32H77xxI/GD32H78xxI) **Note:** For Mode 1 and Mode 2, the size of the self-check repair area must be less than or equal to the data integrity enhancement area size configured by the DIESZ bits.
4:0	Reserved	The reset value must be maintained.

Option Byte Repair Start Address Register (NVMC_OBREPSTADDR_MDF)

Offset Address: 0x0A0

Reset value: 0x08XX XXXX

This register holds the modification value of the corresponding option bits. After a system reset, the value reflects the effective value of the corresponding option bits.

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											STARTADDR[15:11]				
													rw		

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

Bits	Fields	Descriptions
31:21	Reserved	The reset value must be maintained.
20:5	STARTADDR[15:0]	Self-repair start offset address configuration bits. The address is located in the CNVM address area with a granularity of 256 bits. This bit is used not only for self-repair area configuration but also for CRC calculation area configuration. GD32H77xxW/GD32H78xxW/GD32H77xxP/GD32H78xxP support self-repair functionality, and GD32H77xxW/GD32H78xxW/GD32H77xxP/GD32H78xxP/GD32H77xxI/GD32H78xxI support CRC verification functionality. The address of the first byte in the region = STARTADDR[15:0] * 32 + 0x0800 0000. When STARTADDR[15:0] > ENDADDR[15:0], the region is invalid. The minimum region configuration is 32 bytes; in this case, STARTADDR[15:0] = ENDADDR[15:0]. The maximum region configuration is the entire CNVM block; in this case,

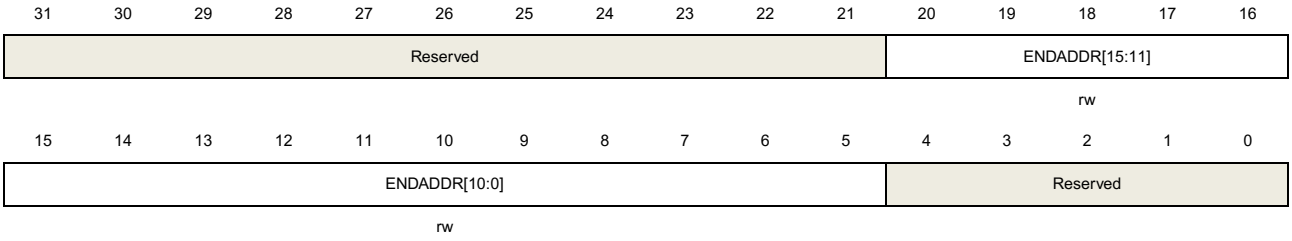
		STARTADDR[15:0] = 0, ENDADDR[15:0] = 0xFFFF (GD32H77xxW/GD32H78xxW/GD32H77xxP/GD32H78xxP) STARTADDR[15:0] = 0, ENDADDR[15:0] = 0xCFFF (GD32H77xxI/GD32H78xxI) **Note:** Software can clear STARTADDR[15:0] by writing 32 zeros to the NVMC_OBREPSTADDR_MDF register. **Note:** For Mode 1 and Mode 2, the size of the self-check repair area must be less than or equal to the data integrity enhancement area size configured by the DIESZ bits.
4:0	Reserved	The reset value must be maintained.

Option Byte Repair End Address Register (NVMC_OBREPENDADDR_MDF)

Offset Address: 0x0A4
Reset value: 0x08XX XXXX

This register holds the modification value of the corresponding option bits. After a system reset, the value reflects the effective value of the corresponding option bits.

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



Bits	Fields	Descriptions
31:21	Reserved	The reset value must be maintained.
20:5	ENDADDR[15:0]	Self-repair end offset address configuration bits The address is located in the CNVM address area with a granularity of 256 bits. This bit is used not only for self-repair area configuration but also for CRC calculation area configuration. GD32H77xxW/GD32H78xxW/GD32H77xxP/GD32H78xxP support self-repair functionality, and GD32H77xxW/GD32H78xxW/GD32H77xxP/GD32H78xxP/GD32H77xxI/GD32H78xxI support CRC verification functionality. Region last byte address = (ENDADDR[15:0] + 1) * 32 - 1 + 0x0800 0000. When STARTADDR[15:0] > ENDADDR[15:0], the region is invalid. The minimum region configuration is 32 bytes; in this case, STARTADDR[15:0] = ENDADDR[15:0]. The maximum region configuration is the entire CNVM block; in this case, STARTADDR[15:0] = 0, ENDADDR[15:0] = 0xFFFF (GD32H77xxW/GD32H78xxW/GD32H77xxP/GD32H78xxP) STARTADDR[15:0] = 0, ENDADDR[15:0] = 0xCFFF (GD32H77xxI/GD32H78xxI)

		Software can clear ENDADDR[15:0] by writing 32 zeros to the NVMC_OBREPENDADDR_MDF register. **Note:** For Mode 1 and Mode 2, the size of the self-check repair area must be less than or equal to the data integrity enhancement area size configured by the DIESZ bits.
4:0	Reserved	The reset value must be maintained.

Option Byte Bad Block Address Register x (NVMC_OBBBADDRx_EFT) (x = 0…3)

Offset Address: 0x0A8 + 0x4 * x
Reset Value: 0x08XX XXXX, Default Value: 0x0800 0000

This register holds the effective value of the corresponding option bits. After reset, it loads the value from the option bytes.

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



Bits	Fields	Descriptions
31:21	Reserved	The reset value must be maintained.
20:5	OBBBADDR[15:0]	Bad block address status bit Option bytes record bad block address, the address is located in the CNVM address area with a granularity of 256 bits. When the self-check process encounters the address recorded in OBBBADDR[15:0], if BBREPLEN is configured as 1, the address is skipped, and no self-check operation is performed; if BBREPLEN is configured as 0, the current bad block setting is ignored, and the self-check operation is performed. **Note:** The bad block address can only be recorded in this register if the CNVM contains the ECNVM data integrity enhancement area.
4:1	Reserved	The reset value must be maintained.
0	OBBBADDRVLD	Valid status bit for bad block address recorded in option bytes 0: Bad block address is invalid. 1: Bad block address is valid.

Option byte bad block address register x (NVMC_OBBBADDRx_MDF) (x = 0…3)

GD32H77xxI/GD32H78xxI do not have this register function, do not set.
Offset address: 0x0B8 + 0x4 * x
Reset value: 0x08XX XXXX

This register holds the modification value of the corresponding option bits. After a system reset, the value reflects the effective value of the corresponding option bits.

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



Bits	Fields	Descriptions
31:21	Reserved	The reset value must be maintained.
20:5	OBBBADDR[15:0]	Bad block address configuration bit Option bytes record bad block address, the address is located in the CNVM address area with a granularity of 256 bits. When the self-check process encounters the address recorded in OBBBADDR[15:0], if BBREPLEN is configured as 1, the address is skipped, and no self-check operation is performed; if BBREPLEN is configured as 0, the current bad block setting is ignored, and the self-check operation is performed. **Note:** The software can clear OBBBADDR[15:0] by writing 32 zeros to NVMC_OBBBADDRx_MDF. When the address is cleared, the status bit OBBBADDRVLD is also cleared. **Note:** The bad block address can only be recorded in this register if the CNVM contains the ECNVM data integrity enhancement area.
4:1	Reserved	The reset value must be maintained.
0	OBBBADDRVLD	Valid configuration bit for bad block address recorded in option bytes. 0: Bad block address is invalid. 1: Bad block address is valid.

Blank address register (NVMC_BLANKADDR)

GD32H77xxI/GD32H78xxI do not have this register function, do not set.
Address offset: 0x0FC
Reset value: 0x0800 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											BLANKADDR [15:11]				
													rc_r		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BLANKADDR [10:0]											Reserved				
					rc_r										

Bits	Fields	Descriptions
31:21	Reserved	The reset value must be maintained.
20:5	BLANKADDR[15:0]	Blank corresponding address for the ECNVM data integrity enhancement area. The address is located in the CNVM address area with a granularity of 256 bits. This bit only records the last occurring address.
4:0	Reserved	The reset value must be maintained.

Product ID register x (NVMC_PIDx) (x = 0,1)

Address offset: 0x100 + 0x4 * x

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

Bits	Fields	Descriptions
31:0	PID[31:0]	Product Reserved ID Register. This register is read-only. These bits remain unchanged after power-on, and this register is programmed once during production.

NVM control and status register (NVMC_CS)

Address offset: 0x330

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

Reserved

Bits	Fields	Descriptions
31	ECBBUSY	Empty CNVM buffer busy flag. This bit is set to 1 when the CNVM buffer emptying operation is in progress. This bit is cleared to 0 when the emptying operation ends or when there is no emptying operation. 0: No CNVM buffer emptying operation is in progress. 1: CNVM buffer emptying operation is in progress.
30:0	Reserved	The reset value must be maintained.

7.4.2. CNVM register

CNVM unlock register (NVMC_CNVM_KEY)

Offset address: 0x004

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
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]	NVMC_CNVM_CTL Unlock Register These bits can only be written by software. Writing the unlock value to KEY[31:0] unlocks the NVMC_CNVM_CTL register.

CNVM Control Register (NVMC_CNVM_CTL)

Offset address: 0x00C

Reset value: 0x0000 0001

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REPRESC	Reserved		CRCRERRIE	CRCENDCALIE	AUTOECCDERRDETIE	Reserved	RSERRIE	RPERRIE	Reserved		OTAFAILCNT[1:0]		PGSERRIE	WPERRIE	ENDIE
w			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
CRCSTA	Reserved				CNVM_EC	CNVM_ECC_MODE[		START	Reserved		CER	MER	SER	PG	LK

RT		C_LK	1:0]							
rw		rw	rw	rs		rw	rw	rw	rw	rs

Bits	Fields	Descriptions
31	REPRESC	Clear the results of the self-check repair function, including counters, error addresses, and status (including ROBBCNT in the NVMC_CNVM_REPCS register, ECNVM_BLANK bit, and NVMC_CNVM_ROBBADDRx registers). The repair function refers to the self-check repair function after reset. This bit can only be set to 1 or cleared to 0 by software when LK is set to 0. (Not applicable to GD32H77xxI/GD32H78xxI) 0: Do not clear the results of the repair function. 1: Clear the results of the repair function. **Note:** A new start of repair function will also clear the current repair results.
30:29	Reserved	The reset value must be maintained.
28	CRCRERRIE	CNVM CRC Read Error Interrupt Enable Bit An interrupt is generated when CRC calculation detects a protected area (DCRP or secure area) and the CRCRERRIE bit is set to 1. This bit can be set to 1 and cleared to 0 by software only when LK is set to 0. 0: Disable CRC read error interrupt 1: Enable CRC read error interrupt
27	CRCENDCALIE	CNVM CRC calculation complete interrupt enable bit This bit can be set to 1 and cleared to 0 by software only when LK is set to 0. 0: Disable CRC calculation complete interrupt 1: Enable CRC calculation complete interrupt
26	AUTOECCDERRDETIE	CNVM ECC double error detection interrupt enable bit This bit can be set to 1 and cleared to 0 by software only when LK is set to 0. 0: Disable double-bit error detection interrupt 1: Enable double-bit error detection interrupt
25	Reserved	The reset value must be maintained.
24	RSERRIE	Read secure error interrupt enable bit This bit can be set to 1 and cleared to 0 by software only when LK is set to 0. 0: Disable read secure error interrupt 1: Enable read secure error interrupt
23	RPERRIE	Read protection error interrupt enable bit This bit can be set to 1 and cleared to 0 by software only when LK is set to 0. 0: Disable read protection error interrupt 1: Enable read protection error interrupt
22:21	Reserved	The reset value must be maintained.
20:19	OTAFAILCNT[1:0]	OTA failure count. This bit is reset on power reset and is not affected by system

		reset. During the OTA upgrade process, check_bootloader will read and write this bit. (Not applicable for GD32H77xxI/GD32H78xxI)
18	PGSERRIE	Programming sequence error interrupt enable bit This bit can be set to 1 and cleared to 0 by software only when LK is set to 0. 0: Disable programming sequence error interrupt 1: Enable programming sequence error interrupt
17	WPERRIE	Erase/Program protection error interrupt enable bit This bit can be set to 1 and cleared to 0 by software only when LK is set to 0. 0: Disable erase/program protection error interrupt 1: Enable erase/program protection error interrupt
16	ENDIE	Operation complete interrupt enable bit This bit can be set to 1 and cleared to 0 by software only when LK is set to 0. 0: Disable operation complete interrupt 1: Enable Operation End Interrupt
15	CRCSTART	Enable CRC check to verify the CNVM content between NVMC_OBREPSTADDR_MDF and NVMC_OBREPENDADDR_MDF. This bit is cleared by hardware when the check is complete or an error occurs.
14:11	Reserved	The reset value must be maintained.
10	CNVM_ECC_LK	CNVM_ECC_LK can only be written as 1. After writing 1, CNVM_ECC_MODE cannot be modified. CNVM_ECC_MODE is used for testing-related functions. Upon test completion, users are advised to enable and lock the CNVM_ECC_LK function.
9:8	CNVM_ECC_MODE[1:0]	CNVM ECC Mode Configuration Bit When CNVM_ECC_OVRD is set to 0, the CNVM ECC mode can be configured, and this bit must be modified when CNVM is idle. 00: Enable ECC function 01: ECC code value is 0 during write, ECC is skipped during read; considering CNVM redundant rows. 10: Only valid data corresponding bits are written during write, ECC is skipped during read. 11: ECC code value is 0 during write, ECC is skipped during read; CNVM redundant rows are not considered. **Note:** After switching CNVM ECC mode, EMPTY_CNVM_BUF should be set to 1 to clear the CNVM buffer and prevent reading incorrect data.
7	START	Send the erase command bit to NVMC. This bit can be set to 1 by software to send an erase command to NVMC only when LK is set to 0. This bit is cleared to 0 by hardware when the BUSY bit is cleared to 0.
6:5	Reserved	The reset value must be maintained.
4	CER	Chip erase command bit

This bit can be set to 1 and cleared to 0 by software only when LK is set to 0.
0: No effect
1: Chip erase command
If both mass erase and chip erase are requested simultaneously, the chip erase will override the mass erase operation. This bit is used to erase the entire CNVM and ECNVM storage areas.

3	MER	CNVM mass erase command bit (not applicable to GD32H77xxI/GD32H78xxI) This bit can be set to 1 and cleared to 0 by software only when LK is set to 0. 0: No effect 1: CNVM mass erase command If both mass erase and sector erase are requested simultaneously, the mass erase will override the sector erase operation. This bit is used to erase the entire CNVM area.
2	SER	CNVM sector erase command bit This bit can be set to 1 and cleared to 0 by software only when LK is set to 0. 0: No effect 1: CNVM sector erase command If both mass erase and sector erase are requested simultaneously, the mass erase will replace the sector erase operation. This bit is used to erase the entire CNVM sector.
1	PG	CNVM programming command bit This bit can be set to 1 and cleared to 0 by software only when LK is set to 0. 0: No effect 1: CNVM programming command
0	LK	NVMC_CNVM_CTL register lock flag bit When the correct sequence is written to the NVMC_CNVM_KEY register, this bit is cleared to 0 by hardware. This bit can be set to 1 by software.

Note: This register must be in the reset state after the corresponding CNVM operation is completed.

CNVM Status Register (NVMC_CNVM_STAT)

Offset address: 0x010
Reset value: 0x8000 X000

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
CNVMECCF	BOOT_AT_CHECK_SYS_EFT	BOOT_AT_CHECK_SYS_MDF	CRCRERR	CRCENDCAL	AUTOECCDERRDET	Reserved	RSERR	RPERR	UPDATESTRERR	ECNVM_ENDF	OTASUC	OTAFAIL	PGSERR	WPERR	ENDF
r	r	r	rc_w1	rc_w1	r		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
NVMVER[1:0]		Reserved										CRCBUSY	Reserved		BUSY

r r r

Bits	Fields	Descriptions
31	CNVMECCF	CNVM ECC feature flag bit This bit indicates whether the ECC feature is enabled or disabled. 0: CNVM ECC feature disabled 1: CNVM ECC feature enabled **Note:** When CNVM_ECC_OVRD is set to 0 and CNVM_ECC_MODE[1:0] is set to 1 / 2 / 3, CNVMECCF is set to 0 by hardware. Otherwise, CNVMECCF is set to 1 by hardware. When CNVMECCF switches from 1 to 0, the error address and status (including the AUTOECCDERRDET bit in the NVMC_CNVM_STAT register, ECNVM_BLANK and REPERRSTAT bits in the NVMC_CNVM_REPCS register, and NVMC_CNVM_AUTOERR2ADDRx registers) are cleared. It is recommended to perform CNVM ECC toggle operations in SRAM (controlled by CNVM_ECC_OVRD and CNVM_ECC_MODE) to prevent program execution anomalies.
30	BOOT_AT_CHECK_SYS_EFT	The flag generated by option byte EFT indicates a startup from check_bootLoader (not applicable to GD32H77xxI/GD32H78xxI).
29	BOOT_AT_CHECK_SYS_MDF	The flag generated by option byte MDF indicates a startup from check_bootLoader. This bit only contains FW_UPDATE. (not applicable to GD32H77xxI/GD32H78xxI).
28	CRCRERR	CRC read error flag bit This bit is set by hardware when CRC calculation detects a protected area (DCRP or secure area), and cleared by software by writing 1. 0: CRC calculation did not detect a protected area 1: CRC calculation detected a protected area, CRC result may be incorrect. **Note:** This flag is effective only when the CRCENDCAL bit is set to 1.
27	CRCENDCAL	CRC calculation end flag bit This bit is set by hardware when CRC calculation is complete, and cleared by software by writing 1. 0: CRC calculation not completed 1: CRC calculation completed **Note:** It is not necessary to reset CRCENDCAL to restart the CRC calculation.
26	AUTOECCDERRDET	Dual-bit error automatic identification detection flag bit This bit is set to 1 when ECC detects a dual-bit error. Software can clear this bit by clearing all AUTOERR2ADDRSTAT in the NVMC_CNVM_AUTOERR2ADDRx register (x = 0…3). 0: Double-bit ECC error not detected. 1: Double-bit ECC error detected.
25	Reserved	The reset value must be maintained.

24	RSERR	Read security error flag bit This bit is set by hardware when accessing a protected secure user address and cleared by writing 1 to it in software. 0: No read security error occurred 1: Read security error occurred
23	RPERR	Read protection error flag bit This bit is set by hardware when accessing an address protected by DCRP or SPC and cleared by writing 1 to it in software. 0: No read protection error occurred 1: Read protection error occurred
22	UPDATESTRERR	Data update start failure error flag bit. The following conditions are checked before data update: 1) Whether CNVM is write-protected (WP). If yes, this bit is set by hardware. 2) Whether the security configuration (secure area / DCRP) of CNVM and ECNVM bank is consistent. If inconsistent, this bit is set by hardware and cleared by writing 1 to it in software. 0: Data update start failure error did not occur 1: Data update start failure error occurred
21	ECNVM_ENDF	ECNVM operation end flag. (Not applicable to GD32H77xxI/GD32H78xxI) This bit is set by hardware when the ECNVM operation is successfully executed. **Note:** This bit functions the same as the ENDF bit in NVMC_ECNVM_STAT. When the data integrity enhancement mode is enabled, after a successful CNVM programming operation, only the ENDF bit in the NVMC_CNVM_STAT register is set. The ENDF bit in the NVMC_ECNVM_STAT register and the ECNVM_ENDF bit in the NVMC_CNVM_STAT register are not set. When the data integrity enhancement mode is enabled, after a successful CNVM programming operation, only the ENDF bit in the NVMC_CNVM_STAT register is set. If PGCHEN is set, hardware will check whether the entire ECNVM programming area is 0xFF before programming. If not, the PGSERR bit in the NVMC_ECNVM_STAT register will be set, while the PGSERR bit in the NVMC_CNVM_STAT register will not be set. After the successful execution of an ECNVM erase (excluding chip erase)/program operation (data integrity enhancement mode disabled), the ENDF bit in the NVMC_ECNVM_STAT register and the ECNVM_ENDF bit in the NVMC_CNVM_STAT register will be set. The software can only clear the ECNVM operation end flag by writing 1 to the ENDF bit in the NVMC_ECNVM_STAT register. After a successful chip erase operation, the ENDF bit in the NVMC_ECNVM_STAT register and the ECNVM_ENDF bit in the NVMC_CNVM_STAT register will be set. The software can clear the ECNVM operation end flag by writing 1 to the ENDF bit

		in the NVMC_ECNVM_STAT register. Alternatively, it can also be cleared by writing 1 to the ECNVM_ENDF bit in the NVMC_CNVM_STAT register. However, this method requires CER to be set for the write to the ECNVM_ENDF bit to take effect, but it is not recommended for users to clear the ECNVM operation end flag using this method.
20	OTASUC	OTA upgrade, check_bootloader verification success flag. (Not applicable to GD32H77xxI/GD32H78xxI) During an OTA upgrade, if the check_bootloader verification succeeds, this bit is set by hardware. This bit is reset on power-on reset, and system reset has no effect on this bit. 0: OTA verification not successful 1: OTA verification successful **Note:** When both OTASUC and OTAFAIL are set to 0, it indicates that no OTA upgrade operation is in progress.
19	OTAFAIL	OTA upgrade, check_bootloader verification failure flag. (Not applicable to GD32H77xxI/GD32H78xxI) During OTA upgrade, if a check_bootloader verification failure occurs, this bit is set by hardware. This bit is reset on power-up, and system reset does not affect this bit. 0: OTA verification not failed 1: OTA verification failed
18	PGSERR	Programming sequence error flag bit This bit is set by hardware and cleared by software by writing 1. 0: No programming sequence error occurred 1: Programming sequence error occurred The following conditions will trigger this bit: 1) Writing a value to the bus without configuring the PG bit. 2) Programming the CNVM area without writing in 64-bit mode. 3) Programming the CNVM area with an address not within the same 256-bit range. **Note:** Data written in 256-bit mode should be sent sequentially.
17	WPERR	Erase/Programming protection error flag bit This bit is set by hardware and cleared by software by writing 1. 0: No erase/programming protection error occurred 1: Erase/Programming protection error occurred
16	ENDF	Operation complete flag bit When the operation is successfully executed, this bit is set to 1 by hardware. Write 1 to clear it to 0 in software.
15:14	NVMVER[1:0]	Nvm 版本 00: GD32H77xxW/GD32H78xxW: 2MB CNVM + 7.5MB ECNVM 01: GD32H77xxP/GD32H78xxP: 2MB CNVM + 3.5MB ECNVM

		10: Reserved 11: GD32H77xxI/GD32H78xxI: 1.625MB CNVM
13:4	Reserved	The reset value must be maintained.
3	CRCBUSY	CNVM CRC Busy Flag This bit is set when CRC calculation is in progress. It is cleared to 0 when the CRC calculation is complete or CRC calculation is disabled. 0: No CRC calculation in progress 1: CRC calculation in progress
2:1	Reserved	The reset value must be maintained.
0	BUSY	CNVM Busy Flag This bit is set to 1 when a CNVM operation is in progress. It is cleared to 0 when the operation is completed or an error occurs.

CNVM Address Register (NVMC_CNVM_ADDR)

Offset address: 0x014
Reset value: 0x0000 0000

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

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

Bits	Fields	Descriptions
31:0	ADDR[31:0]	CNVM Erase Address This bit is set by software. The ADDR bit is the address for the CNVM erase command.

CNVM CRC Result Register (NVMC_CNVM_CRCRESULT)

Offset Address: 0x05C
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
CRC_RESULT[31:16]															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRC_RESULT[15:0]															
r															

Bits	Fields	Descriptions
31:0	CRC_RESULT[31:0]	CRC calculation result: if CRCRERR is set to 1, the value of CRC_RESULT[31:0] will be 0x0000 0000.

CNVM Repair Function Control and Status Register (NVMC_CNVM_REPCS)

GD32H77xxI/GD32H78xxI do not have this register function, do not set.

Offset Address: 0x0C8

Reset value: 0x0000 0000

Note: Reserved bits must maintain the reset value.

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											BLANKMODE	ROBBCNT[3:0]			
											rw	rc_w0			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				REPERRSTAT[3:0]				Reserved	ECNVM_BLANKIE	ECNVM_BLANK	Reserved	REPDONEIE	REPEND	REPSTRRR	REPSTART
				rc_w0					rw	rc_w1		rw	rc_w1	rc_w1	rw

Bits	Fields	Descriptions
31:21	Reserved	The reset value must be maintained.
20	BLANKMODE	ECNVM Data Integrity Enhanced Area Blank Mode 0: During the self-check process, if ECNVM data equals 256'd1, the difference between CNVM data and ECNVM data exceeds 32 bits, CNVM ECC reports an error less than or equal to 1, and CNVM data is not 256'd0 (ECC-corrected value), the ECNVM Data Integrity Enhanced Area is considered blank, and ECNVM_BLANK along with BLANKADDR[15:0] is recorded. In other cases, ECNVM_BLANK and BLANKADDR[15:0] are not recorded. 1: During the self-check process, if ECNVM data equals 256'd1, the difference between CNVM data and ECNVM data exceeds 32 bits, and CNVM ECC reports an error less than or equal to 1, the ECNVM Data Integrity Enhanced Area is considered blank, and ECNVM_BLANK along with BLANKADDR[15:0] is recorded. In other cases, ECNVM_BLANK and BLANKADDR[15:0] are not recorded. **Note:** When ECNVM data equals 256'd1, the difference between CNVM data and ECNVM data exceeds 32 bits, and CNVM ECC reports an error less than or equal to 1, no repair operation is performed. Otherwise, the repair operation is executed.
19:16	ROBBCNT[3:0]	Bad Block Count Bit The number of repair failures during the self-check process. Writing 0 resets the value, and it is automatically reset when the self-check starts.

15:12	Reserved	The reset value must be maintained.
11:8	REPERRSTAT[3:0]	Self-check result, error status. This bit is automatically set to 0 when the repair operation begins, or it can be reset to 0 by software. 0000: No error 0001: Error occurred 0010: Error occurred but fully repaired 0100: Error occurred but corrected with residual ECC-correctable errors 1000: Error occurred but corrected with residual ECC-uncorrectable errors **Note:** Each bit represents a specific meaning and multiple bits can be set simultaneously.
7	Reserved	The reset value must be maintained.
6	ECNVM_BLANKIE	ECNVM data integrity enhanced area content blank interrupt enable bit
5	ECNVM_BLANK	ECNVM data integrity enhanced area content blank flag bit, software can clear this bit by writing 1.
4	Reserved	The reset value must be maintained.
3	REPDONEIE	Self-repair completion interrupt enable bit 0: Disable self-repair completion interrupt 1: Enable self-repair completion interrupt **Note:** If the REPDONEIE bit is set to 1, an interrupt will be generated when the REPEND or REPSTRERR flag is set.
2	REPEND	Self-repair completion flag. When the self-check repair is completed, this bit is set by hardware. Software can clear this bit by writing 1. Errors will not interrupt the repair process and will finish after checking all selected addresses. A new repair operation can only be initiated after software clears this bit. Initiating a new repair operation will clear the results of the previous repair operation.
1	REPSTRERR	Self-check repair configuration error bit Set when the configuration is incorrect, self-check repair is not initiated.
0	REPSTART	Repair start bit When set to 1, initiates self-repair. Compares data consistency between ECNVM and CNVM. ECC is checked simultaneously during the read process. If only a CNVM error occurs, the content of ECNVM is automatically written to CNVM. Software sets this bit, hardware clears it. **Note:** The user can check the REPSTART bit to determine if the self-repair process is busy. When this bit is set to 1, the self-repair operation is in progress. When the self-repair operation is completed, this bit is cleared to 0 by hardware.

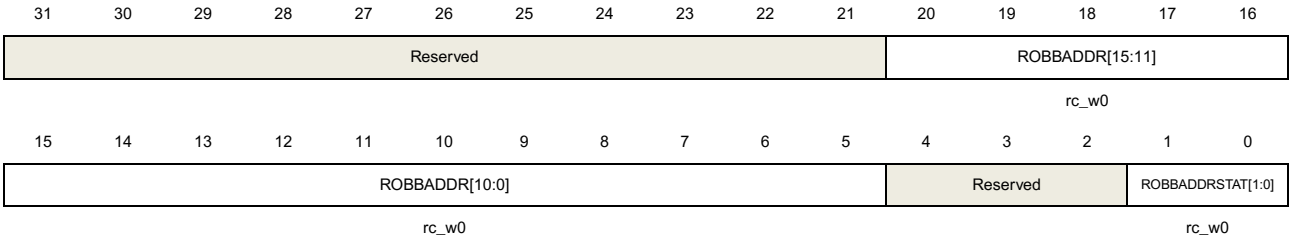
CNVM Read-Only Bad Block Address Register x (NVMC_CNVM_ROBBADDRx) (x = 0…3)

GD32H77xxI/GD32H78xxI do not have this register function, do not set.

Offset Address: 0x0CC + 0x4 * x

Reset value: 0x0800 0000

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



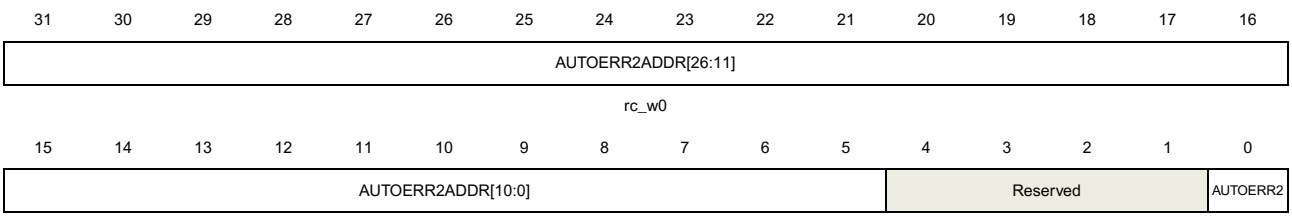
Bits	Fields	Descriptions
31:21	Reserved	The reset value must be maintained.
20:5	ROBBADDR[15:0]	During the self-check process, the register logs the faulty block addresses, recording only the addresses with errors that persist after repair. These addresses are located in the CNVM address area with a granularity of 256 bits. Note: Software can clear ROBBADDR[15:0] by writing 32 zeros to the NVMC_CNVM_ROBBADDRx register. When the address is cleared, the status bits ROBBADDRSTAT[1:0] will also be cleared.
4:2	Reserved	The reset value must be maintained.
1:0	ROBBADDRSTAT[1:0]	ROBBADDR[15:0] status bits 00: No error 01: 1-bit error persists after repair 10: 2-bit error persists after repair 11: There are still 3 or more errors after repair. **Note:** This bit can be cleared individually, and software can clear it by writing 0.

CNVM Auto-Detect ECC Double-Bit Error Address Register x (NVMC_CNVM_AUTOERR2ADDRx) (x = 0…3)

Offset Address: 0x0EC + 0x4 * x

Reset value: 0x0800 0000

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



		ADDRSTAT
rc_w0		rc_w1

Bits	Fields	Descriptions
31:5	AUTOERR2ADDR[26:0]	The address with a double-bit error when the bus reads CNVM. This address is located in the CNVM address range, with a granularity of 256 bits. **Note:** Software can clear AUTOERR2ADDR[26:0] by writing 32 zeros to NVMC_CNVM_AUTOERR2ADDRx. When the address is cleared, the status bit AUTOERR2ADDRSTAT will also be cleared. Only applicable to GD32H77xxI/GD32H78xxI: If an ECC error is detected while reading the option bytes, the value of AUTOERR2ADDR[26:0] will be 0x1FFF_FFE0.
4:1	Reserved	The reset value must be maintained.
0	AUTOERR2ADDRSTAT	AUTOERR2ADDR[26:0] status bit. 0: No error. 1: Double-bit error. **Note:** This bit can be cleared individually, and software can clear it by writing 1.

CNVM OTA Failure Address Register (NVMC_CNVM_OTAFAILADDR)

GD32H77xxI/GD32H78xxI do not have this register function, do not set.

Offset address: 0x128

Reset value: 0x0800 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											OTAFAILADDR[20:16]				
											rc_w0				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OTAFAILADDR[15:0]															
rc_w0															

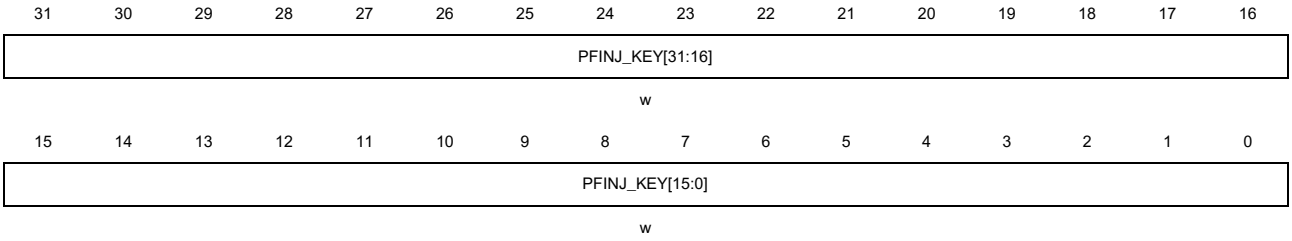
Bits	Fields	Descriptions
31:21	Reserved	The reset value must be maintained.
20:0	OTAFAILADDR[20:0]	Records the address of the last OTA verification failure. **Note:** Software can clear OTAFAILADDR[20:0] by writing 32 zeros to the register NVMC_CNVM_OTAFAILADDR.

CNVM Programming Fault Injection Unlock Register (NVMC_CNVM_PFINJ_KEY)

Offset address: 0x1F0

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:0	PFINJ_KEY[31:0]	NVMC_CNVM_PFINJ unlock register. These bits can only be written by software. Write 0xA55A55AA to PFINJ_KEY[31:0] to unlock the NVMC_CNVM_PFINJ register.

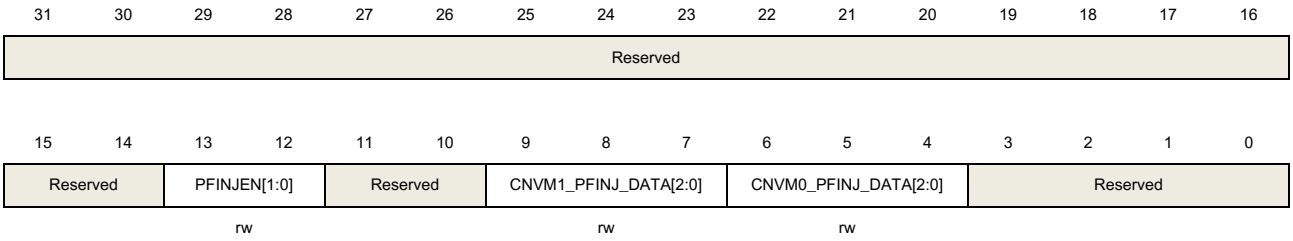
CNVM Programming Fault Injection Register (NVMC_CNVM_PFINJ)

Offset address: 0x1F4

Reset value: 0x0000 0000

Note: The NVMC_CNVM_PFINJ register can only be accessed after writing 0xA55A55AA to the NVMC_CNVM_PFINJ_KEY register.

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



Bits	Fields	Descriptions
31:14	Reserved	The reset value must be maintained.
13:12	PFINJEN[1:0]	Program the fault injection enable bit. Other values: Fault injection disabled. 11: Fault injection enabled.
11:10	Reserved	The reset value must be maintained.
9:7	CNVM1_PFINJ_DATA[2:0]	When PFINJEN[1:0] is set to 0x3 and CNVM1_PFINJ_DATA[*] is set to 1, the corresponding bit of CNVM1_DATAIN[*] will be inverted. **Note:** When CNVM performs a 256-bit programming operation, CNVM1_DATAIN[*] represents bits 128 to 130 to be written to CNVM storage.

6:4	CNVM0_PFINJ_DATA[2:0]	When PFINJEN[1:0] is set to 0x3 and CNVM0_PFINJ_DATA[*] is set to 1, the corresponding bit of CNVM0_DATAIN[*] will be inverted. **Note:** When CNVM performs a 256-bit programming operation, CNVM0_DATAIN[*] represents bits 0 to 2 to be written to CNVM storage.
3:0	Reserved	The reset value must be maintained.

7.4.3. ECNVM Register

ECNVM Unlock Register (NVMC_ECNVM_KEY)

Offset address: 0x150

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
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]	NVMC_ECNVM_CTL unlock register. These bits can only be written by software. Writing the unlock value to KEY[31:0] unlocks the NVMC_ECNVM_CTL register.

ECNVM Control Register (NVMC_ECNVM_CTL)

GD32H77xxI/GD32H78xxI do not have this register function, do not set (except bit31 and bit0).

Offset address: 0x154

Reset value: 0x0000 0011

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
CHIP_REMOVE_EN	Reserved				ECCDETIE	Reserved	RSERRIE	RPERRIE	Reserved				PGSERRIE	WPERRIE	ENDIE
rw					rw		rw	rw					rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								START	PER	Reserved	PGCHEN	MER	SER	PG	LK
								rs	rw		rw	rw	rw	rw	rs

Bits	Fields	Descriptions

31	CHIP_REMOVE_EN	Chip erase enable bit with clear protection. This bit can be set to 1 and cleared to 0 by software only when LK is set to 0. 0: Disable chip erase with clear protection, option bytes cannot be modified. 1: Enable chip erase with clear protection, option bytes can be modified. **Note:** Before setting the CHIP_REMOVE_EN bit in GD32H77xxI/GD32H78xxI, unlock the NVMC_ECNVM_CTL register first.
30:27	Reserved	The reset value must be maintained.
26	ECCDETIE	ECNVM ECC double-bit error detection interrupt enable bit This bit can be set to 1 and cleared to 0 by software only when LK is set to 0. 0: Disable double-bit error detection interrupt 1: Enable double-bit error detection interrupt
25	Reserved	The reset value must be maintained.
24	RSERRIE	Read secure error interrupt enable bit This bit can be set to 1 and cleared to 0 by software only when LK is set to 0. 0: Disable read secure error interrupt 1: Enable read secure error interrupt
23	RPERRIE	Read protection error interrupt enable bit This bit can be set to 1 and cleared to 0 by software only when LK is set to 0. 0: Disable read protection error interrupt 1: Enable read protection error interrupt
22:19	Reserved	The reset value must be maintained.
18	PGSERRIE	Programming sequence error interrupt enable bit This bit can be set to 1 and cleared to 0 by software only when LK is set to 0. 0: Disable programming sequence error interrupt 1: Enable programming sequence error interrupt
17	WPERRIE	Erase/Program protection error interrupt enable bit This bit can be set to 1 and cleared to 0 by software only when LK is set to 0. 0: Disable erase/program protection error interrupt 1: Enable erase/program protection error interrupt
16	ENDIE	Operation complete interrupt enable bit This bit can be set to 1 and cleared to 0 by software only when LK is set to 0. 0: Disable operation complete interrupt 1: Enable Operation End Interrupt
15:8	Reserved	The reset value must be maintained.
7	START	Send the erase command bit to NVMC. This bit can be set to 1 by software to send an erase command to NVMC only when LK is set to 0. This bit is cleared to 0 by hardware when the BUSY bit is cleared to 0.

6	PER	Page Erase Command Bit This bit can be set to 1 and cleared to 0 by software only when LK is set to 0. 0: No effect 1: Page Erase Command If both page erase and sector erase are requested simultaneously, the sector erase will override the page erase operation. This bit is used to erase one page (4KB) in the main ECNVM area.
5	Reserved	The reset value must be maintained.
4	PGCHEN	Programming Area Check Enable Bit This bit can be set to 1 and cleared to 0 by software only when LK is set to 0. 0: Do not check if the programming area data is all 0xFF before programming. 1: Check if the programming area data is all 0xFF before programming. **Note:** If this bit is set and the ECNVM programming area is not all 0xFF, the PGSERR bit in the NVMC_ECNVM_STAT register will be set, and the programming operation will be invalid. The programming area check function is valid for 8-byte and 32-byte programming.
3	MER	Mass Erase Command Bit This bit can be set to 1 and cleared to 0 by software only when LK is set to 0. 0: No effect 1: Mass Erase Command If both mass erase and sector erase are requested simultaneously, the mass erase will replace the sector erase operation. This bit is used to erase the entire ECNVM memory area.
2	SER	Sector Erase Command Bit This bit can be set to 1 and cleared to 0 by software only when LK is set to 0. 0: No effect 1: Sector Erase Command If both mass erase and sector erase are requested simultaneously, the mass erase will replace the sector erase operation. This bit is used to erase a sector of the ECNVM memory area.
1	PG	ECNVM Block Programming Command Bit This bit can be set to 1 and cleared to 0 by software only when LK is set to 0. 0: No effect 1: ECNVM Block Programming Command
0	LK	Lock flag bit of NVMC_ECNVM_CTL register When the correct sequence is written to the NVMC_ECNVM_KEY register, this bit is cleared to 0 by hardware. This bit can be set to 1 by software.

Note: This register must be in the reset state after the corresponding ECNVM operation is completed.

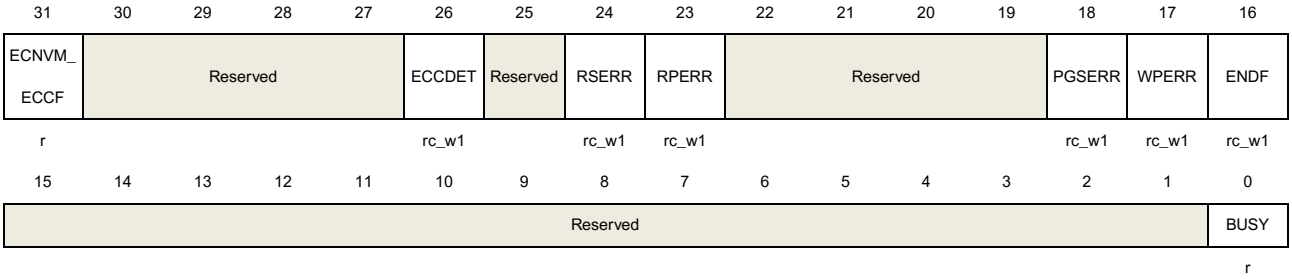
ECNVM Status Register (NVMC_ECNVM_STAT)

GD32H77xxI/GD32H78xxI do not have this register function, do not set.

Offset address: 0x158

Reset value: 0xX000 0000

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



Bits	Fields	Descriptions
31	ECNVM_ECCF	ECNVM ECC function flag bit This bit indicates whether the ECC function is enabled or disabled. 0: ECNVM ECC function disabled 1: ECNVM ECC function enabled When ECC is enabled, write operations must be 64-bit.
30:27	Reserved	The reset value must be maintained.
26	ECCDET	Double error detection flag bit When a double error is detected, this bit is set by hardware and cleared to 0 by writing 1 in software. 0: No double ECC error detected 1: Double ECC error detected
25	Reserved	The reset value must be maintained.
24	RSERR	Read security error flag bit This bit is set by hardware and cleared by software by writing 1. 0: No read security error occurred 1: Read security error occurred
23	RPERR	Read protection error flag bit This bit is set by hardware and cleared by software by writing 1. 0: No read protection error occurred 1: Read protection error occurred
22:19	Reserved	The reset value must be maintained.
18	PGSERR	Programming sequence error flag bit This bit is set by hardware and cleared by software by writing 1. 0: No programming sequence error occurred

		1: Programming sequence error occurred
17	WPERR	Erase/Programming protection error flag bit This bit is set by hardware and cleared by software by writing 1. 0: No erase/programming protection error occurred 1: Erase/Programming protection error occurred
16	ENDF	Operation complete flag bit When the operation is successfully executed, this bit is set to 1 by hardware. Write 1 to clear it to 0 in software.
15:1	Reserved	The reset value must be maintained.
0	BUSY	ECNVM Busy Flag This bit is set to 1 when an ECNVM operation is in progress. It is cleared to 0 when the operation is completed or an error occurs.

ECNVM Address Register (NVMC_ECNVM_ADDR)

GD32H77xxI/GD32H78xxI do not have this register function, do not set.

Offset address: 0x15C

Reset value: 0x0000 0000

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

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

Bits	Fields	Descriptions
31:0	ADDR[31:0]	ECNVM Erase Address This bit is set by software. The ADDR bit is the address for the ECNVM Erase command.

ECNVM ECC Error Address Register (NVMC_ECNVM_ECCADDR)

GD32H77xxI/GD32H78xxI do not have this register function, do not set.

Offset address: 0x160

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

ECCADDR [15:0]
r

Bits	Fields	Descriptions
31:0	ECCADDR[31:0]	The starting address of the read operation when an ECC error is detected. Single Read Operation: When an ECC error (double-bit error detection) is detected, ECCADDR[31:0] records the 256-bit aligned starting address. For example, if the error occurs at 0x0820_001F, the recorded address is 0x0820_0000. When reading option bytes: If an ECC double-bit error is detected during the option byte read, ECCADDR[31:0] is set to 0x1FFF_FFFE. When the error flag (ECCDET) in the NVMC_ECNVM_STAT register is reset, ECCADDR[31:0] will also be reset. When the ECC error flag is set to 1, this register will be updated, meaning ECCADDR[31:0] will store the starting address of the latest read operation that generated an ECC error, overwriting the previous address.

7.4.4. OTP register

Do not configure registers related to ECNVM, repair, and OTA functions.

OTP control register (NVMC_OTP_CTL)

Offset address: 0x200

Reset value: 0xXX00 XX00

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
AES_KEY_CRC[7:0]								Reserved							OTP_REPSTART
r															rs

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DP_CRC[7:0]								Reserved			ENDIE	PGSERRIE	MULTI_DATAERRIE	DATAERRIE	PGLKERRIE
r											rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:24	AES_KEY_CRC[7:0]	8-bit CRC Calculation Result of the AES Key Bits This bit field is used to verify the value of OTP_AES_KEY. The CRC calculation uses the standard CRC-8-CCITT algorithm: X^8 + X^2 + X + 1 (as per the existing algorithm). If AES_KEY_ON is 0, after system reset, the 128-bit OTP_AES_KEY value will be automatically read by the MCU from OTP storage. The CRC computation result is generated after system reset and OTP read completion. If AES_KEY_ON is 1, the read value of this register is 0. **Note**: When AES_KEY_ON is set to 1, regardless of whether a value has been

		written to the OTP_AES_KEY register, the result of AES_KEY_CRC[7:0] is always 0.
23:17	Reserved	The reset value must be maintained.
16	OTP_REPSTART	OTP repair trigger signal When the LK of NVMC_CNVM_CTL is set to 0, this bit is set to 1 by software and cleared to 0 by hardware at the end of the repair. 0: OTP repair not triggered 1: OTP repair enable triggered, starting OTP region repair. Upon completion, this signal is cleared to 0 by hardware.
15:8	DP_CRC[7:0]	8-bit CRC computation result of debug password data This bit field is used to validate the value of DBGPSWD_DATA. CRC computation uses the CRC-8 algorithm: X8 + X2 + X + 1 (based on the existing algorithm). If DPLK is 0, after system reset, the 64-bit debug password data is automatically read by the MCU from OTP storage. The CRC computation result is generated after system reset and OTP read completion. If DPLK is 1, the read value of this register is 0. **Note:** When DPLK is set to 1, regardless of whether a value has been written to the DBGPSWD_DATA register, the result of DP_CRC[7:0] is always 0.
7:5	Reserved	The reset value must be maintained.
4	ENDIE	OTP operation completed interrupt enable bit This bit can be set to 1 and reset to 0 by software only when LK in NVMC_CNVM_CTL is set to 0. 0: Disable OTP operation completed interrupt 1: Enable OTP operation completed interrupt
3	PGSERRIE	OTP programming sequence error interrupt enable bit This bit can be set to 1 and reset to 0 by software only when LK in NVMC_CNVM_CTL is set to 0. 0: Disable OTP programming sequence error interrupt 1: Enable OTP programming sequence error interrupt
2	MULTI_DATAERRIE	Multi-bit redundant data error interrupt enable bit This bit can be set to 1 and reset to 0 by software only when LK in NVMC_CNVM_CTL is set to 0. 0: Disable multi-bit redundant data error interrupt 1: Enable multi-bit redundant data error interrupt
1	DATAERRIE	Data error interrupt enable bit This bit can be set to 1 and reset to 0 by software only when LK in NVMC_CNVM_CTL is set to 0. 0: Disable redundant data error interrupt

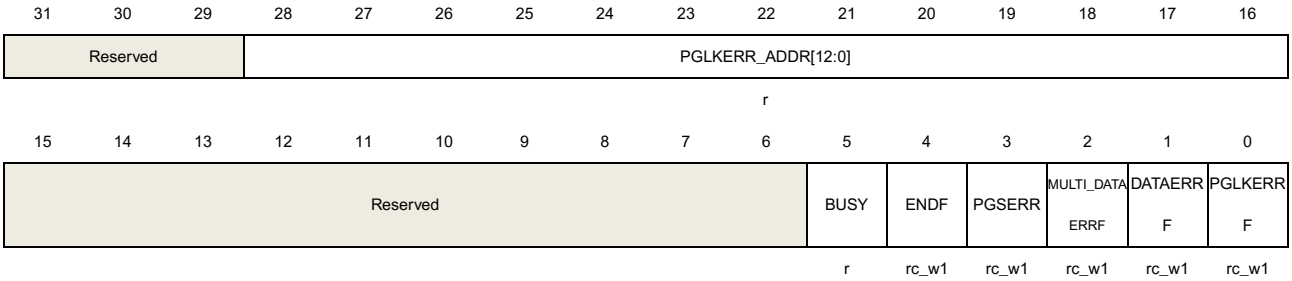
		1: Enable redundant data error interrupt
0	PGLKERRIE	Programming error interrupt enable bit in locked state This bit can be set to 1 and reset to 0 by software only when LK in NVMC_CNVM_CTL is set to 0. 0: Disable write error interrupt in locked state 1: Enable write error interrupt in locked state

OTP status register (NVMC_OTP_STAT)

Offset address: 0x204

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:29	Reserved	The reset value must be maintained.
28:16	PGLKERR_ADDR[12:0]	When PGLKERRF is set to 1, this bit records the lower 13 bits of the logical address corresponding to the error. PGLKERR_ADDR[12:0] is updated when a new error occurs.
15:6	Reserved	The reset value must be maintained.
5	BUSY	OTP busy flag bit This bit is set to 1 when an operation is processing OTP or OTP repair. It is cleared to 0 when the operation is completed or an error occurs.
4	ENDF	Operation complete flag bit This bit is set by hardware when a repair or programming operation is successfully performed, and cleared to 0 by software writing 1.
3	PGSERR	OTP programming sequence error flag bit This bit is set by hardware when a programming sequence error occurs and cleared to 0 by software writing 1. 0: No OTP programming sequence error occurred 1: OTP programming sequence error occurred The following conditions will trigger this bit: (1) Writing a value to the bus without configuring the PG bit. (2) Only single 16-bit writes are supported and must be performed at OTP address

0x1FFF_0000 + 0x20 * x (x=0 ... 207). Otherwise, PGSERR is set.

2	MULTI_DATAERRF	Multi-bit redundancy data error flag This bit is set by hardware when a multi-bit redundancy data error occurs and is cleared by software by writing 1. 0: No multi-bit redundancy data error occurred 1: A multi-bit redundancy data error occurred
1	DATAERRF	Redundancy data error flag This bit is set by hardware when a redundancy data error occurs and is cleared by software by writing 1. 0: No redundancy data error occurred 1: A redundancy data error occurred
0	PGLKERRF	Programming error flag in locked state This bit is set by hardware when a programming error occurs in the locked state and is cleared by software by writing 1. 0: No write error occurred in the locked state 1: A write error occurred in locked state.

OTP user control register 0 (NVMC_OTP_USER_CTL0)

Offset address: 0x20C
Reset value: 0xXXXX XXXX, load the value from the OTP storage unit after reset.

(1) NVMC_OTP_USER_CTLx parameters have two sets of registers: hardware registers and software registers. Users can read and write hardware registers based on the AXI bus and read software registers based on the AHB bus. The following register descriptions are for software registers on the AHB bus.

(2) In the NVMC_OTP_USER_CTLx parameters, the software registers DCRP_EFFCTL, SCR_EFFCTL, SCR, SPC_H, SPC_L, NDBG, JTAGNSW, SCR_AREA_START, SCR_AREA_END, DCRP_AREA_START, DCRP_AREA_END, ECNVM_BANK1_SCR_EN, ECNVM_BANK0_SCR_EN, CNVM_SCR_EN, ECNVM_BANK1_DCRP_EN, ECNVM_BANK0_DCRP_EN, CNVM_DCRP_EN, and all lock bits are reset only during a power reset. After a system reset and power reset, data is loaded. The values of other registers are reset during a system reset and data is loaded after the system reset. Additionally, the configurations of NDBG and JTAGNSW take effect after a power reset.

Note: JTAGNSW configured from 0 to 1 takes effect after a power reset; JTAGNSW configured from 1 to 0 takes effect after a system reset.

(3) In the NVMC_OTP_USER_CTLx parameter, once VFALL, VFIMG, DISLFI, DCRP_EFFCTL, SCR_EFFCTL, SCR, SPC_H, SPC_L, NDBG, ECNVM_BANK1_SCR_EN, ECNVM_BANK0_SCR_EN, CNVM_SCR_EN, ECNVM_BANK1_DCRP_EN, ECNVM_BANK0_DCRP_EN, and CNVM_DCRP_EN are configured to 1, their corresponding lock bits will also automatically be set to 1. When reading the software registers, they will

display as 1, but the values stored in OTP will not be modified. The effective values are those in the software registers. Apart from the configuration bits mentioned above, the remaining configuration bits can revert to 0 after writing 1 if not locked; however, if locked, they cannot be modified.

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
DCRP_EFFCTL_LK	SCR_EFFCTL_LK	SCR_LK	SPC_H_LK	SPC_L_LK	JTAGNSW_LK	NDBG_LK[1:0]	DCRP_EFFCTL	SCR_EFFCTL	SCR	SPC_H	SPC_L	JTAGNSW	NDBG[1:0]
r	r	r	r	r	r	r	r	r	r	r	r	r	r

15:11	10	9	8	7:6	5:4	3	2	1	0
Reserved	VFALL_LK	VFIMG_LK	DISLFI_LK	Reserved			VFALL	VFIMG	DISLFI
	r	r	r				r	r	r

Bits	Fields	Descriptions
31	DCRP_EFFCTL_LK	DCRP_EFFCTL lock bit This bit is factory set to 0. 0: Unlock DCRP_EFFCTL, allowing modification of DCRP_EFFCTL. 1: Lock DCRP_EFFCTL, preventing modification of DCRP_EFFCTL.
30	SCR_EFFCTL_LK	SCR_EFFCTL lock bit This bit is factory set to 0. 0: Unlock SCR_EFFCTL, allowing modification of SCR_EFFCTL. 1: Lock SCR_EFFCTL, preventing modification of SCR_EFFCTL.
29	SCR_LK	SCR lock bit This bit is factory set to 0. 0: Unlock SCR, allowing modification of SCR. 1: Lock SCR, preventing modification of SCR.
28	SPC_H_LK	SPC_H lock bit This bit is factory set to 0. 0: Unlock SPC_H, SPC_H can be modified. 1: Lock SPC_H, SPC_H cannot be modified.
27	SPC_L_LK	SPC_L lock bit This bit is factory set to 0. 0: Unlock SPC_L, SPC_L can be modified. 1: Lock SPC_L, SPC_L cannot be modified.
26	JTAGNSW_LK	JTAGNSW lock bit This bit is factory set to 0. 0: Unlock JTAGNSW, JTAGNSW can be modified. 1: Lock JTAGNSW, JTAGNSW cannot be modified.
25:24	NDBG_LK[1:0]	NDBG lock bit

		This bit is factory set to 0. 0: Unlock NDBG, NDBG can be modified. 1: Lock NDBG, NDBG cannot be modified.
23	DCRP_EFFCTL	DCRP area enable control bit 0: DCRP region configuration takes effect in the option bytes. 1: DCRP region configuration takes effect in OTP.
22	SCR_EFFCTL	SCR region enable control bit 0: SCR region configuration takes effect in the option bytes. 1: SCR region configuration takes effect in OTP.
21	SCR	Secure mode enable This bit is factory set to 0. 0: Disable secure mode. 1: Enable secure mode. **Note:** Secure mode is enabled as long as this bit or the SCR bit in the NVMC_OBSTAT0_EFT register is set to 1.
20	SPC_H	Configure the security protection level to high protection level. This bit is factory set to 0. **Note:** If both SPC_H and SPC_L bits in OTP are set to 1, SPC is configured to high protection level.
19	SPC_L	Configure the security protection to low protection level. This bit is factory set to 0. **Note:** If SPC_L in OTP is set to 1, downgrading SPC protection level to unprotected state is disabled. If both SPC_H and SPC_L in OTP are set to 1, SPC is at a high protection level.
18	JTAGNSW	SW or JTAG debugger selection This bit is factory set to 0. 0: SW 1: JTAG **Note:** When NDBG[1:0] is selected as no debugging function, the JTAGNSW bit is invalid, and the debugging function is disabled.
17:16	NDBG[1:0]	Debugging permission setting The factory value of these bits are 0. 00: Normal JTAG (only valid when JTAGNSW bit is 1, otherwise SW debugger is selected) 01: Secure JTAG (only valid when JTAGNSW bit is 1, otherwise SW debugger is selected) 10~11: No debug (debug function is closed regardless of the JTAGNSW bit)
15:11	Reserved	The reset value must be maintained.
10	VFALL_LK	VFALL lock bit

		This bit is factory set to 0. 0: Unlock VFALL, modification of VFALL is allowed. 1: Lock VFALL, VFALL cannot be modified.
9	VFIMG_LK	VFIMG lock bit This bit is factory set to 0. 0: Unlock VFIMG, VFIMG can be modified. 1: Lock VFIMG, VFIMG cannot be modified.
8	DISLFI_LK	DISLFI lock bit This bit is factory set to 0. 0: Unlock DISLFI, DISLFI can be modified. 1: Lock DISLFI, DISLFI cannot be modified.
7:3	Reserved	The reset value must be maintained.
2	VFALL	Firmware image (image must include certificate) authentication enable bit The factory default value of this bit is 0. 0: Disable firmware image authentication 1: Enable firmware image authentication
1	VFIMG	Verify firmware image enable bit This bit is factory set to 0. 0: Disable firmware image authentication function. 1: Enable firmware image authentication function.
0	DISLFI	Firmware installation authorization function setting. This bit is factory set to 0. 0: Enable authorized firmware installation. 1: Disable authorized firmware installation.

OTP User Control Register 1 (NVMC_OTP_USER_CTL1)

Offset Address: 0x210

Reset value: 0xXXXX XXXX, load the value from the OTP storage unit after reset.

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	ECNVM_DIE_DISABLE_LK	BBREPL_DISABLE_LK	SWAP_DISABLE_LK	DIE_MODE_LOCK_LK	DIE_EN_LK	REPAIR_DISABLE_LK	UPDATE_DISABLE_LK	Reserved	ECNVM_DIE_DISABLE	BBREPL_DISABLE	SWAP_DISABLE	DIE_MODE_LOCK	DIE_EN	REPAIR_DISABLE	UPDATE_DISABLE
	r	r	r	r	r	r	r		r	r	r	r	r	r	r

Bits	Fields	Descriptions

31:15	Reserved	The reset value must be maintained.
14	ECNVM_DIE_DISABLE_LK	ECNVM_DIE_DISABLE lock bit. This bit is factory set to 0. 0: Unlock ECNVM_DIE_DISABLE, allowing modification of ECNVM_DIE_DISABLE. 1: Lock ECNVM_DIE_DISABLE, preventing modification of ECNVM_DIE_DISABLE.
13	BBREPL_DISABLE_LK	BBREPL_DISABLE lock bit. This bit is factory set to 0. 0: Unlock BBREPL_DISABLE, allowing modification of BBREPL_DISABLE. 1: Lock BBREPL_DISABLE, cannot modify BBREPL_DISABLE.
12	SWAP_DISABLE_LK	SWAP_DISABLE lock bit This bit is factory set to 0. 0: Unlock SWAP_DISABLE, SWAP_DISABLE can be modified. 1: Lock SWAP_DISABLE, cannot modify SWAP_DISABLE.
11	DIE_MODE_LOCK_LK	DIE_MODE_LOCK lock bit This bit is factory set to 0. 0: Unlock DIE_MODE_LOCK, DIE_MODE_LOCK can be modified. 1: Lock DIE_MODE_LOCK, cannot modify DIE_MODE_LOCK.
10	DIE_EN_LK	DIE_EN lock bit This bit is factory set to 0. 0: Unlock DIE_EN, DIE_EN can be modified. 1: Lock DIE_EN, cannot modify DIE_EN.
9	REPAIR_DISABLE_LK	REPAIR_DISABLE lock bit This bit is factory set to 0. 0: Unlock REPAIR_DISABLE, can modify REPAIR_DISABLE. 1: Lock REPAIR_DISABLE, cannot modify REPAIR_DISABLE.
8	UPDATE_DISABLE_LK	UPDATE_DISABLE lock position This bit is factory set to 0. 0: Unlock UPDATE_DISABLE, can modify UPDATE_DISABLE. 1: Lock UPDATE_DISABLE, cannot modify UPDATE_DISABLE.
7	Reserved	The reset value must be maintained.
6	ECNVM_DIE_DISABLE	ECNVM data integrity enhancement disable bit This bit is factory set to 0. 0: No impact on ECNVM data integrity enhancement functionality. 1: Disable ECNVM data integrity enhancement functionality. **Note:** When this bit is set to 1, function bits associated with CNVM and ECNVM will be disabled, including self-repair and data integrity enhancement functionality.
5	BBREPL_DISABLE	Force disable bad block replacement functionality bit

This bit is factory set to 0.
0: No impact on bad block replacement functionality.
1: Force disable bad block replacement function
Note: When this bit is set to 1, regardless of the BBREPLEN option byte configuration, the final effective value is disable.

4	SWAP_DISABLE	Disable ECNVM data integrity enhancement area swap function bit This bit is factory set to 0. 0: No effect on ECNVM data integrity enhancement area swap function 1: Disable ECNVM data integrity enhancement area swap function **Note:** When this bit is set to 1, regardless of the SWAP option byte configuration, the final effective value is disable.
3	DIE_MODE_LOCK	Lock the current data integrity enhancement mode configuration This bit is factory set to 0. 0: No effect on the current data integrity enhancement mode configuration 1: Lock the current data integrity enhancement mode configuration **Note:** When this bit is set to 1, the DIEEN and DIESZ in the option bytes and DIE_EN in the OTP will be locked and cannot be modified. If an attempt is made to modify DIE_EN, PGLKERRF will be set.
2	DIE_EN	Force enable data integrity enhancement mode This bit is factory set to 0. 0: No effect on the data integrity enhancement mode 1: Force enable Data Integrity Enhancement mode **Note:** The final effective value is obtained by performing an OR operation between DIE_EN and DIEEN in the option bytes.
1	REPAIR_DISABLE	Disable self-check and repair functionality This bit is factory set to 0. 0: No impact on self-check and repair functionality 1: Disable self-check and repair functionality **Note:** When this bit is set to 1, the final effective value is disable regardless of the configuration of option bytes PORRSTREPDIS, SYSRSTREPDIS, and HWREPMOD.
0	UPDATE_DISABLE	Disable OTA upgrade data update functionality This bit is factory set to 0. 0: No impact on OTA upgrade data update functionality 1: Disable OTA upgrade data update functionality **Note:** When this bit is set to 1, the final effective value is disable regardless of the configuration of the FW_UPDATE option byte.

OTP User Control Register 2 (NVMC_OTP_USER_CTL2)

Offset Address: 0x214

Reset value: 0xXXXX XXXX, load the value from the OTP storage unit after reset.

Note: For different devices, the registers related to SCR and DCRP in OTP_USER_CTL2 and OTP_USER_CTL3 have different restrictions:

（1） SCR_AREA_START[8:0], SCR_AREA_END[8:0] in NVMC_OTP_USER_CTL2 and DCRP_AREA_START[8:0], DCRP_AREA_END[8:0] in NVMC_OTP_USER_CTL3: In GD32H77xxI/GD32H78xxI, the range of these bits is reduced. For GD32H77xxI/GD32H78xxI, the range is 0~0x19F. If configured beyond the maximum value of the range, it will be fixed to the maximum value within the range.

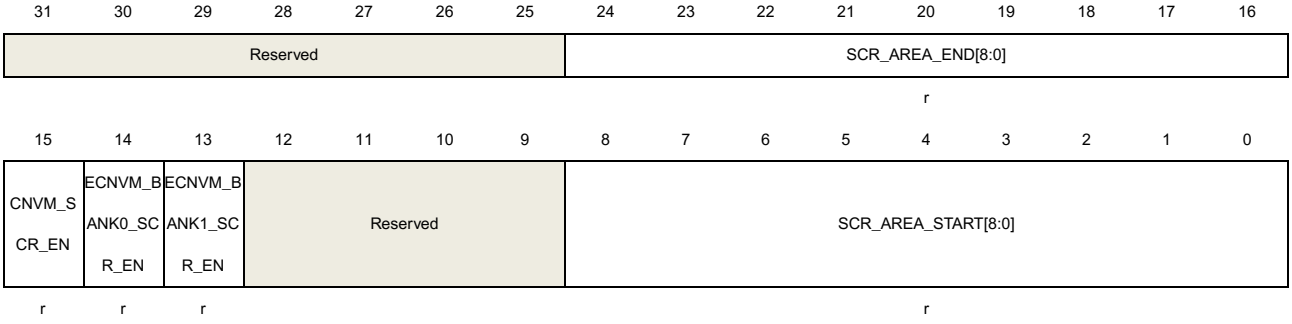
（2） ECNVM_BANK1_SCR_EN, ECNVM_BANK1_DCRP_EN: In GD32H77xxP/GD32H78xxP/GD32H77xxI/GD32H78xxI, if the OTA function is not available, all the above bits in the OTP output will be 0.

（3） ECNVM_BANK0_SCR_EN, ECNVM_BANK0_DCRP_EN: In GD32H77xxI/GD32H78xxI, without data integrity enhancement function, all the above bits in the OTP output will be 0.

（4） CNVM_SCR_EN, CNVM_DCRP_EN: In GD32H77xxI/GD32H78xxI, without data integrity enhancement function, all the above bits in the OTP and option bytes are fixed to 1.

For the above restrictions on different devices, when determining programming errors under the locked state for SCR and DCRP-related registers, it is based on the parameters in the actual AHB registers and the lock bits in the actual AHB registers. The values stored in the OTP are not considered. The values of SCR and DCRP-related registers read from the AHB software registers are the effective values.

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



Bits	Fields	Descriptions
31:25	Reserved	The reset value must be maintained.
24:16	SCR_AREA_END[8:0]	Secure user area end address bit This bit is factory set to 0. This bitfield contains the last 4K byte block of the secure user area. The secure user area is defined in OTP with a 4KB granularity. The last byte address of the area = (SCR_AREA_END[8:0] + 1) * 4096 - 1 + base address.

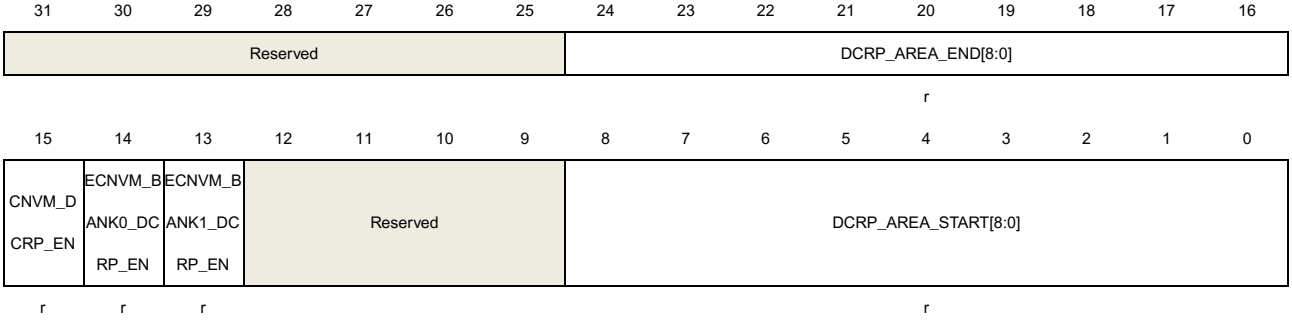
		When CNVM_SCR_EN is set to 1, the base address is the starting address of the CNVM region. When ECNVM_BANK0_SCR_EN is set to 1, the base address is the starting address of the bank0 region. When ECNVM_BANK1_SCR_EN is set to 1, the base address is the starting address of bank1 in the bank1 region. If SCR_AREA_END[8:0] and SCR_AREA_START[8:0] are both 0, the secure user area is undefined. If SCR_AREA_END[8:0] equals SCR_AREA_START[8:0] and is greater than 0, the entire CNVM / bank0 / bank1 is the secure user area. If SCR_AREA_END[8:0] is less than SCR_AREA_START[8:0], the secure user area is empty.
15	CNVM_SCR_EN	CNVM area SCR function enable configuration bit. GD32H77xxW/GD32H78xxW/GD32H77xxP/GD32H78xxP: The factory default value of this bit is 0. GD32H77xxI/GD32H78xxI: The factory default value of this bit is 1. 0: Disable CNVM area SCR function. 1: Enable CNVM area SCR function.
14	ECNVM_BANK0_SCR_EN	ECNVM bank0 area SCR function enable configuration bit. This bit is factory set to 0. 0: Disable the SCR function of ECNVM bank0 region. 1: Enable the SCR function of ECNVM bank0 region.
13	ECNVM_BANK1_SCR_EN	SCR function enable configuration bit for ECNVM bank1 region. This bit is factory set to 0. 0: Disable the SCR function of ECNVM bank1 region. 1: Enable the SCR function of ECNVM bank1 region.
12:9	Reserved	The reset value must be maintained.
8:0	SCR_AREA_START[8:0]	Start address bit of the secure user area. This bit is factory set to 0. This bit field contains the starting 4KB block of the secure user area. The secure user area is defined in OTP with a 4KB granularity. The address of the first byte in the region = SCR_AREA_START[8:0] * 4096 + base address. When CNVM_SCR_EN is set to 1, the base address is the starting address of the CNVM region. When ECNVM_BANK0_SCR_EN is set to 1, the base address is the starting address of the bank0 region. When ECNVM_BANK1_SCR_EN is set to 1, the base address is the starting address of bank1 in the bank1 region. If SCR_AREA_END[8:0] and SCR_AREA_START[8:0] are both 0, the secure user area is undefined.

If SCR_AREA_END[8:0] equals SCR_AREA_START[8:0] and is greater than 0, the entire CNVM / bank0 / bank1 is the secure user area.
If SCR_AREA_END[8:0] is less than SCR_AREA_START[8:0], the secure user area is invalid.

OTP User Control Register 3 (NVMC_OTP_USER_CTL3)

Offset Address: 0x218
Reset value: 0xXXXX XXXX, load the value from the OTP storage unit after reset.

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



Bits	Fields	Descriptions
31:25	Reserved	The reset value must be maintained.
24:16	DCRP_AREA_END[8:0]	DCRP domain end address bits This bit is factory set to 0. This bitfield contains the last 4KB block of the DCRP. DCRP is defined in OTP with a granularity of 4K bytes. The last byte address of the region = (DCRP_AREA_END[8:0] + 1) * 4096 - 1 + base address When CNVM_DCRP_EN is set to 1, the base address is the CNVM start address in the CNVM area. When ECNVM_BANK0_DCRP_EN is set to 1, the base address is the bank0 start address in the bank0 area. When ECNVM_BANK1_DCRP_EN is set to 1, the base address is the bank1 start address in the bank1 area. If both DCRP_AREA_END[8:0] and DCRP_AREA_START[8:0] are 0, DCRP is undefined. If DCRP_AREA_END[8:0] equals DCRP_AREA_START[8:0] and is greater than 0, the entire CNVM / bank0 / bank1 is DCRP. If DCRP_AREA_END[8:0] is less than DCRP_AREA_START[8:0], DCRP is empty.
15	CNVM_DCRP_EN	CNVM area DCRP function enable configuration bit GD32H77xxW/GD32H78xxW/GD32H77xxP/GD32H78xxP: The factory default value of this bit is 0.

		GD32H77xxI/GD32H78xxI: The factory default value of this bit is 1. 0: Disable CNVM area DCRP function 1: Enable CNVM area DCRP function
14	ECNVM_BANK0_DCRP_EN	ECNVM bank0 area DCRP function enable configuration bit This bit is factory set to 0. 0: Disable ECNVM bank0 area DCRP function 1: Enable ECNVM bank0 area DCRP function
13	ECNVM_BANK1_DCRP_EN	ECNVM bank1 area DCRP function enable configuration bit This bit is factory set to 0. 0: Disable ECNVM bank1 area DCRP function 1: Enable ECNVM bank1 area DCRP function
12:9	Reserved	The reset value must be maintained.
8:0	DCRP_AREA_START[8:0]	DCRP field start address bit This bit is factory set to 0. This field contains the starting 4K byte block of DCRP. DCRP is defined in OTP with a granularity of 4K bytes. Region first byte address = DCRP_AREA_START[8:0] * 4096 + base address When CNVM_DCRP_EN is set to 1, the base address is the CNVM start address in the CNVM area. When ECNVM_BANK0_DCRP_EN is set to 1, the base address is the bank0 start address in the bank0 area. When ECNVM_BANK1_DCRP_EN is set to 1, the base address is the bank1 start address in the bank1 area. If both DCRP_AREA_END[8:0] and DCRP_AREA_START[8:0] are 0, DCRP is undefined. If DCRP_AREA_END[8:0] equals DCRP_AREA_START[8:0] and is greater than 0, the entire CNVM / bank0 / bank1 is DCRP. If DCRP_AREA_END[8:0] is less than DCRP_AREA_START[8:0], DCRP is invalid.

OTP User Control Register 4 (NVMC_OTP_USER_CTL4)

Offset address: 0x21C

Reset value: 0xXXXX XXXX, load the value from the OTP storage unit after reset.

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			CNVM_DCRP_ENLK	ECNVM_BANK0_DC	ECNVM_BANK1_DC	Reserved			CNVM_SCR_ENLK	ECNVM_BANK0_SC	ECNVM_BANK1_SC	AES_KEY_ON	DCRP_AREALK	SCR_AREALK	DPLK

		RP_ENLK	RP_ENLK			R_ENLK	R_ENLK					
	r	r	r		r	r	r	r	r	r	r	

Bits	Fields	Descriptions
31:13	Reserved	The reset value must be maintained.
12	CNVM_DCRP_ENLK	CNVM_DCRP_EN lock bit in the NVMC_OTP_USER_CTL3 register GD32H77xxW/GD32H78xxW/GD32H77xxP/GD32H78xxP: The factory default value of this bit is 0. GD32H77xxI/GD32H78xxI: The factory default value of this bit is 1. 0: Unlock CNVM_DCRP_EN, can modify CNVM_DCRP_EN. 1: Lock CNVM_DCRP_EN, cannot modify CNVM_DCRP_EN.
11	ECNVM_BANK0_DCRP_ENLK	ECNVM_BANK0_DCRP_EN lock bit in NVMC_OTP_USER_CTL3 register. This bit is factory set to 0. 0: Unlock ECNVM_BANK0_DCRP_EN, can modify ECNVM_BANK0_DCRP_EN. 1: Lock ECNVM_BANK0_DCRP_EN, cannot modify ECNVM_BANK0_DCRP_EN.
10	ECNVM_BANK1_DCRP_ENLK	ECNVM_BANK1_DCRP_EN lock bit in NVMC_OTP_USER_CTL3 register. This bit is factory set to 0. 0: Unlock ECNVM_BANK1_DCRP_EN, can modify ECNVM_BANK1_DCRP_EN. 1: Lock ECNVM_BANK1_DCRP_EN, cannot modify ECNVM_BANK1_DCRP_EN.
9:7	Reserved	The reset value must be maintained.
6	CNVM_SCR_ENLK	CNVM_SCR_EN lock bit in NVMC_OTP_USER_CTL2 register. GD32H77xxW/GD32H78xxW/GD32H77xxP/GD32H78xxP: The factory default value of this bit is 0. GD32H77xxI/GD32H78xxI: The factory default value of this bit is 1. 0: Unlock CNVM_SCR_EN, can modify CNVM_SCR_EN. 1: Lock CNVM_SCR_EN, unable to modify CNVM_SCR_EN.
5	ECNVM_BANK0_SCR_ENLK	ECNVM_BANK0_SCR_EN lock bit in NVMC_OTP_USER_CTL2 register. This bit is factory set to 0. 0: Unlock ECNVM_BANK0_SCR_EN, able to modify ECNVM_BANK0_SCR_EN. 1: Lock ECNVM_BANK0_SCR_EN, unable to modify ECNVM_BANK0_SCR_EN.
4	ECNVM_BANK1_SCR_ENLK	ECNVM_BANK1_SCR_EN lock bit in NVMC_OTP_USER_CTL2 register. This bit is factory set to 0. 0: Unlock ECNVM_BANK1_SCR_EN, able to modify ECNVM_BANK1_SCR_EN. 1: Lock ECNVM_BANK1_SCR_EN, unable to modify ECNVM_BANK1_SCR_EN.
3	AES_KEY_ON	OTP_AES_KEY lock bit. This bit is factory set to 0. 0: Unlock OTP_AES_KEY, able to write the value of OTP_AES_KEY. 1: Lock OTP_AES_KEY, unable to write the value of OTP_AES_KEY.
2	DCRP_AREALK	DCRP region address lock bit.

		This bit is factory set to 0. 0: Unlock the 16 bits of DCRP_AREA_END[8:0] and DCRP_AREA_START[8:0] in NVMC_OTP_USER_CTL2, these bits can be modified. 1: Lock the 16 bits of DCRP_AREA_END[8:0] and DCRP_AREA_START[8:0] in NVMC_OTP_USER_CTL2, these bits cannot be modified.
1	SCR_AREALK	Secure user area address lock bit This bit is factory set to 0. 0: Unlock the 16 bits of SCR_AREA_END[8:0] and SCR_AREA_START[8:0] in NVMC_OTP_USER_CTL1, these bits can be modified. 1: Lock the 16 bits of SCR_AREA_END[8:0] and SCR_AREA_START[8:0] in NVMC_OTP_USER_CTL1, these bits cannot be modified.
0	DPLK	DBGPSWD_DATA lock bit This bit is factory set to 0. 0: Unlock the DBGPSWD_DATA register, allowing the value of DBGPSWD_DATA to be written. 1: Lock the DBGPSWD_DATA register, preventing the value of DBGPSWD_DATA from being written.

OTP User Control Register 5 (NVMC_OTP_USER_CTL5)

Offset address: 0x220

Reset value: 0xXXXX XXXX, load the value from the OTP storage unit after reset.

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DUMMY1_LK[15:8]								DUMMY1[15:8]							
r								r							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DUMMY0_LK[7:0]								DUMMY0[7:0]							
r								r							

Bits	Fields	Descriptions
31:24	DUMMY1_LK[7:0]	DUMMY1[7:0] lock bit The default value of this bit field is 0. 0: Unlock the 8 bits of DUMMY1[7:0], these bits can be modified. 1: Lock the 8 bits of DUMMY1[7:0], these bits cannot be modified.
23:16	DUMMY1[7:0]	The default value of this bit field is 0. Reserve space for other operations.
15:8	DUMMY0_LK[7:0]	DUMMY0[7:0] lock bit The default value of this bit field is 0. 0: Unlock the 8 bits of DUMMY0[7:0], these bits can be modified. 1: Lock the 8 bits of DUMMY0[7:0], these bits cannot be modified.

7:0	DUMMY0[7:0]	The default value of this bit field is 0. Reserve space for other operations.

8. RAM ECC monitor unit (RAMECCMU)

The GD32H77x_78x device features two RAM ECC monitor units (RAMECCMU) in Region 0 and Region1 separately. It provides a method to verify ECC status for applications and execute error handling when errors occur.

8.1. Characteristics

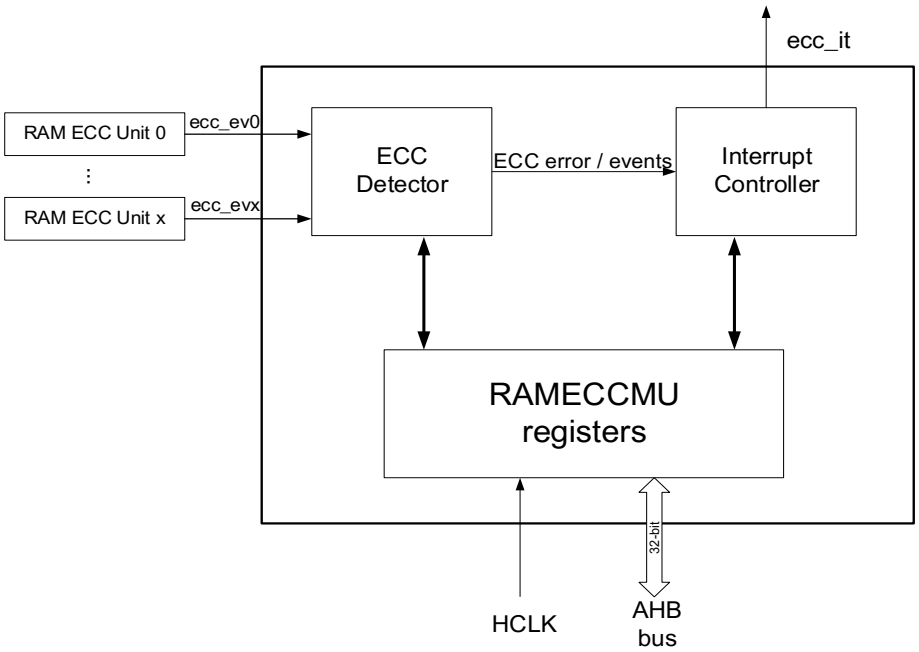
The main functions of RAMECCMU are the following:

- RAM ECC monitor for each Region
- Identification for RAM failing address/data

8.2. Function overview

GD32H77x_78x features two RAMECC monitor units mounted on AHB3 in Region 0 and AHB2 in Region 1 separately. The block architecture of RAMECCMU is shown as ***Figure 8-1. Block architecture of RAMECCMU***.

Figure 8-1. Block architecture of RAMECCMU



There are two RAMECC monitor units in GD32H77x_78x series which are described in ***Table 8-1. RAMECC monitor x unit for Region 0 (x=0..4)*** and ***Table 8-2. RAMECC monitor x unit for Region 1 (x=0..2)***.

Table 8-1. RAMECC monitor x unit for Region 0 (x=0..4)

RAMECC monitor number	RAMECC monitor status
0	AXI SRAM ECC
1	ITCM-RAM ECC
2	DTCM-RAM ECC(D0TCM)
3	DTCM-RAM ECC(D1TCM)
4	RAM shared(ITCM/AXI SRAM) ECC

Table 8-2. RAMECC monitor x unit for Region 1 (x=0..2)

RAMECC monitor number	RAMECC monitor status
0	SRAM0 ECC
1	SRAM1 ECC
2	Backup RAM(BKPSRAM) ECC

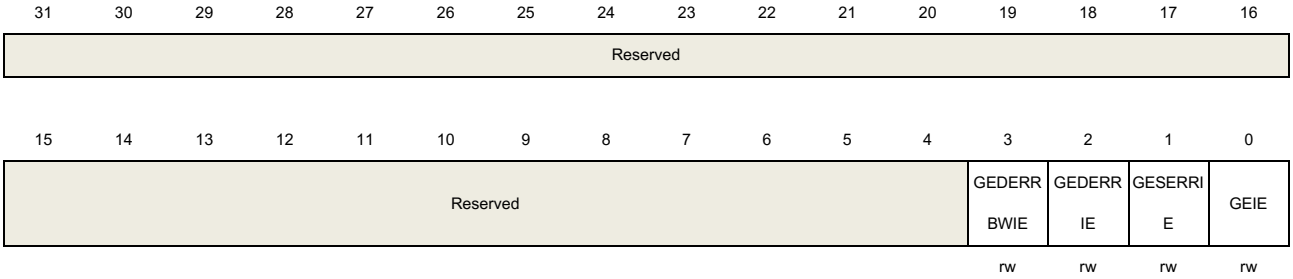
8.3. Register definition

RAMECCMU Region 0 base address: 0x5200 9000
RAMECCMU Region 1 base address: 0x4802 3000

8.3.1. RAMECCMU global interrupt register (RAMECCMU_INT)

Address offset: 0x00
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	GEDERRBWIE	Global ECC double error on byte write interrupt enable 0: No interrupt generated 1: Interrupt generated when ECC double detection error occurs during a byte write operation to RAM
2	GEDERRIE	Global ECC double error interrupt enable 0: No interrupt generated 1: Interrupt generated when ECC double detection error occurs during a read operation from RAM
1	GESERRIE	Global ECC single error interrupt enable 0: No interrupt generated 1: Interrupt generated when ECC single error occurs during a read operation from RAM
0	GEIE	Global ECC interrupt enable 0: No interrupt generated 1: Interrupt generated when one of GEDERRBWIE, GEDERRIE or GESERRIE error occurs

8.3.2. RAMECCMU monitor x control register (RAMECCMU_MxCTL)

Address offset: 0x20 * (x+1), (x is ECC monitoring number, x=0..4 for Region 0, while x=0..2

for Region 1)

Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										ECCERRLATEN	ECCDERRBWIE	ECCDERRIE	ECCSERRIE	Reserved	
										rw	rw	rw	rw		

Bits	Fields	Descriptions
31:6	Reserved	Must be kept at reset value.
5	ECCERRLATEN	ECC error latching enable 0: No error context are latched to their respective registers when ECC error occurs 1: Error context are latched to their respective registers when ECC error occurs
4	ECCDERRBWIE	ECC double error on byte write interrupt enable 0: No interrupt generated 1: Interrupt generated when ECC double error occurs on byte write operation to RAM
3	ECCDERRIE	ECC double error interrupt enable 0: No interrupt generated 1: Interrupt generated when ECC double error occurs on read operation from RAM
2	ECCSERRIE	ECC single error interrupt enable 0: No interrupt generated 1: Interrupt generated when ECC single error occurs on read operation from RAM
1:0	Reserved	Must be kept at reset value.

8.3.3. RAMECCMU monitor x status register (RAMECCMU_MxSTAT)

Address offset: 0x24 + 0x20 * x, (x= ECC monitoring number, x=0..4 for Region 0, while x=0..2 for Region 1)

Reset value: 0x0000 0000

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

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

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

rc_w0 rc_w0 rc_w0

Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value.
2	ECCDERRBWDF	ECC double error on byte write detected flag This bit is set by hardware and cleared by writing 0. 0: Double error is detected during read when ECCDERRDF is set 1: Double error detected during unaligned write
1	ECCDERRDF	ECC double error detected flag 0: No double error detected 1: Double error detected
0	ECCSERRDCF	ECC single error detected and corrected flag 0: No error detected and corrected 1: Error detected and corrected

8.3.4. RAMECCMU monitor x failing address register (RAMECCMU_MxFADDR)

Address offset: 0x28 + 0x20 * x, (x= ECC monitoring number, x=0..4 for Region 0, while x=0..2 for Region 1)

Reset value: 0x2400 0000 for AXI SRAM
0x0000 0000 for ITCM
0x2000 0000 for D0TCM
0x2000 0004 for D1TCM
0x2408 0000 for RAM shared(ITCM/AXI SRAM)

0x3000 0000 for SRAM0
0x3000 4000 for SRAM1
0x3880 0000 for BKPSRAM

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

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

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

Bits	Fields	Descriptions
31:0	ECCFADDR[31:0]	ECC error failing address This register contains the address which generated ECC error when error occurs.

8.3.5. RAMECCMU monitor x failing data low register (RAMECCMU_MxFDL)

Address offset: 0x2C + 0x20 * x, (x= ECC monitoring number, x=0..4 for Region 0, while x=0..2 for Region 1)
Reset value: 0x0000 0000

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

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

Bits	Fields	Descriptions
31:0	ECCFDL[31:0]	ECC failing data low bits This register contains the LSB of data which generated ECC error when error occurs or the full memory word for 32-bit word SRAM.

8.3.6. RAMECCMU monitor x failing data high register (RAMECCMU_MxFDH)

Address offset: 0x30 + 0x20 * x, (x= ECC monitoring number, x=0..4 for Region 0, while x=0..2 for Region 1)
Reset value: 0x0000 0000

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

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

Bits	Fields	Descriptions
31:0	ECCFDH[31:0]	ECC failing data high bits (64-bit) This register contains the MSB of data which generated ECC error when error occurs.

8.3.7. RAMECCMU monitor x failing ECC error code register (RAMECCMU_MxFECODE)

Address offset: 0x34 + 0x20 * x, (x= ECC monitoring number, x=0..4 for Region 0, while x=0..2 for Region 1)

Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:0	ECCFECODE[31:0]	ECC failing error code This register contains the index where the bit error occurs and the ECC code.

9. Power management unit (PMU)

9.1. Overview

The power consumption is regarded as one of the most important issues for the devices of GD32H77x_78x 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 GD32H77x_78x series devices, there are three power domains, including V_{DD} / V_{DDA} domain, V_{CORE} domain, and Backup domain, as is shown in ***Figure 9-1. Power supply overview***. The power of the V_{DD} / V_{DDA} domain is supplied directly by power source. A Step-down voltage stabilizer, which is low power switched-mode power supply (SMPS step-down voltage stabilizer) is used to supply the V_{CORE} domain power. A power switch is implemented for the Backup domain. It can be powered from the VBAT voltage when the main VDD supply is shut down. Peripheral supply regulation USB regulator and MIPI (Mobile Industry Processor Interface) regulator.

The GD32H78x series supports two operating modes: General Mode and High-Performance Mode.

In General Mode, the maximum frequency is 700 MHz, while in High-Performance Mode, the maximum frequency is 750 MHz (provided that the operating temperature is within 85°C and the Vcore voltage operates at 0.97V).

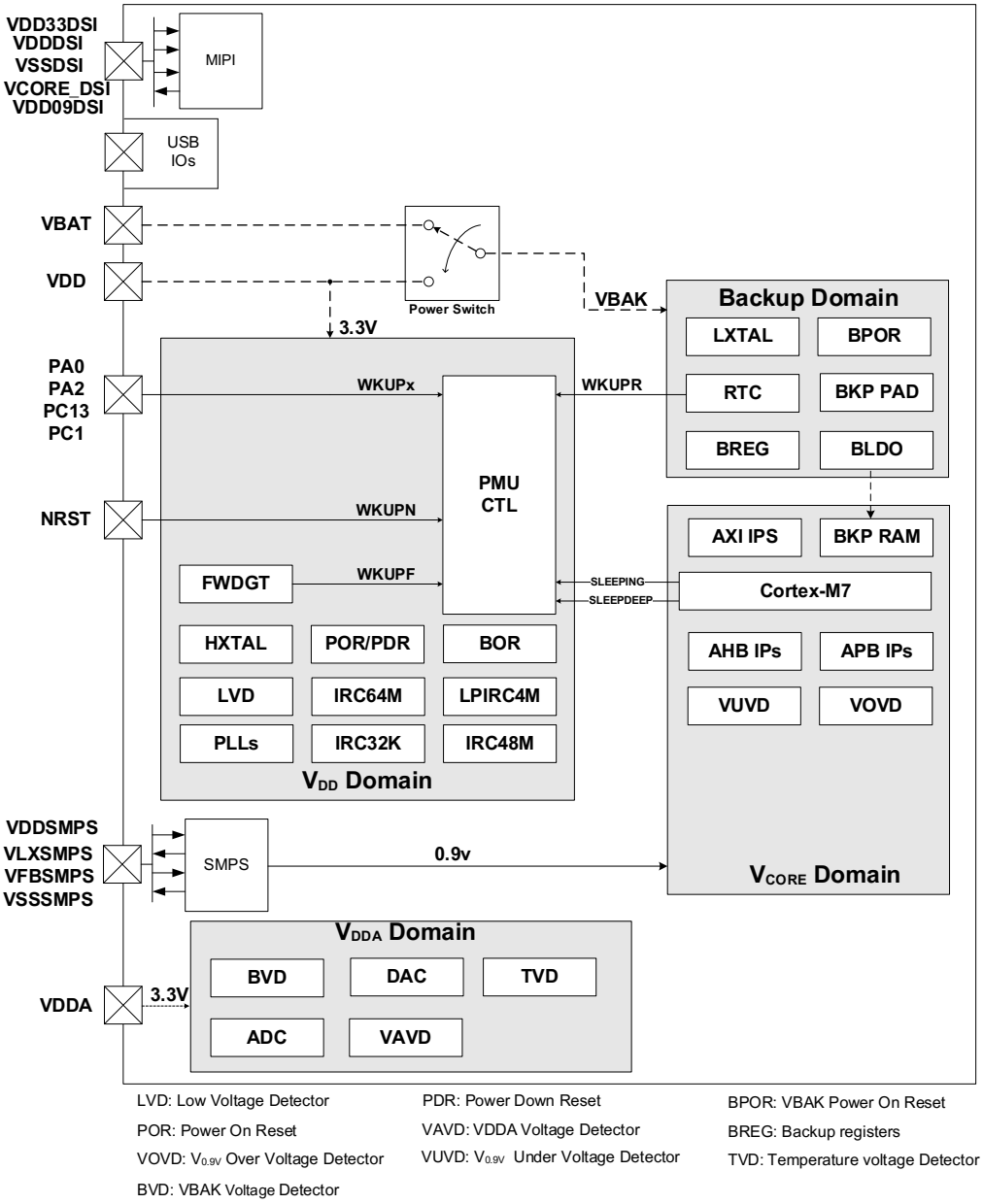
9.2. Characteristics

- Three power domains: V_{BAK}, V_{DD} / V_{DDA} and V_{CORE} power domains.
- Three power saving modes: Sleep, Deep-sleep and Standby modes.
- Low Voltage Detector (LVD) can issue an interrupt or event when the power is lower than a programmed threshold.
- Battery power (VBAT) for Backup domain when VDD is shut down.
- USB supply regulator.
- MIPI supply regulator.
- Power supply supervision: POR / PDR monitor / BOR monitor / LVD monitor / V_{DDA} Voltage Detector (VAVD) monitor / V_{CORE} Under Voltage Detector(VUVD) monitor / V_{CORE} Over Voltage Detector(VOVD) monitor /V_{BAK} thresholds / Temperature thresholds.
- VBAT battery charging / Operating modes / Voltage scaling control / Low-power modes management.
- Step-down voltage stabilizer, which is low power switched-mode power supply (SMPS step-down voltage stabilizer.

9.3. Function overview

Figure 9-1. Power supply overview provides details on the internal configuration of the PMU and the relevant power domains.

Figure 9-1. Power supply overview



Note: The SMPS power supply is not supported on all devices. Please refer to the datasheet for details.

9.3.1. Backup domain

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

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

The clock source of the Real Time Clock (RTC) circuit can be derived from the Internal 32KHz RC oscillator (IRC32K) or the Low Speed Crystal oscillator (LXTAL), or HXTAL divided by RTCDIV[5:0] (in RCU_CFG0 register). When VDD is shut down, only LXTAL is valid for RTC. Before entering the power saving mode by executing the WFI / WFE instruction, the Cortex®-M7 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 the ***Real time clock (RTC)***.

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

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

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

- PC13 can be used as RTC function pin described in the ***Real time clock (RTC)*** 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).

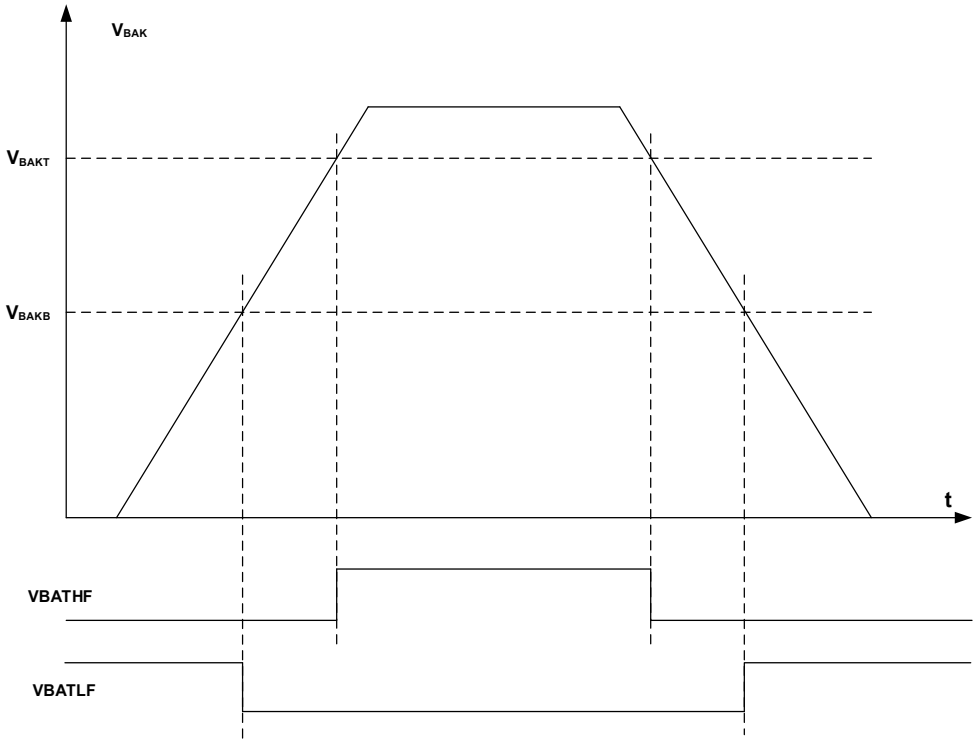
The external VBAT battery can be charged by the VDD through an internal resistor. The charging resistor can be selected by configuring the VCRSEL bit in PMU_CTL3 register. A 5kOhms resistor or a 1.5kOhms resistor can be selected for external V_{BAT} battery charging. The external VBAT battery charging is enabled by setting the VCEN bit in PMU_CTL3 register. When in BKP only mode, the VBAT battery charging is disabled by hardware.

Note: In BKP only mode, VDD is power-off, and the backup domain is power by VBAT pin.

Backup domain voltage thresholds

The internal power switch can select between the voltage sources of the Backup domain VBAT and VDD. The supply voltage for Backup domain (VBAK) 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} or lower than V_{BAKB}, the flag bit VBATHF / VBATLF will set. And this is only available when BKPVSEN bit is set. Backup domain voltage thresholds, as is shown in ***Figure 9-2. Waveform of the Backup domain voltage thresholds***.

Figure 9-2. Waveform of the Backup domain voltage thresholds



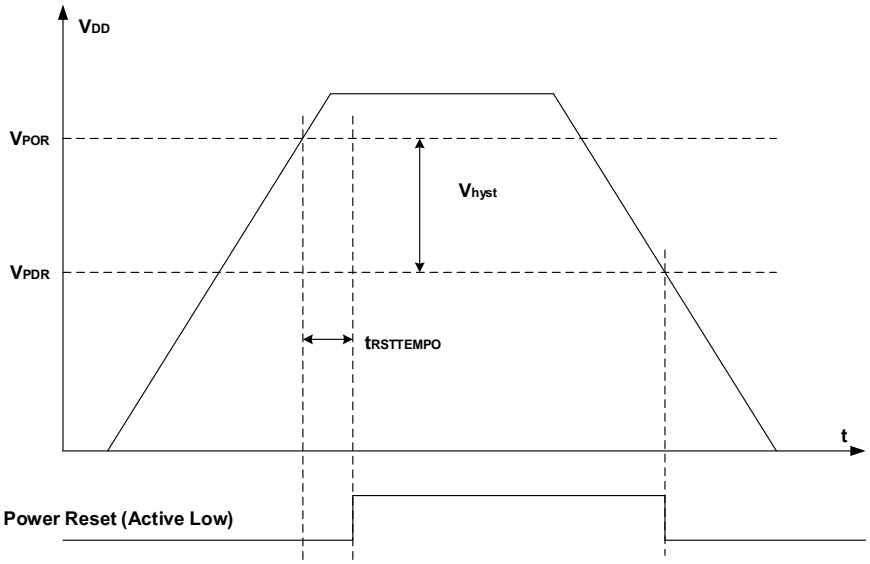
9.3.2. V_{DD} / V_{DDA} power domain

V_{DD} / V_{DDA} domain includes two parts: V_{DD} domain and V_{DDA} domain. V_{DD} domain includes HXTAL (high speed crystal oscillator), POR / PDR (power on / down reset), BOR (brownout reset), LVD (low voltage detector), IRC32K (internal 32KHz RC oscillator), LPIRC4M (low power internal 4MHz RC oscillator), IRC48M (internal 48MHz RC oscillator at 48MHz frequency), IRC64M (internal 64MHz RC oscillator at 64MHz frequency), PLLs (phase locking loop), FWDGT (free watchdog timer), all pads except PC13 / PC14 / PC15, etc. V_{DDA} domain includes ADC / DAC (AD / DA Converter), VAVD (V_{DDA} voltage detector), BVD (V_{BAK} voltage detector), TVD (temperature voltage detector).

V_{DD} domain

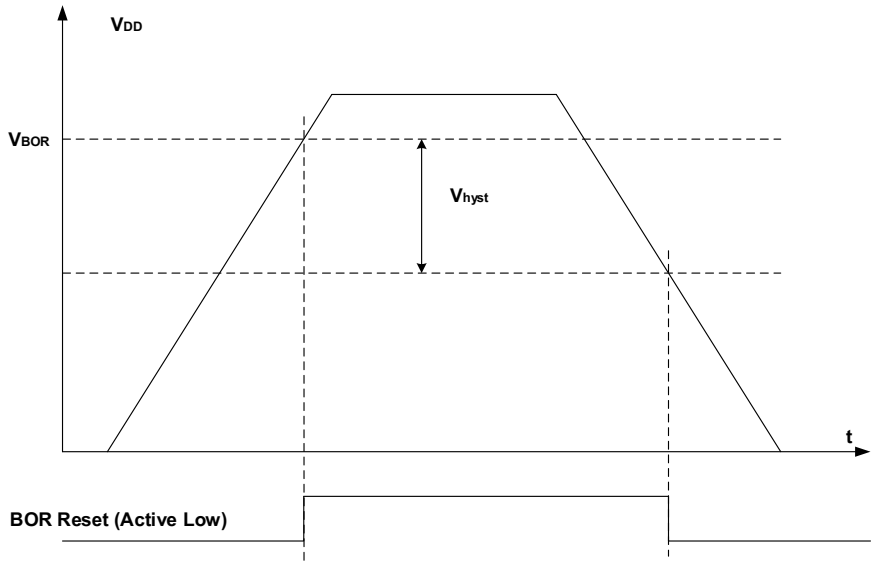
The POR / PDR circuit is implemented to detect VDD and generate the power reset signal which resets the whole chip except the Backup domain when the supply voltage is lower than the specified threshold. ***Figure 9-3. Waveform of the POR / PDR*** shows the relationship between the supply voltage and the power reset signal. V_{POR} indicates the threshold of power on reset, while V_{PDR} means the threshold of power down reset. The hysteresis voltage (Vhyst) is refer to the datasheet.

Figure 9-3. Waveform of the POR / PDR



The BOR circuit is used to detect VDD 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 0b00 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 0b00 or not. ***Figure 9-4. 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 (Vhyst) is refer to the datasheet.

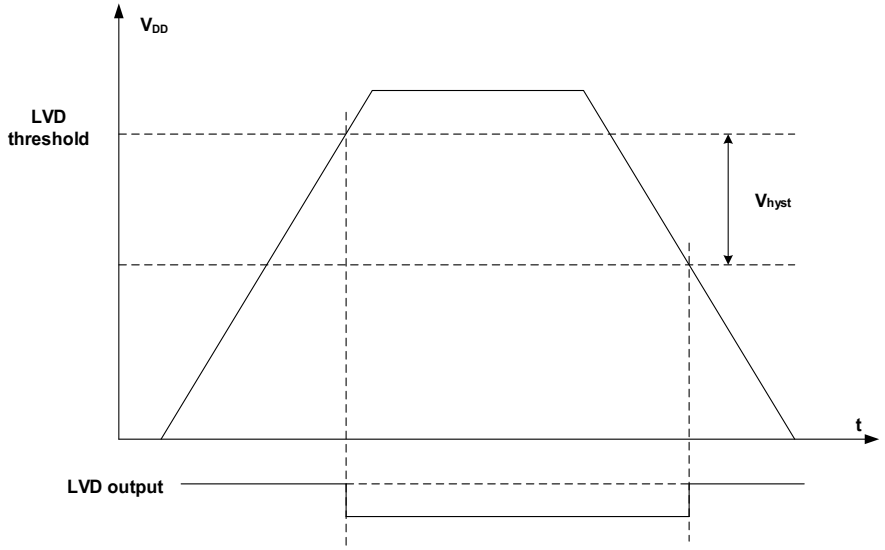
Figure 9-4. Waveform of the BOR



V_{DDA} domain

The LVD is used to detect whether the VDD supply voltage is lower than a programmed threshold selected by the LVDT[2:0] bits in the Power control register(PMU_CTL0). The LVD is enabled by setting the LVDEN bit, and LVDF bit, which in PMU_CS, indicates if VDD 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 9-5. Waveform of the LVD threshold*** shows the relationship between the V_{DD} supply voltage and the LVD output (LVD interrupt signal depends on EXTI line 16 rising or falling edge configuration). The hysteresis voltage (V_{hyst}) is refer to the datasheet.

Figure 9-5. Waveform of the LVD threshold



Generally, digital circuits are powered by VDD, while most of analog circuits are powered by VDDA. To improve the ADC and DAC conversion accuracy, the independent power supply

VDDA is implemented to achieve better performance of analog circuits. VDDA can be externally connected to VDD through the external filtering circuit that avoids noise on VDDA, and VSSA should be connected to VSS through the specific circuit independently. Otherwise, if VDDA is different from VDD, VDDA 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 25-2. ADC input pins definition*** and ***Table 26-1. DAC I/O description***

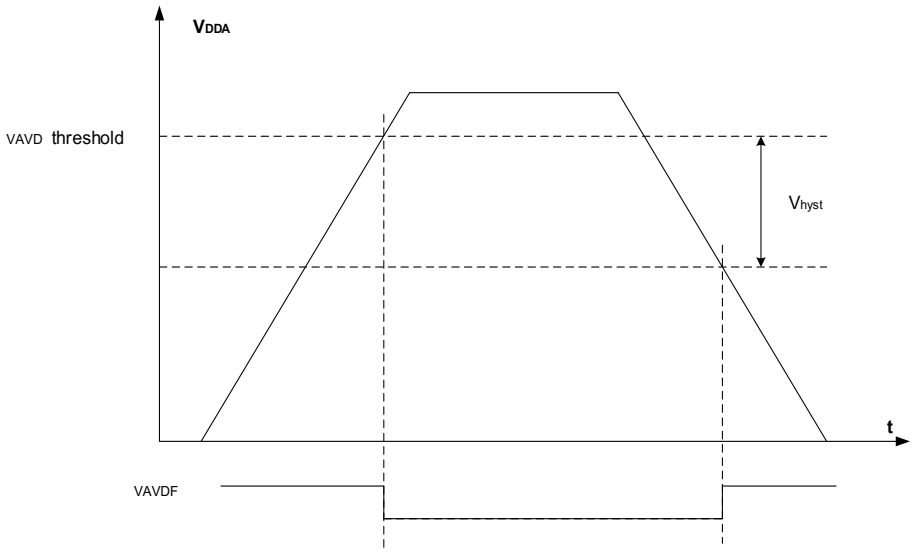
, VREFN pin must be connected to VSSA pin. The VREFP pin is only available on no less than 100-pin packages, or else the VREFP pin is not available and internally connected to VDDA. The VREFN pin is only available on no less than 100-pin packages, or else the VREFN pin is not available and internally connected to VSSA.

V_{DDA} domain voltage thresholds

The V_{DDA} analog voltage detector (VAVD) is used to detect whether the VDDA supply voltage is lower than a programmed threshold selected by the VAVDVC[1:0] bits in the power control register(PMU_CTL0). The VAVD is enabled by setting the VAVDEN bit, and VAVDF bit, which in PMU_CS, indicates if V_{DDA} is higher or lower than the specified VAVD threshold. This event is internally connected to the EXTI line 16 and can generate an interrupt if it is enabled through the EXTI registers.

Figure 9-6. Waveform of the VAVD threshold shows the relationship between the VAVD threshold and the VAVDF. The hysteresis voltage (Vhyst) is refer to the datasheet.

Figure 9-6. Waveform of the VAVD threshold



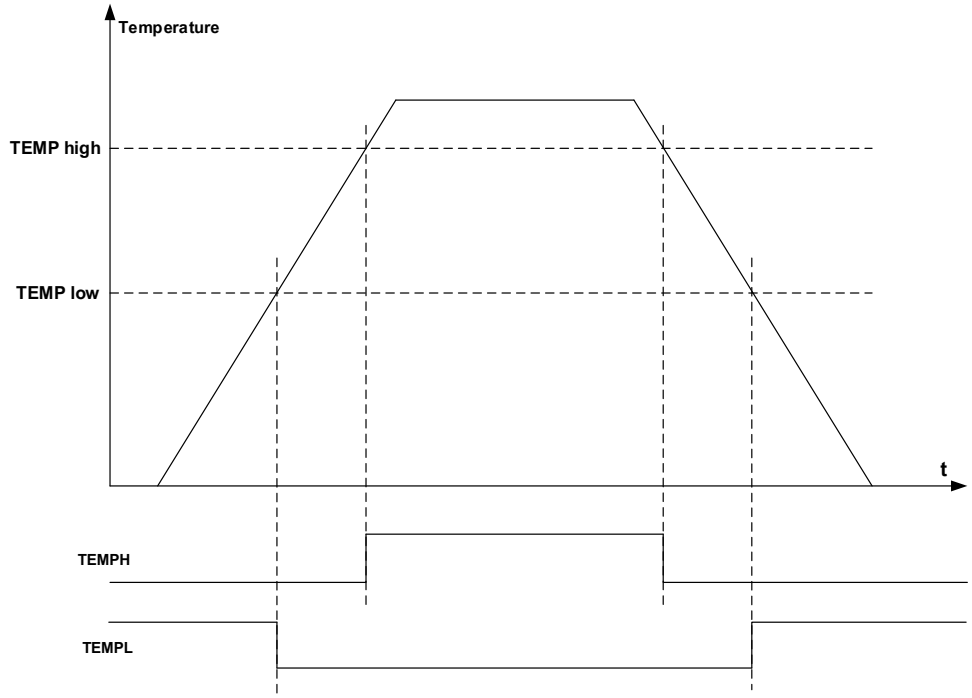
Temperature voltage thresholds

As will as Backup domain voltage thresholds, The junction temperature can be monitored by

comparing it with two threshold levels, TEMP high and TEMP low. TEMPH and TEMPL flags, in the PMU_CTL1, indicate whether the device temperature is higher or lower than the threshold. The temperature voltage thresholds 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 backup regulator is enabled (VBTMEN it set in the PMU_CTL1 register).

TEMPH and TEMPL wakeup interrupts are available on the RTC tamper signals.

Figure 9-7. Temperature thresholds



9.3.3. V_{CORE} power domain

The main functions that include Cortex®-M7 logic, AHB, APB, AXI peripherals, the APB interfaces for the Backup domain and the V_{DD} / V_{DDA} domain, BKP RAM (backup ram), VOVD (V_{CORE} over voltage detector), VUVD (V_{CORE} under voltage detector) ,etc, are located in this power domain. Once the V_{CORE} is powered up, the POR will generate a reset sequence on the V_{CORE} power domain. If need to enter the expected power saving mode, the associated control bits must be configured. Then, once a WFI (Wait for Interrupt) or WFE (Wait for Event) instruction is executed, the device will enter an expected power saving mode.

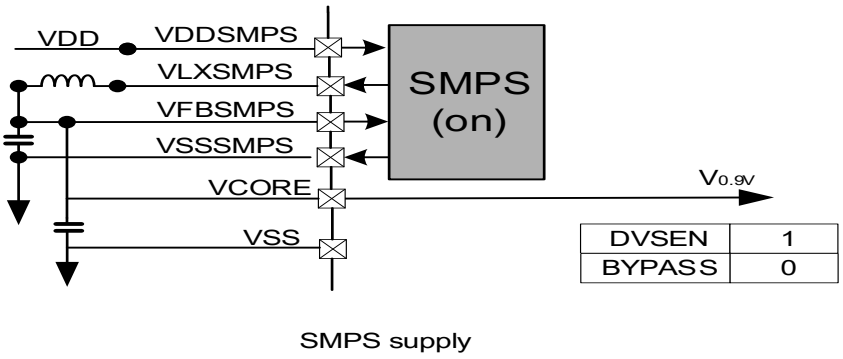
V_{CORE} power supply

With the SMPS step-down stabilizer, it is possible to set the power source of the V_{CORE} power domain. Different configurations can provide two effective V_{CORE} power supply modes.

- SMPS power supplies mode

The configuration method to enter this mode is that the DVSEN bit is 0b1. At this time, the SMPS step-down stabilizer is on. It also supplies power for the V_{CORE} power domain. The BYPASS bit is 0b0, and the V_{CORE} power domain is not powered through V_{CORE} (external direct power supplies). ***Figure 9-8. SMPS supplies for VCORE power domain*** shows this power supplies mode.

Figure 9-8. SMPS supplies for V_{CORE} power domain



■ Bypass mode

The configuration method to enter this mode is that the DVSEN bit is 0b0. At this time, the SMPS step-down stabilizer is turned off, the BYPASS bit is 0b1, and the V_{CORE} power domain is powered through V_{CORE} (external direct power supplies). ***Figure 9-9. Bypass*** shows this power supplies mode.

Figure 9-9. Bypass

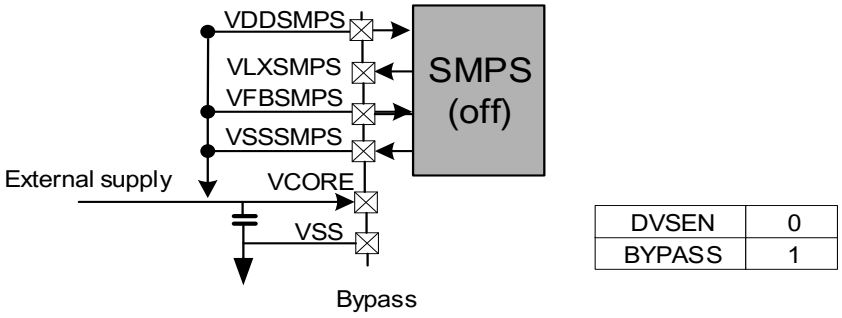


Table 9-1. supply mode

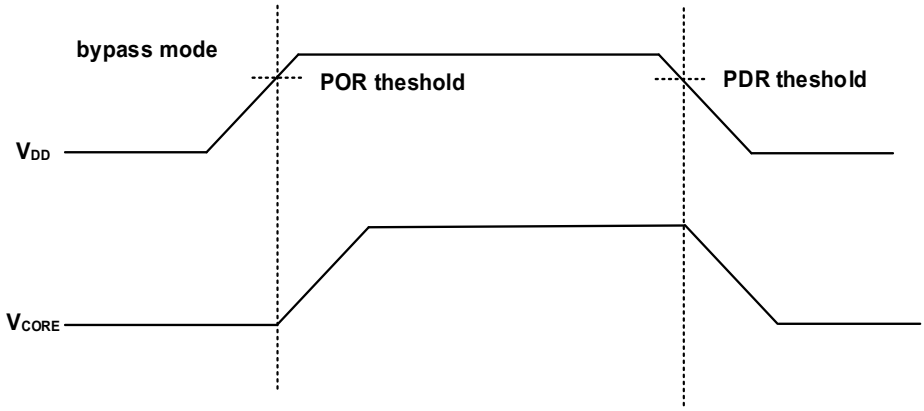
Mode	Supply configuration	DVSEN	BYPASS
0	SMPS power supplies mode	1	0
1	bypass mode	0	1

Note: In addition to the above valid combinations, other DVSEN, BYPASS bits or bit field configuration combinations are invalid. The power supplies state of the V_{CORE} power domain will remain the state after reset (no configure the power supplies mode).

Note: The maximum working frequency is related to the supply voltage. Please refer to the datasheet for details.

Note: The SMPS power supply is not supported on all devices. Please refer to the datasheet for details.

Figure 9-10. Power on and power down timing requirements for V_{DD} and V_{CORE} in bypass mode



V_{CORE} power thresholds detector

There is an internal V_{CORE} power over voltage detector, when VOVDEN is 0b1, it means V_{CORE} power over voltage detector is enabled. Once V_{CORE} power domain is over than a programmed threshold selected by the VOVDVC[1:0] bits in the power control register(PMU_CTL0), 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] bits in PMU_CTL2 register, VOVDF1 can trigger the IRQ219 interrupt. This allows to suppress spikes with a programmable length of 1 to 255 of 1024* T_{pclk4}. The hysteresis voltage (Vhyst) please refer to device datasheet.

Figure 9-11. Waveform of the VOVD0

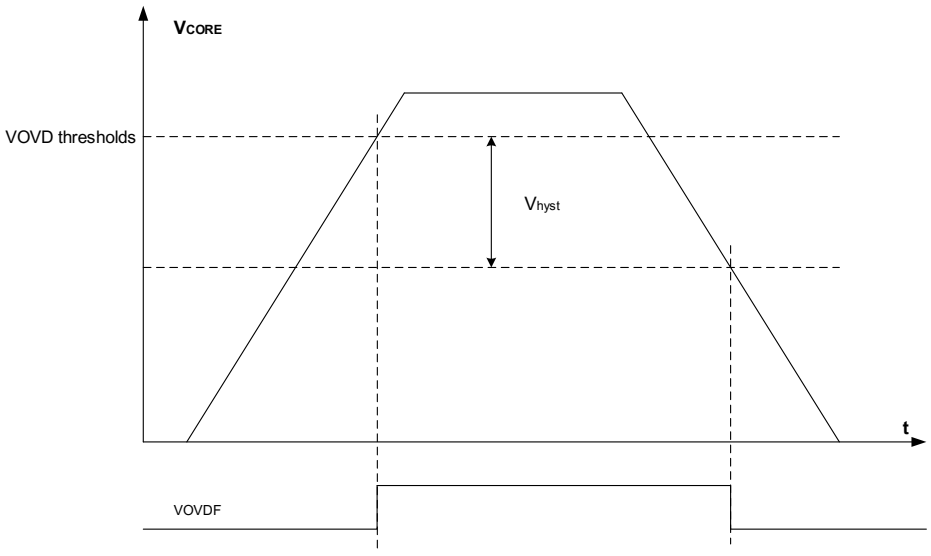
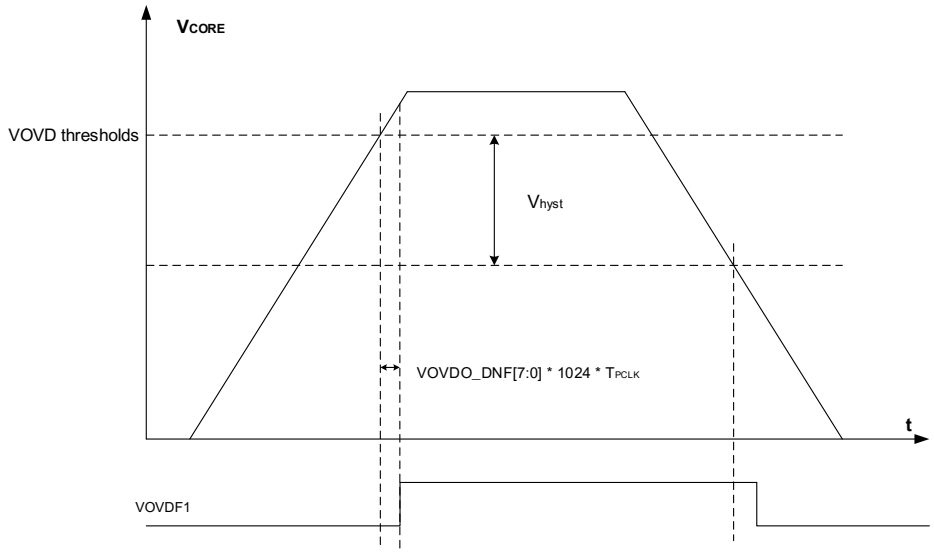


Figure 9-12. Waveform of the VOVD1



There is an internal V_{CORE} power under voltage detector, when VUVDEN is 0b1, it means V_{CORE} power under voltage detector is enabled. Once V_{CORE} power domain is lower than a programmed threshold selected by the VUVDVC[1:0] bits in the power control register(PMU_CTL0), VUVDF0 will be set immediately after two flip-flops synchronization of the analog output. 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_CTL2 register, VUVDF1 can trigger the IRQ219 interrupt. This allows to suppress spikes with a programmable length of 1 to 255 of 1024* T_{pclk4}. The hysteresis voltage (Vhyst) please refer to device datasheet.

Figure 9-13. waveform of VUVD0

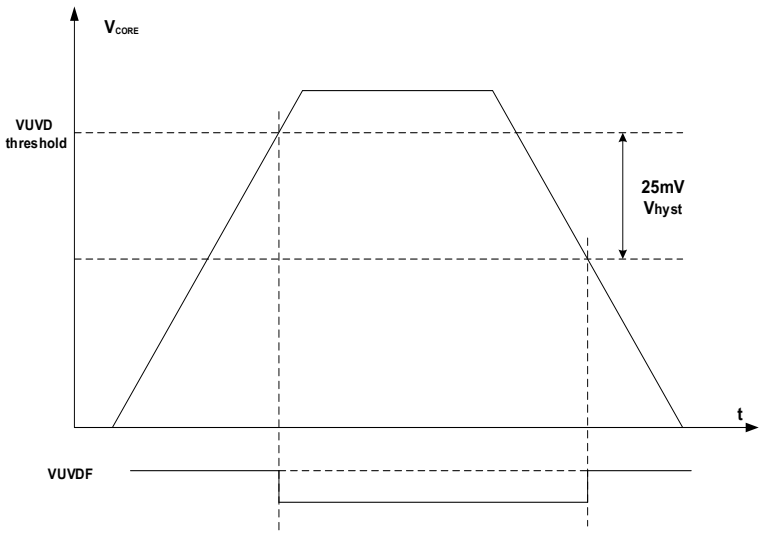
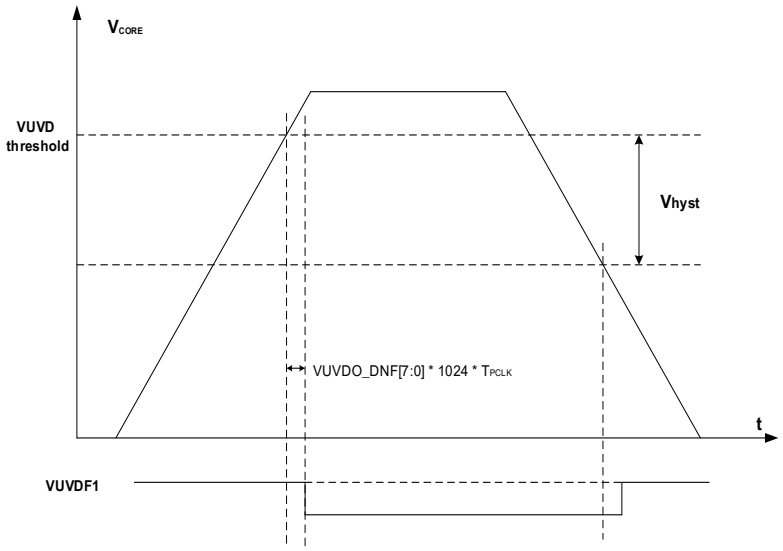


Figure 9-14. waveform of VUVD1



9.3.4. USB Power Management

The GD32H77x_78x series USB integrates an internal regulator. Users can choose to enable this regulator by connecting the VDD50USB pin to a 5V power supply to provide power for the USB module, as shown in ***Figure 9-15. Connection diagram when USB regulator is enabled***. Alternatively, users can bypass this regulator by connecting the VDD33USB pin to a 3.3V power supply to provide power for the USB module, as shown in ***Figure 9-16. Connection diagram when USB regulator is disabled***.

Figure 9-15. Connection diagram when USB regulator is enabled

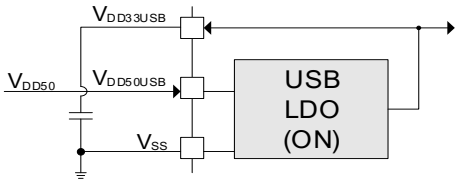
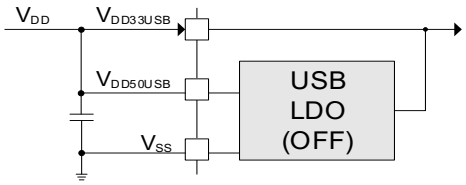


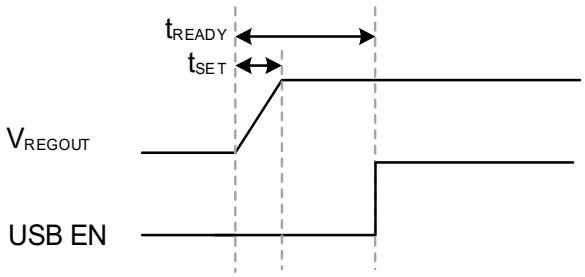
Figure 9-16. Connection diagram when USB regulator is disabled



Note: When VDD >= 3V, the USB power can operate in either USB regulator supply mode or bypass mode; when VDD < 3V, the USB power can only operate in USB regulator supply mode.

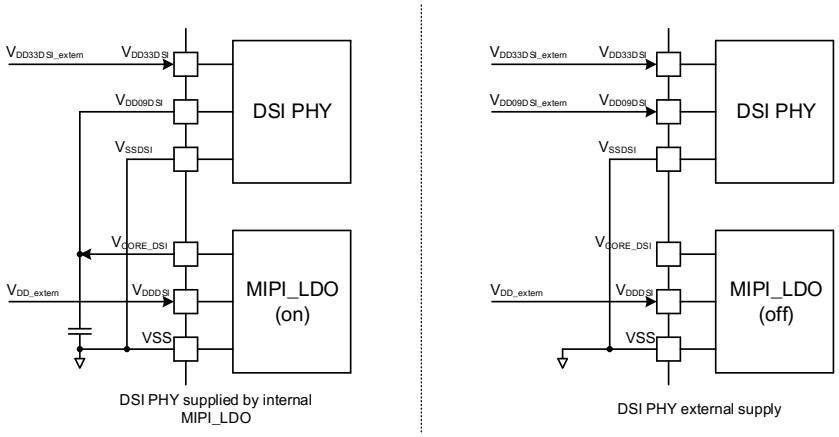
Note: When using the USB module or ISP download function, VDD33USB must be powered, which can be supplied through one of the two methods mentioned above.

Figure 2 8. The power-on sequence of the USB voltage regulator



9.3.5. DSI Power domain

Figure 9-17. Schematic diagram of MIPI_LDO power supply



The DSI module supports independent power management function and can achieve separate power-on and power-off control. This module consists of a DSI controller and a DSI PHY. The DSI controller is powered by VCORE_DSI, while the DSI DPHY are powered by VDD33DSI and VDD09DSI respectively. The two parts are isolated and protected through LVL (Level Shifter) and ISO (Isolation Cell). Because DSI has a dual power domain structure of digital and analog and is equipped with an independent analog LDO, the power-on and power-off operations must follow specific timing requirements.

When powering on, the module should first be powered on, then the isolation should be released. After the power-on is completed, the reset of the module can be revoked through the RCU control to bring the module back to the normal working state. Among them, the DSI PHY must be powered on before the DSI controller. The signal for controlling the power supply of the DSI controller is MIPISW. When this signal is set, the system will indicate through the MIPIHTF feedback signal whether the power-off or power-on has been completed. Users need to wait for the change of this flag bit to confirm the successful switching of the power supply status.

The operation sequence for power-off is the opposite of that for power-on. First, the software of the module should be reset through the RCU, then isolation should be enabled, and finally the power should be turned off. In high-temperature environments, to protect the chip and reduce power consumption, the system needs to turn off the power of some peripheral

devices, including the DSI. At this time, the above-mentioned power-off process should also be followed. After the temperature returns to normal, the module should be restarted according to the power-on process. Throughout the entire power management process, the reset operation of the module is controlled by the RCU register. Users only need to power on in the order of first powering on, then releasing isolation, and finally releasing the reset, and power off in the order of first reset, then isolation, and finally power-off, to achieve reliable power management of the MIPI module.

9.3.6. Power saving modes

After a system reset or a power reset, the MCU operates at full function and all power domains are active. Users can achieve lower power consumption through slowing down the system clocks (HCLK, PCLK1, PCLK2, PCLK3 and PCLK4) 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.

After system reset / power reset or wakeup from standby mode. The MCU enter normal run mode, no effect on all clocks, all power on.

Sleep mode

The Sleep mode is corresponding to the SLEEPING mode of the Cortex®-M7. In Sleep mode, only clock of Cortex®-M7 is off. To enter the Sleep mode, it is only necessary to clear the SLEEPDEEP bit in the Cortex®-M7 System Control Register, and execute a WFI or WFE instruction. If the Sleep mode is entered by executing a WFI instruction, any interrupt can wake up the system. If it is entered by executing a WFE instruction, any wakeup event can wake up the system (If SEVONPEND is 1, any interrupt can wake up the system, refer to Cortex-M7 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®-M7 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 APB system reset, 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®-M7. In Deep-sleep mode, all clocks in the V_{CORE} domain are off, and all of LPIRC4M, IRC48M, IRC64M, HXTAL and PLLs are disabled. Before entering the Deep-sleep mode, it is necessary to set the SLEEPDEEP bit in the Cortex®-M7 System Control Register, set STBMOD bit to 0 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 0b1, any interrupt from EXTI lines can wake up the system, refer to Cortex®-M7 Technical Reference Manual).

Note: If power-on or exit from standby, it need to wait more than 300us before entering Deep-sleep mode.

Standby mode

The Standby mode is based on the SLEEPDEEP mode of the Cortex®-M7, too. In Standby mode, the whole V_{CORE} domain is power off, and all of LPIRC4M, IRC48M, IRC64M, HXTAL and PLLs are disabled. Before entering the Standby mode, it is necessary to set the SLEEPDEEP bit in the Cortex®-M7 System Control Register, and set the STBMOD bit to 1 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 and tamper), the FWDGT reset, LCKMD, 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 V_{CORE} power domain are lost in Standby mode. When exiting from the Standby mode, a power-on reset occurs and the Cortex®-M7 will execute instruction code from the 0x00000000 address.

Table 9-2. Power saving mode summary

Mode	Description	Entry	Wakeup	Wakeup status	Wakeup Latency
Sleep	Only CPU clock is off	SLEEPDEEP = 0, WFI or WFE from run	Any interrupt for WFI Any event (or interrupt when SEVONPEND is 1) for WFE	normal run mode	-
Deep-sleep	1. All clocks in the V_{CORE} domain are off 2. Disable LPIRC4M, IRC48M, IRC64M, HXTAL and PLLs	SLEEPDEEP = 1, STBMOD = 0, WFI or WFE	Any interrupt from EXTI lines for WFI Any event(or interrupt when SEVONPEND is 1) from EXTI for WFE	normal run mode	LPIRC4M /IRC64M (confirmed by DSPWUSSEL) wakeup time, + NVM wakeup time
Standby	1. The V_{CORE} domain is power off 2. Disable LPIRC4M, IRC48M, IRC64M, HXTAL and PLLs	SLEEPDEEP = 1, STBMOD = 1, WFI or WFE	1. NRST pin 2. WKUP pins 3. FWDGT reset 4. RTC (alarm and tamper) 5. LCKMD	normal run mode	IRC64M wakeup time, + SMPS wakeup time+NVM wakeup time
BKP only	All VDD / V_{CORE} domain	VDD off	VDD on	normal	VDD power on

Mode	Description	Entry	Wakeup	Wakeup status	Wakeup Latency
mode	power off			run mode	sequence

Note: In Standby mode, all I/Os are in high-impedance state except NRST pin, PC13 pin when configured for RTC function, PC14 and PC15 pins when used as LXTAL crystal oscillator pins, and WKUP pins if enabled,. PC7 with CORE POR OFF enabled.

9.4. Register definition

PMU base address: 0x5800 5800

9.4.1. Control register 0 (PMU_CTL0)

Address offset: 0x00
Reset value: 0x1600 8000

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
SRAM_PRDYN	SRAM_MODE[1:0]		VUVDVC[1:0]		VOVDVC[1:0]		VUVDEN	Reserved				VOVDEN	VAVDVC[1:0]		VAVDEN
r	rw		rw		rw		rw					rw	rw		rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSVOS[1:0]		Reserved			COREPDREN	COREPOROFFP	BKPWEN	LVDT[2:0]			LVDEN	STBRST	WURST	STBMOD	COREPOROFFOUT
rs					rw	rw	rw	rw			rw	rc_w1	rc_w1	rw	rw

Bits	Fields	Descriptions
31	SRAM_PRDYN	Indicates when the internal memory instance power supplies are stable after returning from retention mode. 1: Indicates that the device has entered Deep-sleep mode 0: Indicates that the device has exited Deep-sleep mode
30:29	SRAM_MODE[1:0]	Sram_mode in Deep-sleep 00: Disable 01: Precharge 10: Retention 11: Reserved
28:27	VUVDVC[1:0]	VCORE under voltage detector voltage level configure bits These bits are set and cleared by software. 00: 0.825V rising threshold, 0.8V falling threshold 01: 0.725V rising threshold, 0.7V falling threshold 10: 0.675V rising threshold, 0.65V falling threshold 11: 0.9V rising threshold, 0.875V falling threshold
26:25	VOVDVC[1:0]	V_{CORE} over voltage detector voltage level configure bits. These bits are set and cleared by software. 00: Configure V_{CORE} over voltage detector voltage level to 0.9V 01: Configure V_{CORE} over voltage detector voltage level to 1.0V 10: Configure V_{CORE} over voltage detector voltage level to 1.1V

		11: Configure V_{CORE} over voltage detector voltage level to 1.2V
24	VUVDEN	V_{CORE} under voltage detector enable 0: Disable V_{CORE} under voltage detector 1: Enable V_{CORE} under voltage detector
23:20	Reserved	Must be kept at reset value
19	VOVDEN	V_{CORE} over voltage detector voltage level enable bit This bit is set and cleared by software. 0: V_{CORE} over voltage detector disabled 1: V_{CORE} over voltage detector enabled
18:17	VAVDVC[1:0]	VDDA analog voltage detector voltage level configure bits These bits are set and cleared by software. 00: Configure VDDA analog voltage detector voltage level to 1.7V 01: Configure VDDA analog voltage detector voltage level to 2.1V 10: Configure VDDA analog voltage detector voltage level to 2.5V 11: Configure VDDA analog voltage detector voltage level to 2.8V
16	VAVDEN	VDDA analog voltage detector voltage enable bit This bit is set and cleared by software. 0: VDDA analog voltage detector voltage disabled. 1: VDDA analog voltage detector voltage enabled
15:14	DSVOS[1:0]	Deep-sleep mode voltage scaling selection These bits control the V_{CORE} voltage level in system Deep-sleep mode, to obtain the best trade-off between power consumption and performance. 00: Reserved 01: DSVOS scale 0.7V 10: DSVOS scale 0.8V (default) 11: DSVOS scale 0.9V
13:11	Reserved	Must be kept at reset value.
10	COREPDREN	CORE PDR circuit enable signal 0: The CORE PDR circuit is disabled 1: The CORE PDR circuit is enabled
9	COREPOROFFP	CORE POR OFF output polarity selection (Standby mode output from PC7) 0:Not negation 1:Negation
8	BKPWEN	Backup domain write enable bit This bit is set and cleared by software. 0: Disable write access to the registers in backup domain 1: Enable write access to the registers in backup domain After reset, any write access to the registers in Backup domain is ignored. This bit

		has to be set to enable write access to these registers.
7:5	LVDT[2:0]	Low voltage detector threshold 000: 2.1V 001: 2.3V 010: 2.4V 011: 2.6V 100: 2.7V 101: 2.9V 110: 3.0V 111: PB7 is used as an analog input pin to receive analog voltage for comparison with 0.8V reference, and must be configured in analog mode.
4	LVDEN	Low voltage detector enable bit This bit is set and cleared by software. 0: Disable low voltage detector 1: Enable low voltage detector
3	STBRST	Standby flag reset bit This bit is set by software. 0: No effect 1: Reset the standby flag This bit is always read as 0.
2	WURST	Wakeup flag reset bit This bit is set by software. 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®-M7 enters SLEEPDEEP mod 1: Enter the Standby mode when the Cortex®-M7 enters SLEEPDEEP mode
0	COREPOROFFOUT	CORE POR OFF output is enabled 0: Output not from PC7 1: Output from PC7

9.4.2. Control and status register (PMU_CS)

Address offset: 0x04

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											VOVDF0	Reserved			VAVDF
											r				r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		WUPEN5	Reserved	WUPEN3	Reserved	WUPEN1	WUPEN0	VUVDF1	VOVDF1	VUVDF0	Reserved	UVDF	LVDF	STBF	WUF
		rw		rw		rw	rw	r	r	r		r	r	r	r

Bits	Fields	Descriptions
31:21	Reserved	Must be kept at reset value.
20	VOVDF0	Peripheral voltage on V_{CORE} detector flag bit This bit is set and cleared by hardware. It is valid only if VOVDEN is enabled. 0: V_{CORE} is lower than VOVD threshold. 1: V_{CORE} is equal or higher than VOVD threshold.
19:17	Reserved	Must be kept at reset value
16	VAVDF	Analog voltage detector output on VDDA This bit is set and cleared by hardware. It is valid only if AVD on VDDA is enabled by the VAVDEN bit. 0: VDDA is equal or higher than the AVD threshold selected with the VAVDVC bits. 1: VDDA is lower than the AVD threshold selected with the VAVDVC bits.
15:14	Reserved	Must be kept at reset value.
13	WUPEN5	WKUP Pin5 (PC1) enable 0: Disable WKUP pin5 function 1: Enable WKUP pin5 function If WUPEN5 is set before entering the power saving mode, a rising edge on the WKUP pin5 wakes up the system from the power saving mode. As the WKUP pin5 is active high, the WKUP pin5 is internally configured to input pull down mode. And set this bit will trigger a wakeup event when the input is already high.
12	Reserved	Must be kept at reset value.
11	WUPEN3	WKUP pin3 (PC13) enable 0: Disable WKUP pin3 function 1: Enable WKUP pin3 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	Reserved	Must be kept at reset value.
9	WUPEN1	WKUP pin1 (PA2) 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 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_{CORE} under voltage detector flag bit after digital filter This bit is set and cleared by hardware. It is valid only if VUVDEN is enabled. 0: V_{CORE} is higher than VUVD threshold. 1: V_{CORE} is equal or lower than VUVD threshold.
6	VOVDF1	V_{CORE} over voltage detector flag bit after digital filter This bit is set and cleared by hardware. It is valid only if VOVDEN is enabled. 0: V_{CORE} is lower than VOVD threshold. 1: V_{CORE} is equal or higher than VOVD threshold.
5	VUVDF0	V_{CORE} under voltage detector flag bit This bit is set and cleared by hardware. It is valid only if VUVDEN is enabled 0: V_{CORE} is higher than VUVD threshold 1: V_{CORE} is equal or lower than VUVD threshold
4	Reserved	Must be kept at reset value.
3	UVDF	Record the undervoltage reset of the V_{CORE} 0: No undervoltage reset happened 1: Undervoltage reset happened
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.

9.4.3. 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: VBAT level below high threshold level 1: VBAT level equal or above high threshold level
20	VBATLF	VBAT level monitoring versus low threshold 0: VBAT level above low threshold level 1: VBAT level equal or below low threshold level
19:17	Reserved	Must be kept at reset value.
16	BKPVSRF	Backup voltage stabilizer ready flag This bit is set by hardware to indicate that the backup regulator is ready. 0: Backup voltage stabilizer not ready 1: Backup voltage stabilizer ready
15:5	Reserved	Must be kept at reset value.
4	VBTMEN	VBAT and temperature monitoring enable.

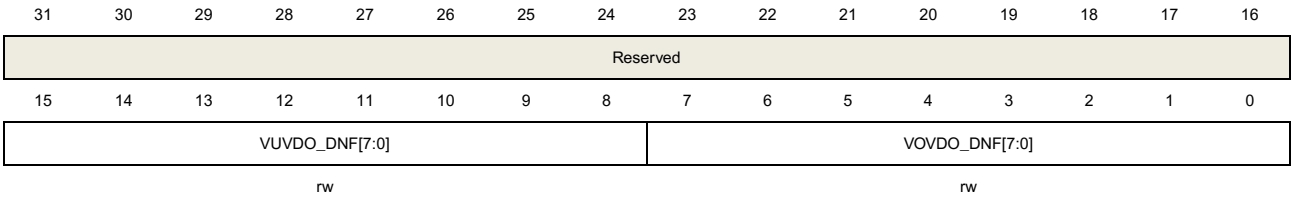
		When set, the VBAT supply and temperature monitoring is enabled. 0: VBAT and temperature monitoring disabled 1: VBAT and temperature monitoring enabled Note: VBAT and temperature monitoring are only available when the Backup voltage stabilizer ready (BKPVSRF bit set to 1).
3:1	Reserved	Must be kept at reset value.
0	BKPVSEN	Backup voltage stabilizer enable This bit is set and cleared by software. When set, the backup voltage stabilizer (used to maintain the backup RAM content in Standby and VBAT modes) is enabled. When reset, the backup voltage stabilizer is off. The backup RAM can still be used in Run and Deep-sleep modes. However, its content will be lost in Standby and VBAT modes. If BREN is set, the application must wait till the backup voltage stabilizer ready flag (BKPVSRF) is set. Which indicate that the data written into the SRAM will be maintained in Standby and VBAT modes. 0: Backup voltage stabilizer disabled. 1: Backup voltage stabilizer enabled.

9.4.4. Control register 2 (PMU_CTL2)

Address offset: 0x0C

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:8	VUVDO_DNF[7:0]	VUVD analog output digital noise filter These bits are used to configure the digital noise filter on VUVD analog output. The digital filter will filter spikes with a length of up to VUVDO_DNF[7:0] * 1024 * T_{pclk4}. 00000000: Digital filter is disabled. 00000001: Digital filter is enabled and filter spikes with a length of up to 1024 * T_{pclk4}. ... 11111111: Digital filter is enabled and filter spikes with a length of up to 255*1024 * T_{pclk4}.
7:0	VOVDO_DNF[7:0]	VOVD analog output digital noise filter

These bits are used to configure the digital noise filter on VOVD analog output. The digital filter will filter spikes with a length of up to VOVDO_DNF[7:0] * 1024 * T_{pclk4}

00000000: Digital filter is disabled.

00000001: Digital filter is enabled and filter spikes with a length of up to 1024 * T_{pclk4}.

...

11111111: Digital filter is enabled and filter spikes with a length of up to 255*1024 * T_{pclk4}.

9.4.5. Control register 3 (PMU_CTL3)

Address offset: 0x10

Reset value: 0x0000 0004

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
ESCHTF	PKCAUHTF	MIPI_LDO_RDY	MIPI_LDO_BYP	MIPI_LDO_EN	USB33RF	USBSEN	VUSB33DEN	ESCHTEN	PKCAUHTEN	MIPIHTEN	MIPIHTF	ESCSW	PKCAUSW	MIPISW	Reserved
r	r	r	rw	rw	r	rw	rw	rw	rw	rw	r	rw	rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				MIPI_PDR_EN	MIPI_PDR_LEV	VCEN	VCRSEL	Reserved	MIPI_RDY_FALL	Reserved			DVSEN	Reserved	BYPASS
				rw	rw	rw	rw		rc_w1				rw		rw

Bits	Fields	Descriptions
31	ESCHTF	ESC power switch ack flag 0: Power-off complete 1: Power-on complete
30	PKCAUHTF	PKCAU power switch ack flag 0: Power-off complete 1: Power-on complete
29	MIPI_LDO_RDY	The MIPI_LDO output voltage ready signal. 0: Switch off, no output of internal VDD09DSI。 1: Switch on, output internal VDD09DSI.
28	MIPI_LDO_BYP	MIPI_LDO BYPASS mode enable signal 0: BYPASS mode disabled 1: BYPASS mode enabled
27	MIPI_LDO_EN	MIPI_LDO enable signal 0: Enable MIPI LDO 1: Disable MIPI LDO
26	USB33RF	USB supply ready flag bit 0: USB33 supply not ready.

		1: USB33 supply ready.
25	USBSEN	USB voltage stabilizer enable. This bit is set and cleared by software. 0: USB voltage stabilizer disabled 1: USB voltage stabilizer enabled
24	VUSB33DEN	VDD33USB voltage level detector enable bit This bit is set and cleared by software. 0: VDD33USB voltage level detector disabled 1: VDD33USB voltage level detector enabled
23	ESCHTEN	ESC output isolation enable 0:Output isolation 1:Output connection
22	PKCAUHTEN	PKCAU output isolation enable 0:Output isolation 1:Output connection
21	MIPIHTEN	MIPI output isolation enable 0:Output isolation 1:Output connection
20	MIPIHTF	Mipi power switch ack flag 0: Power-off complete 1: Power-on complete
19	ESCSW	Powers off the ESC. 0: Disconnect the ESC power 1: Connect the ESC power
18	PKCAUSW	Powers off the PKCAU. 0: Disconnect the PKCAU power 1: Connect the PKCAU power
17	MIPISW	Powers off the MIPI. 0: Disconnect the MIPI power 1: Connect the MIPI power
16:12	Reserved	Must be kept at reset value.
11	MIPI_PDR_EN	MIPI_LDO VDD09DSI PDR function enable signal 0: VDD09DSI PDR function disabled 1: VDD09DSI PDR function enabled
10	MIPI_PDR_LEV	MIPI_LDO VDD09DSI PDR threshold selection control signal: 0: PDR falling threshold is 0.775V, POR rising threshold is 0.8V 1: PDR falling threshold is 0.675V, POR rising threshold is 0.7V

9	VCEN	VBAT battery charging enable 0: Disable VBAT battery charging 1: Enable VBAT battery charging
8	VCRSEL	VBAT battery charging resistor selection 0: 5kOhms resistor is selected for charging VBAT battery 1: 1.5kOhms resistor is selected for charging VBAT battery
6	MIPI_RDY_FALL	The MIPI_LDO_RDY signal performs falling edge detection, sets and stores a flag, which is software-readable, to record the occurrence of undervoltage reset on VDD09DSI.
7:3	Reserved	Must be kept at reset value.
2	DVSEN	Step-down voltage stabilizer enable bit This bit is set and cleared by software. 0: Step-down voltage stabilizer disabled 1: Step-down voltage stabilizer enabled
1	Reserved	Must be kept at reset value.
0	BYPASS	Power management unit bypass control bit This bit is set and cleared by software. 0: Power management unit normal operation. 1: Power management unit bypassed, voltage monitoring still active.

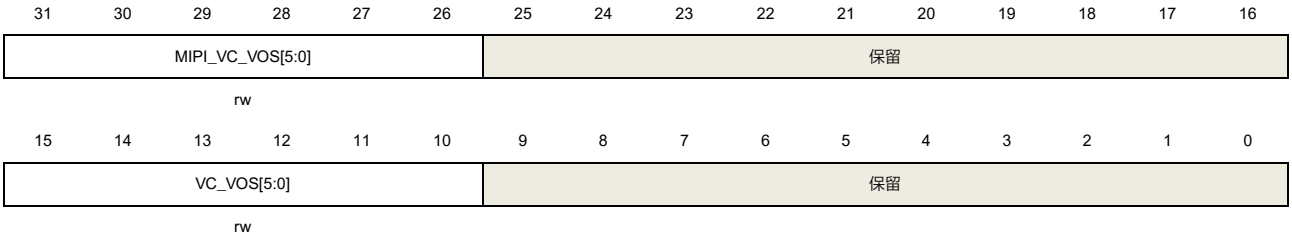
9.4.6. Control register 4 (PMU_CTL4)

Address offset: 0x14

Reset value: 0x0000 0000 (when VOSCFGEN = 0) / 0xB400 B400 (when VOSCFGEN = 1)

This register can be configured to any value only when VOSCFGEN in SYSCFG_PMCFG1 is enabled

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



Bits	Fields	Descriptions
31:26	MIPI_VC_VOS[5:0]	VCORE_DSI voltage level selection: 10mV step 00_0000: 0.52V 00_0001: 0.52V ..... 00_0111:0.52V 00_1000:0.53V

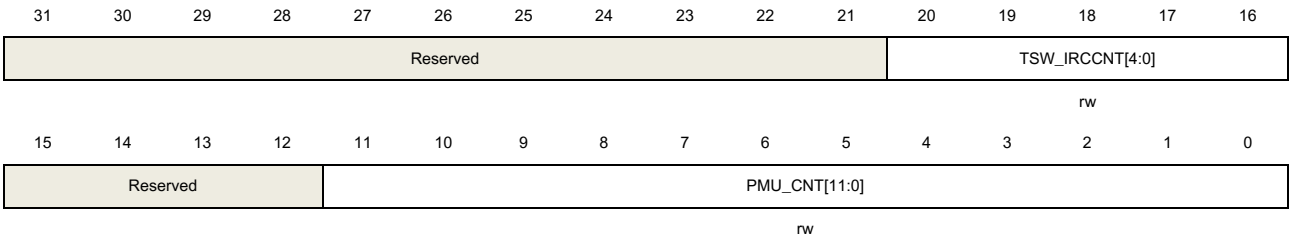
		..... 10_1100:0.89V 10_1101:0.90V 10_1110:0.91V ..... 11_1110:1.07V 11_1111:1.08V"
25:16	Reserved	Must be kept at reset value.
15:10	VC_VOS[5:0]	VCORE voltage level selection: 10mV step 00_0000: 0.52V 00_0001: 0.52V ..... 00_0111:0.52V 00_1000:0.53V ..... 10_1100:0.89V 10_1101:0.90V 10_1110:0.91V ..... 11_1110:1.07V 11_1111:1.08V
9:0	Reserved	Must be kept at reset value.

9.4.7. Parameter register (PMU_PAR)

Address offset: 0x18

Reset value: 0x000A 0000

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



Bits	Fields	Descriptions
31:21	Reserved	Must be kept at reset value.
20:16	TSW_IRCCNT[4:0]	When enter Deep-sleep, switch to LPIRC4M / IRC64M (confirmed by DSPWUSSEL) clock. Wait the LPIRC4M / IRC64M (confirmed by DSPWUSSEL) counter and then set stop_state signal. The default is 10 clocks.

15:12	Reserved	Must be kept at reset value.
11:0	PMU_CNT[11:0]	Exit Deep-sleep mode wait time count configure bits When exit Deep-sleep mode, before open system clock, it recommend 5us~50us delay.

10. Reset and clock unit (RCU)

10.1. Reset control unit (RCTL)

10.1.1. Overview

GD32H77x_78x reset control unit 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. The resets can be triggered by an external signal, internal events and the reset generators. More information about these resets will be described in the following sections.

10.1.2. Function overview

Power reset

The power reset is generated by either an external reset as power on and power down reset (POR / PDR reset), brownout reset (BOR 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 and it will be de-asserted when the power reset completed. 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:

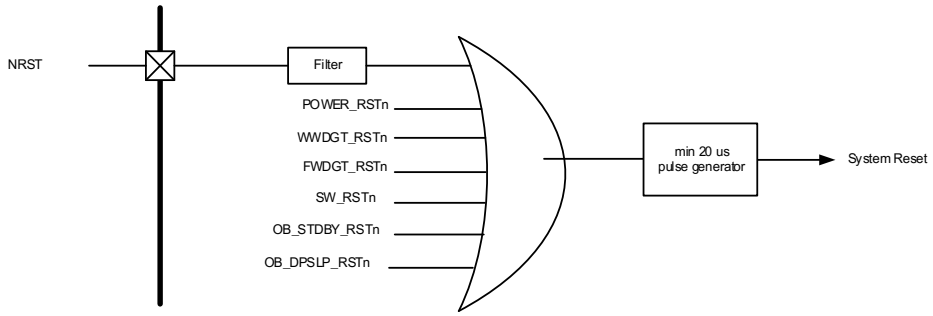
- A power reset (POWER_RSTn).
- A external pin reset (NRST).
- A window watchdog timer reset (WWDGT_RSTn).
- A free watchdog timer reset (FWDGT_RSTn).
- The SYSRESETREQ bit in Cortex®-M7 application interrupt and reset control register is set (SW_RSTn).
- Reset generated when entering Standby mode when resetting nRST_STDBY bit in user option bytes (OB_STDBY_RSTn).
- Reset generated when entering Deep-sleep mode when resetting nRST_DPSLP bit in user option bytes (OB_DPSLP_RSTn).

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

A system reset pulse generator guarantees low level pulse duration of 20 μs for each reset

source (external or internal reset).

Figure 10-1. 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 (VDD or VBAT power on, if both supplies have previously been powered off).

Note: The BKPSRAM is not reset by backup domain reset.

10.2. Clock control unit (CCTL)

10.2.1. Overview

The clock control unit provides a range of frequencies and clock functions. These include an Internal 64M RC oscillator (IRC64M), an Internal 48M RC oscillator (IRC48M), a High Speed crystal oscillator (HXTAL), a Low Speed Internal 32K RC oscillator (IRC32K), a Low Speed crystal oscillator (LXTAL), a Low Power Internal 4M RC oscillator (LPIRC4M), four Phase Lock Loop (PLL), a HXTAL clock monitor, a LXTAL clock monitor, clock prescalers, clock multiplexers and clock gating circuitry.

The clocks of the AXI, AHB, APB and Cortex®-M7 are derived from the system clock (CK_SYS) which can source from the IRC64M, HXTAL, LPIRC4M or PLL0. The maximum operating frequency of the system clock (CK_SYS) can be up to 600 MHz (applicable to GD32H77x), 700 MHz (applicable to the general mode of GD32H78x) or 750 MHz (applicable to the high-performance mode of GD32H78x).

Figure 10-2. Clock tree of GD32H77x

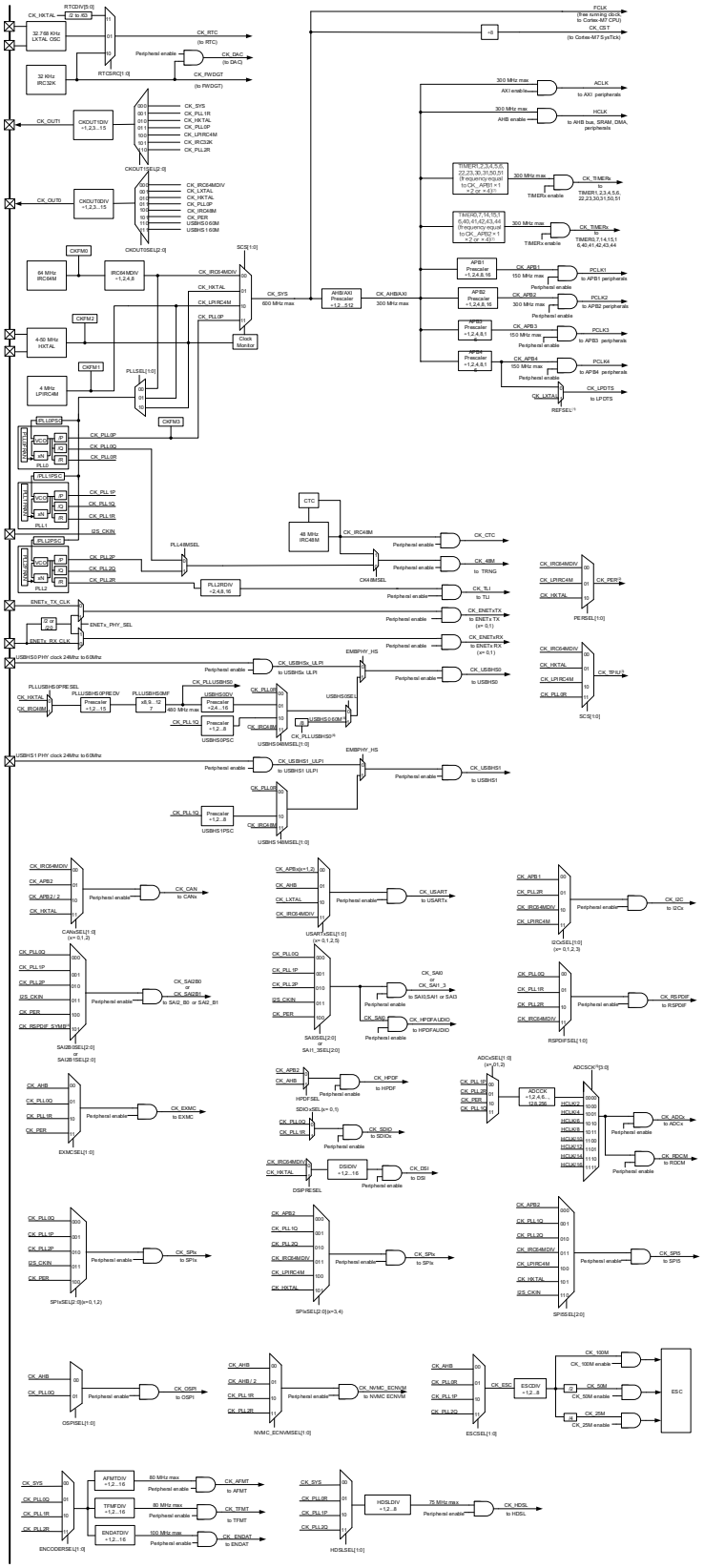
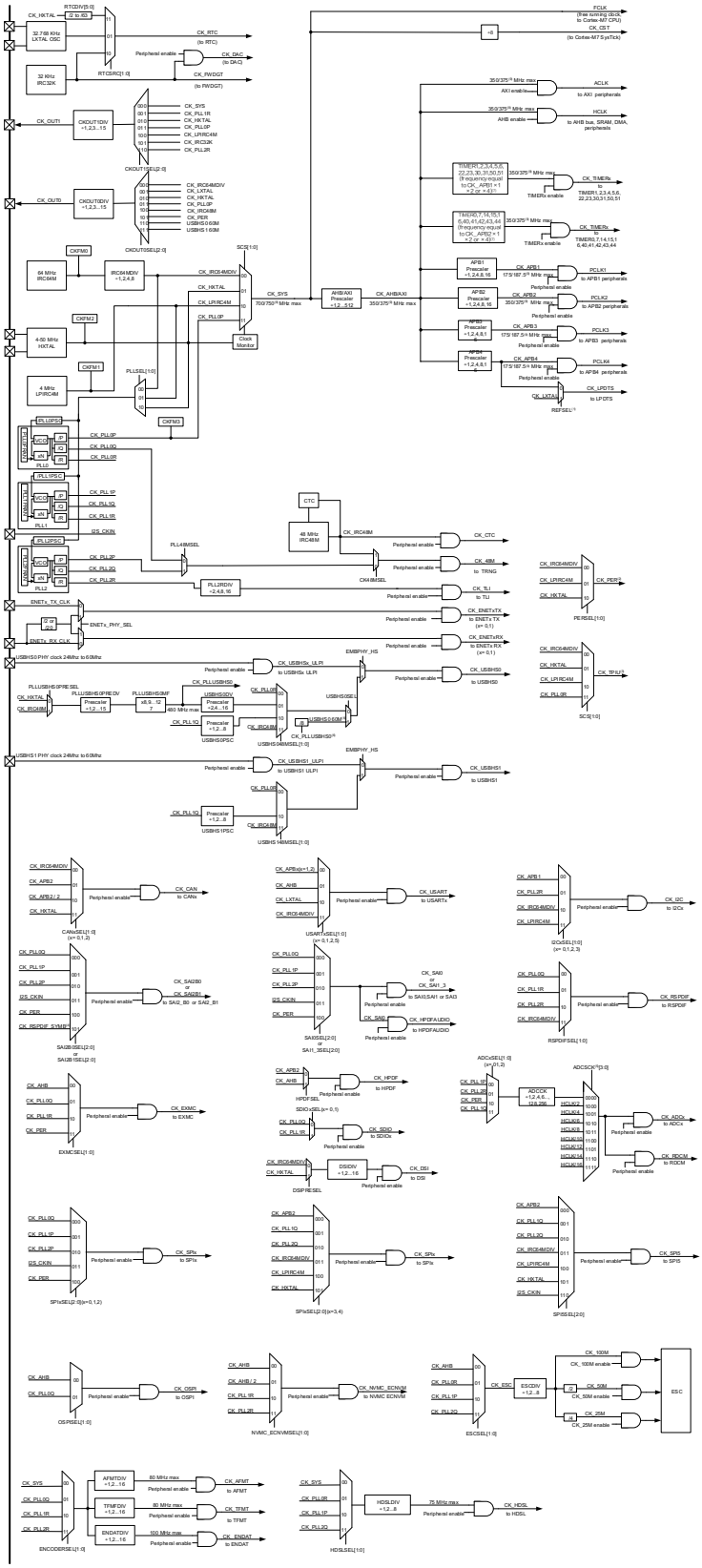


Figure 10-3. Clock tree of GD32H78x



(1). The REFSEL is LPDTS reference clock which is clocked by CK_APB4 or CK_LXTAL, refer to ***Figure 29-1. LPDTS block diagram***.

(2). The CK_PER is peripheral clock which is clocked by CK_IRC64MDIV, CK_LPIRC4M or CK_HXTAL.

(3). The CK_TPIU is Trace Port Interface Unit (TPIU) clock which is clocked by CK_IRC64MDIV, CK_HXTAL, CK_LPIRC4M or CK_PLL0R.

(4). The CK_RSPDIF_SYMB is RSPDIF channel symbol clock, refer to ***Figure 51-1. RSPDIF block diagram***.

(5). The ADCSCK is ADC synchronization clock selection bits, refer to ***Sync control register (ADC_SYNCCTL)***

(6). USBHS0 60M is USBHS0 internal PHY 60M input source clock, refer to ***Internal embedded PHY***. When USBHS0 60M is selected as the USBHS0 clock,CK_PLLUSBHS0 should be configured at 480Mhz.

(7). TIMER clock frequency, refer to the TIMERSEL bit in the RCU_CFG1 register.

(8). The RDCM clock is from ADC0.

(9). Applicable to the high-performance mode of GD32H78x.

The frequency of AXI, AHB, APB4, APB3, APB2 and the APB1 domains can be configured by each prescaler. The maximum frequency of the AXI / AHB and the APB4 / APB3 / APB2 / APB1 domains is 300 MHz / 300 MHz / 150 MHz / 150 MHz / 300 MHz / 150 MHz (applicable to GD32H77x), 350 MHz / 350 MHz / 175 MHz / 175 MHz / 350 MHz / 175 MHz (applicable to the general mode of GD32H78x) or 375 MHz / 375 MHz / 187.5 MHz / 187.5 MHz / 375 MHz / 187.5 MHz (applicable to the high-performance mode of GD32H78x). The Cortex System Timer (SysTick) external clock is clocked with the system clock (CK_SYS) divided by 8. The SysTick can work either with this clock or with the system clock (CK_SYS), configurable in the SysTick control and status register.

The ADCs are clocked by the clock of PLL1P, PLL2R, CK_PER ,CK_PLL1Q selected by the ADCxSEL and then prescaler by ADCCK, or by the clock of AHB divided by 2, 4, 6, 8, 10, 12, 14, 16.

The TIMERs are clocked by the clock divided from CK_AHB. The frequency of TIMERs clock is equal to CK_APBx, twice the CK_APBx or four times the CK_APBx. Please refer to TIMERSEL bit in RCU_CFG1 for detail.

The TRNG are clocked by the clock of CK_48M. The CK_48M is selected from the clock of PLL0Q, the clock of PLL2P or the clock of IRC48M by PLL48MSEL and CK48MSEL bit in RCU_ADDCTL0 register. The TRNG supports clock switch dynamically.

The USBHS ULPI is clocked by external ULPI PHY clock or the internal clock , which select by EMBPHY_HS in USBHS_GUSBCS register. The USBHS internal clock is configured by the USBHSxSEL bits in RCU_USBCLKCTL register.

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

The USART is clocked by the clock of CK_APBx, CK_AHB, CK_LXTAL or CK_IRC64MDIV which defined by USARTxSEL bits in RCU_CFG1 register. The USART supports clock switch dynamically.

The I2C is clocked by the clock of CK_APB1, CK_PLL2R, CK_IRC64MDIV or CK_LPIRC4M which defined by I2CxSEL bits in RCU_CFG0 register or RCU_CFG3 register. The I2C

supports clock switch dynamically.

The SPI0(I2S0), SPI1(I2S1) and SPI2(I2S2) are clocked by the clock of CK_PLL0Q, CK_PLL1P, CK_PLL2P, I2S_CKIN or CK_PER which defined by SPIxSEL(x = 0,1,2) bits in RCU_CFG5 register. The SPI0(I2S0), SPI1(I2S1) and SPI2(I2S2) support clock switch dynamically.

The SPI3 and SPI4 are clocked by the clock of CK_APB2, CK_PLL1Q, CK_PLL2Q, CK_IRC64MDIV, CK_LPIRC4M or CK_HXTAL which defined by SPIxSEL(x = 3,4) bits in RCU_CFG5 register. The SPI3 / SPI4 supports clock switch dynamically.

The SPI5(I2S5) is clocked by the clock of CK_APB2, CK_PLL1Q, CK_PLL2Q, CK_IRC64MDIV, CK_LPIRC4M, CK_HXTAL or I2S_CKIN which defined by SPI5SEL bits in RCU_CFG5 register. The SPI5 / I2S5 supports clock switch dynamically.

The OSPI is clocked by the clock of CK_AHB or CK_PLL0Q which defined by OSPISEL bits in RCU_CFG4 register. The OSPI supports clock switch dynamically.

The LPDTS is clocked by the clock of CK_APB4 or CK_LXTAL.

The CAN is clocked by the clock of CK_HXTAL, CK_APB2, CK_APB2 / 2 or CK_IRC64MDIV which defined by CANxSEL bits in RCU_CFG1 register. The CAN supports clock switch dynamically.

The RSPDIF is clocked by the clock of CK_PLL0Q, CK_PLL1R, CK_PLL2R or CK_IRC64MDIV which defined by RSPDIFSEL bits in RCU_CFG1 register. The RSPDIF supports clock switch dynamically.

The SAI2 is clocked by the clock of CK_PLL0Q, CK_PLL1P, CK_PLL2P, I2S_CKIN, CK_PER or CK_RSPDIF_SYMB which defined by SAI2B0SEL bits or SAIB1SEL bits in RCU_CFG2 register. The SAI2 supports clock switch dynamically.

The SAI0, SAI1and SAI3 are clocked by the clock of CK_PLL0Q, CK_PLL1P, CK_PLL2P, I2S_CKIN or CK_PER which defined by SAI0SEL bits or SAI1_3SEL bits in RCU_CFG2 register. The SAI0, SAI1and SAI3 supports clock switch dynamically.

The HPDF is clocked by the clock of CK_AHB or CK_APB2 which defined by HPDFSEL bit in RCU_CFG1 register. The HPDF supports clock switch dynamically.

The HPDF_AUDIO is clocked by the clock of CK_PLL0Q, CK_PLL1P, CK_PLL2P, I2S_CKIN or CK_PER which defined by SAI0SEL bits in RCU_CFG2 register.

The EXMC is clocked by the clock of CK_AHB, CK_PLL0Q, CK_PLL1R or CK_PER which defined by EXMCSEL bits in RCU_CFG4 register. The EXMC supports clock switch dynamically.

The SDIO is clocked by the clock of CK_PLL0Q or CK_PLL1R which defined by SDIOxSEL bits in RCU_CFG3 or RCU_CFG4 register. The SDIO supports clock switch dynamically.

The TLI is clocked by the clock of PLL2R divided by 2, 4, 8, 16 which defined by PLL2RDIV bits in RCU_CFG1 register.

The ESC is clocked by the clock of CK_AHB, CK_PLL0R, CK_PLL1P or CK_PLL2Q which defined by ESCSEL bits in RCU_CFG1 register and then divided by ESCDIV. The ESC supports clock switch dynamically.

The RSPDIF is clocked by the clock of CK_PLL0Q, CK_PLL1R, CK_PLL2R or CK_IRC64MDIV which defined by RSPDIFSEL bits in RCU_CFG1 register. The RSPDIF supports clock switch dynamically.

The NVMC is clocked by the clock of CK_AHB, CK_AHB / 2, CK_PLL1R or CK_PLL2R which defined by NVMC_ECNVMSEL bits in RCU_CFG1 register. The NVMC supports clock switch dynamically.

The ENCODER (AFMT, TFMT, ENDAT) is clocked by the clock of CK_SYS, CK_PLL0Q, CK_PLL1R or CK_PLL2R which defined by ENCODERSEL bits in RCU_CFG6 register and then divided by AFMTDIV, TFMTDIV or ENDATDIV. The ENCODER (AFMT, TFMT, ENDAT) supports clock switch dynamically.

The HDSL is clocked by the clock of CK_SYS, CK_PLL0R, CK_PLL1P or CK_PLL2Q which defined by HDSLSEL bits in RCU_CFG6 register and then divided by HDSLDIV. The HDSL supports clock switch dynamically.

The ENETx TX / RX are clocked by External PIN (ENETx_TX_CLK / ENETx_RX_CLK), which select by ENET0_PHY_SEL bit or ENET1_PHY_SEL bit in SYSCFG_PMCFG0 register.

The RTC is clocked by LXTAL clock or IRC32K clock or HXTAL clock divided by 2 to 63 (defined by RTCDIV bits in RCU_CFG0) which select by RTCSRC bit in backup domain control register (RCU_BDCTL). After the RTC select HXTAL clock divided by 2 to 63 (defined by RTCDIV bits in RCU_CFG0), the clock disappeared when the 0.9V core domain power off. After the RTC select IRC32K, the clock disappeared when V_{DD} power off. When 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.

10.2.2. Characteristics

- 4 to 50 MHz High Speed crystal oscillator (HXTAL).
- Internal 64 MHz RC oscillator (IRC64M).
- Internal 48 MHz RC oscillator (IRC48M).
- 32,768 Hz Low Speed crystal oscillator (LXTAL).
- Internal 32KHz RC oscillator (IRC32K).
- Low Power Internal 4M RC oscillator (LPIRC4M).
- PLL clock source can be HXTAL, LPIRC4M or IRC64M.
- Integer and fraction factors of PLLs.
- PLLs fraction factors can be modified at runtime.
- Peripheral clock switch dynamically.
- HXTAL clock monitor.
- LXTAL clock monitor.

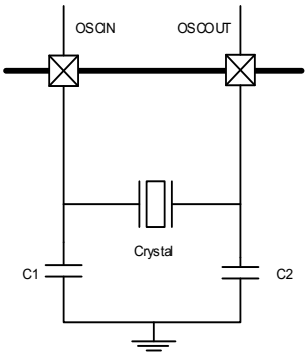
- Clock frequency monitor.

10.2.3. Function overview

High speed crystal oscillator (HXTAL)

The high speed external crystal oscillator (HXTAL), which has a frequency from 4 to 50 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.

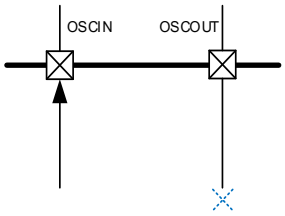
Figure 10-4. HXTAL clock source



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

Select external clock bypass mode by setting the HXTALBPS and HXTALEN bits in the control register RCU_CTL. The CK_HXTAL is equal to the external clock which drives the OSCIN pin. During bypass mode, the signal is connected to OSCIN, and OSCOUT remains in the suspended state, as shown in ***Figure 10-5. HXTAL clock source in bypass mode***. The CK_HXTAL is equal to the external clock which drives the OSCIN pin.

Figure 10-5. HXTAL clock source in bypass mode



Internal 64M RC oscillators (IRC64M)

The internal 64M RC oscillator, IRC64M, has a fixed frequency of 64 MHz and is the default clock source selection for the CPU when the device is powered up. If the IRC64MDIV[1:0] bits in RCU_ADDCTL1 register is configured, the CK_IRC64MDIV can output 8, 16, 32, 64MHz clock. The IRC64M oscillator provides a lower cost type clock source as no external components are required. The IRC64M RC oscillator can be switched on or off using the IRC64MEN bit in the control register RCU_CTL. The IRC64MSTB flag in the control register RCU_CTL is used to indicate if the internal 64M RC oscillator is stable. The start-up time of the IRC64M oscillator is shorter than the HXTAL crystal oscillator. An interrupt can be generated if the related interrupt enable bit, IRC64MSTBIE, in the Clock Interrupt Register, RCU_INT, is set when the IRC64M becomes stable. The CK_IRC64MDIV clock can also be used as the system clock source or the PLL input clock.

The frequency accuracy of the IRC64M can be calibrated by the manufacturer, but its operating frequency is still less accurate than HXTAL. The application requirements, environment and cost will determine which oscillator type is selected.

If the HXTAL or PLL0P is the system clock source, to minimize the time required for the system to recover from the Deep-sleep Mode, the hardware forces the CK_IRC64MDIV or CK_LPIR4M clock to be the system or kernel clock when the system initially wakes-up.

Internal 48M RC oscillators (IRC48M)

The internal 48M RC oscillator, IRC48M, has a fixed frequency of 48 MHz. The IRC48M oscillator provides a lower cost type clock source as no external components are required when USBHS / TRNG used. The IRC48M RC oscillator can be switched on or off using the IRC48MEN bit in the RCU_ADDCTL0 Register. The IRC48MSTB flag in the RCU_ADDCTL0 Register is used to indicate if the internal 48M RC oscillator is stable. An interrupt can be generated if the related interrupt enable bit, IRC48MSTBIE, in the RCU_ADDINT Register, is set when the IRC48M becomes stable. The IRC48M clock is used for the clocks of USBHS / TRNG.

The frequency accuracy of the IRC48M can be calibrated by the manufacturer, but its operating frequency is still not enough accurate because the USBHS need the frequency must between 48MHz+/-1%. A hardware automatically dynamic trim performed in CTC unit adjust the IRC48M to the needed frequency.

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.

Low Power Internal 4M RC oscillator (LPIRC4M)

The low power internal 4M RC oscillator has a frequency of about 4 MHz and is a low power clock source for use as the system clock or PLL input clock. The LPIRC4M offers a low cost clock source as no external components are required. The LPIRC4M RC oscillator can be switched on or off by using the LPIRC4MEN bit in the additional clock control register 1 (RCU_ADDCTL1). The LPIRC4MSTB flag in the clock interrupt register (RCU_INT) will indicate if the LPIRC4M clock is stable. An interrupt can be generated if the related interrupt enable bit LPIRC4MSTBIE in the clock interrupt register (RCU_INT) is set when the LPIRC4M becomes stable.

The frequency accuracy of the LPIRC4M can be calibrated by the manufacturer. After reset, the calibration is loaded in LPIRC4MCALIB bits in RCU_ADDCTL1 register.

If the HXTAL or PLL0P is the system clock source, to minimize the time required for the system to recover from the Deepsleep Mode, the hardware forces the CK_IRC64MDIV or CK_LPIR4M clock to be the system or kernel clock when the system initially wakes-up.

Phase locked loop (PLL)

There are four internal Phase Locked Loop, the PLL0, PLL1, PLL2 and PLLUSBHS0. The PLL0, PLL1 and PLL2 support integer and fraction factors and the fraction factors can be modified at run time. Otherwise, the PLL0, PLL1 and PLL2 can generate P / Q / R clock output respectively. The PLL0P could be used to generator system clock (no more than maximum operating frequency of the system clock).

For each PLL, When PLLxFRAEN bit is '1' and PLLxFRAN value is not '0' in RCU_PLLxFRA register, the PLLx is in fraction mode, for example:

$$CK_PLL0VCO = CK_PLL0VCOSRC * \left(PLL0N + \frac{PLL0FRAN}{2^{13}}\right) \quad (10\text{-}1)$$

Otherwise, the PLLx is in integer mode, for example:

$$CK_PLL0VCO = CK_PLL0VCOSRC * PLL0N \quad (10\text{-}2)$$

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

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

The PLL2 can be switched on or off by using the PLL2EN bit in the RCU_CTL register. The PLL2STB flag in the RCU_CTL register will indicate if the PLL2 clock is stable. An interrupt can be generated if the related interrupt enable bit, PLL2STBIE, in the RCU_INT register, is set as the PLL2 becomes stable.

The PLLUSBHS0 can be switched on or off by using the PLLUSBHS0EN bit in the RCU_ADDCTL1 register. The PLLUSBHS0STB flag in the RCU_ADDCTL1 register will indicate if the PLLUSBHS0 clock is stable. An interrupt can be generated if the related interrupt enable bit, PLLUSBHS0STBIE, in the RCU_ADDINT register, is set as the PLLUSBHS0 becomes stable.

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

Peripheral clock switch dynamically

If the peripheral has two more source clock selection, the peripheral can switches from the running clock to another present clock dynamically. Otherwise, peripheral clock can not be switched. Only TRNG / USART / I2C / SPI / RSPDIF / SAI / SDIO / EXMC / CAN / HPDF peripherals support clock switch dynamically.

System clock (CK_SYS) selection

After the system reset, the default CK_SYS source will be IRC64M and can be switched to HXTAL or LPIRC4M or CK_PLL0P by changing the system clock switch bits, SCS, in the clock configuration register 0, RCU_CFG0. When the SCS value is changed, the CK_SYS will continue to operate using the original clock source until the target clock source is stable. When a clock source is directly or indirectly (by PLL0P) 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®-M7. If the HXTAL is selected as the clock source of CK_SYS or PLL0 and CK_PLL0P used as system clock, the HXTAL failure will force the CK_SYS source to IRC64M and the PLL0 will be disabled automatically. If the HXTAL is selected as the clock source of any PLLs, the HXTAL failure will force the PLL closed automatically.

LXTAL clock monitor (LCKM)

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

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

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

A 4-bits plus one counter will work at IRC32K domain when LCKMD enable. If the LXTAL clock has stuck at 0/1 error or when it is outside 20 KHz-40 KHz, the counter will overflow. The LXTAL clock failure will been found. Once the LXTAL failure is detected, the LXTAL clock stuck interrupt flag, LCKMIF, in the clock interrupt register, RCU_INT, will be set and the LXTAL failure event will be generated.

Clock frequency monitor (CKFM)

The clock frequency monitor can use IRC48M to monitor IRC64M, LPIRC4M, HXTAL and PLL0P clock frequency range. For IRC64M and PLL0P, 100 IRC48M clock cycle is used as the monitor window. For LPIRC4M and HXTAL, 1000 IRC48M clock cycle is used as the monitor window. User can configure the monitoring range of the clock frequency by configuring the RCU_CKFMCFGx (x = 0, 1, 2) register. If the corresponding interrupt is enabled and the clock frequency failure flag is set, the interrupt will be occurred.

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

Clock output capability

The clock output capability is ranging from 32 kHz to maximum operating frequency of the system clock. There are several clock signals can be selected via the CK_OUT0 clock source selection bits, CKOUT0SEL, in the Clock configuration register 2 (RCU_CFG2). The corresponding GPIO pin should be configured in the properly alternate function I/O (AFIO) mode to output the selected clock signal. The CK_OUT1 is selected by CKOUT1SEL, in the clock configuration register 2 (RCU_CFG2).

Table 10-1. Clock output 0 source select

Clock Source 0 Selection bits	Clock Source
000	CK_IRC64MDIV
001	CK_LXTAL
010	CK_HXTAL
011	CK_PLL0P
100	CK_IRC48M
101	CK_PER
110	USBHS0 60M

Table 10-2. Clock output 1 source select

Clock Source 1 Selection bits	Clock Source
000	CK_SYS
001	CK_PLL1R
010	CK_HXTAL
011	CK_PLL0P
100	CK_LPIRC4M
101	CK_IRC32K
110	CK_PLL2R

The CK_OUT0 frequency can be reduced by a configurable binary divider, controlled by the CKOUT0DIV bits, in the clock configuration register (RCU_CFG2).

The CK_OUT1 frequency can be reduced by a configurable binary divider, controlled by the CKOUT1DIV bits, in the clock configuration register (RCU_CFG2).

RTC clock measure

The three clock source of RTC clock, LXTAL, IRC32K, HXTAL divided by 2 to 63 (defined by RTCDIV bits in RCU_CFG0), can be measured by TIMER. Then the user can get the clocks frequency, and adjust the RTC and FWDGT counter. Please refer to TIMER15_CI0_SEL bits and TIMER16_CI0_SEL bits *Timer input selection register 6 (SYSCFG_TIMERCISEL6)* register for detail.

10.3. Register definition

RCU base address: 0x5802 4400

10.3.1. Control register (RCU_CTL)

Address offset: 0x00
Reset value: 0xC000 8040

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
IRC64MSTB	IRC64MEN	PLL2STB	PLL2EN	PLL1STB	PLL1EN	PLL0STB	PLL0EN	Reserved			HXTALRDYRST	CKMEN	HXTALBPS	HXTALSTB	HXTALEN
r	rw	r	rw	r	rw	r	rw				rw	rw	rw	r	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IRC64MCALIB[8:0]								IRC64MADJ[6:0]							
r								rw							

Bits	Fields	Descriptions
31	IRC64MSTB	IRC64M internal 64MHz RC oscillator stabilization flag Set by hardware to indicate if the IRC64M oscillator is stable and ready for use. 0: IRC64M oscillator is not stable 1: IRC64M oscillator is stable
30	IRC64MEN	Internal 64MHz RC oscillator enable Set and reset by software. This bit cannot be reset if the IRC64M 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 64 MHz RC oscillator disabled 1: Internal 64 MHz RC oscillator enabled
29	PLL2STB	PLL2 clock stabilization flag Set by hardware to indicate if the PLL2 output clock is stable and ready for use. 0: PLL2 is not stable 1: PLL2 is stable
28	PLL2EN	PLL2 enable Set and reset by software. Reset by hardware when entering Deep-sleep or Standby mode. 0: PLL2 is switched off 1: PLL2 is switched on
27	PLL1STB	PLL1 clock stabilization flag Set by hardware to indicate if the PLL1 output clock is stable and ready for use. 0: PLL1 is not stable

		1: PLL1 is stable
26	PLL1EN	PLL1 enable Set and reset by software. Reset by hardware when entering Deep-sleep or Standby mode. 0: PLL1 is switched off 1: PLL1 is switched on
25	PLL0STB	PLL0 clock stabilization flag Set by hardware to indicate if the PLL0 output clock is stable and ready for use. 0: PLL0 is not stable 1: PLL0 is stable
24	PLL0EN	PLL0 enable Set and reset by software. This bit cannot be reset if the PLL0 clock is used as the system clock. Reset by hardware when entering Deep-sleep or Standby mode. 0: PLL0 is switched off 1: PLL0 is switched on
23:21	Reserved	Must be kept at reset value.
20	HXTALRDYRST	HXTAL ready reset 0: HXTAL ready no reset 1: HXTAL ready reset
19	CKMEN	HXTAL clock monitor enable 0: Disable the High speed 4 ~ 50 MHz crystal oscillator (HXTAL) clock monitor 1: Enable the High speed 4 ~ 50 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 IRC64M 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 IRC64M internal RC oscillator regardless of the control bit, IRC64MEN, state.
18	HXTALBPS	High speed crystal oscillator (HXTAL) clock bypass mode enable The HXTALBPS bit can be written only if the HXTALEN is 0. 0: Disable the HXTAL Bypass mode 1: Enable the HXTAL Bypass mode in which the HXTAL output clock is equal to the input clock.
17	HXTALSTB	High speed crystal oscillator (HXTAL) clock stabilization flag Set by hardware to indicate if the HXTAL oscillator is stable and ready for use. 0: HXTAL oscillator is not stable 1: HXTAL oscillator is stable
16	HXTALEN	High speed crystal oscillator (HXTAL) enable

		Set and reset by software. This bit cannot be reset if the HXTAL clock is used as the system clock or the PLL0 input clock when PLL0P clock is selected to the system clock. Reset by hardware when entering Deep-sleep or Standby mode. 0: High speed 4 ~ 50 MHz crystal oscillator disabled 1: High speed 4 ~ 50 MHz crystal oscillator enabled
15:7	IRC64MCALIB[8:0]	Internal 64MHz RC Oscillator calibration value These bits are load automatically at power on.
6:0	IRC64MADJ[6:0]	Internal 64MHz RC Oscillator clock trim adjust value These bits are set by software. The trimming value is these bits IRC64MADJ[6:0] added to the IRC64MCALIB[8:0] bits. The trimming value should trim the IRC64M to 64 MHz +/- 1%.

10.3.2. PLL0 register (RCU_PLL0)

Address offset: 0x04

Reset value: 0x0100 2020

To configure the PLL0 clock, refer to the following formula:

CK_PLL0VCOSRC = CK_PLL0SRC / PLL0PSC

CK_PLL0VCO = CK_PLL0VCOSRC × (PLL0N + PLL0FRAN / 2^{13})

CK_PLL0R = CK_PLL0VCO / PLL0R

CK_PLL0P = CK_PLL0VCO / PLL0P

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
PLLSTBSRC	PLL0R[6:0]							Reserved	PLL0P[6:0]						
w	rw								rw						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	PLL0N[8:0]									PLL0PSC[5:0]					
	rw									rw					

Bits	Fields	Descriptions
31	PLLSTBSRC	PLLs stabilization signal sources. 0: Analog logic 1: Digital logic
30:24	PLL0R[6:0]	The PLL0R output frequency division factor from PLL0 VCO clock Set and reset by software when the PLL0 is disable. These bits used to generator PLL0R output clock (CK_PLL0R) from PLL0 VCO clock (CK_PLL0VCO). The CK_PLL0VCO is described in PLL0N bits in RCU_PLL0 register. 0000000: CK_PLL0R = CK_PLL0VCO 0000001: CK_PLL0R = CK_PLL0VCO / 2 (default) 0000010: CK_PLL0R = CK_PLL0VCO / 3.

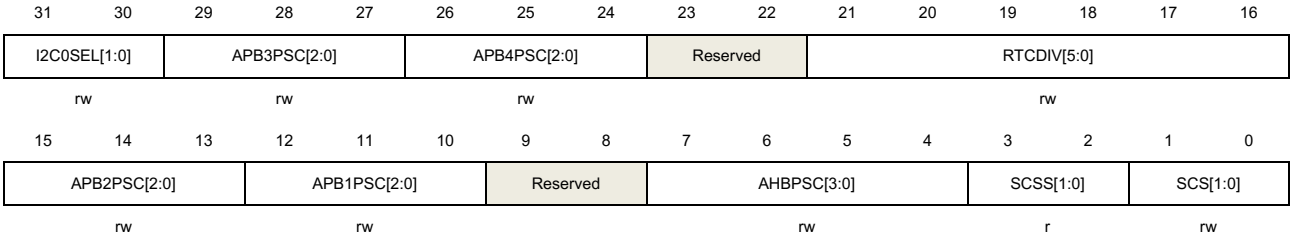
		0000011: CK_PLL0R = CK_PLL0VCO / 4 0000100: CK_PLL0R = CK_PLL0VCO / 5 … 1111111: CK_PLL0R = CK_PLL0VCO / 128
23	Reserved	Must be kept at reset value.
22:16	PLL0P[6:0]	The PLL0P output frequency division factor from PLL0 VCO clock Set and reset by software when the PLL0 is disable. These bits used to generator PLL0P output clock (CK_PLL0P) from PLL0 VCO clock (CK_PLL0VCO). The CK_PLL0P is used to system clock (no more than maximum operating frequency of the system clock). The CK_PLL0VCO is described in PLL0N bits in RCU_PLL0 register. 0000000: CK_PLL0P = CK_PLL0VCO (default) 0000001: CK_PLL0P = CK_PLL0VCO / 2 0000010: CK_PLL0P = CK_PLL0VCO / 3 0000011: CK_PLL0P = CK_PLL0VCO / 4 0000100: CK_PLL0P = CK_PLL0VCO / 5 … 1111111: CK_PLL0P = CK_PLL0VCO / 128
15	Reserved	Must be kept at reset value.
14:6	PLL0N[8:0]	The PLL0 VCO clock multiplication factor Set and reset by software (only use word / half-word write) when the PLL0 is disable. These bits used to generator PLL0 VCO clock (CK_PLL0VCO) from PLL0 VCO source clock (CK_PLL0VCOSRC). The CK_PLL0VCOSRC is described in PLL0PSC bits in RCU_PLL0 register. **Note:** The frequency of CK_PLL0VCO is between 150MHz to 836MHz. When PLL0 fractional latch is disabled, the value of PLL0N must: 9<=PLL0N<=512 When PLL0 fractional latch is enabled, the value of PLL0N must: 12<=PLL0N<=508 000000000: Reserved … 000000111: Reserved 000001000: PLL0N = 9 … 001000000: PLL0N = 65 001000001: PLL0N = 66 … 111111111: PLL0N = 512
5:0	PLL0PSC[5:0]	The PLL0 VCO source clock prescaler Set and reset by software when the PLL0 is disable. These bits used to generate the clock of PLL0 VCO source clock (CK_PLL0VCOSRC) from PLL0 source clock (CK_PLL0SRC) which described in PLLSEL in RCU_PLLALL register. The VCO source clock is between 1M to 16MHz

000000: reserved
000001: CK_PLL0SRC
000010: CK_PLL0SRC / 2
000011: CK_PLL0SRC / 3
…
111111: CK_PLL0SRC / 63

10.3.3. Clock configuration register 0 (RCU_CFG0)

Address offset: 0x08
Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:30	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_PLL2R selected as I2C0 source clock 10: CK_IRC64MDIV selected as I2C0 source clock 11: CK_LPIRC4M selected as I2C0 source clock
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	APB4PSC[2:0]	APB4 prescaler selection Set and reset by software to control the APB4 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

23:22	Reserved	Must be kept at reset value.
21:16	RTCDIV[5:0]	RTC clock divider factor Set and reset by software. These bits is used to generator clock for RTC (no more than 1MHz) from HXTAL clock. 000000: no clock for RTC 000001: no clock for RTC 000010: CK_HXTAL / 2 000011: CK_HXTAL / 3 … 111111: CK_HXTAL / 63
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 and AXI prescaler selection Set and reset by software to control the AHB and AXI 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_IRC64MDIV as the CK_SYS source 01: Select CK_HXTAL as the CK_SYS source 10: Select CK_LPIRC4M as the CK_SYS source

		11: Select CK_PLL0P 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 IRC64MDIV 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_IRC64MDIV as the CK_SYS source 01: Select CK_HXTAL as the CK_SYS source 10: Select CK_LPIRC4M as the CK_SYS source 11: Select CK_PLL0P as the CK_SYS source

10.3.4. 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	LPIRC4MSTBIC	LPIRC4MSTBIE	LPIRC4MSTBIF	CKMIC	PLL2STBIC	PLL1STBIC	PLL0STBIC	HXTALSTBIC	IRC64MSTBIC	LXTALSTBIC	IRC32KSTBIC
			w	r	w	rw	r	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	PLL2STBIE	PLL1STBIE	PLL0STBIE	HXTALSTBIE	IRC64MSTBIE	LXTALSTBIE	IRC32KSTBIE	CKMIF	PLL2STBIF	PLL1STBIF	PLL0STBIF	HXTALSTBIF	IRC64MSTBIF	LXTALSTBIF	IRC32KSTBIF
	rw	rw	rw	rw	rw	rw	rw	r	r	r	r	r	r	r	r

Bits	Fields	Descriptions
31:29	Reserved	Must be kept at reset value.
28	LCKMIC	LXTAL clock stuck interrupt clear Write 1 by software to reset the LCKMIF flag. 0: Not reset LCKMIF flag 1: Reset LCKMIF flag
27	LCKMIF	LXTAL clock stuck interrupt flag Set by hardware when the LXTAL clock is stuck. Reset when setting the LCKMIC bit by software. 0: Clock operating normally 1: LXTAL clock stuck
26	LPIRC4MSTBIC	LPIRC4M stabilization interrupt clear Write 1 by software to reset the LPIRC4MSTBIF flag. 0: Not reset LPIRC4MSTBIF flag

		1: Reset LPIRC4MSTBIF flag
25	LPIRC4MSTBIE	LPIRC4M stabilization interrupt enable Set and reset by software to enable/disable the LPIRC4M stabilization interrupt. 0: Disable the LPIRC4M stabilization interrupt 1: Enable the LPIRC4M stabilization interrupt
24	LPIRC4MSTBIF	LPIRC4M stabilization interrupt flag Set by hardware when the LPIRC4M is stable and the LPIRC4MSTBIE bit is set. Reset when setting the IRC64MSTBIC bit by software. 0: No IRC64M stabilization interrupt generated 1: IRC64M stabilization interrupt generated
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	PLL2STBIC	PLL2 stabilization interrupt clear Write 1 by software to reset the PLL2STBIF flag. 0: Not reset PLL2STBIF flag 1: Reset PLL2STBIF flag
21	PLL1STBIC	PLL1 stabilization interrupt clear Write 1 by software to reset the PLL1STBIF flag. 0: Not reset PLL1STBIF flag 1: Reset PLL1STBIF flag
20	PLL0STBIC	PLL0 stabilization interrupt clear Write 1 by software to reset the PLL0STBIF flag. 0: Not reset PLL0STBIF flag 1: Reset PLL0STBIF flag
19	HXTALSTBIC	HXTAL stabilization interrupt clear Write 1 by software to reset the HXTALSTBIF flag. 0: Not reset HXTALSTBIF flag 1: Reset HXTALSTBIF flag
18	IRC64MSTBIC	IRC64M stabilization interrupt clear Write 1 by software to reset the IRC64MSTBIF flag. 0: Not reset IRC64MSTBIF flag 1: Reset IRC64MSTBIF 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	Reserved	Must be kept at reset value.
14	PLL2STBIE	PLL2 stabilization interrupt enable Set and reset by software to enable/disable the PLL2 stabilization interrupt. 0: Disable the PLL2 stabilization interrupt 1: Enable the PLL2 stabilization interrupt
13	PLL1STBIE	PLL1 stabilization interrupt enable Set and reset by software to enable/disable the PLL1 stabilization interrupt. 0: Disable the PLL1 stabilization interrupt 1: Enable the PLL1 stabilization interrupt
12	PLL0STBIE	PLL0 stabilization interrupt enable Set and reset by software to enable/disable the PLL0 stabilization interrupt. 0: Disable the PLL0 stabilization interrupt 1: Enable the PLL0 stabilization interrupt
11	HXTALSTBIE	HXTAL stabilization interrupt enable Set and reset by software to enable/disable the HXTAL stabilization interrupt 0: Disable the HXTAL stabilization interrupt 1: Enable the HXTAL stabilization interrupt
10	IRC64MSTBIE	IRC64M stabilization interrupt enable Set and reset by software to enable/disable the IRC64M stabilization interrupt 0: Disable the IRC64M stabilization interrupt 1: Enable the IRC64M stabilization interrupt
9	LXTALSTBIE	LXTAL stabilization interrupt enable LXTAL stabilization interrupt enable/disable control 0: Disable the LXTAL stabilization interrupt 1: Enable the LXTAL stabilization interrupt
8	IRC32KSTBIE	IRC32K stabilization interrupt enable IRC32K stabilization interrupt enable/disable control 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	PLL2STBIF	PLL2 stabilization interrupt flag Set by hardware when the PLL2 is stable and the PLL2STBIE bit is set.

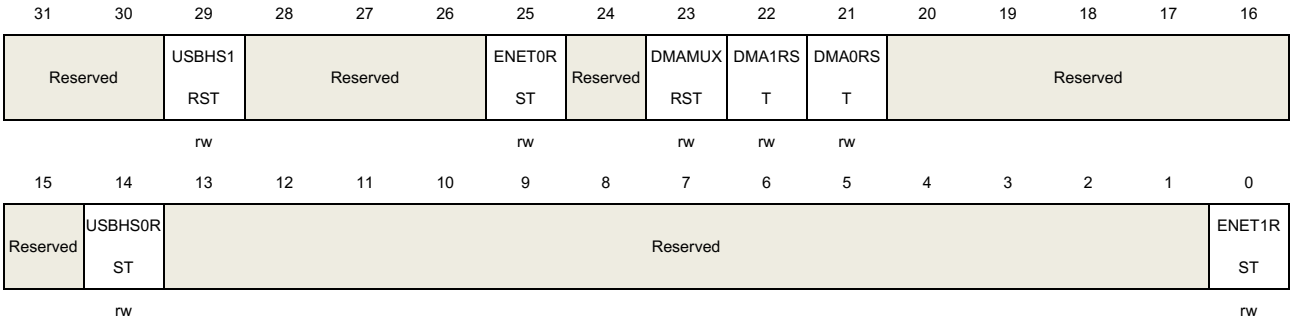
		Reset when setting the PLL2STBIC bit by software. 0: No PLL2 stabilization interrupt generated 1: PLL2 stabilization interrupt generated
5	PLL1STBIF	PLL1 stabilization interrupt flag Set by hardware when the PLL1 is stable and the PLL1STBIE bit is set. Reset when setting the PLL1STBIC bit by software. 0: No PLL1 stabilization interrupt generated 1: PLL1 stabilization interrupt generated
4	PLL0STBIF	PLL0 stabilization interrupt flag Set by hardware when the PLL0 is stable and the PLL0STBIE bit is set. Reset when setting the PLL0STBIC bit by software. 0: No PLL0 stabilization interrupt generated 1: PLL0 stabilization interrupt generated
3	HXTALSTBIF	HXTAL stabilization interrupt flag Set by hardware when the High speed 4 ~ 50 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	IRC64MSTBIF	IRC64M stabilization interrupt flag Set by hardware when the Internal 64 MHz RC oscillator clock is stable and the IRC64MSTBIE bit is set. Reset when setting the IRC64MSTBIC bit by software. 0: No IRC64M stabilization interrupt generated 1: IRC64M 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

10.3.5. 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).



Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29	USBHS1RST	USBHS1 reset This bit is set and reset by software. 0: No reset 1: Reset the USBHS1
28:26	Reserved	Must be kept at reset value.
25	ENET0RST	Ethernet0 reset This bit is set and reset by software. 0: No reset 1: Reset the Ethernet0
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
20:15	Reserved	Must be kept at reset value.
14	USBHS0RST	USBHS0 reset This bit is set and reset by software.

		0: No reset 1: Reset the USBHS0
13:1	Reserved	Must be kept at reset value.
0	ENET1RST	Ethernet1 reset This bit is set and reset by software. 0: No reset 1: Reset the Ethernet1

10.3.6. 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															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			ESCRST	Reserved	FFTRST	PKCAURST	Reserved	TMURST	TRNGRST	Reserved	HAURST	CAURST	SDIO1RST	FACRST	DCIRST
			rw		rw	rw		rw	rw		rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:13	Reserved	Must be kept at reset value.
12	ESCRST	ESC reset This bit is set and reset by software. 0: No reset 1: Reset the ESC
11	Reserved	Must be kept at reset value.
10	FFTRST	FFT reset This bit is set and reset by software. 0: No reset 1: Reset the FFT
9	PKCAURST	PKCAU reset This bit is set and reset by software. 0: No reset 1: Reset the PKCAU
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	Reserved	Must be kept at reset value.
4	HAURST	HAU reset This bit is set and reset by software. 0: No reset 1: Reset the HAU
3	CAURST	CAU reset This bit is set and reset by software. 0: No reset 1: Reset the CAU
2	SDIO1RST	SDIO1 reset This bit is set and reset by software. 0: No reset 1: Reset the SDIO1
1	FACRST	FAC reset This bit is set and reset by software. 0: No reset 1: Reset the FAC
0	DCIRST	DCI reset This bit is set and reset by software. 0: No reset 1: Reset the DCI and GPSI

10.3.7. 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						RTDEC1RST	RTDEC0RST	DSIRST	OSPI1RST	OSPI0RST	OSPIMRST	MDMARST	SDIO0RST	IPARST	EXMCRST

rw rw rw rw rw rw rw rw rw rw

Bits	Fields	Descriptions
31:10	Reserved	Must be kept at reset value.
9	RTDEC1RST	RTDEC1 reset This bit is set and reset by software. 0: No reset 1: Reset the RTDEC1
8	RTDEC0RST	RTDEC0 reset This bit is set and reset by software. 0: No reset 1: Reset the RTDEC0
7	DSIRST	DSI reset This bit is set and reset by software. 0: No reset 1: Reset the DSI
6	OSPI1RST	OSPI1 reset This bit is set and reset by software. 0: No reset 1: Reset the OSPI1
5	OSPI0RST	OSPI0 reset This bit is set and reset by software. 0: No reset 1: Reset the OSPI0
4	OSPIMRST	OSPIM reset This bit is set and reset by software. 0: No reset 1: Reset the OSPIM
3	MDMARST	MDMA reset This bit is set and reset by software. 0: No reset 1: Reset the MDMA
2	SDIO0RST	SDIO0 reset This bit is set and reset by software. 0: No reset 1: Reset the SDIO0
1	IPARST	IPA reset This bit is set and reset by software. 0: No reset

		1: Reset the IPA
0	EXMCRST	EXMC reset This bit is set and reset by software. 0: No reset 1: Reset the EXMC

10.3.8. AHB4 reset register (RCU_AHB4RST)

Address offset: 0x1C

Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HWSEMRST	CRCRST	Reserved				PKRST	PJRST	PHRST	PGRST	PFRST	PERST	PDRST	PCRST	PBRST	PARST
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	HWSEMRST	HWSEM reset This bit is set and reset by software. 0: No reset 1: Reset the HWSEM
14	CRCRST	CRC reset This bit is set and reset by software. 0: No reset 1: Reset the CRC
13:10	Reserved	Must be kept at reset value.
9	PKRST	GPIO port K reset This bit is set and reset by software. 0: No reset 1: Reset the GPIO port K
8	PJRST	GPIO port J reset This bit is set and reset by software. 0: No reset 1: Reset the GPIO port J
7	PHRST	GPIO port H reset

This bit is set and reset by software.
0: No reset
1: Reset the GPIO port H

6	PGRST	GPIO port G reset This bit is set and reset by software. 0: No reset 1: Reset the GPIO port G
5	PFRST	GPIO port F reset This bit is set and reset by software. 0: No reset 1: Reset the GPIO port F
4	PERST	GPIO port E reset This bit is set and reset by software. 0: No reset 1: Reset the GPIO port E
3	PDRST	GPIO port D reset This bit is set and reset by software. 0: No reset 1: Reset the GPIO port D
2	PCRST	GPIO port C reset This bit is set and reset by software. 0: No reset 1: Reset the GPIO port C
1	PBRST	GPIO port B reset This bit is set and reset by software. 0: No reset 1: Reset the GPIO port B
0	PARST	GPIO port A reset This bit is set and reset by software. 0: No reset 1: Reset the GPIO port A

10.3.9. 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

UART7RST	UART6RST	DACRST	Reserved	CTCRST	Reserved		I2C3RST	I2C2RST	I2C1RST	I2C0RST	UART4RST	UART3RST	USART2RST	USART1RST	MDIORST
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

SPI2RST	SPI1RST	RSPDIFRST	Reserved	TIMER51RST	TIMER50RST	TIMER31RST	TIMER30RST	TIMER23RST	TIMER22RST	TIMER6RST	TIMER5RST	TIMER4RST	TIMER3RST	TIMER2RST	TIMER1RST
rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31	UART7RST	UART7 reset This bit is set and reset by software. 0: No reset 1: Reset the UART7
30	UART6RST	UART6 reset This bit is set and reset by software. 0: No reset 1: Reset the UART6
29	DACRST	DAC reset This bit is set and reset by software. 0: No reset 1: Reset the DAC
28	Reserved	Must be kept at reset value.
27	CTCRST	CTC reset This bit is set and reset by software. 0: No reset 1: Reset the CTC
26:25	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	MDIORST	MDIO reset This bit is set and reset by software. 0: No reset 1: Reset the MDIO
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	RSPDIFRST	RSPDIF reset This bit is set and reset by software. 0: No reset 1: Reset the RSPDIF
12	Reserved	Must be kept at reset value.
11	TIMER51RST	TIMER51 reset This bit is set and reset by software.

		0: No reset 1: Reset the TIMER51
10	TIMER50RST	TIMER50 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER50
9	TIMER31RST	TIMER31 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER31
8	TIMER30RST	TIMER30 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER30
7	TIMER23RST	TIMER23 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER23
6	TIMER22RST	TIMER22 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER22
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

10.3.10. 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	EDOUTRST	TIMER44RST	TIMER43RST	TIMER42RST	TIMER41RST	TIMER40RST	SAI2RST	SAI1RST	SAI0RST	SPI5RST	SPI4RST	HPDFRST	TIMER16RST	TIMER15RST	TIMER14RST
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
SAI3RST	Reserved	SPI3RST	SPI0RST	Reserved	ADC2RST	ADC1RST	ADC0RST	Reserved		USART5RST	USART0RST	Reserved		TIMER7RST	TIMER0RST
rw		rw	rw		rw	rw	rw			rw	rw			rw	rw

Bits	Fields	Descriptions
31	TRIGSELRST	TRIGSEL reset This bit is set and reset by software. 0: No reset 1: Reset the TRIGSEL
30	EDOUTRST	EDOUT reset This bit is set and reset by software. 0: No reset 1: Reset the EDOUT
29	TIMER44RST	TIMER44 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER44
28	TIMER43RST	TIMER43 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER43
27	TIMER42RST	TIMER42 reset This bit is set and reset by software.

		0: No reset 1: Reset the TIMER42
26	TIMER41RST	TIMER41 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER41
25	TIMER40RST	TIMER40 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER40
24	SAI2RST	SAI2 reset This bit is set and reset by software. 0: No reset 1: Reset the SAI2
23	SAI1RST	SAI1 reset This bit is set and reset by software. 0: No reset 1: Reset the SAI1
22	SAI0RST	SAI0 reset This bit is set and reset by software. 0: No reset 1: Reset the SAI0
21	SPI5RST	SPI5 reset This bit is set and reset by software. 0: No reset 1: Reset the SPI5
20	SPI4RST	SPI4 reset This bit is set and reset by software. 0: No reset 1: Reset the SPI4
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 TIMER15R
16	TIMER14RST	TIMER14 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER14
15	SAI3RST	SAI3 reset This bit is set and reset by software. 0: No reset 1: Reset the SAI3
14	Reserved	Must be kept at reset value.
13	SPI3RST	SPI3 reset This bit is set and reset by software. 0: No reset 1: Reset the SPI3
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	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
7:6	Reserved	Must be kept at reset value.
5	USART5RST	USART5 reset This bit is set and reset by software. 0: No reset 1: Reset the USART5
4	USART0RST	USART0 reset This bit is set and reset by software.

		0: No reset 1: Reset the USART0
3:2	Reserved	Must be kept at reset value.
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

10.3.11. 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															

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

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	WWDGTRST	WWDGT reset This bit is set and reset by software. 0: No reset 1: Reset the WWDGT
0	TLIRST	TLI reset This bit is set and reset by software. 0: No reset 1: Reset the TLI

10.3.12. APB4 reset register (RCU_APB4RST)

Address offset: 0x2C
Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											PMURST	LPDTSRST	VREFRST	CMPRST	SYSCFGRST
											rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:5	Reserved	Must be kept at reset value.
4	PMURST	PMU reset This bit is set and reset by software. 0: No reset 1: Reset the PMU
3	LPDTSRST	LPDTS reset This bit is set and reset by software. 0: No reset 1: Reset the LPDTS
2	VREFRST	VREF reset This bit is set and reset by software. 0: No reset 1: Reset the VREF
1	CMPRST	CMP reset This bit is set and reset by software. 0: No reset 1: Reset the CMP
0	SYSCFGRST	SYSCFG reset This bit is set and reset by software. 0: No reset 1: Reset the SYSCFG

10.3.13. 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
Reserved	USBHS1ULPIEN	USBHS1EN	ENET0PTPEN	ENET0RXEN	ENET0TXEN	ENET0EN	Reserved	DMAMUXEN	DMA1EN	DMA0EN	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

USBHS0ULPIEN	USBHS0EN	Reserved	ENET1PTPEN	ENET1RXEN	ENET1TXEN	ENET1EN
rw	rw		rw	rw	rw	rw

Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30	USBHS1ULPIEN	USBHS1 ULPI clock enable This bit is set and reset by software. 0: Disabled USBHS1 ULPI clock 1: Enabled USBHS1 ULPI clock
29	USBHS1EN	USBHS1 clock enable This bit is set and reset by software. 0: Disabled USBHS1 clock 1: Enabled USBHS1 clock
28	ENET0PTPEN	Ethernet0 PTP clock enable This bit is set and reset by software. 0: Disabled Ethernet0 PTP clock 1: Enabled Ethernet0 PTP clock
27	ENET0RXEN	Ethernet0 RX clock enable This bit is set and reset by software. 0: Disabled Ethernet0 RX clock 1: Enabled Ethernet0 RX clock
26	ENET0TXEN	Ethernet0 TX clock enable This bit is set and reset by software. 0: Disabled Ethernet0 TX clock 1: Enabled Ethernet0 TX clock
25	ENET0EN	Ethernet0 clock enable This bit is set and reset by software. 0: Disabled Ethernet0 clock 1: Enabled Ethernet0 clock
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	USBHS0ULPIEN	USBHS0 ULPI clock enable This bit is set and reset by software. 0: Disabled USBHS0 ULPI clock 1: Enabled USBHS0 ULPI clock
14	USBHS0EN	USBHS0 clock enable This bit is set and reset by software. 0: Disabled USBHS0 clock 1: Enabled USBHS0 clock
13:4	Reserved	Must be kept at reset value.
3	ENET1PTPEN	Ethernet1 PTP clock enable This bit is set and reset by software. 0: Disabled Ethernet1 PTP clock 1: Enabled Ethernet1 PTP clock
2	ENET1RXEN	Ethernet1 RX clock enable This bit is set and reset by software. 0: Disabled Ethernet1 RX clock 1: Enabled Ethernet1 RX clock
1	ENET1TXEN	Ethernet1 TX clock enable This bit is set and reset by software. 0: Disabled Ethernet1 TX clock 1: Enabled Ethernet1 TX clock
0	ENET1EN	Ethernet1 clock enable This bit is set and reset by software. 0: Disabled Ethernet1 clock 1: Enabled Ethernet1 clock

10.3.14. AHB2 enable register (RCU_AHB2EN)

Address offset: 0x34

Reset value: 0x0000 0100

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			ESCEN	Reserved	FFTEN	PKCAUEN	RAMECCMU1EN	TMUEN	TRNGEN	GPSIEN	HAUEN	CAUEN	SDIO1EN	FACEN	DCIEN
			rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:13	Reserved	Must be kept at reset value.
12	ESCEN	ESC clock enable This bit is set and reset by software. 0: Disabled ESC clock 1: Enabled ESC clock
11	Reserved	Must be kept at reset value.
10	FFTEN	FFT clock enable This bit is set and reset by software. 0: Disabled FFT clock 1: Enabled FFT clock
9	PKCAUEN	PKCAU clock enable This bit is set and reset by software. 0: Disabled PKCAU clock 1: Enabled PKCAU clock
8	RAMECCMU1EN	RAMECCMU1 clock enable This bit is set and reset by software. 0: Disabled RAMECCMU1 clock 1: Enabled RAMECCMU1 clock
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	GPSIEN	GPSI clock enable This bit is set and reset by software. 0: Disabled GPSI clock 1: Enabled GPSI clock
4	HAUEN	HAU clock enable

		This bit is set and reset by software. 0: Disabled HAU clock 1: Enabled HAU clock
3	CAUEN	CAU clock enable This bit is set and reset by software. 0: Disabled CAU clock 1: Enabled CAU clock
2	SDIO1EN	SDIO1 clock enable This bit is set and reset by software. 0: Disabled SDIO1 clock 1: Enabled SDIO1 clock
1	FACEN	FAC clock enable This bit is set and reset by software. 0: Disabled FAC clock 1: Enabled FAC clock
0	DCIEN	DCI clock enable This bit is set and reset by software. 0: Disabled DCI clock 1: Enabled DCI clock

10.3.15. AHB3 enable register (RCU_AHB3EN)

Address offset: 0x38

Reset value: 0x0000 8400

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															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPUEN	Reserved				RAMECCMU0EN	RTDEC1EN	RTDEC0EN	DSIEN	OSPI1EN	OSPI0EN	OSPIMEN	MDMAEN	SDIO0EN	IPAEN	EXMCEN
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	CPUEN	CPU clock enable This bit is set and reset by software. 0: Disabled CPU clock 1: Enabled CPU clock

14:11	Reserved	Must be kept at reset value.
10	RAMECCMU0EN	RAMECCMU0 clock enable This bit is set and reset by software. 0: Disabled RAMECCMU0 clock 1: Enabled RAMECCMU0 clock
9	RTDEC1EN	RTDEC1 clock enable This bit is set and reset by software. 0: Disabled RTDEC1 clock 1: Enabled RTDEC1 clock
8	RTDEC0EN	RTDEC0 clock enable This bit is set and reset by software. 0: Disabled RTDEC0 clock 1: Enabled RTDEC0 clock
7	DSIEN	DSI clock enable This bit is set and reset by software. 0: Disabled DSI clock 1: Enabled DSI clock
6	OSPI1EN	OSPI1 clock enable This bit is set and reset by software. 0: Disabled OSPI1 clock 1: Enabled OSPI1 clock
5	OSPI0EN	OSPI0 clock enable This bit is set and reset by software. 0: Disabled OSPI0 clock 1: Enabled OSPI0 clock
4	OSPIMEN	OSPIM clock enable This bit is set and reset by software. 0: Disabled OSPIM clock 1: Enabled OSPIM clock
3	MDMAEN	MDMA clock enable This bit is set and reset by software. 0: Disabled MDMA clock 1: Enabled MDMA clock
2	SDIO0EN	SDIO0 clock enable This bit is set and reset by software. 0: Disabled SDIO0 clock 1: Enabled SDIO0 clock
1	IPAEN	IPA clock enable This bit is set and reset by software.

		0: Disabled IPA clock 1: Enabled IPA clock
0	EXMCEN	EXMC clock enable This bit is set and reset by software. 0: Disabled EXMC clock 1: Enabled EXMC clock

10.3.16. AHB4 enable register (RCU_AHB4EN)

Address offset: 0x3C

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															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HWSEMEN	CRCEN	BKPSRAMEN	Reserved			PKEN	PJEN	PHEN	PGEN	PFEN	PEEN	PDEN	PCEN	PBEN	PAEN
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	HWSEMEN	HWSEM clock enable This bit is set and reset by software. 0: Disabled HWSEM clock 1: Enabled HWSEM clock
14	CRCEN	CRC clock enable This bit is set and reset by software. 0: Disabled CRC clock 1: Enabled CRC clock
13	BKPSRAMEN	Backup SRAM clock enable This bit is set and reset by software. 0: Disabled Backup SRAM clock 1: Enabled Backup SRAM clock
12:10	Reserved	Must be kept at reset value.
9	PKEN	GPIO port K clock enable This bit is set and reset by software. 0: Disabled GPIO port K clock 1: Enabled GPIO port K clock

8	PJEN	GPIO port J clock enable This bit is set and reset by software. 0: Disabled GPIO port J clock 1: Enabled GPIO port J clock
7	PHEN	GPIO port H clock enable This bit is set and reset by software. 0: Disabled GPIO port H clock 1: Enabled GPIO port H clock
6	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
5	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
4	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
3	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
2	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
1	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
0	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

10.3.17. 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
UART7EN	UART6EN	DACEN	DACHOLDEN	CTCEN	Reserved		I2C3EN	I2C2EN	I2C1EN	I2C0EN	UART4EN	UART3EN	USART2EN	USART1EN	MDIOEN
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
SPI2EN	SPI1EN	RSPDIFEN	Reserved	TIMER51EN	TIMER50EN	TIMER31EN	TIMER30EN	TIMER23EN	TIMER22EN	TIMER6EN	TIMER5EN	TIMER4EN	TIMER3EN	TIMER2EN	TIMER1EN
rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31	UART7EN	UART7 clock enable This bit is set and reset by software. 0: Disabled UART7 clock 1: Enabled UART7 clock
30	UART6EN	UART6 clock enable This bit is set and reset by software. 0: Disabled UART6 clock 1: Enabled UART6 clock
29	DACEN	DAC clock enable This bit is set and reset by software. 0: Disabled DAC clock 1: Enabled DAC clock
28	DACHOLDEN	DAC hold 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
27	CTCEN	CTC clock enable This bit is set and reset by software. 0: Disabled CTC clock 1: Enabled CTC clock
26:25	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	MDIOEN	MDIO clock enable This bit is set and reset by software. 0: Disabled MDIO clock 1: Enabled MDIO clock
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	RSPDIFEN	RSPDIF clock enable

		This bit is set and reset by software. 0: Disabled RSPDIF clock 1: Enabled RSPDIF clock
12	Reserved	Must be kept at reset value.
11	TIMER51EN	TIMER51 clock enable This bit is set and reset by software. 0: Disabled TIMER51 clock 1: Enabled TIMER51 clock
10	TIMER50EN	TIMER50 clock enable This bit is set and reset by software. 0: Disabled TIMER50 clock 1: Enabled TIMER50 clock
9	TIMER31EN	TIMER31 clock enable This bit is set and reset by software. 0: Disabled TIMER31 clock 1: Enabled TIMER31 clock
8	TIMER30EN	TIMER30 clock enable This bit is set and reset by software. 0: Disabled TIMER30 clock 1: Enabled TIMER30 clock
7	TIMER23EN	TIMER23 clock enable This bit is set and reset by software. 0: Disabled TIMER23 clock 1: Enabled TIMER23 clock
6	TIMER22EN	TIMER22 clock enable This bit is set and reset by software. 0: Disabled TIMER22 clock 1: Enabled TIMER22 clock
5	TIMER6EN	TIMER6 clock enable This bit is set and reset by software. 0: Disabled TIMER6 clock 1: Enabled TIMER6 clock
4	TIMER5EN	TIMER5 clock enable This bit is set and reset by software. 0: Disabled TIMER5 clock 1: Enabled TIMER5 clock
3	TIMER4EN	TIMER4 clock enable This bit is set and reset by software. 0: Disabled TIMER4 clock

		1: Enabled TIMER4 clock
2	TIMER3EN	TIMER3 clock enable This bit is set and reset by software. 0: Disabled TIMER3 clock 1: Enabled TIMER3 clock
1	TIMER2EN	TIMER2 clock enable This bit is set and reset by software. 0: Disabled TIMER2 clock 1: Enabled TIMER2 clock
0	TIMER1EN	TIMER1 clock enable This bit is set and reset by software. 0: Disabled TIMER1 clock 1: Enabled TIMER1 clock

10.3.18. 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
TRIGSEL EN	EDOUTE N	TIMER44 EN	TIMER43 EN	TIMER42 EN	TIMER41 EN	TIMER40 EN	SAI2EN	SAI1EN	SAI0EN	SPI5EN	SPI4EN	HPDFEN	TIMER16 EN	TIMER15 EN	TIMER14 EN
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
SAI3EN	Reserved	SPI3EN	SPI0EN	Reserved	ADC2EN	ADC1EN	ADC0EN	Reserved		USART5 EN	USART0 EN	Reserved		TIMER7E N	TIMER0E N
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	EDOUTEN	EDOUT clock enable This bit is set and reset by software. 0: Disabled EDOUT clock 1: Enabled EDOUT clock
29	TIMER44EN	TIMER44 clock enable This bit is set and reset by software. 0: Disabled TIMER44 clock

		1: Enabled TIMER44 clock
28	TIMER43EN	TIMER43 clock enable This bit is set and reset by software. 0: Disabled TIMER43 clock 1: Enabled TIMER43 clock
27	TIMER42EN	TIMER42 clock enable This bit is set and reset by software. 0: Disabled TIMER42 clock 1: Enabled TIMER42 clock
26	TIMER41EN	TIMER41 clock enable This bit is set and reset by software. 0: Disabled TIMER41 clock 1: Enabled TIMER41 clock
25	TIMER40EN	TIMER40 clock enable This bit is set and reset by software. 0: Disabled TIMER40 clock 1: Enabled TIMER40 clock
24	SAI2EN	SAI2 clock enable This bit is set and reset by software. 0: Disabled SAI2 clock 1: Enabled SAI2 clock
23	SAI1EN	SAI1 clock enable This bit is set and reset by software. 0: Disabled SAI1 clock 1: Enabled SAI1 clock
22	SAI0EN	SAI0 clock enable This bit is set and reset by software. 0: Disabled SAI0 clock 1: Enabled SAI0 clock
21	SPI5EN	SPI5 clock enable This bit is set and reset by software. 0: Disabled SPI5 clock 1: Enabled SPI5 clock
20	SPI4EN	SPI4 clock enable This bit is set and reset by software. 0: Disabled SPI4 clock 1: Enabled SPI4 clock
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	SAI3EN	SAI3 clock enable This bit is set and reset by software. 0: Disabled SAI3 clock 1: Enabled SAI3 clock
14	Reserved	Must be kept at reset value.
13	SPI3EN	SPI3 clock enable This bit is set and reset by software. 0: Disabled SPI3 clock 1: Enabled SPI3 clock
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	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
7:6	Reserved	Must be kept at reset value.
5	USART5EN	USART5 clock enable This bit is set and reset by software. 0: Disabled USART5 clock 1: Enabled USART5 clock
4	USART0EN	USART0 clock enable This bit is set and reset by software. 0: Disabled USART0 clock 1: Enabled USART0 clock
3:2	Reserved	Must be kept at reset value.
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

10.3.19. 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															

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

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	WWDGTEN	WWDGT clock enable This bit is set and reset by software. 0: Disabled WWDGT clock

		1: Enabled WWDGT clock
0	TLIEN	TLI clock enable This bit is set and reset by software. 0: Disabled TLI clock 1: Enabled TLI clock

10.3.20. APB4 enable register (RCU_APB4EN)

Address offset: 0x4C

Reset value: 0x0000 0010

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											PMUEN	LPDTSEN	VREFEN	CMPEN	SYSCFGEN
											rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:5	Reserved	Must be kept at reset value.
4	PMUEN	PMU clock enable This bit is set and reset by software. 0: Disabled PMU clock 1: Enabled PMU clock
3	LPDTSEN	LPDTS clock enable This bit is set and reset by software. 0: Disabled LPDTS clock 1: Enabled LPDTS clock
2	VREFEN	VREF clock enable This bit is set and reset by software. 0: Disabled VREF clock 1: Enabled VREF clock
1	CMPEN	CMP clock enable This bit is set and reset by software. 0: Disabled CMP clock 1: Enabled CMP clock
0	SYSCFGEN	SYSCFG clock enable This bit is set and reset by software. 0: Disabled SYSCFG clock 1: Enabled SYSCFG clock

10.3.21. AHB1 sleep mode enable register (RCU_AHB1SPEN)

Address offset: 0x50

Reset value: 0x7EE3 C00F

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	USBHS1ULPISPEN	USBHS1SPEN	ENET0PTPSPEN	ENET0RXSPEN	ENET0TXSPEN	ENET0SPEN	Reserved	DMAMUXSPEN	DMA1SPEN	DMA0SPEN	Reserved			SRAM1SPEN	SRAM0SPEN
	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
USBHS0ULPISPEN	USBHS0SPEN	Reserved										ENET1PTPSPEN	ENET1RXSPEN	ENET1TXSPEN	ENET1SPEN
rw	rw											rw	rw	rw	rw

Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30	USBHS1ULPISPEN	USBHS1 ULPI clock enable when sleep mode This bit is set and reset by software. 0: Disabled USBHS1 ULPI clock when sleep mode 1: Enabled USBHS1 ULPI clock when sleep mode
29	USBHS1SPEN	USBHS1 clock enable when sleep mode This bit is set and reset by software. 0: Disabled USBHS1 clock when sleep mode 1: Enabled USBHS1 clock when sleep mode
28	ENET0PTPSPEN	Ethernet0 PTP clock enable when sleep mode This bit is set and reset by software. 0: Disabled Ethernet0 PTP clock when sleep mode 1: Enabled Ethernet0 PTP clock when sleep mode
27	ENET0RXSPEN	Ethernet0 RX clock enable when sleep mode This bit is set and reset by software. 0: Disabled Ethernet0 RX clock when sleep mode 1: Enabled Ethernet0 RX clock when sleep mode
26	ENET0TXSPEN	Ethernet0 TX clock enable when sleep mode This bit is set and reset by software. 0: Disabled Ethernet0 TX clock when sleep mode 1: Enabled Ethernet0 TX clock when sleep mode
25	ENET0SPEN	Ethernet0 clock enable when sleep mode This bit is set and reset by software.

		0: Disabled Ethernet0 clock when sleep mode 1: Enabled Ethernet0 clock when sleep mode
24	Reserved	Must be kept at reset value.
23	DMAMUXSPEN	DMAMUX clock enable when sleep mode This bit is set and reset by software. 0: Disabled DMAMUX clock when sleep mode 1: Enabled DMAMUX clock when sleep mode
22	DMA1SPEN	DMA1 clock enable when sleep mode This bit is set and reset by software. 0: Disabled DMA1 clock when sleep mode 1: Enabled DMA1 clock when sleep mode
21	DMA0SPEN	DMA0 clock enable when sleep mode This bit is set and reset by software. 0: Disabled DMA0 clock when sleep mode 1: Enabled DMA0 clock when sleep mode
20:18	Reserved	Must be kept at reset value.
17	SRAM1SPEN	SRAM1 clock enable when sleep mode This bit is set and reset by software. 0: Disabled SRAM1 clock when sleep mode 1: Enabled SRAM1 clock when sleep mode
16	SRAM0SPEN	SRAM0 clock enable when sleep mode This bit is set and reset by software. 0: Disabled SRAM0 clock when sleep mode 1: Enabled SRAM0 clock when sleep mode
15	USBHS0ULPISPEN	USBHS0 ULPI clock enable when sleep mode This bit is set and reset by software. 0: Disabled USBHS0 ULPI clock when sleep mode 1: Enabled USBHS0 ULPI clock when sleep mode
14	USBHS0SPEN	USBHS0 clock enable when sleep mode This bit is set and reset by software. 0: Disabled USBHS0 clock when sleep mode 1: Enabled USBHS0 clock when sleep mode
13:4	Reserved	Must be kept at reset value.
3	ENET1PTPSPEN	Ethernet1 PTP clock enable when sleep mode This bit is set and reset by software. 0: Disabled Ethernet1 PTP clock when sleep mode 1: Enabled Ethernet1 PTP clock when sleep mode
2	ENET1RXSPEN	Ethernet1 RX clock enable when sleep mode

		This bit is set and reset by software. 0: Disabled Ethernet1 RX clock when sleep mode 1: Enabled Ethernet1 RX clock when sleep mode
1	ENET1TXSPEN	Ethernet1 TX clock enable when sleep mode This bit is set and reset by software. 0: Disabled Ethernet1 TX clock when sleep mode 1: Enabled Ethernet1 TX clock when sleep mode
0	ENET1SPEN	Ethernet1 clock enable when sleep mode This bit is set and reset by software. 0: Disabled Ethernet1 clock when sleep mode 1: Enabled Ethernet1 clock when sleep mode

10.3.22. AHB2 sleep mode enable register (RCU_AHB2SPEN)

Address offset: 0x54

Reset value: 0x0000 1FFF

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			ESCSPEN	Reserved	FFTSPEN	PKCAUSPEN	RAMECCMU1SPEN	TMUSPEN	TRNGSPEN	GPSISPEN	HAUSPEN	CAUSPEN	SDIO1SPEN	FACSPEN	DCISPEN
			rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:13	Reserved	Must be kept at reset value.
12	ESCSPEN	ESC clock enable when sleep mode This bit is set and reset by software. 0: Disabled ESC clock when sleep mode 1: Enabled ESC clock when sleep mode
11	Reserved	Must be kept at reset value.
10	FFTSPEN	FFT clock enable when sleep mode This bit is set and reset by software. 0: Disabled FFT clock when sleep mode 1: Enabled FFT clock when sleep mode
9	PKCAUSPEN	PKCAU clock enable when sleep mode This bit is set and reset by software. 0: Disabled PKCAU clock when sleep mode

		1: Enabled PKCAU clock when sleep mode
8	RAMECCMU1SPEN	RAMECCMU1 clock enable when sleep mode This bit is set and reset by software. 0: Disabled RAMECCMU1 clock when sleep mode 1: Enabled RAMECCMU1 clock when sleep mode
7	TMUSPEN	TMU clock enable when sleep mode This bit is set and reset by software. 0: Disabled TMU clock when sleep mode 1: Enabled TMU clock when sleep mode
6	TRNGSPEN	TRNG clock enable when sleep mode This bit is set and reset by software. 0: Disabled TRNG clock when sleep mode 1: Enabled TRNG clock when sleep mode
5	GPSISPEN	GPSI enable when sleep mode This bit is set and reset by software. 0: Disabled GPSI clock when sleep mode 1: Enabled GPSI clock when sleep mode
4	HAUSPEN	HAU enable when sleep mode This bit is set and reset by software. 0: Disabled HAU clock when sleep mode 1: Enabled HAU clock when sleep mode
3	CAUSPEN	CAU clock enable when sleep mode This bit is set and reset by software. 0: Disabled CAU clock when sleep mode 1: Enabled CAU clock when sleep mode
2	SDIO1SPEN	SDIO1 clock enable when sleep mode This bit is set and reset by software. 0: Disabled SDIO1 clock when sleep mode 1: Enabled SDIO1 clock when sleep mode
1	FACSPEN	FAC clock enable when sleep mode This bit is set and reset by software. 0: Disabled FAC clock when sleep mode 1: Enabled FAC clock when sleep mode
0	DCISPEN	DCI clock enable when sleep mode This bit is set and reset by software. 0: Disabled DCI clock when sleep mode 1: Enabled DCI clock when sleep mode

10.3.23. AHB3 sleep mode enable register (RCU_AHB3SPEN)

Address offset: 0x58

Reset value: 0x0000 C7FF

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
NVMCSPEN	AXISRAMSPEN	Reserved			RAMECCMU0SPEN	RTDEC1SPEN	RTDEC0SPEN	DSISPEN	OSPI1SPEN	OSPI0SPEN	OSPIMSPEN	MDMASPEN	SDIO0SPEN	IPASPEN	EXMCSPEN
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	NVMCSPEN	NVMC clock enable when sleep mode This bit is set and reset by software. 0: Disabled NVMC clock when sleep mode 1: Enabled NVMC clock when sleep mode
14	AXISRAMSPEN	AXI SRAM clock enable when sleep mode This bit is set and reset by software. 0: Disabled AXI SRAM clock when sleep mode 1: Enabled AXI SRAM clock when sleep mode
13:11	Reserved	Must be kept at reset value.
10	RAMECCMU0SPEN	RAMECCMU0 clock enable when sleep mode This bit is set and reset by software. 0: Disabled RAMECCMU0 clock when sleep mode 1: Enabled RAMECCMU0 clock when sleep mode
9	RTDEC1SPEN	RTDEC1 clock enable when sleep mode This bit is set and reset by software. 0: Disabled RTDEC1 clock when sleep mode 1: Enabled RTDEC1 clock when sleep mode
8	RTDEC0SPEN	RTDEC0 clock enable when sleep mode This bit is set and reset by software. 0: Disabled RTDEC0 clock when sleep mode 1: Enabled RTDEC0 clock when sleep mode
7	DSISPEN	DSI clock enable when sleep mode This bit is set and reset by software. 0: Disabled DSI clock when sleep mode

		1: Enabled DSI clock when sleep mode
6	OSPI1SPEN	OSPI1 clock enable when sleep mode This bit is set and reset by software. 0: Disabled OSPI1 clock when sleep mode 1: Enabled OSPI1 clock when sleep mode
5	OSPI0SPEN	OSPI0 clock enable when sleep mode This bit is set and reset by software. 0: Disabled OSPI0 clock when sleep mode 1: Enabled OSPI0 clock when sleep mode
4	OSPIMSPEN	OSPIM clock enable when sleep mode This bit is set and reset by software. 0: Disabled OSPIM clock when sleep mode 1: Enabled OSPIM clock when sleep mode
3	MDMASPEN	MDMA clock enable when sleep mode This bit is set and reset by software. 0: Disabled MDMA clock when sleep mode 1: Enabled MDMA clock when sleep mode
2	SDIO0SPEN	SDIO0 clock enable when sleep mode This bit is set and reset by software. 0: Disabled SDIO0 clock when sleep mode 1: Enabled SDIO0 clock when sleep mode
1	IPASPEN	IPA clock enable when sleep mode This bit is set and reset by software. 0: Disabled IPA clock when sleep mode 1: Enabled IPA clock when sleep mode
0	EXMCSPEN	EXMC clock enable when sleep mode This bit is set and reset by software. 0: Disabled EXMC clock when sleep mode 1: Enabled EXMC clock when sleep mode

10.3.24. AHB4 sleep mode enable register (RCU_AHB4SPEN)

Address offset: 0x5C

Reset value: 0x0000 63FF

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	CRCSPEN	BKPSRAMSPEN	Reserved	PKSPEN	PJSPEN	PHSPEN	PGSPEN	PFSPEN	PESPEN	PDSPEN	PCSPEN	PBSPEN	PASPEN
	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	CRCSPEN	CRC clock enable when sleep mode This bit is set and reset by software. 0: Disabled CRC clock when sleep mode 1: Enabled CRC clock when sleep mode
13	BKPSRAMSPEN	Backup SRAM clock enable when sleep mode This bit is set and reset by software. 0: Disabled Backup SRAM clock when sleep mode 1: Enabled Backup SRAM clock when sleep mode
12:10	Reserved	Must be kept at reset value.
9	PKSPEN	GPIO port K clock enable when sleep mode This bit is set and reset by software. 0: Disabled GPIO port K clock when sleep mode 1: Enabled GPIO port K clock when sleep mode
8	PJSPEN	GPIO port J clock enable when sleep mode This bit is set and reset by software. 0: Disabled GPIO port J clock when sleep mode 1: Enabled GPIO port J clock when sleep mode
7	PHSPEN	GPIO port H clock enable when sleep mode This bit is set and reset by software. 0: Disabled GPIO port H clock when sleep mode 1: Enabled GPIO port H clock when sleep mode
6	PGSPEN	GPIO port G clock enable when sleep mode This bit is set and reset by software. 0: Disabled GPIO port G clock when sleep mode 1: Enabled GPIO port G clock when sleep mode
5	PFSPEN	GPIO port F clock enable when sleep mode This bit is set and reset by software. 0: Disabled GPIO port F clock when sleep mode 1: Enabled GPIO port F clock when sleep mode
4	PESPEN	GPIO port E clock enable when sleep mode This bit is set and reset by software. 0: Disabled GPIO port E clock when sleep mode

		1: Enabled GPIO port E clock when sleep mode
3	PDSPEN	GPIO port D clock enable when sleep mode This bit is set and reset by software. 0: Disabled GPIO port D clock when sleep mode 1: Enabled GPIO port D clock when sleep mode
2	PCSPEN	GPIO port C clock enable when sleep mode This bit is set and reset by software. 0: Disabled GPIO port C clock when sleep mode 1: Enabled GPIO port C clock when sleep mode
1	PBSPEN	GPIO port B clock enable when sleep mode This bit is set and reset by software. 0: Disabled GPIO port B clock when sleep mode 1: Enabled GPIO port B clock when sleep mode
0	PASPEN	GPIO port A clock enable when sleep mode This bit is set and reset by software. 0: Disabled GPIO port A clock when sleep mode 1: Enabled GPIO port A clock when sleep mode

10.3.25. APB1 sleep mode enable register (RCU_APB1SPEN)

Address offset: 0x60

Reset value: 0xF9FF EFFF

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
UART7SPEN	UART6SPEN	DACSPEN	DACHOLDSPEN	CTCSPEN	Reserved		I2C3SPEN	I2C2SPEN	I2C1SPEN	I2C0SPEN	UART4SPEN	UART3SPEN	USART2SPEN	USART1SPEN	MDIOSPEN
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
SPI2SPEN	SPI1SPEN	RSPDIFSPEN	Reserved	TIMER51SPEN	TIMER50SPEN	TIMER31SPEN	TIMER30SPEN	TIMER23SPEN	TIMER22SPEN	TIMER6SPEN	TIMER5SPEN	TIMER4SPEN	TIMER3SPEN	TIMER2SPEN	TIMER1SPEN
rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31	UART7SPEN	UART7 clock enable when sleep mode This bit is set and reset by software. 0: Disabled UART7 clock when sleep mode 1: Enabled UART7 clock when sleep mode
30	UART6SPEN	UART6 clock enable when sleep mode This bit is set and reset by software. 0: Disabled UART6 clock when sleep mode

		1: Enabled UART6 clock when sleep mode
29	DACSPEN	DAC clock enable when sleep mode This bit is set and reset by software. 0: Disabled DAC clock when sleep mode 1: Enabled DAC clock when sleep mode
28	DACHOLDSPEN	DAC hold clock enable when sleep mode This bit is set and reset by software. The hold clock source is IRC32K. 0: Disabled DAC hold clock when sleep mode 1: Enabled DAC hold clock when sleep mode
27	CTCSPEN	CTC clock enable when sleep mode This bit is set and reset by software. 0: Disabled CTC clock when sleep mode 1: Enabled CTC clock when sleep mode
26:25	Reserved	Must be kept at reset value.
24	I2C3SPEN	I2C3 clock enable when sleep mode This bit is set and reset by software. 0: Disabled I2C3 clock when sleep mode 1: Enabled I2C3 clock when sleep mode
23	I2C2SPEN	I2C2 clock enable when sleep mode This bit is set and reset by software. 0: Disabled I2C2 clock when sleep mode 1: Enabled I2C2 clock when sleep mode
22	I2C1SPEN	I2C1 clock enable when sleep mode This bit is set and reset by software. 0: Disabled I2C1 clock when sleep mode 1: Enabled I2C1 clock when sleep mode
21	I2C0SPEN	I2C0 clock enable when sleep mode This bit is set and reset by software. 0: Disabled I2C0 clock when sleep mode 1: Enabled I2C0 clock when sleep mode
20	UART4SPEN	UART4 clock enable when sleep mode This bit is set and reset by software. 0: Disabled UART4 clock when sleep mode 1: Enabled UART4 clock when sleep mode
19	UART3SPEN	UART3 clock enable when sleep mode This bit is set and reset by software. 0: Disabled UART3 clock when sleep mode 1: Enabled UART3 clock when sleep mode

18	USART2SPEN	USART2 clock enable when sleep mode This bit is set and reset by software. 0: Disabled USART2 clock when sleep mode 1: Enabled USART2 clock when sleep mode
17	USART1SPEN	USART1 clock enable when sleep mode This bit is set and reset by software. 0: Disabled USART1 clock when sleep mode 1: Enabled USART1 clock when sleep mode
16	MDIOSPEN	MDIO clock enable when sleep mode This bit is set and reset by software. 0: Disabled MDIO clock when sleep mode 1: Enabled MDIO clock when sleep mode
15	SPI2SPEN	SPI2 clock enable when sleep mode This bit is set and reset by software. 0: Disabled SPI2 clock when sleep mode 1: Enabled SPI2 clock when sleep mode
14	SPI1SPEN	SPI1 clock enable when sleep mode This bit is set and reset by software. 0: Disabled SPI1 clock when sleep mode 1: Enabled SPI1 clock when sleep mode
13	RSPDIFSPEN	RSPDIF clock enable when sleep mode This bit is set and reset by software. 0: Disabled RSPDIF clock when sleep mode 1: Enabled RSPDIF clock when sleep mode
12	Reserved	Must be kept at reset value.
11	TIMER51SPEN	TIMER51 clock enable when sleep mode This bit is set and reset by software. 0: Disabled TIMER51 clock when sleep mode 1: Enabled TIMER51 clock when sleep mode
10	TIMER50SPEN	TIMER50 clock enable when sleep mode This bit is set and reset by software. 0: Disabled TIMER50 clock when sleep mode 1: Enabled TIMER50 clock when sleep mode
9	TIMER31SPEN	TIMER31 clock enable when sleep mode This bit is set and reset by software. 0: Disabled TIMER31 clock when sleep mode 1: Enabled TIMER31 clock when sleep mode
8	TIMER30SPEN	TIMER30 clock enable when sleep mode This bit is set and reset by software.

		0: Disabled TIMER30 clock when sleep mode 1: Enabled TIMER30 clock when sleep mode
7	TIMER23SPEN	TIMER23 clock enable when sleep mode This bit is set and reset by software. 0: Disabled TIMER23 clock when sleep mode 1: Enabled TIMER23 clock when sleep mode
6	TIMER22SPEN	TIMER22 clock enable when sleep mode This bit is set and reset by software. 0: Disabled TIMER22 clock when sleep mode 1: Enabled TIMER22 clock when sleep mode
5	TIMER6SPEN	TIMER6 clock enable when sleep mode This bit is set and reset by software. 0: Disabled TIMER6 clock when sleep mode 1: Enabled TIMER6 clock when sleep mode
4	TIMER5SPEN	TIMER5 clock enable when sleep mode This bit is set and reset by software. 0: Disabled TIMER5 clock when sleep mode 1: Enabled TIMER5 clock when sleep mode
3	TIMER4SPEN	TIMER4 clock enable when sleep mode This bit is set and reset by software. 0: Disabled TIMER4 clock when sleep mode 1: Enabled TIMER4 clock when sleep mode
2	TIMER3SPEN	TIMER3 clock enable when sleep mode This bit is set and reset by software. 0: Disabled TIMER3 clock when sleep mode 1: Enabled TIMER3 clock when sleep mode
1	TIMER2SPEN	TIMER2 clock enable when sleep mode This bit is set and reset by software. 0: Disabled TIMER2 clock when sleep mode 1: Enabled TIMER2 clock when sleep mode
0	TIMER1SPEN	TIMER1 clock enable when sleep mode This bit is set and reset by software. 0: Disabled TIMER1 clock when sleep mode 1: Enabled TIMER1 clock when sleep mode

10.3.26. APB2 sleep mode enable register (RCU_APB2SPEN)

Address offset: 0x64

Reset value: 0xFFFF B733

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
TRIGSELSPEN	EDOUTSPEN	TIMER44SPEN	TIMER43SPEN	TIMER42SPEN	TIMER41SPEN	TIMER40SPEN	SAI2SPEN	SAI1SPEN	SAI0SPEN	SPI5SPEN	SPI4SPEN	HPDFSPEN	TIMER16SPEN	TIMER15SPEN	TIMER14SPEN
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
SAI3SPEN	Reserved	SPI3SPEN	SPI0SPEN	Reserved	ADC2SPEN	ADC1SPEN	ADC0SPEN	Reserved		USART5SPEN	USART0SPEN	Reserved		TIMER7SPEN	TIMER0SPEN
rw		rw	rw		rw	rw	rw			rw	rw			rw	rw

Bits	Fields	Descriptions
31	TRIGSELSPEN	TRIGSEL clock enable when sleep mode This bit is set and reset by software. 0: Disabled TRIGSEL clock when sleep mode 1: Enabled TRIGSEL clock when sleep mode
30	EDOUTSPEN	EDOUT clock enable when sleep mode This bit is set and reset by software. 0: Disabled EDOUT clock when sleep mode 1: Enabled EDOUT clock when sleep mode
29	TIMER44SPEN	TIMER44 clock enable when sleep mode This bit is set and reset by software. 0: Disabled TIMER44 clock when sleep mode 1: Enabled TIMER44 clock when sleep mode
28	TIMER43SPEN	TIMER43 clock enable when sleep mode This bit is set and reset by software. 0: Disabled TIMER43 clock when sleep mode 1: Enabled TIMER43 clock when sleep mode
27	TIMER42SPEN	TIMER42 clock enable when sleep mode This bit is set and reset by software. 0: Disabled TIMER42 clock when sleep mode 1: Enabled TIMER42 clock when sleep mode
26	TIMER41SPEN	TIMER41 clock enable when sleep mode This bit is set and reset by software. 0: Disabled TIMER41 clock when sleep mode 1: Enabled TIMER41 clock when sleep mode
25	TIMER40SPEN	TIMER40 clock enable when sleep mode This bit is set and reset by software. 0: Disabled TIMER40 clock when sleep mode 1: Enabled TIMER40 clock when sleep mode
24	SAI2SPEN	SAI2 clock enable when sleep mode

		This bit is set and reset by software. 0: Disabled SAI2 clock when sleep mode 1: Enabled SAI2 clock when sleep mode
23	SAI1SPEN	SAI1 clock enable when sleep mode This bit is set and reset by software. 0: Disabled SAI1 clock when sleep mode 1: Enabled SAI1 clock when sleep mode
22	SAI0SPEN	SAI0 clock enable when sleep mode This bit is set and reset by software. 0: Disabled SAI0 clock when sleep mode 1: Enabled SAI0 clock when sleep mode
21	SPI5SPEN	SPI5 clock enable when sleep mode This bit is set and reset by software. 0: Disabled SPI5 clock when sleep mode 1: Enabled SPI5 clock when sleep mode
20	SPI4SPEN	SPI4 clock enable when sleep mode This bit is set and reset by software. 0: Disabled SPI4 clock when sleep mode 1: Enabled SPI4 clock when sleep mode
19	HPDFSPEN	HPDF clock enable when sleep mode This bit is set and reset by software. 0: Disabled HPDF clock when sleep mode 1: Enabled HPDF clock when sleep mode
18	TIMER16SPEN	TIMER16 clock enable when sleep mode This bit is set and reset by software. 0: Disabled TIMER16 clock when sleep mode 1: Enabled TIMER16 clock when sleep mode
17	TIMER15SPEN	TIMER15 clock enable when sleep mode This bit is set and reset by software. 0: Disabled TIMER15 clock when sleep mode 1: Enabled TIMER15 clock when sleep mode
16	TIMER14SPEN	TIMER14 clock enable when sleep mode This bit is set and reset by software. 0: Disabled TIMER14 clock when sleep mode 1: Enabled TIMER14 clock when sleep mode
15	SAI3SPEN	SAI3 clock enable when sleep mode This bit is set and reset by software. 0: Disabled SAI3 clock when sleep mode 1: Enabled SAI3 clock when sleep mode

14	Reserved	Must be kept at reset value.
13	SPI3SPEN	SPI3 clock enable when sleep mode This bit is set and reset by software. 0: Disabled SPI3 clock when sleep mode 1: Enabled SPI3 clock when sleep mode
12	SPI0SPEN	SPI0 clock enable when sleep mode This bit is set and reset by software. 0: Disabled SPI0 clock when sleep mode 1: Enabled SPI0 clock when sleep mode
11	Reserved	Must be kept at reset value.
10	ADC2SPEN	ADC2 clock enable when sleep mode This bit is set and reset by software. 0: Disabled ADC2 clock when sleep mode 1: Enabled ADC2 clock when sleep mode
9	ADC1SPEN	ADC1 clock enable when sleep mode This bit is set and reset by software. 0: Disabled ADC1 clock when sleep mode 1: Enabled ADC1 clock when sleep mode
8	ADC0SPEN	ADC0 clock enable when sleep mode This bit is set and reset by software. 0: Disabled ADC0 clock when sleep mode 1: Enabled ADC0 clock when sleep mode
7:6	Reserved	Must be kept at reset value.
5	USART5SPEN	USART5 clock enable when sleep mode This bit is set and reset by software. 0: Disabled USART5 clock when sleep mode 1: Enabled USART5 clock when sleep mode
4	USART0SPEN	USART0 clock enable when sleep mode This bit is set and reset by software. 0: Disabled USART0 clock when sleep mode 1: Enabled USART0 clock when sleep mode
3:2	Reserved	Must be kept at reset value.
1	TIMER7SPEN	TIMER7 clock enable when sleep mode This bit is set and reset by software. 0: Disabled TIMER7 clock when sleep mode 1: Enabled TIMER7 clock when sleep mode
0	TIMER0SPEN	TIMER0 clock enable when sleep mode This bit is set and reset by software.

0: Disabled TIMER0 clock when sleep mode

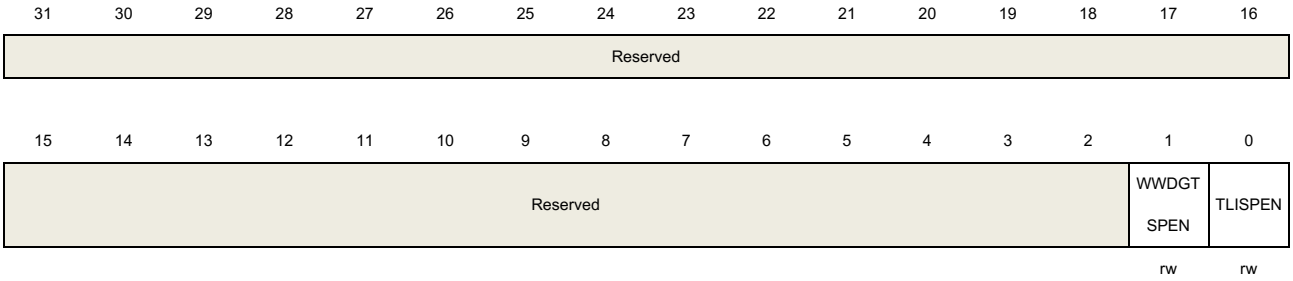
1: Enabled TIMER0 clock when sleep mode

10.3.27. APB3 sleep mode enable register (RCU_APB3SPEN)

Address offset: 0x68

Reset value: 0x0000 0003

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



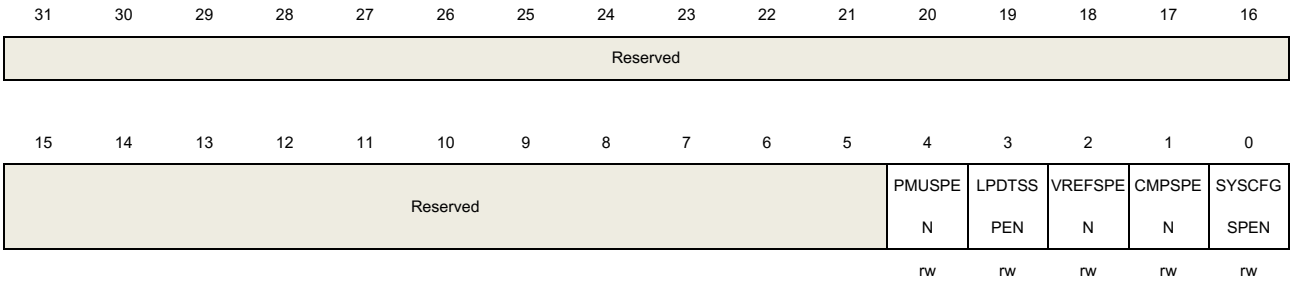
Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	WWDGTSPEN	WWDGT clock enable when sleep mode This bit is set and reset by software. 0: Disabled WWDGT clock when sleep mode 1: Enabled WWDGT clock when sleep mode
0	TLISPEN	TLI clock enable when sleep mode This bit is set and reset by software. 0: Disabled TLI clock when sleep mode 1: Enabled TLI clock when sleep mode

10.3.28. APB4 sleep mode enable register (RCU_APB4SPEN)

Address offset: 0x6C

Reset value: 0x0000 001F

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



Bits	Fields	Descriptions

31:5	Reserved	Must be kept at reset value.
4	PMUSPEN	PMU clock enable when sleep mode This bit is set and reset by software. 0: Disabled PMU clock when sleep mode 1: Enabled PMU clock when sleep mode
3	LPDTSSPEN	LPDTS clock enable when sleep mode This bit is set and reset by software. 0: Disabled LPDTS clock when sleep mode 1: Enabled LPDTS clock when sleep mode
2	VREFSPEN	VREF clock enable when sleep mode This bit is set and reset by software. 0: Disabled VREF clock when sleep mode 1: Enabled VREF clock when sleep mode
1	CMPSPEN	CMP clock enable when sleep mode This bit is set and reset by software. 0: Disabled CMP clock when sleep mode 1: Enabled CMP clock when sleep mode
0	SYSCFGSPEN	SYSCFG clock enable when sleep mode This bit is set and reset by software. 0: Disabled SYSCFG clock when sleep mode 1: Enabled SYSCFG clock when sleep mode

10.3.29. Backup domain control register (RCU_BDCTL)

Address offset: 0x70
Reset value: 0x0000 0018, reset by Backup domain Reset.

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

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

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

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

Bits	Fields	Descriptions

31:17	Reserved	Must be kept at reset value.
16	BKPRST	Backup domain reset This bit is set and reset by software. 0: No reset 1: Resets backup domain
15	RTCEN	RTC clock enable This bit is set and reset by software. 0: Disabled RTC clock 1: Enabled RTC clock
14:11	Reserved	Must be kept at reset value.
10	LXTALSTARTON	LXTAL start circuit on control 0: Close the start circuit when LXTAL is ready 1: Hold the start circuit when LXTAL is ready
9:8	RTCSRC[1:0]	RTC clock entry selection Set and reset by software to control the RTC clock source. Once the RTC clock source has been selected, it cannot be changed anymore unless the backup domain is reset. 00: No clock selected 01: CK_LXTAL selected as RTC source clock 10: CK_IRC32K selected as RTC source clock 11: (CK_HXTAL / RTCDIV) selected as RTC source clock, please refer to RTCDIV bits in RCU_CFG0 register.
7	LXTALRDYRST	LXTAL ready reset 0: LXTAL ready no reset 1: LXTAL ready reset
6	LCKMD	LXTAL clock failure detection Set by hardware to indicate when a failure has been detected by the clock security system on the external 32 kHz oscillator (LXTAL). It can be clean by disable LCKMEN or disable LXTALEN. 0: No failure detected on LXTAL (32 kHz oscillator) 1: Failure detected on LXTAL (32 kHz oscillator)
5	LCKMEN	LXTAL clock monitor enable 0: Disable the LXTAL clock monitor 1: Enable the LXTAL clock monitor Set by software to enable the clock security system on LXTAL (32 kHz oscillator). LCKMEN should be enabled only on the LXTAL is enabled (LXTALEN bit enabled) and ready (LXTALSTB flag set by hardware). **Note:** Once LCKMEN bit is set, this bit can be reset by backup domain reset or resetting this bit after detecting LXTAL clock failure (LCKMD = 1).

4:3	LXTALDRI[1:0]	LXTAL drive capability Set and reset by software. Backup domain reset resets this value. 00: Lower driving capability 01: Medium low driving capability 10: Medium high driving capability 11: Higher driving capability **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

10.3.30. 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	Reserved							
r	r	r	r	r	r	r	rw								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														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
23:2	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

10.3.31. PLL clock additional control register (RCU_PLLADDCTL)

Address offset: 0x80
Reset value: 0xFF81 0101

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
PLL2PEN	PLL2REN	PLL2QEN	PLL1PEN	PLL1REN	PLL1QEN	PLL0PEN	PLL0REN	PLL0QEN	PLL2Q[6:0]						
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	PLL1Q[6:0]							Reserved	PLL0Q[6:0]						
	rw								rw						

Bits	Fields	Descriptions
31	PLL2PEN	PLL2P divider output enable This bit is set and reset by software. The PLL2PEN bit can be written only if the PLL2EN is 0. 0: Disable the CK_PLL2P output 1: Enable the CK_PLL2P output
30	PLL2REN	PLL2R divider output enable This bit is set and reset by software. The PLL2REN bit can be written only if the PLL2EN is 0. 0: Disable the CK_PLL2R output 1: Enable the CK_PLL2R output
29	PLL2QEN	PLL2Q divider output enable This bit is set and reset by software. The PLL2QEN bit can be written only if the PLL2EN is 0. 0: Disable the CK_PLL2Q output 1: Enable the CK_PLL2Q output
28	PLL1PEN	PLL1P divider output enable This bit is set and reset by software. The PLL1PEN bit can be written only if the PLL1EN is 0. 0: Disable the CK_PLL1P output 1: Enable the CK_PLL1P output
27	PLL1REN	PLL1R divider output enable This bit is set and reset by software. The PLL1REN bit can be written only if the PLL1EN is 0.

		0: Disable the CK_PLL1R output 1: Enable the CK_PLL1R output
26	PLL1QEN	PLL1Q divider output enable This bit is set and reset by software. The PLL1QEN bit can be written only if the PLL1EN is 0. 0: Disable the CK_PLL1Q output 1: Enable the CK_PLL1Q output
25	PLL0PEN	PLL0P divider output enable This bit is set and reset by software. The PLL0PEN bit can be written only if the PLL0EN is 0. 0: Disable the CK_PLL0P output 1: Enable the CK_PLL0P output
24	PLL0REN	PLL0R divider output enable This bit is set and reset by software. The PLL0REN bit can be written only if the PLL0EN is 0. 0: Disable the CK_PLL0R output 1: Enable the CK_PLL0R output
23	PLL0QEN	PLL0Q divider output enable This bit is set and reset by software. The PLL0QEN bit can be written only if the PLL0EN is 0. 0: Disable the CK_PLL0Q output 1: Enable the CK_PLL0Q output
22:16	PLL2Q[6:0]	The PLL2Q output frequency division factor from PLL2 VCO clock Set and reset by software when the PLL2 is disable. These bits used to generator PLL2Q output clock (CK_PLL2Q) from PLL2 VCO clock (CK_PLL2VCO). The CK_PLL2VCO is described in PLL2N bits in RCU_PLL2 register. 0000000: CK_PLL2Q = CK_PLL2VCO 0000001: CK_PLL2Q = CK_PLL2VCO / 2 0000010: CK_PLL2Q = CK_ PLL2VCO / 3. 0000011: CK_PLL2Q = CK_ PLL2VCO / 4 0000100: CK_PLL2Q = CK_ PLL2VCO / 5 … 1111111: CK_PLL2Q = CK_ PLL2VCO / 128
15	Reserved	Must be kept at reset value.
14:8	PLL1Q[6:0]	The PLL1Q output frequency division factor from PLL1 VCO clock Set and reset by software when the PLL1 is disable. These bits used to generator PLL1Q output clock (CK_PLL1Q) from PLL1 VCO clock (CK_PLL1VCO). The CK_PLL1VCO is described in PLL1N bits in RCU_PLL1 register. 0000000: CK_PLL1Q = CK_PLL1VCO 0000001: CK_ PLL1Q = CK_PLL1VCO / 2

		0000010: CK_PLL1Q = CK_PLL1VCO / 3. 0000011: CK_PLL1Q = CK_PLL1VCO / 4 0000100: CK_PLL1Q = CK_PLL1VCO / 5 … 1111111: CK_PLL1Q = CK_PLL1VCO / 128
7	Reserved	Must be kept at reset value.
6:0	PLL0Q[6:0]	The PLL0Q output frequency division factor from PLL0 VCO clock Set and reset by software when the PLL0 is disable. These bits used to generator PLL0Q output clock (CK_PLL0Q) from PLL0 VCO clock (CK_PLL0VCO). The CK_PLL0Q is used to USBHS (48MHz), TRNG(48MHz) and SDIO. The CK_PLL0VCO is described in PLL0N bits in RCU_PLL0 register. 0000000: CK_PLL0Q = CK_PLL0VCO 0000001: CK_PLL0Q = CK_PLL0VCO / 2 0000010: CK_PLL0Q = CK_PLL0VCO / 3 0000011: CK_PLL0Q = CK_PLL0VCO / 4 0000100: CK_PLL0Q = CK_PLL0VCO / 5 … 1111111: CK_PLL0Q = CK_PLL0VCO / 128

10.3.32. PLL1 register (RCU_PLL1)

Address offset: 0x84

Reset value: 0x0101 2020

To configure the PLL1 clock, refer to the following formula:

CK_PLL1VCOSRC = CK_PLL1SRC / PLL1PSC

CK_PLL1VCO = CK_PLL1VCOSRC × (PLL1N + PLL1FRAN / 2^{13})

CK_PLL1P = CK_ PLL1VCO / PLL1P

CK_PLL1R = CK_PLL1VCO / PLL1R

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	PLL1R[6:0]							Reserved	PLL1P[6:0]						
	rw								rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	PLL1N[8:0]									PLL1PSC[5:0]					
	rw									rw					

Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30:24	PLL1R[6:0]	The PLL1R output frequency division factor from PLL1 VCO clock Set and reset by software when the PLL1 is disable. These bits used to generate PLL1R output clock (CK_PLL1R) from PLL1 VCO clock (CK_PLL1VCO). The

		CK_PLL1VCO is described in PLL1N bits in RCU_PLL1 register. 0000000: CK_PLL1R = CK_PLL1VCO 0000001: CK_PLL1R = CK_PLL1VCO / 2 0000010: CK_PLL1R = CK_PLL1VCO / 3. 0000011: CK_PLL1R = CK_PLL1VCO / 4 0000100: CK_PLL1R = CK_PLL1VCO / 5 … 1111111: CK_PLL1R = CK_PLL1VCO / 128
23	Reserved	Must be kept at reset value.
22:16	PLL1P[6:0]	The PLL1P output frequency division factor from PLL1 VCO clock Set and reset by software when the PLL1 is disable. These bits used to generator PLL1P output clock (CK_PLL1P) from PLL1 VCO clock (CK_PLL1VCO). The CK_PLL1VCO is described in PLL1N bits in RCU_ PLL1 register. 0000000: CK_PLL1P = CK_PLL1VCO 0000001: CK_PLL1P = CK_PLL1VCO / 2 0000010: CK_PLL1P = CK_PLL1VCO / 3 0000011: CK_PLL1P = CK_PLL1VCO / 4 0000100: CK_PLL1P = CK_PLL1VCO / 5 … 1111111: CK_PLL1P = CK_PLL1VCO / 128
15	Reserved	Must be kept at reset value.
14:6	PLL1N[8:0]	The PLL1 VCO clock multiplication factor Set and reset by software (only use word / half-word write) when the PLL1 is disable. These bits used to generate PLL1 VCO clock (CK_PLL1VCO) from PLL1 VCO source clock (CK_PLL1VCOSRC). The CK_PLL1VCOSRC is described in PLLPSC bits in RCU_PLL register **Note:** The frequency of CK_PLL1VCO is between 150MHz to 836MHz. When PLL1 fractional latch is disabled, the value of PLL1N must: 9<=PLL1N<=512 When PLL1 fractional latch is enabled, the value of PLL1N must: 12<=PLL1N<=508 000000000: Reserved … 000000111: Reserved 000001000: PLL1N = 9. … 001000000: PLL1N = 65 001000001: PLL1N = 66 … 111111111: PLL1N = 512
5:0	PLL1PSC[5:0]	The PLL1 VCO source clock prescaler Set and reset by software when the PLL1 is disable. These bits used to generate the clock of PLL1 VCO source clock (CK_PLL1VCOSRC) from PLL1 source clock

(CK_PLL1SRC) which is described in PLLSEL in RCU_PLLALL register.
The VCO source clock is between 1M to 16MHz.
000000: Reserved.
000001: CK_PLL1SRC
000010: CK_PLL1SRC / 2
000011: CK_PLL1SRC / 3
…
111111: CK_PLL1SRC / 63

10.3.33. PLL2 register (RCU_PLL2)

Address offset: 0x88
Reset value: 0x0101 2020
To configure the PLL2 clock, refer to the following formula:
CK_PLL2VCOSRC = CK_PLL2SRC / PLL2PSC
CK_PLL2VCO = CK_ PLL2VCOSRC × (PLL2N + PLL2FRAN / 2^{13})
CK_PLL2P = CK_ PLL2VCO / PLL2P
CK_PLL2R = CK_ PLL2VCO / PLL2R

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	PLL2R[6:0]							Reserved	PLL2P[6:0]						
	rw								rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	PLL2N[8:0]									PLL2PSC[5:0]					
	rw									rw					

Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30:24	PLL2R[6:0]	The PLL2R output frequency division factor from PLL2 VCO clock Set and reset by software when the PLL2 is disable. These bits used to generate PLL2R output clock (CK_PLL2R) from PLL2 VCO clock (CK_PLL2VCO). The CK_PLL2R is used to generate TLI clock (<=216MHz). The CK_PLL2VCO is described in PLL2N bits in RCU_PLL2 register. 0000000: CK_PLL2R = CK_PLL2VCO 0000001: CK_PLL2R = CK_PLL2VCO / 2 0000010: CK_PLL2R = CK_PLL2VCO / 3 0000011: CK_PLL2R = CK_PLL2VCO / 4 0000100: CK_PLL2R = CK_PLL2VCO / 5 … 1111111: CK_PLL2R = CK_PLL2VCO / 128
23	Reserved	Must be kept at reset value.

22:16	PLL2P[6:0]	The PLL2 P output frequency division factor from PLL2 VCO clock Set and reset by software when the PLL2 is disable. These bits used to generator PLL2 P output clock (CK_PLL2P) from PLL2 VCO clock (CK_PLL2VCO). The CK_PLL2P is used to USBHS (48MHz), TRNG (48MHz), or SDIO (<=48MHz). The CK_PLL2VCO is described in PLL2N bits in RCU_PLL2 register. 0000000: CK_PLL2P = CK_PLL2VCO 0000001: CK_PLL2P = CK_PLL2VCO / 2 0000010: CK_PLL2P = CK_PLL2VCO / 3 0000011: CK_PLL2P = CK_PLL2VCO / 4 0000100: CK_PLL2R = CK_PLL2VCO / 5 … 1111111: CK_PLL2R = CK_PLL2VCO / 128
15	Reserved	Must be kept at reset value.
14:6	PLL2N[8:0]	The PLL2 VCO clock multiplication factor Set and reset by software (only use word / half-word write) when the PLL2 is disable. These bits used to generate PLL2 VCO clock (CK_PLL2VCO) from PLL2 VCO source clock (CK_PLL2VCOSRC). The CK_PLL2VCOSRC is described in PLLPSC bits in RCU_PLL register. **Note:** The frequency of CK_PLL2VCO is between 150MHz to 836MHz. When PLL2 fractional latch is disabled, the value of PLL2N must: 9<=PLL2N<=512 When PLL2 fractional latch is enabled, the value of PLL2N must: 12<=PLL2N<=508 000000000: Reserved … 000000111: Reserved 000001000: PLL2N = 9. … 001000000: PLL2N = 65 001000001: PLL2N = 66 … 111111111: PLL2N = 512
5:0	PLL2PSC[5:0]	The PLL2 VCO source clock prescaler Set and reset by software when the PLL2 is disable. These bits used to generate the clock of PLL2 VCO source clock (CK_PLL2VCOSRC) from PLL2 source clock (CK_PLL2SRC) which is described in PLLSEL in RCU_PLLALL register. The VCO source clock is between 1M to 16MHz. 000000: Reserved . 000001: CK_PLL2SRC 000010: CK_PLL2SRC / 2 000011: CK_PLL2SRC / 3 … 111111: CK_PLL2SRC / 63

10.3.34. 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).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HPDFSEL	ESCSEL[1:0]		ESCDIV[2:0]			Reserved	TIMERSEL	USART5SEL[1:0]		USART2SEL[1:0]		USART1SEL[1:0]		PLL2RDIV[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
PERSEL[1:0]		CAN2SEL[1:0]		CAN1SEL[1:0]		CAN0SEL[1:0]		NVMC_ECNVMSEL[1:0]		RSPDIFSEL[1:0]		DSIPRESEL	Reserved	USART0SEL[1:0]	
rw		rw		rw		rw		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	ESCSEL[1:0]	ESC clock source selection Set and reset by software to control the ESC clock source 00: CK_AHB selected as ESC source clock 01: CK_PLL0R selected as ESC source clock 10: CK_PLL1P selected as ESC source clock 11: CK_PLL2Q selected as ESC source clock
28:26	ESCDIV[2:0]	ESC clock divider factor Set and reset by software. 000: CK_ESC / 1 001: CK_ESC / 2 010: CK_ESC / 3 011: CK_ESC / 4 101: CK_ESC / 5 101: CK_ESC / 6 110: CK_ESC / 7 111: CK_ESC / 8
25	Reserved	Must be kept at reset value.
24	TIMERSEL	TIMER clock selection This bit is set and reset by software. This bit defined all timer clock selection. 0: If APB1PSC / APB2PSC in RCU_CFG0 register is 0b0xx(CK_APBx = CK_AHB) or 0b100(CK_APBx = CK_AHB/2), the TIMER clock is equal to CK_AHB(CK_TIMERx = CK_AHB). Or else, the TIMER clock is twice the

		corresponding APB clock (TIMER in APB1 domain: CK_TIMERx = 2 x CK_APB1; TIMER in APB2 domain: CK_TIMERx = 2 x CK_APB2). 1: If APB1PSC / APB2PSC in RCU_CFG0 register is 0b0xx(CK_APBx = CK_AHB), 0b100(CK_APBx = CK_AHB / 2), or 0b101(CK_APBx = CK_AHB / 4), the TIMER clock is equal to CK_AHB(CK_TIMERx = CK_AHB). Or else, the TIMER clock is four times the corresponding APB clock (TIMER in APB1 domain: CK_TIMERx = 4 x CK_APB1, TIMER in APB2 domain: CK_TIMERx = 4 x CK_APB2).
23:22	USART5SEL[1:0]	USART5 clock source selection Set and reset by software to control the USART5 clock source 00: CK_APB2 output clock selected as USART5 source clock 01: CK_AHB output clock selected as USART5 source clock 10: CK_LXTAL output clock selected as USART5 source clock 11: CK_IRC64MDIV output clock selected as USART5 source clock
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_AHB selected as USART2 source clock 10: CK_LXTAL selected as USART2 source clock 11: CK_IRC64MDIV 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_AHB selected as USART1 source clock 10: CK_LXTAL selected as USART1 source clock 11: CK_IRC64MDIV selected as USART1 source clock
17:16	PLL2RDIV[1:0]	The divider factor from PLL2R clock These bits are set and reset by software when PLL2 is disabled. These bits used to generate clock for TLI clock. 00: CK_PLL2R / 2 01: CK_PLL2R / 4 10: CK_PLL2R / 8 11: CK_PLL2R / 16
15:14	PERSEL[1:0]	CK_PER clock selection Set and reset by software to control the CK_PER clock source 00: CK_IRC64MDIV selected as CK_PER source clock 01: CK_LPIRC4M selected as CK_PER source clock 10: CK_HXTAL selected as CK_PER source clock 11: reserved
13:12	CAN2SEL[1:0]	CAN2 clock source selection Set and reset by software to control the CAN2 clock source

		00: CK_IRC64MDIV selected as CAN2 source clock 01: CK_APB2 selected as CAN2 source clock 10: CK_APB2 / 2 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_IRC64MDIV selected as CAN1 source clock 01: CK_APB2 selected as CAN1 source clock 10: CK_APB2 / 2 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_IRC64MDIV selected as CAN0 source clock 01: CK_APB2 selected as CAN0 source clock 10: CK_APB2 / 2 selected as CAN0 source clock 11: CK_HXTAL selected as CAN0 source clock
7:6	NVMC_ECNVMSEL[1:0]	NVMC ECNVM clock source selection Set and reset by software to control the NVMC clock source 00: CK_AHB selected as NVMC ECNVM source clock 01: CK_AHB / 2 selected as NVMC ECNVM source clock 10: CK_PLL1R selected as NVMC ECNVM source clock 11: CK_PLL2R selected as NVMC ECNVM source clock
5:4	RSPDIFSEL[1:0]	RSPDIF clock source selection Set and reset by software to control the RSPDIF clock source 00: CK_PLL0Q selected as RSPDIF source clock 01: CK_PLL1R selected as RSPDIF source clock 10: CK_PLL2R selected as RSPDIF source clock 11: CK_IRC64MDIV selected as RSPDIF source clock
3	DSIPRESEL	DSI preceding stage clock source selection Set and reset by software to control the DSI preceding stage clock source 0: CK_IRC64MDIV selected as DSI preceding stage source clock 1: CK_HXTAL selected as DSI preceding stage source clock
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_AHB selected as USART0 source clock 10: CK_LXTAL selected as USART0 source clock 11: CK_IRC64MDIV selected as USART0 source clock

10.3.35. 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	SAI2B1SEL[2:0]			Reserved	SAI2B0SEL[2:0]			Reserved	SAI1_3SEL[2:0]			Reserved	SAI0SEL[2:0]		
	rw				rw				rw				rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	CKOUT1SEL[2:0]			CKOUT1DIV[3:0]				Reserved	CKOUT0SEL[2:0]			CKOUT0DIV[3:0]			
	rw			rw					rw			rw			

Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30:28	SAI2B1SEL[2:0]	SAI2 Block 1 clock source selection Set and reset by software to control the SAI2 BLOCK 1 clock source 000: CK_PLL0Q selected as SAI2 BLOCK 1 source clock 001: CK_PLL1P selected as SAI2 BLOCK 1 source clock 010: CK_PLL2P selected as SAI2 BLOCK 1 source clock 011: I2S_CKIN selected as SAI2 BLOCK 1 source clock 100: CK_PER selected as SAI2 BLOCK 1 source clock 101: CK_RSPDIF_SYMBselected as SAI2 BLOCK 1 source clock 11x: reserved
27	Reserved	Must be kept at reset value.
26:24	SAI2B0SEL[2:0]	SAI2 Block 0 clock source selection Set and reset by software to control the SAI2 BLOCK 0 clock source 000: CK_PLL0Q selected as SAI2 BLOCK 0 source clock 001: CK_PLL1P selected as SAI2 BLOCK 0 source clock 010: CK_PLL2P selected as SAI2 BLOCK 0 source clock 011: I2S_CKIN selected as SAI2 BLOCK 0 source clock 100: CK_PER selected as SAI2 BLOCK 0 source clock 101: CK_RSPDIF_SYMBselected as SAI2 BLOCK 0 source clock 11x: reserved
23	Reserved	Must be kept at reset value.
22:20	SAI1_3SEL[2:0]	SAI1 and SAI3 clock source selection Set and reset by software to control the SAI1 and SAI3 clock source 000: CK_PLL0Q selected as SAI1 and SAI3 source clock 001: CK_PLL1P selected as SAI1 and SAI3 source clock 010: CK_PLL2P selected as SAI1 and SAI3 source clock 011: I2S_CKIN selected as SAI1 and SAI3 source clock

		100: CK_PER selected as SAI1 and SAI3 source clock others: Reserved
19	Reserved	Must be kept at reset value.
18:16	SAI0SEL[2:0]	SA0 and HPDF audio clock source selection Set and reset by software to control the SAI0 and HPDF audio clock source 000: CK_PLL0Q selected as SAI0 and HPDF audio source clock 001: CK_PLL1P selected as SAI0 and HPDF audio source clock 010: CK_PLL2P selected as SAI0 and HPDF audio source clock 011: I2S_CKIN selected as SAI0 and HPDF audio source clock 100: CK_PER selected as SAI0 and HPDF audio source clock others: Reserved
15	Reserved	Must be kept at reset value.
14:12	CKOUT1SEL[2:0]	CKOUT1 clock source selection Set and reset by software. 000: CK_SYS clock selected 001: CK_PLL1R clock selected 010: CK_HXTAL clock selected 011: CK_PLL0P clock selected 100: CK_LPIRC4M clock selected 101: CK_IRC32K clock selected 110: CK_PLL2R clock selected 111: reserved **Note:** Configuration of this bit field may cause interference with CK_OUT1, and it is strongly recommended to configure these bits only after resetting but before enabling HXTAL and PLLs.
11:8	CKOUT1DIV[3:0]	The CK_OUT1 divider which the CK_OUT1 frequency can be reduced see bits14:12 of RCU_CFG2 for CK_OUT1 0000: inhibit predividers 0001: CK_OUT1 is divided by 1 0010: CK_OUT1 is divided by 2 0011: CK_OUT1 is divided by 3 0100: CK_OUT1 is divided by 4 … 1111: CK_OUT1 is divided by 15 **Note:** Configuration of this bit field may cause interference with CK_OUT1, and it is strongly recommended to configure these bits only after resetting but before enabling HXTAL and PLLs.
7	Reserved	Must be kept at reset value.
6:4	CKOUT0SEL[2:0]	CKOUT0 clock source selection

		Set and reset by software 000: CK_OUT0 selects CK_IRC64MDIV 001: CK_OUT0 selects CK_LXTAL 010: CK_OUT0 selects CK_HXTAL 011: CK_OUT0 selects CK_PLL0P 100: CK_OUT0 selects CK_IRC48M 101: CK_OUT0 selects CK_PER 110: CK_OUT0 selects USBHS0 60M 111: Reserved **Note:** Configuration of this bit field may cause interference with CK_OUT0, and it is strongly recommended to configure these bits only after resetting but before enabling HXTAL and PLLs.
3:0	CKOUT0DIV[3:0]	The CK_OUT0 divider which the CK_OUT0 frequency can be reduced see bits 6:4 of RCU_CFG2 for CK_OUT0 0000: inhibit predividers 0001: The CK_OUT0 is divided by 1 0010: The CK_OUT0 is divided by 2 0011: The CK_OUT0 is divided by 3 0100: The CK_OUT0 is divided by 4 … 1111: The CK_OUT0 is divided by 15 **Note:** Configuration of this bit field may cause interference with CK_OUT0, and it is strongly recommended to configure these bits only after resetting but before enabling HXTAL and PLLs.

10.3.36. Clock configuration register 3 (RCU_CFG3)

Address offset: 0x94

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		ADC2SEL[1:0]		ADC01SEL[1:0]		Reserved	DSPWUSSEL	Reserved							
		rw		rw			rw								

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			SDIO1SEL	Reserved		DSIDIV[3:0]				I2C3SEL[1:0]		I2C2SEL[1:0]		I2C1SEL[1:0]	
			rw			rw				rw		rw		rw	

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29:28	ADC2SEL[1:0]	ADC2 clock source selection Set and reset by software to control the ADC2 clock source

		00: CK_PLL1P selected as ADC2 source clock 01: CK_PLL2R selected as ADC2 source clock 10: CK_PER selected as ADC2 source clock 11: CK_PLL1Q selected as ADC2 source clock
27:26	ADC01SEL[1:0]	ADC0 and ADC1 clock source selection Set and reset by software to control the ADC0 and ADC1 clock source 00: CK_PLL1P selected as ADC0 and ADC1 source clock 01: CK_PLL2R selected as ADC0 and ADC1 source clock 10: CK_PER selected as ADC0 and ADC1 source clock 11: CK_PLL1Q selected as ADC0 and ADC1 source clock
25	Reserved	Must be kept at reset value.
24	DSPWUSSEL	Deep-sleep wakeup system clock source selection Set and reset by software to select the system wakeup clock from deep-sleep mode. The selected clock is also used as emergency clock for the Clock stuck on HXTAL. 0: The CK_IRC64MDIV is selected as wake up system clock from deep-sleep mode 1: The CK_LPIRC4M is selected as wake up system clock from deep-sleep mode **Note:** If DSPWUSSEL = '1' and peripheral clock source select CK_IRC64MDIV, when system is wakeup form deep-sleep mode by this peripheral, if peripheral wakeup function turns off, the peripheral will have no clock because of IRC64M is turn off. In this case, user needs to set IRC64MEN in RCU_CTL register to turn on IRC64M clock again. When the CKMEN bit is set and the system clock is CK_HXTAL or switched to HXTAL, this bit cannot be changed.
23:13	Reserved	Must be kept at reset value.
12	SDIO1SEL	SDIO1 clock source selection Set and reset by software to control the SDIO1 clock source. 0: CK_PLL0Q selected as SDIO1 source clock 1: CK_PLL1R selected as SDIO1 source clock
11:10	Reserved	Must be kept at reset value.
9:6	DSIDIV[3:0]	DSI clock divider factor Set and reset by software. 0000: CK_DSI / 1 0001: CK_DSI / 2 ... 1111: CK_DSI / 16
5:4	I2C3SEL[1:0]	I2C3 clock source selection

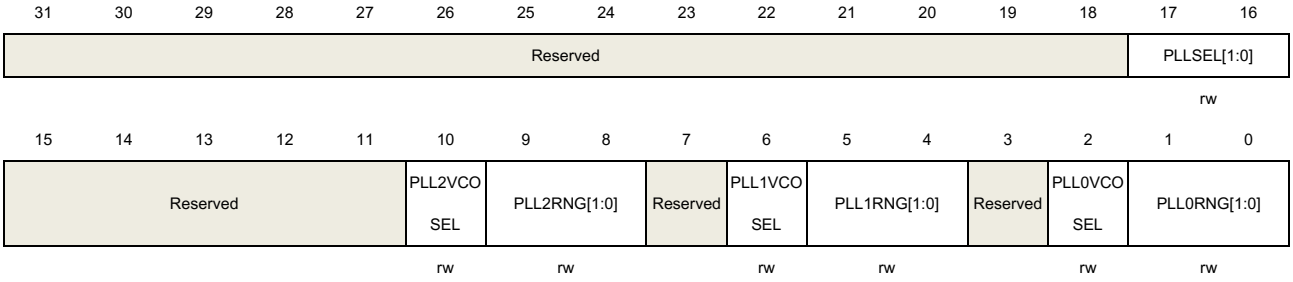
		Set and reset by software to control the I2C3 clock source. 00: CK_APB1 selected as I2C3 source clock 01: CK_PLL2R selected as I2C3 source clock 10: CK_IRC64MDIV selected as I2C3 source clock 11: CK_LPIRC4M selected as I2C3 source clock
3:2	I2C2SEL[1:0]	I2C2 clock source selection Set and reset by software to control the I2C2 clock source. 00: CK_APB1 selected as I2C2 source clock 01: CK_PLL2R selected as I2C2 source clock 10: CK_IRC64MDIV selected as I2C2 source clock 11: CK_LPIRC4M selected as I2C2 source clock
1:0	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_PLL2R selected as I2C1 source clock 10: CK_IRC64MDIV selected as I2C1 source clock 11: CK_LPIRC4M selected as I2C1 source clock

10.3.37. PLL all configuration register (RCU_PLLALL)

Address offset: 0x98

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:18	Reserved	Must be kept at reset value.
17:16	PLLSEL[1:0]	PLLs clock source selection Set and reset by software to control the PLLs clock source. 00: CK_IRC64MDIV selected as source clock of PLL0, PLL1, PLL2 01: CK_LPIRC4M selected as source clock of PLL0, PLL1, PLL2 10: CK_HXTAL selected as source clock of PLL0, PLL1, PLL2 11: No clock selected as source clock of PLL0, PLL1, PLL2
15:11	Reserved	Must be kept at reset value.

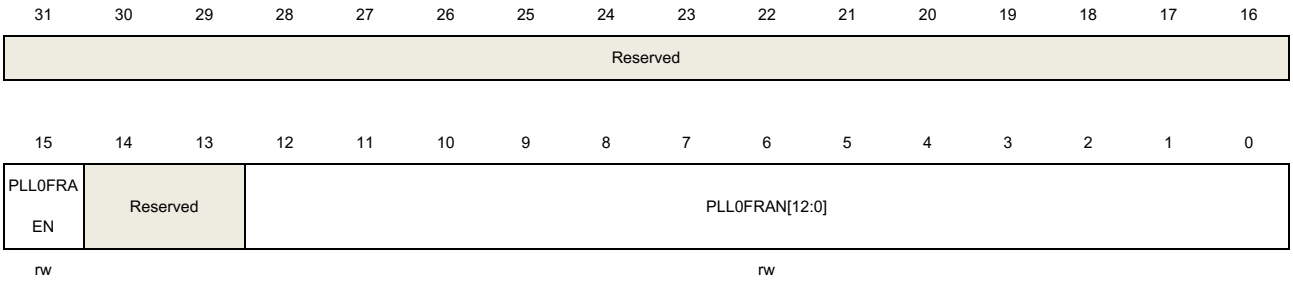
10	PLL2VCOSEL	PLL2 VCO selection Set and reset by software when the PLL2 is disable. 0: select wide VCO, range: 192 - 836MHz, PLL2 input clock frequency range should be 2-16M 1: select narrow VCO, range: 150 - 420MHz, PLL2 input clock frequency range should be 1-2M
9:8	PLL2RNG[1:0]	PLL2 input clock range Set and reset by software when the PLL2 is disable. 00: input clock frequency: 1 - 2MHz 01: input clock frequency: 2 - 4MHz 10: input clock frequency: 4 - 8MHz 11: input clock frequency: 8 - 16MHz
7	Reserved	Must be kept at reset value.
6	PLL1VCOSEL	PLL1 VCO selection Set and reset by software when the PLL1 is disable. 0: select wide VCO, range: 192 - 836MHz, PLL1 input clock frequency range should be 2-16M 1: select narrow VCO, range: 150 - 420MHz, PLL1 input clock frequency range should be 1-2M
5:4	PLL1RNG[1:0]	PLL1 input clock range Set and reset by software when the PLL1 is disable. 00: input clock frequency: 1 - 2MHz 01: input clock frequency: 2 - 4MHz 10: input clock frequency: 4 - 8MHz 11: input clock frequency: 8 - 16MHz
3	Reserved	Must be kept at reset value.
2	PLL0VCOSEL	PLL0 VCO selection Set and reset by software when the PLL0 is disable. 0: select wide VCO, range: 192 - 836MHz, PLL0 input clock frequency range should be 2-16M 1: select narrow VCO, range: 150 - 420MHz, PLL0 input clock frequency range should be 1-2M
1:0	PLL0RNG[1:0]	PLL0 input clock range Set and reset by software when the PLL0 is disable.. 00: input clock frequency: 1 - 2MHz 01: input clock frequency: 2 - 4MHz 10: input clock frequency: 4 - 8MHz 11: input clock frequency: 8 - 16MHz

10.3.38. PLL0 fraction configuration register (RCU_PLL0FRA)

Address offset: 0x9C

Reset value: 0x0000 0000

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



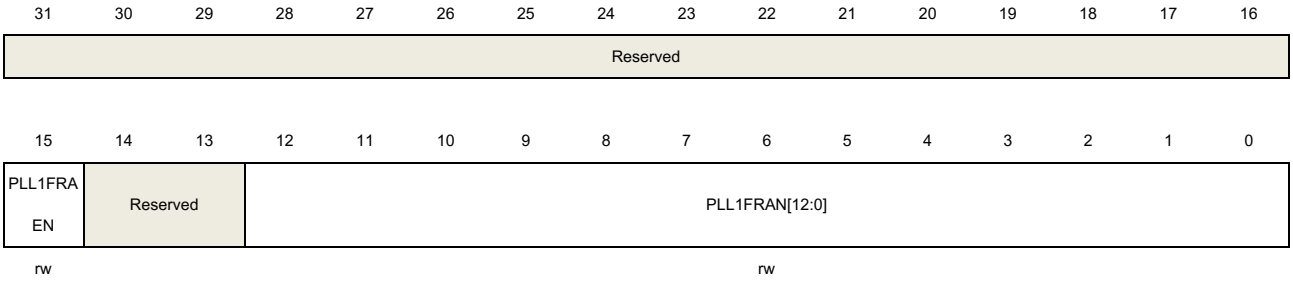
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	PLL0FRAEN	PLL0 fractional latch enable. These bits are set and reset by software. These bits can lock the PLL0FRAN value into a Sigma-Delta modulator .When PLL0FRAEN bit switchs from "0" to "1", the PLL0FRAN value will transfer to modulator.
14:13	Reserved	Must be kept at reset value.
12:0	PLL0FRAN[12:0]	Fractional part of the multiplication factor for PLL0 VCO These bits are set and reset by software. These bits can control the fractional part of the multiplication for PLL0 VCO. This bit field can be modified dynamically to fine-tune the PLL0 VCO. These bits must configure the PLL0 VCO out frequency to the following range: When PLL0VCOSEL = 0, the range is 192MHz to 836MHz; When PLL0VCOSEL = 1, the range is 150MHz to 420MHz.

10.3.39. PLL1 fraction configuration register (RCU_PLL1FRA)

Address offset: 0xA0

Reset value: 0x0000 0000

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



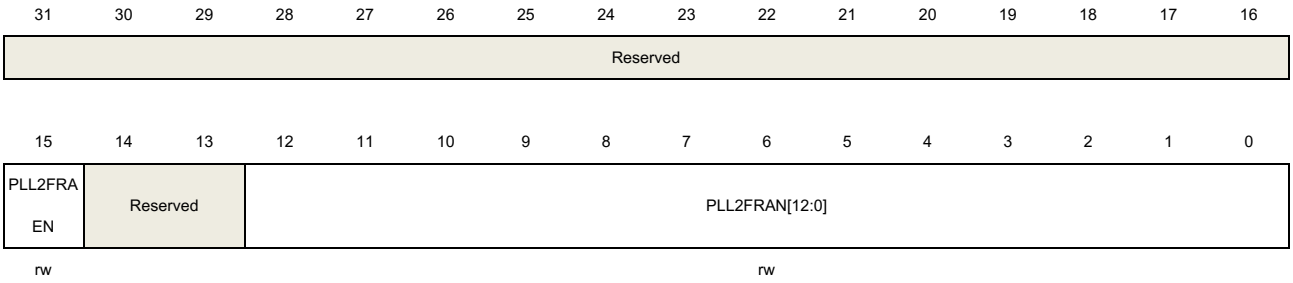
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	PLL1FRAEN	PLL1 fractional latch enable. These bits are set and reset by software. These bits can lock the PLL1FRAN value into a Sigma-Delta modulator .When PLL1FRAEN bit switchs from "0" to "1", the PLL1FRAN value will transfer to modulator.
14:13	Reserved	Must be kept at reset value.
12:0	PLL1FRAN [12:0]	Fractional part of the multiplication factor for PLL1 VCO These bits are set and reset by software. These bits can control the fractional part of the multiplication for PLL1 VCO. This bit field can be modified dynamically to fine-tune the PLL1 VCO. These bits must configure the PLL1 VCO out frequency to the following range: When PLL1VCOSEL = 0, the range is 192MHz to 836MHz; When PLL1VCOSEL = 1, the range is 150MHz to 420MHz.

10.3.40. PLL2 fraction configuration register (RCU_PLL2FRA)

Address offset: 0xA4

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	PLL2FRAEN	PLL2 fractional latch enable. These bits are set and reset by software. These bits can lock the PLL2FRAN value into a Sigma-Delta modulator .When PLL2FRAEN bit switchs from "0" to "1", the PLL2FRAN value will transfer to modulator.
14:13	Reserved	Must be kept at reset value.
12:0	PLL2FRAN[12:0]	Fractional part of the multiplication factor for PLL2 VCO These bits are set and reset by software. These bits can control the fractional part of the multiplication for PLL2 VCO. This bit field can be modified dynamically to fine-tune the PLL2 VCO. These bits must configure the PLL2 VCO out frequency to the following range:

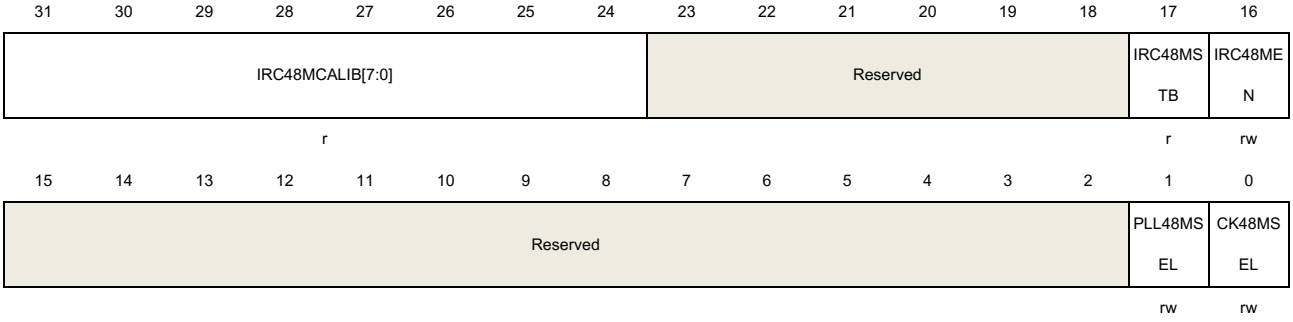
When PLL2VCOSEL = 0, the range is 192MHz to 836MHz; When PLL2VCOSEL = 1, the range is 150MHz to 420MHz.

10.3.41. Additional clock control register 0 (RCU_ADDCTL0)

Address offset: 0xC0

Reset value: 0x8000 0000

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



Bits	Fields	Descriptions
31:24	IRC48MCALIB [7:0]	Internal 48MHz RC oscillator calibration value register These bits are load automatically at power on.
23:18	Reserved	Must be kept at reset value.
17	IRC48MSTB	Internal 48MHz RC oscillator clock stabilization Flag Set by hardware to indicate if the IRC48M oscillator is stable and ready for use. 0: IRC48M is not stable 1: IRC48M is stable
16	IRC48MEN	Internal 48MHz RC oscillator enable Set and reset by software. Reset by hardware when entering Deep-sleep or Standby mode. 0: IRC48M disable 1: IRC48M enable
15:2	Reserved	Must be kept at reset value.
1	PLL48MSEL	PLL48M clock selection Set and reset by software. This bit used to generate PLL48M clock which select CK_PLL0Q or CK_PLL2P clock. 0: Select CK_PLL0Q clock 1: Select CK_PLL2P clock
0	CK48MSEL	48MHz clock selection Set and reset by software. This bit used to generate CK48M clock which select IRC48M clock or PLL48M clock. The CK48M clock used for TRNG/USBHS. The PLL48M clock refer to PLL48MSEL bit in RCU_ADDCTL register.

0: Don't select IRC48M clock (use CK_PLL0Q clock or CK_PLL2P clock select by PLL48MSEL)

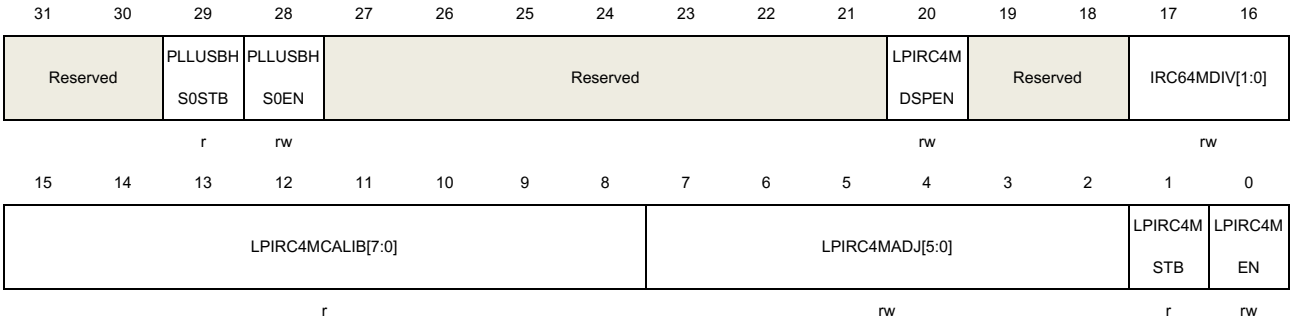
1: Select IRC48M clock

10.3.42. Additional clock control register 1(RCU_ADDCTL1)

Address offset: 0xC4

Reset value: 0x0200 7080

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



Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29	PLLUSBHS0STB	PLLUSBHS0 clock stabilization flag Set by hardware to indicate if the PLLUSBHS0 clock is stable and ready for use 0: PLLUSBHS0 clock is not stable 1: PLLUSBHS0 clock is stabilized
28	PLLUSBHS0EN	PLLUSBHS0 clock enable Set and reset by software. Reset by hardware when enter deep-sleep or standby mode. 0: Disabled PLLUSBHS0 clock 1: Enabled PLLUSBHS0 clock
27:21	Reserved	Must be kept at reset value.
20	LPIRC4MDSPEN	LPIRC4M clock enable in deepsleep mode Set and reset by software. LPIRC4M can be forced on even in deepsleep mode to quickly be used as a kernel clock for some peripherals. This bit has no effect on the value of LPIRC4MEN. 0: Has no impact on LPIRC4M 1: Force LPIRC4M to run even in deepsleep mode
19:18	Reserved	Must be kept at reset value.
17:16	IRC64MDIV[1:0]	IRC64M clock divider Set and reset by software. It cannot be written when system clock select CK_IRC64MDIV or IRC64MEN is set.

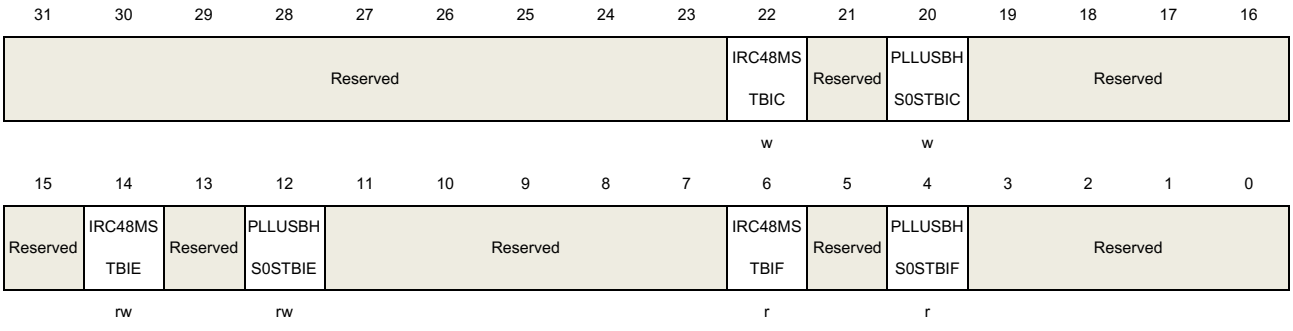
		00: CK_IRC64MDIV = CK_IRC64M / 1 01: CK_IRC64MDIV = CK_IRC64M / 2 10: CK_IRC64MDIV = CK_IRC64M / 4 11: CK_IRC64MDIV = CK_IRC64M / 8
15:8	LPIRC4MCALIB[7:0]	LPIRC4M calibration value These bits are load automatically at power on. The calibration signal step is 0.4%.
7:2	LPIRC4MADJ[5:0]	LPIRC4M frequency clock trim adjust value These bits are set by software. The trimming value is these bits (LPIRC4MADJ[5:0]) added to the LPIRC4MCALIB[7:0] bits. The trimming value should trim the LPIRC4MCALIB to 4 MHz +/- 1%.
1	LPIRC4MSTB	LPIRC4M clock stabilization flag Set by hardware to indicate if the LPIRC4M oscillator is stable and ready for use. 0: LPIRC4M RC oscillator is not stable 1: LPIRC4M RC oscillator is stabilized
0	LPIRC4MEN	LPIRC4M clock enable Set and reset by software. This bit cannot be reset if the LPIRC4M clock is used as the system clock. If DSPWUSSEL = 1, set by hardware when leaving Deep-sleep mode. 0: LPIRC4M RC oscillator disabled 1: LPIRC4M RC oscillator enabled

10.3.43. Additional clock interrupt register (RCU_ADDINT)

Address offset: 0xCC

Reset value: 0x0000 0000

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



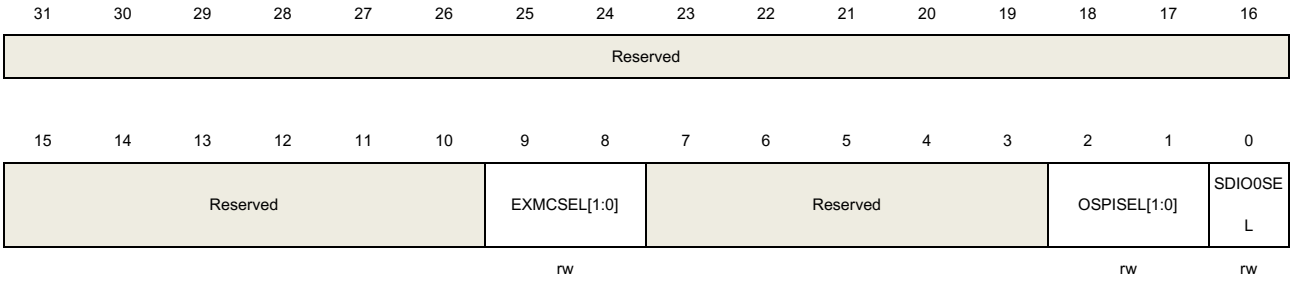
Bits	Fields	Descriptions
31:23	Reserved	Must be kept at reset value.
22	IRC48MSTBIC	Internal 48 MHz RC oscillator Stabilization interrupt clear Write 1 by software to reset the IRC48MSTBIF flag. 0: Not reset IRC48MSTBIF flag

		1: Reset IRC48MSTBIF flag
21	Reserved	Must be kept at reset value.
20	PLLUSBHS0STBIC	Internal PLL of USBHS0 Stabilization interrupt clear Write 1 by software to reset the PLLUSBHS0STBIF flag. 0: Not reset PLLUSBHS0STBIF flag 1: Reset PLLUSBHS0STBIF flag
19:15	Reserved	Must be kept at reset value.
14	IRC48MSTBIE	Internal 48 MHz RC oscillator Stabilization interrupt enable Set and reset by software to enable/disable the IRC48M stabilization interrupt 0: Disable the IRC48M stabilization interrupt 1: Enable the IRC48M stabilization interrupt
13	Reserved	Must be kept at reset value.
12	PLLUSBHS0STBIE	Internal PLL of USBHS0 Stabilization interrupt enable Set and reset by software to enable/disable the USBHS0 PLL stabilization interrupt 0: Disable the USBHS0 PLL stabilization interrupt 1: Enable the USBHS0 PLL stabilization interrupt
11:7	Reserved	Must be kept at reset value.
6	IRC48MSTBIF	IRC48M stabilization interrupt flag Set by hardware when the Internal 48 MHz RC oscillator clock is stable and the IRC48MSTBIE bit is set. Reset by software when setting the IRC48MSTBIC bit. 0: No IRC48M stabilization interrupt generated 1: IRC48M stabilization interrupt generated
5	Reserved	Must be kept at reset value.
4	PLLUSBHS0STBIF	Internal PLL of USBHS0 stabilization interrupt flag Set by hardware when the USBHS0 PLL clock is stable and the PLLUSBHS0STBIE bit is set. Reset by software when setting the PLLUSBHS0STBIC bit. 0: No USBHS0 PLL clock stabilization interrupt generated 1: USBHS0 PLL clock stabilization interrupt generated
3:0	Reserved	Must be kept at reset value.

10.3.44. Clock configuration register 4 (RCU_CFG4)

Address offset: 0xD0
Reset value: 0x0000 0000

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



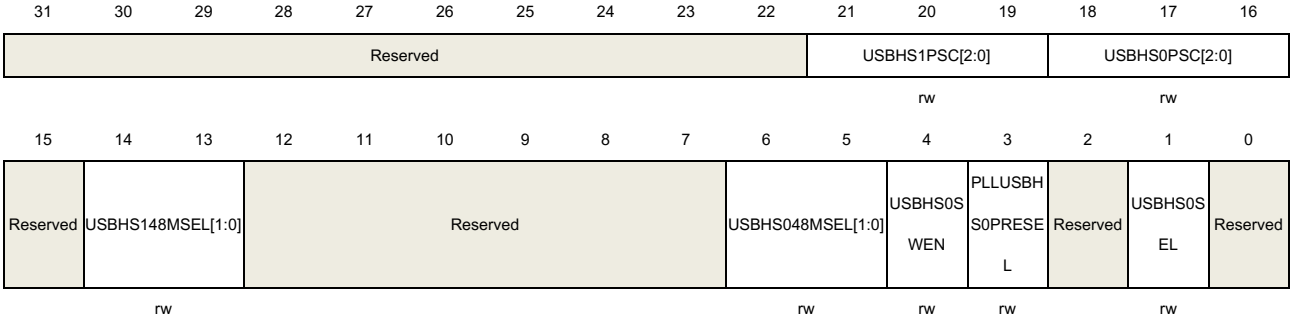
Bits	Fields	Descriptions
31:10	Reserved	Must be kept at reset value.
9:8	EXMCSEL[1:0]	EXMC clock source selection Set and reset by software to control the EXMC clock source 00: CK_AHB selected as EXMC source clock 01: CK_PLL0Q selected as EXMC source clock 10: CK_PLL1R selected as EXMC source clock 11: CK_PER selected as EXMC source clock
7:3	Reserved	Must be kept at reset value.
2:1	OSPISEL[1:0]	OSPI clock source selection Set and reset by software to control the OSPI clock source 00: CK_AHB selected as OSPI source clock 01: CK_PLL0Q selected as OSPI source clock 1x: Reserved
0	SDIO0SEL	SDIO0 clock source selection Set and reset by software to control the SDIO0 clock source. 0: CK_PLL0Q selected as SDIO0 source clock 1: CK_PLL1R selected as SDIO0 source clock

10.3.45. USB clock control register (RCU_USBCLKCTL)

Address offset: 0xD4

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:22	Reserved	Must be kept at reset value.
21:19	USBHS1PSC[2:0]	USBHS1 clock prescaler selection Set and reset by software to control the USBHS1 clock prescaler value. The USBHS1 clock must be 48MHz. These bits can't be reset if the USBHS1 clock is enabled. 000: CK_USBHS1 = CK_PLL1Q / 1 001: CK_USBHS1 = CK_PLL1Q / 2 010: CK_USBHS1 = CK_PLL1Q / 3 011: CK_USBHS1 = CK_PLL1Q / 4 100: CK_USBHS1 = CK_PLL1Q / 5 101: CK_USBHS1 = CK_PLL1Q / 6 110 :CK_USBHS1 = CK_PLL1Q / 7 111 :CK_USBHS1 = CK_PLL1Q / 8
18:16	USBHS0PSC[2:0]	USBHS0 clock prescaler selection Set and reset by software to control the USBHS0 clock prescaler value. The USBHS0 clock must be 48MHz. These bits can't be reset if the USBHS0 clock is enabled. 000: CK_USBHS0 = CK_PLL1Q / 1 001: CK_USBHS0 = CK_PLL1Q / 2 010: CK_USBHS0 = CK_PLL1Q / 3 011: CK_USBHS0 = CK_PLL1Q / 4 100: CK_USBHS0 = CK_PLL1Q / 5 101: CK_USBHS0 = CK_PLL1Q / 6 110 :CK_USBHS0 = CK_PLL1Q / 7 111 :CK_USBHS0 = CK_PLL1Q / 8
15	Reserved	Must be kept at reset value.
14:13	USBHS148MSEL[1:0]	USBHS1 48M clock source selection Set and reset by software. 00: CK_PLL0R selected as USBHS1 48M source clock 01: reserved 10: CK_PLL1Q/USBHS1PSC output selected as USBHS1 48M source clock 11: CK_IRC48M selected as USBHS1 48M source clock
12:7	Reserved	Must be kept at reset value.
6:5	USBHS048MSEL[1:0]	USBHS0 48M clock source selection Set and reset by software. 00: CK_PLL0R selected as USBHS0 48M source clock 01: CK_PLLUSBHS0/USBHS0DV selected as USBHS0 48M source clock 10: CK_PLL1Q/USBHS0PSC selected as USBHS0 48M source clock 11: CK_IRC48M selected as USBHS0 48M source clock

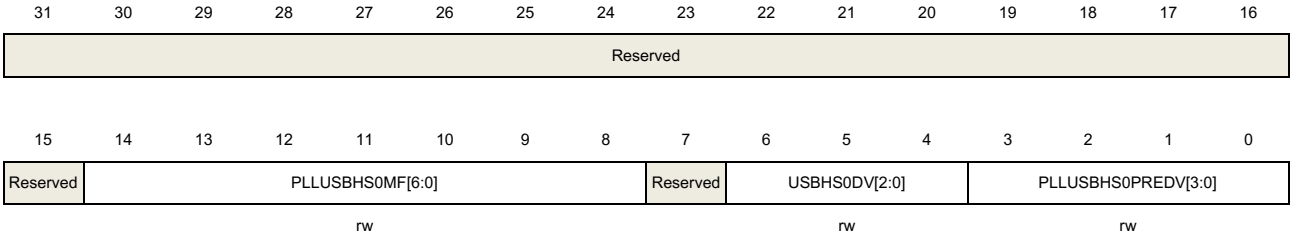
4	USBHS0SWEN	USBHS0 clock source selection enable 0: Hardware switch USBHS0 clock by USBHS0 module 1: Use USBHS0SW to switch USBHS0 clock
3	PLLUSBHS0PRESEL	PLLUSBHS0 clock source preselection Set and reset by software. 0: CK_HXTAL selected as PLLUSBHS0 source clock 1: CK_IRC48M selected as PLLUSBHS0 source clock
2	Reserved	Must be kept at reset value.
1	USBHS0SEL	USBHS0 clock source selection Set and reset by software. 0: 48M selected as USBHS0 source clock 1: 60M selected as USBHS0 source clock
0	Reserved	Must be kept at reset value.

10.3.46. PLLUSB configuration register (RCU_PLLUSBCFG)

Address offset: 0xD8

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:15	Reserved	Must be kept at reset value.
14:8	PLLUSBHS0MF[6:0]	The PLLUSBHS0 clock multiplication factor 0000000: Reserved 0000001: Reserved … 0000111: Reserved 0001000: PLLUSBHS0MF input source clock multiplied by 8 0001001: PLLUSBHS0MF input source clock multiplied by 9 0001010: PLLUSBHS0MF input source clock multiplied by 10 0001011: PLLUSBHS0MF input source clock multiplied by 11 … 1111111: PLLUSBHS0MF input source clock multiplied by 127 **Note:** The frequency of PLLUSBHS0 output clock is no more than 480MHz

7	Reserved	Must be kept at reset value.
6:4	USBHS0DV[2:0]	USBHS0 clock divider factor These bits are set and reset by software. 000: USBHS0DV input source clock divided by 2 001: USBHS0DV input source clock divided by 4 010: USBHS0DV input source clock divided by 6 … 111: USBHS0DV input source clock divided by 16
3:0	PLLUSBHS0PREDV[3:0]	PLLUSBHS0PREDV clock divide factor These bits are set and reset by software. 0000: Reserved 0001: PLLUSBHS0PREDV input source clock divided by 1 0010: PLLUSBHS0PREDV input source clock divided by / 2 … 1111: PLLUSBHS0PREDV input source clock divided by / 15

10.3.47. APB2 additional reset register (RCU_ADDAPB2RST)

Address offset: 0xE0

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										RDCMRST	HDSLRST	BISSRST	ENDATRST	TFMTRST	AFMTRST
										rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													CAN2RST	CAN1RST	CAN0RST
													rw	rw	rw

Bits	Fields	Descriptions
31:22	Reserved	Must be kept at reset value.
21	RDCMRST	RDCM reset This bit is set and reset by software. 0: No reset 1: Reset RDCM unit
20	HDSLRST	HDSL reset This bit is set and reset by software. 0: No reset 1: Reset HDSL unit

19	BISSRST	BISS reset This bit is set and reset by software. 0: No reset 1: Reset BISS unit
18	ENDATRST	ENDAT reset This bit is set and reset by software. 0: No reset 1: Reset ENDAT unit
17	TFMTRST	TFMT reset This bit is set and reset by software. 0: No reset 1: Reset TFMT unit
16	AFMTRST	AFMT reset This bit is set and reset by software. 0: No reset 1: Reset AFMT unit
15:3	Reserved	Must be kept at reset value.
2	CAN2RST	CAN2 reset This bit is set and reset by software. 0: No reset 1: Reset CAN2 unit
1	CAN1RST	CAN1 reset This bit is set and reset by software. 0: No reset 1: Reset CAN1
0	CAN0RST	CAN0 reset This bit is set and reset by software. 0: No reset 1: Reset CAN0

10.3.48. APB2 additional enable register (RCU_ADDAPB2EN)

Address offset: 0xE4

Reset value: 0x0000 0000

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

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

Reserved	CAN2EN	CAN1EN	CAN0EN
	rw	rw	rw

Bits	Fields	Descriptions
31:22	Reserved	Must be kept at reset value.
21	RDCMEN	RDCM clock enable This bit is set and reset by software. 0: Disable RDCM clock 1: Enable RDCM clock
20	HDSLEN	HDSL clock enable This bit is set and reset by software. 0: Disable HDSL clock 1: Enable HDSL clock
19	BISSEN	BISS clock enable This bit is set and reset by software. 0: Disable BISS clock 1: Enable BISS clock
18	ENDATEN	ENDAT clock enable This bit is set and reset by software. 0: Disable ENDAT clock 1: Enable ENDAT clock
17	TFMTEN	TFMT clock enable This bit is set and reset by software. 0: Disable TFMT clock 1: Enable TFMT clock
16	AFMTEN	AFMT clock enable This bit is set and reset by software. 0: Disable AFMT clock 1: Enable AFMT clock
15:3	Reserved	Must be kept at reset value.
2	CAN2EN	CAN2 clock enable This bit is set and reset by software. 0: Disable CAN2 clock 1: Enable CAN2 clock
1	CAN1EN	CAN1 clock enable This bit is set and reset by software. 0: Disable CAN1 clock 1: Enable CAN1 clock

0	CAN0EN	CAN0 clock enable This bit is set and reset by software. 0: Disable CAN0 clock 1: Enable CAN0 clock

10.3.49. APB2 additional sleep enable register (RCU_ADDAPB2SPEN)

Address offset: 0xE8

Reset value: 0x0000 0007

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										RDCMSPEN	HDSLSPEN	BISSSPEN	ENDATSPEN	TFMTSPEN	AFMTSPEN
										rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													CAN2SPEN	CAN1SPEN	CAN0SPEN
													rw	rw	rw

Bits	Fields	Descriptions
31:22	Reserved	Must be kept at reset value.
21	RDCMSPEN	RDCM clock enable in sleep mode This bit is set and reset by software. 0: Disable RDCM clock in sleep mode 1: Enable RDCM clock in sleep mode
20	HDSLSPEN	HDSL clock enable in sleep mode This bit is set and reset by software. 0: Disable HDSL clock in sleep mode 1: Enable HDSL clock in sleep mode
19	BISSSPEN	BISS clock enable in sleep mode This bit is set and reset by software. 0: Disable BISS clock in sleep mode 1: Enable BISS clock in sleep mode
18	ENDATSPEN	ENDAT clock enable in sleep mode This bit is set and reset by software. 0: Disable ENDAT clock in sleep mode 1: Enable ENDAT clock in sleep mode
17	TFMTSPEN	TFMT clock enable in sleep mode This bit is set and reset by software. 0: Disable TFMT clock in sleep mode

		1: Enable TFMT clock in sleep mode
16	AFMTSPEN	AFMT clock enable in sleep mode This bit is set and reset by software. 0: Disable AFMT clock in sleep mode 1: Enable AFMT clock in sleep mode
15:3	Reserved	Must be kept at reset value.
2	CAN2SPEN	CAN2 clock enable in sleep mode This bit is set and reset by software. 0: Disable CAN2 clock in sleep mode 1: Enable CAN2 clock in sleep mode
1	CAN1SPEN	CAN1 clock enable in sleep mode This bit is set and reset by software. 0: Disable CAN1 clock in sleep mode 1: Enable CAN1 clock in sleep mode
0	CAN0SPEN	CAN0 clock enable in sleep mode This bit is set and reset by software. 0: Disable CAN0 clock in sleep mode 1: Enable CAN0 clock in sleep mode

10.3.50. Clock configuration register 5 (RCU_CFG5)

Address offset: 0xEC

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									SPI5SEL[2:0]			Reserved	SPI4SEL[2:0]		
									rw				rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	SPI3SEL[2:0]			Reserved	SPI2SEL[2:0]			Reserved	SPI1SEL[2:0]			Reserved	SPI0SEL[2:0]		
	rw				rw				rw				rw		

Bits	Fields	Descriptions
31:23	Reserved	Must be kept at reset value.
22:20	SPI5SEL[2:0]	SPI5 / I2S5 clock source selection Set and reset by software to control the SPI5 / I2S5 clock source 000: CK_APB2 selected as SPI5 / I2S5 source clock 001: CK_PLL1Q selected as SPI5 / I2S5 source clock 010: CK_PLL2Q selected as SPI5 / I2S5 source clock 011: CK_IRC64MDIV selected as SPI5 / I2S5 source clock 100: CK_LPIRC4M selected as SPI5 / I2S5 source clock

		101: CK_HXTAL selected as SPI5 / I2S5 source clock 110: I2S_CKIN selected as SPI5 / I2S5 source clock 111: reserved
19	Reserved	Must be kept at reset value.
18:16	SPI4SEL[2:0]	SPI4 clock source selection Set and reset by software to control the SPI4 clock source 000: CK_APB2 selected as SPI4 source clock 001: CK_PLL1Q selected as SPI4 source clock 010: CK_PLL2Q selected as SPI4 source clock 011: CK_IRC64MDIV selected as SPI4 source clock 100: CK_LPIRC4M selected as SPI4 source clock 101: CK_HXTAL selected as SPI4 source clock others: reserved
15	Reserved	Must be kept at reset value.
14:12	SPI3SEL[2:0]	SPI3 clock source selection Set and reset by software to control the SPI3 clock source 000: CK_APB2 selected as SPI3 source clock 001: CK_PLL1Q selected as SPI3 source clock 010: CK_PLL2Q selected as SPI3 source clock 011: CK_IRC64MDIV selected as SPI3 source clock 100: CK_LPIRC4M selected as SPI3 source clock 101: CK_HXTAL selected as SPI3 source clock others: reserved
11	Reserved	Must be kept at reset value.
10:8	SPI2SEL[2:0]	SPI2 / I2S2 clock source selection Set and reset by software to control the SPI2 / I2S2 clock source 000: CK_PLL0Q selected as SPI2 / I2S2 source clock 001: CK_PLL1P selected as SPI2 / I2S2 source clock 010: CK_PLL2P selected as SPI2 / I2S2 source clock 011: I2S_CKIN selected as SPI2 / I2S2 source clock 100: CK_PER selected as SPI2 / I2S2 source clock others: reserved
7	Reserved	Must be kept at reset value.
6:4	SPI1SEL[2:0]	SPI1 / I2S1 clock source selection Set and reset by software to control the SPI1 / I2S1 clock source 000: CK_PLL0Q selected as SPI1 / I2S1 source clock 001: CK_PLL1P selected as SPI1 / I2S1 source clock 010: CK_PLL2P selected as SPI1 / I2S1 source clock 011: I2S_CKIN selected as SPI1 / I2S1 source clock 100: CK_PER selected as SPI1 / I2S1 source clock

		others: reserved
3	Reserved	Must be kept at reset value.
2:0	SPI0SEL[2:0]	SPI0 / I2S0 clock source selection Set and reset by software to control the SPI0 / I2S0 clock source 000: CK_PLL0Q selected as SPI0 / I2S0 source clock 001: CK_PLL1P selected as SPI0 / I2S0 source clock 010: CK_PLL2P selected as SPI0 / I2S0 source clock 011: I2S_CKIN selected as SPI0 / I2S0 source clock 100: CK_PER selected as SPI0 / I2S0 source clock others: reserved

10.3.51. Clock configuration register 6 (RCU_CFG6)

Address offset: 0xF0

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					HDSLDIV[2:0]			Reserved							
					rw										

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENDATDIV[3:0]				TFMTDIV[3:0]				AFMTDIV[3:0]				HDSLSEL[1:0]		ENCODERSEL[1:0]	
rw				rw				rw				rw		rw	

Bits	Fields	Descriptions
31:27	Reserved	Must be kept at reset value.
26:24	HDSLDIV[2:0]	HDSL divider The divider is HDSLDIV +1.
23:16	Reserved	Must be kept at reset value.
15:12	ENDATDIV[3:0]	ENDAT divider The divider is ENDATDIV +1.
11:8	TFMTDIV[3:0]	TFMT divider The divider is TFMTDIV +1.
7:4	AFMTDIV[3:0]	AFMT divider The divider is AFMTDIV +1.
3:2	HDSLSEL[1:0]	HDSL clock source selection select 00: CK_SYS selected as HDSL source clock 01: CK_PLL0R selected as HDSL source clock 10: CK_PLL1P selected as HDSL source clock

		11: CK_PLL2Q selected as HDSL source clock
1:0	ENCODERSEL[1:0]	ENCODER clock source selection select 00: CK_SYS selected as ENCODER source clock 01: CK_PLL0Q selected as ENCODER source clock 10: CK_PLL1R selected as ENCODER source clock 11: CK_PLL2R selected as ENCODER source clock

10.3.52. Clock frequency monitor configuration register 0 (RCU_CKFMCFG0)

Address offset: 0xF4

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										LPIRC4MCKFFIE	LPIRC4MCKFFC	LPIRC4MCKFFF	LPIRC4MCKFMC[1:0]		LPIRC4MCKFMEN
										rw	rw	r	rw		rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										IRC64MCKFFIE	IRC64MCKFFC	IRC64MCKFFF	IRC64MCKFMC[1:0]		IRC64MCKFMEN
										rw	rw	r	rw		rw

Bits	Fields	Descriptions
31:22	Reserved	Must be kept at reset value.
21	LPIRC4MCKFFIE	LPIRC4M clock frequency failure interrupt enable 0: Disable LPIRC4M clock failure interrupt 1: Enable LPIRC4M clock failure interrupt
20	LPIRC4MCKFFC	LPIRC4M clock frequency failure flag clear Write 1 by software to reset the LPIRC4MCKFFF flag. 0: Not reset LPIRC4MCKFFF flag 1: Reset LPIRC4MCKFFF flag
19	LPIRC4MCKFFF	LPIRC4M clock frequency failure flag This bit indicates whether LPIRC4M clock frequency is out of monitoring range. 0: LPIRC4M clock frequency is not out of monitoring range 1: LPIRC4M clock frequency is out of monitoring range
18:17	LPIRC4MCKFMC	LPIRC4M clock frequency monitor configuration This bit is set and reset by software. 00: The clock frequency monitoring range is +/- 5% 01: The clock frequency monitoring range is +/- 10% 10: The clock frequency monitoring range is +/- 15% 11: The clock frequency monitoring range is +/- 20%

16	LPIRC4MCKFMEN	LPIRC4M clock frequency monitor enable This bit is set and reset by software. 0: Disable the Low Power Internal 4M RC oscillator (LPIRC4M) clock frequency monitor 1: Enable the Low Power Internal 4M RC oscillator (LPIRC4M) clock frequency monitor
15:6	Reserved	Must be kept at reset value.
5	IRC64MCKFFIE	IRC64M clock frequency failure interrupt enable 0: Disable IRC64M clock frequency failure interrupt 1: Enable IRC64M clock frequency failure interrupt
4	IRC64MCKFFC	IRC64M clock frequency failure flag clear Write 1 by software to reset the IRC64MCKFFF flag. 0: Not reset IRC64MCKFFF flag 1: Reset IRC64MCKFFF flag
3	IRC64MCKFFF	IRC64M clock frequency failure flag This bit indicates whether IRC64M clock is out of monitoring range. 0: IRC64M clock frequency is not out of monitoring range 1: IRC64M clock frequency is out of monitoring range
2:1	IRC64MCKFMC[1:0]	IRC64M clock frequency monitor configuration This bit is set and reset by software. 00: The clock frequency monitoring range is +/- 5% 01: The clock frequency monitoring range is +/- 10% 10: The clock frequency monitoring range is +/- 15% 11: The clock frequency monitoring range is +/- 20%
0	IRC64MCKFMEN	IRC64M clock frequency monitor enable This bit is set and reset by software. 0: Disable the Internal 64M RC oscillators (IRC64M) clock frequency monitor 1: Enable the Internal 64M RC oscillators (IRC64M) clock frequency monitor

10.3.53. Clock frequency monitor configuration register 1 (RCU_CKFMCFG1)

Address offset: 0xF8

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
HXTALCKFFC	HXTALCKFFF	Reserved						HXTALCKFFIE	HXTALCKFMIN[10:0]						
rw	r							rw	rw						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

HXTALCKFMIN[10:0]	HXTALCKFMAX[10:0]	HXTALCKFMEN
rw	rw	rw

Bits	Fields	Descriptions
31	HXTALCKFFC	HXTAL clock frequency failure flag clear Write 1 by software to reset the HXTALCKFFF flag. 0: Not reset HXTALCKFFF flag 1: Reset HXTALCKFFF flag
30	HXTALCKFFF	HXTAL clock frequency failure flag This bit indicates whether HXTAL clock frequency is out of monitoring range. 0: HXTAL clock frequency is not out of monitoring range 1: HXTAL clock frequency is out of monitoring range
29:24	Reserved	Must be kept at reset value.
23	HXTALCKFFIE	HXTAL clock frequency failure interrupt enable 0: Disable HXTAL clock frequency failure interrupt 1: Enable HXTAL clock frequency failure interrupt
22:12	HXTALCKFMIN[10:0]	HXTAL clock frequency monitor configuration minimum value This bit is set and reset by software.
11:1	HXTALCKFMAX[10:0]	HXTAL clock frequency monitor configuration maximum value This bit is set and reset by software.
0	HXTALCKFMEN	HXTAL clock frequency monitor enable This bit is set and reset by software. 0: Disable the HXTAL clock frequency monitor 1: Enable the HXTAL clock frequency monitor

10.3.54. Clock frequency monitor configuration register 2 (RCU_CKFMCFG2)

Address offset: 0xFC

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
PLL0PCKFFC	PLL0PCKFFF	Reserved						PLL0PCKFFIE	PLL0PCKFMIN[10:0]						
rw	r							rw	rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PLL0PCKFMIN[10:0]								PLL0PCKFMAX[10:0]							PLL0PCKFMEN
rw								rw							rw

Bits	Fields	Descriptions
31	PLL0PCKFFC	PLL0P clock frequency failure flag clear Write 1 by software to reset the PLL0PCKFFF flag. 0: Not reset PLL0PCKFFF flag 1: Reset PLL0PCKFFF flag
30	PLL0PCKFFF	PLL0P clock frequency failure flag This bit indicates whether PLL0P clock frequency is out of monitoring range. 0: PLL0P clock frequency is not out of monitoring range 1: PLL0P clock frequency is out of monitoring range
29:24	Reserved	Must be kept at reset value.
23	PLL0PCKFFIE	PLL0P clock frequency failure interrupt enable 0: Disable PLL0P clock frequency failure interrupt 1: Enable PLL0P clock frequency failure interrupt
22:12	PLL0PCKFMIN[10:0]	PLL0P clock frequency monitor configuration minimum value This bit is set and reset by software.
11:1	PLL0PCKFMAX[10:0]	PLL0P clock frequency monitor configuration maximum value This bit is set and reset by software.
0	PLL0PCKFMEN	PLL0P clock frequency monitor enable This bit is set and reset by software. 0: Disable the PLL0P clock frequency monitor 1: Enable the PLL0P clock frequency monitor

11. Clock trim controller (CTC)

11.1. Overview

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

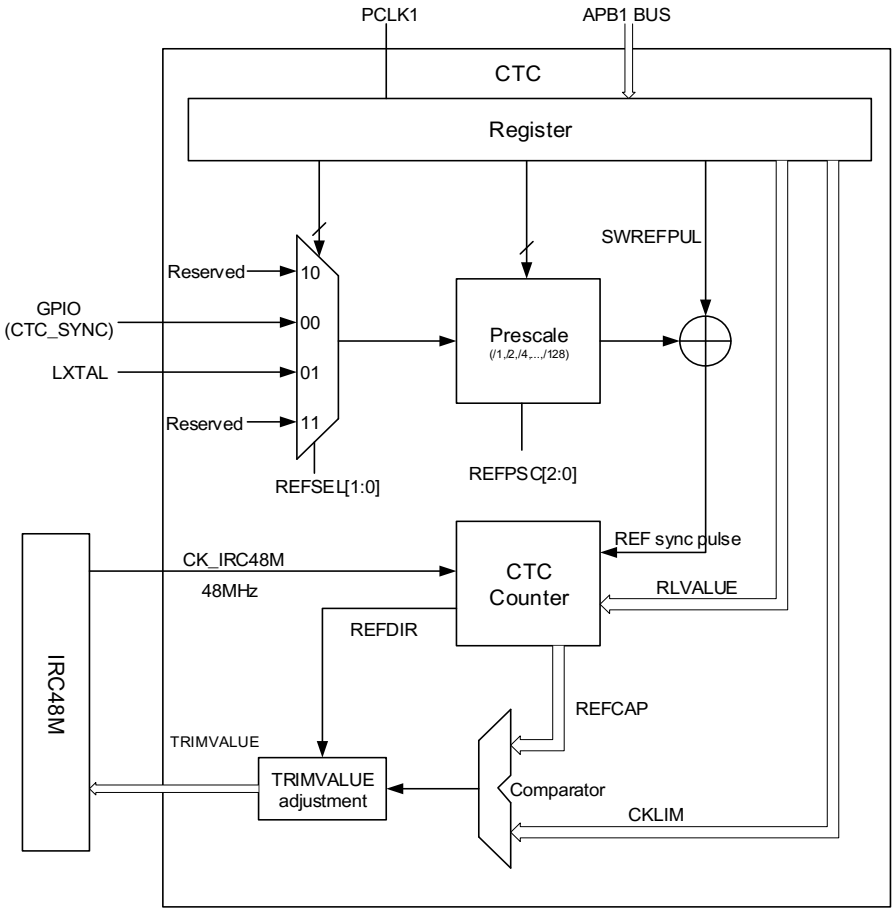
11.2. Characteristics

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

11.3. Function overview

Figure 11-1. CTC overview provides details on the internal configuration of the CTC.

Figure 11-1. CTC overview



11.3.1. REF sync pulse generator

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

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

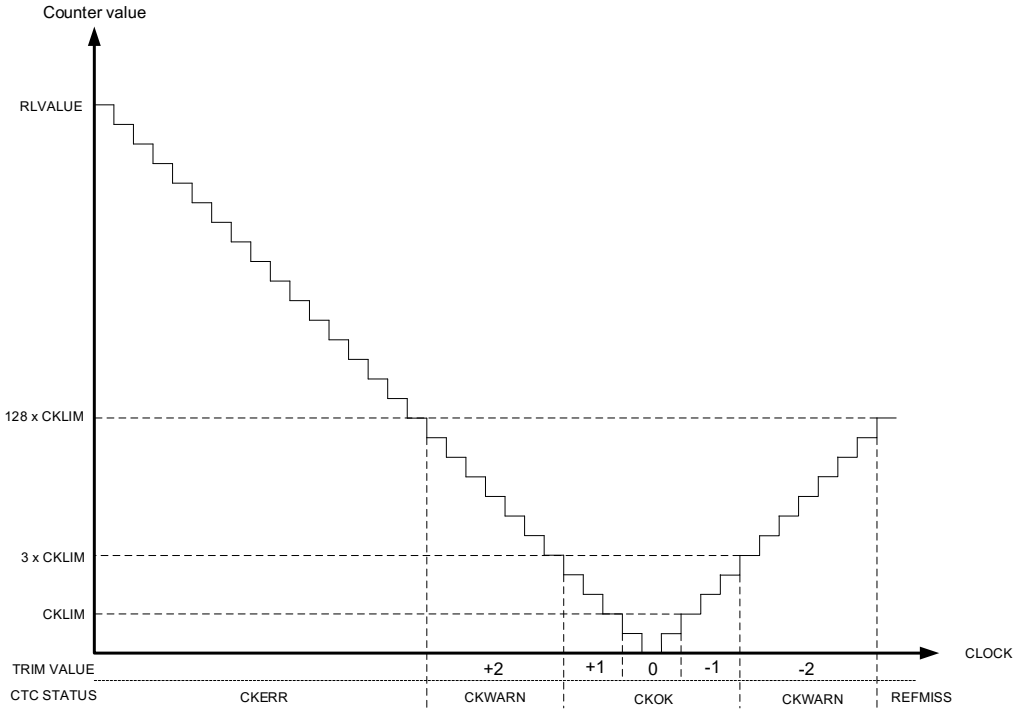
Thirdly, if a software reference pulse needed, write 1 to SWREFPUL bit in CTC_CTL0 register. The software reference pulse generated in last step is logical OR with the external reference pulse.

11.3.2. CTC trim counter

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

and then up- counting to 128 x CKLIM (defined in CTC_CTL1 register), and then stop until next REF sync pulse detected. If any REF sync pulse detected, the current CTC trim counter value is captured to REFCAP in status register (CTC_STAT), and the counter direction is captured to REFDIR in status register (CTC_STAT). The detail is showing in ***Figure 11-2. CTC trim counter***.

Figure 11-2. CTC trim counter



11.3.3. Frequency evaluation and automatically trim process

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

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

- Counter < CKLIM when REF sync pulse is detected.

  The CKOKIF in CTC_STAT register set, and an interrupt generated if CKOKIE bit in

CTC_CTL0 register is 1.

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

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

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

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

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

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

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

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

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

  The TRIMVALUE in CTC_CTL0 register is not changed

- Counter = 128 x CKLIM when up-counting.

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

  The TRIMVALUE in CTC_CTL0 register is not changed.

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

11.3.4. Software program guide

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

$$RLVALUE = (F_{clock} / F_{REF}) - 1 \quad (11\text{-}1)$$

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

$$CKLIM=(Fclock/\ FREF\ )\ \times 0.12\%\ /2 \quad (11\text{-}2)$$

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

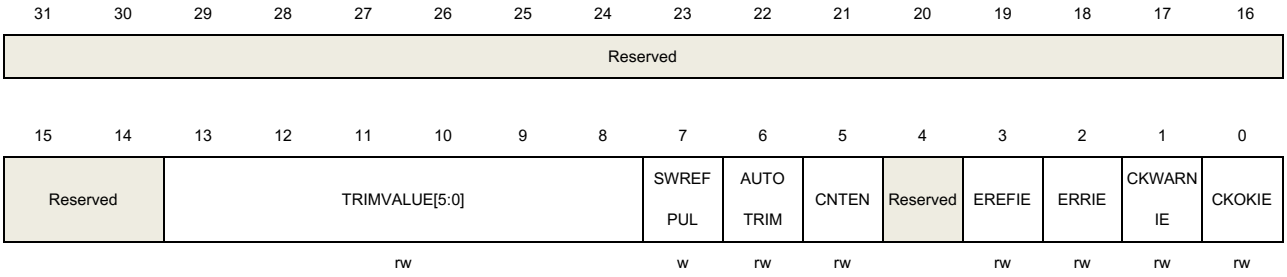
11.4. Register definition

CTC base address: 0x4000 8400

11.4.1. Control register 0 (CTC_CTL0)

Address offset: 0x00
Reset value: 0x0000 2000

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



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

		1: CTC trim counter enabled.
4	Reserved	Must be kept at reset value.
3	EREFIE	EREFIF interrupt enable 0: EREFIF interrupt disable 1: EREFIF interrupt enable
2	ERRIE	Error (ERRIF) interrupt enable 0: ERRIF interrupt disable 1: ERRIF interrupt enable
1	CKWARNIE	Clock trim warning (CKWARNIF) interrupt enable 0: CKWARNIF interrupt disable 1: CKWARNIF interrupt enable
0	CKOKIE	Clock trim OK (CKOKIF) interrupt enable 0: CKOKIF interrupt disable 1: CKOKIF interrupt enable

11.4.2. Control register 1 (CTC_CTL1)

Address offset: 0x04
Reset value: 0x2022 BB7F

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

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

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

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

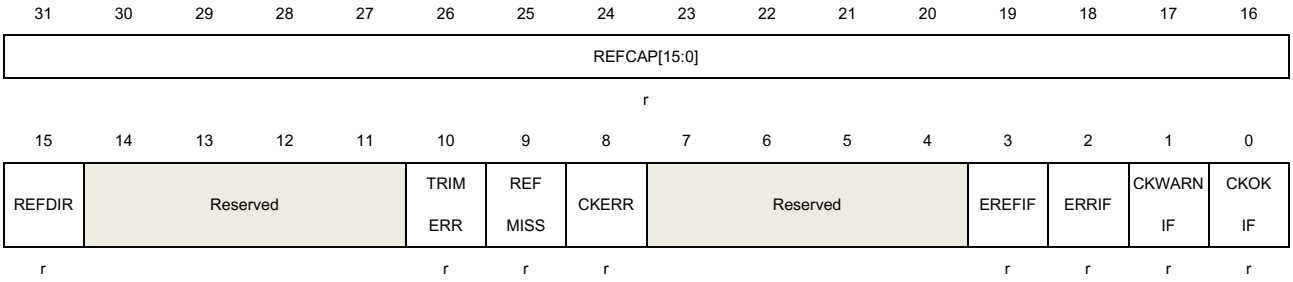
		Other values are reserved.
27	Reserved	Must be kept at reset value.
26:24	REFPSC[2:0]	Reference signal source prescaler These bits are set and cleared by software 000: Reference signal not divided 001: Reference signal divided by 2 010: Reference signal divided by 4 011: Reference signal divided by 8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

11.4.3. Status register (CTC_STAT)

Address offset: 0x08

Reset value: 0x0000 0000

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



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

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

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

11.4.4. Interrupt clear register (CTC_INTC)

Address offset: 0x0C
Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												EREFIC	ERRIC	CKWARNIC	CKOKIC
												w	w	w	w

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

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

12. Clock phase delay module (CPDM)

12.1. 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.

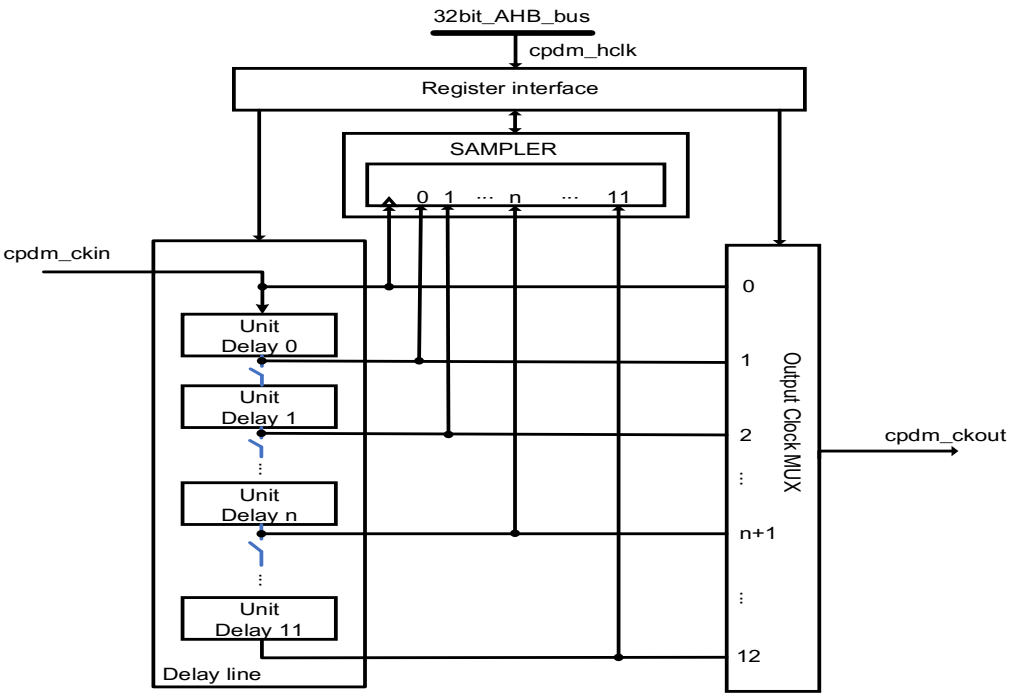
12.2. Characteristics

- Supports the input clock frequency ranges: 25 MHz ~ 208MHz.
- Supports up to 12 output clock phase selections.

12.3. 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.

Figure 12-1. 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.

12.3.1. 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[6:0] 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.

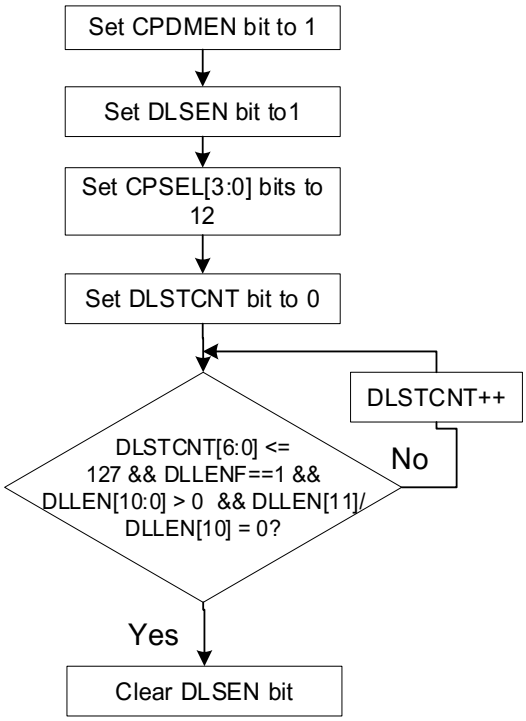
Note: The unit delay and output clock phase can only be changed when DLSEN is 1.

12.3.2. 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 12-2. CPDM delay line length configuration flowchart*** shows the configuration process.

Figure 12-2. CPDM delay line length configuration flowchart



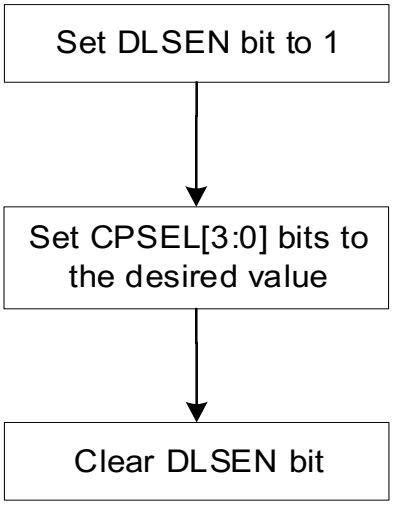
Note:

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 12-3. CPDM output clock phase configuration flowchart*.

Figure 12-3. CPDM output clock phase configuration flowchart



12.4. Register definition

CPDM(SDIO0) base address: 0x5200 8000

CPDM(SDIO1) base address: 0x4802 2800

CPDM(OSPI0) base address: 0x5200 6000

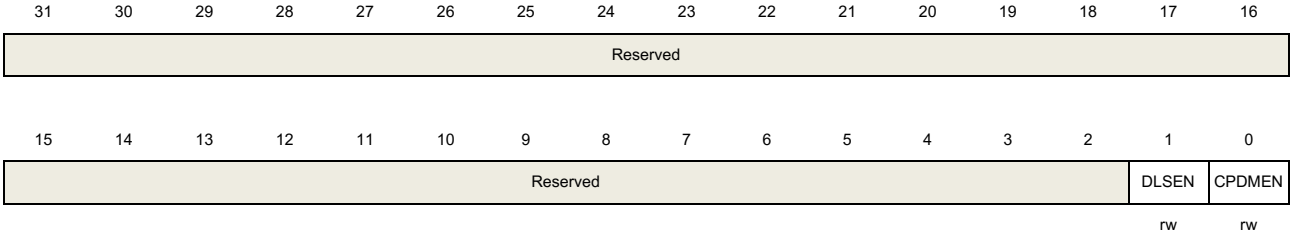
CPDM(OSPI1) base address: 0x5200 B000

12.4.1. Control register (CPDM_CTL)

Address offset: 0x00

Reset value: 0x0000 0000

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



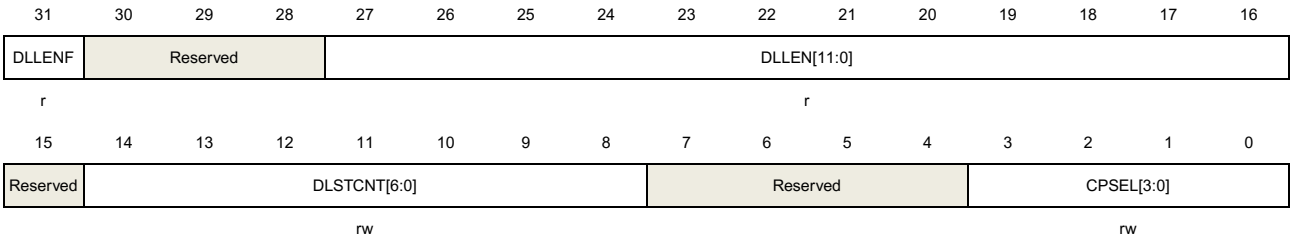
Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	DLSEN	CPDM delay line sample module enable bit 0: disable CPDM delay line sample module 1: enable CPDM delay line sample module
0	CPDMEN	CPDM enable bit 0: disable CPDM 1: enable CPDM

12.4.2. 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

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

13.1. Overview

There are up to 135 general purpose I/O pins (GPIO), named PA0~PA15, PB0~PB15, PC0~PC15, PD0~PD15, PE0~PE15, PF0~PF15, PG0~PG15, PH0~PH15, PJ8~PJ11, PK0~PK2 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.

13.2. 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.

13.3. 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

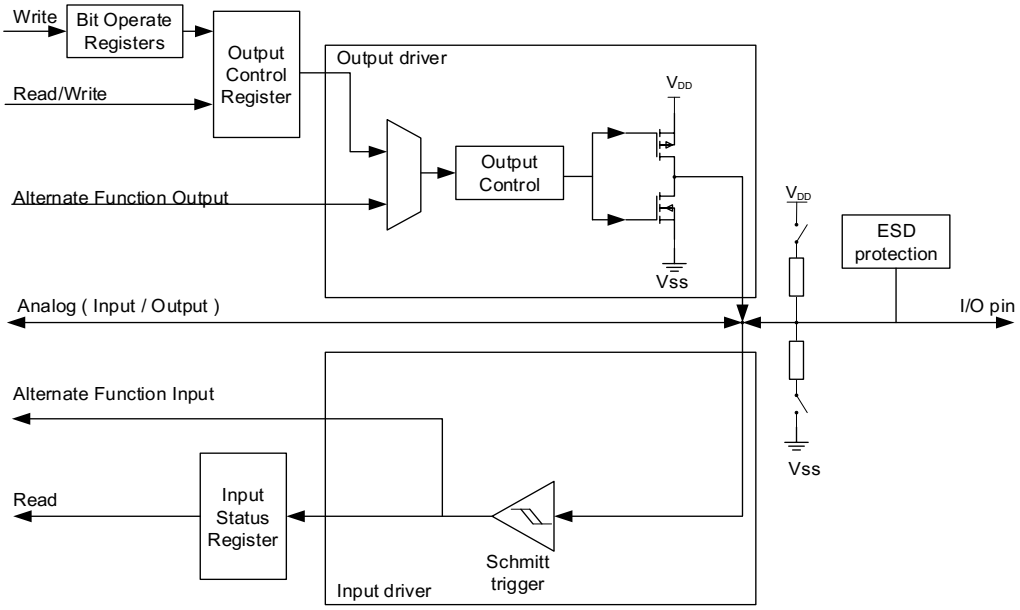
be configured by GPIO output speed registers (GPIOx_OSPD). Each port can be configured as floating (no pull-up and pull-down), pull-up or pull-down function by GPIO pull-up/pull-down registers (GPIOx_PUD).

Table 13-1. 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 13-1. Basic structure of a standard I/O port bit shows the basic structure of an I/O port bit.

Figure 13-1. Basic structure of a standard I/O port bit



13.3.1. GPIO pin configuration

During or just after the reset period, the alternative functions are all inactive and the GPIO ports (except ports A/B) are configured into the analog 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 output mode

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

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

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

13.3.2. External interrupt/event lines

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

13.3.3. 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 (port B, C, D, E has thirty-two) alternate functions can be configured by GPIO alternate functions selected registers (GPIOx_AFSELy (y = 0..2)). The detail alternate function assignments for each port are in the device datasheet.

13.3.4. 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.

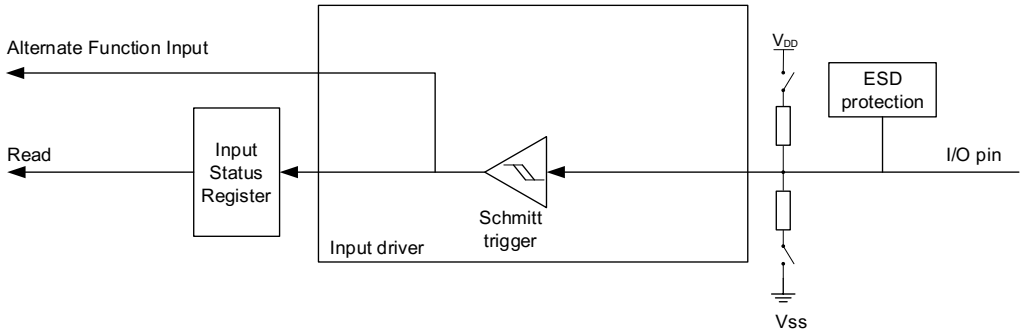
13.3.5. Input configuration

When GPIO pin is configured as input:

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

Figure 13-2. Input configuration shows the input configuration.

Figure 13-2. Input configuration



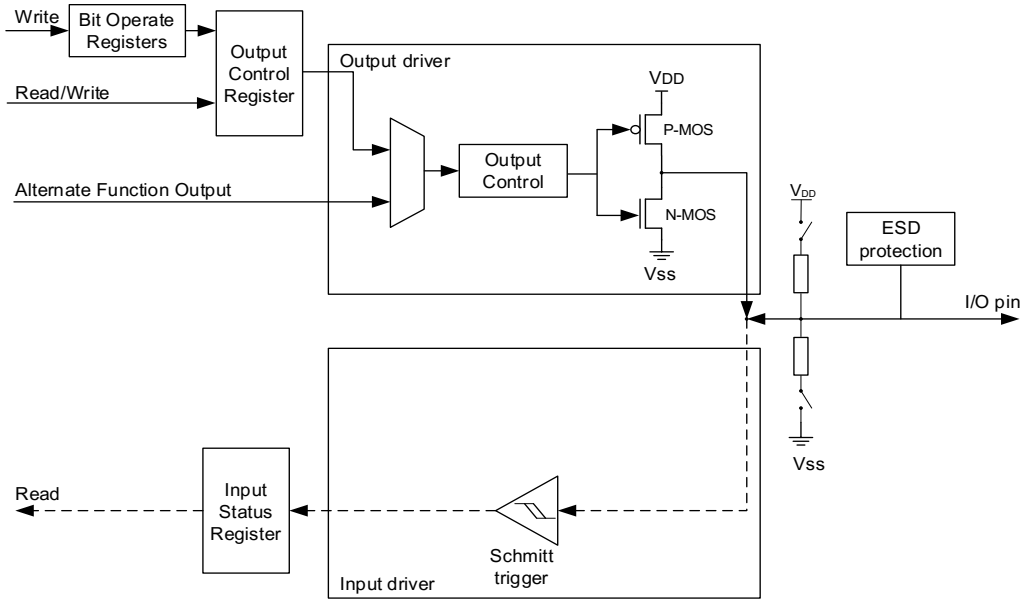
13.3.6. Output configuration

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 13-3. Output configuration shows the output configuration.

Figure 13-3. Output configuration



13.3.7. 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 13-4. Analog configuration shows the analog configuration.

Figure 13-4. Analog configuration



13.3.8. Alternate function (AF) configuration

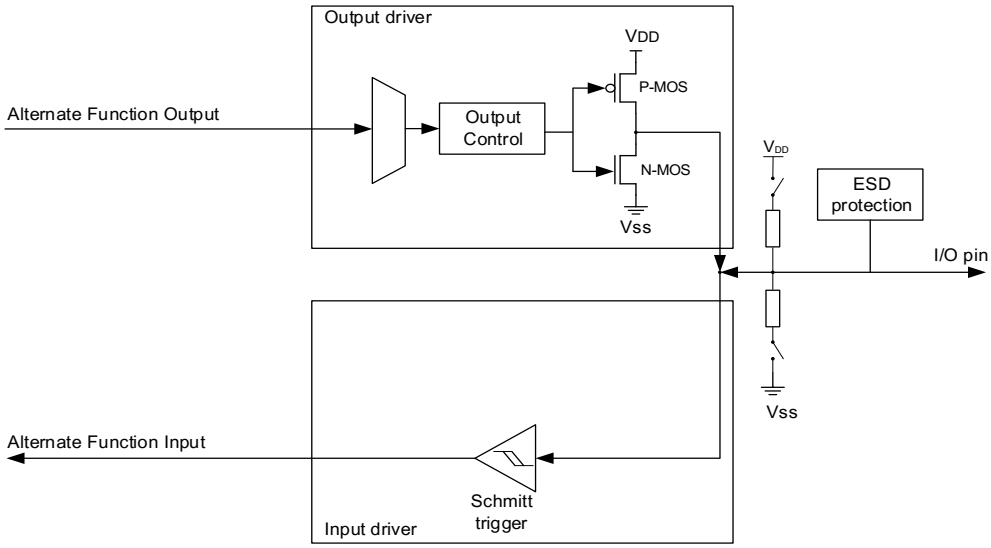
To suit for different device packages, the GPIO supports some alternate functions mapped to some other pins by software.

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 13-5. Alternate function configuration shows the alternate function configuration.

Figure 13-5. Alternate function configuration



13.3.9. 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..2). 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.

13.3.10. 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.

13.3.11. I/O compensation unit

(1) The compensation cell is used to control the slew rate (rise time/fall time) of I/O switching, in order to reduce I/O noise on the power supply. This cell provides the optimal compensation code for the current temperature, environment, and other conditions. When the CPS_RDY bit in the ***I/O compensation control register (SYSCFG_CPSCTL)*** is set, the compensation code stored in this area can be read. Users may also configure the compensation code by programming the ***I/O compensation cell code configuration register (SYSCFG_CPSCCCFG)*** in software.

The I/O compensation cell supports 2 voltage ranges: 1.71 V to 2.7 V and 2.7 V to 3.6 V. Both voltage ranges refer to the VDD supply voltage of the chip. When VDD is in the range of 1.71 V to 2.7 V, the VDD voltage is relatively low, resulting in slower I/O switching speeds and lower I/O-induced noise; therefore, configuring CPS is not required. When VDD is in the range of 2.7 V to 3.6 V, the VDD voltage is higher and I/O switching is faster; in this case, CPS can be configured to 1 to reduce I/O noise.

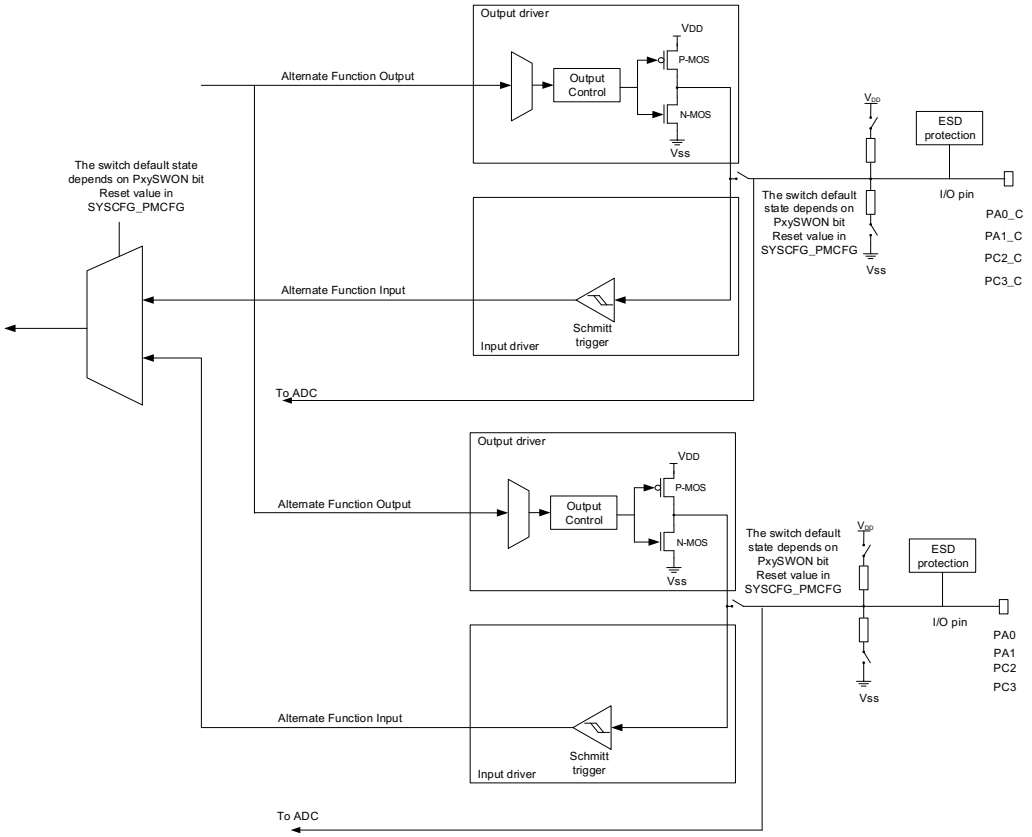
(2) When the supply voltage is below 2.7 V, if the I/O switching speed is insufficient, the IOSPDOP bit in the ***I/O compensation control register (SYSCFG_CPSCTL)*** can be set to 1 to increase the I/O speed at low voltage.

13.3.12. Analog configuration for ADC

Some pins are connected directly to PA0_C、PA1_C、PC2_C and PC3_C ADC analog input as show ***Figure 13-6. Analog configuration for ADC***.

When PxySWON=1 (refer to SYSCFG_PMCFG0 register), Pxy enables input/output functionality, and Pxy_C disables input/output functionality (Pxy_C can only use analog functionality). When PxySWON=0, Pxy_C enables input/output functionality, and Pxy disables input/output functionality (Pxy can only use analog functionality).

Figure 13-6. Analog configuration for ADC



13.3.13. 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.

Note: When GPIO filtering is enabled, the status of the GPIOx_ISTAT register and the signal output to EXTI are the filtered GPIO signals. When GPIOx filtering is disabled, the status of the GPIOx_ISTAT register and the signal output to EXTI are the unfiltered GPIO signals.

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.

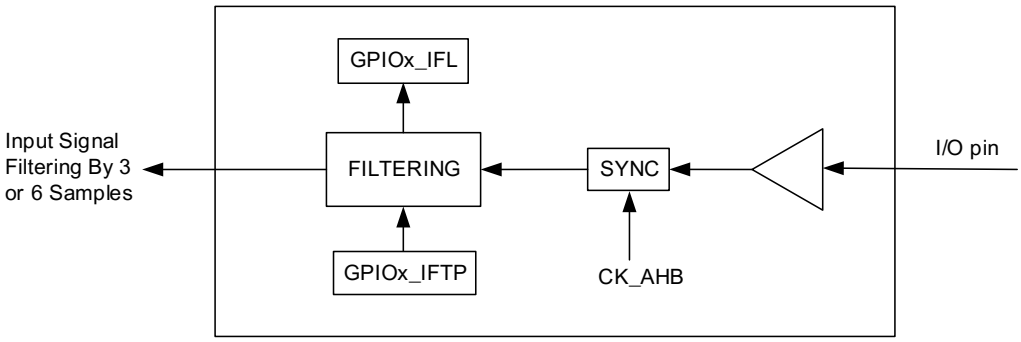
Synchronization to CK_AHB only

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 13-7. 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 sampling 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, GPIOx0 to GPIOx7 use FLPRD0, GPIOx8 to GPIOx15 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 13-8. 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 13-8. Input filtering clock cycle***, the input filtering configuration is as follows:

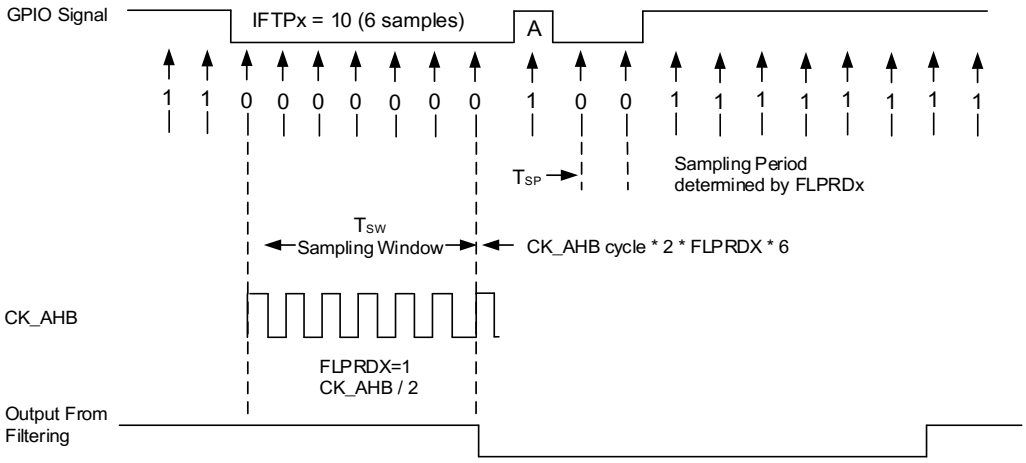
- IFTP0[1:0]=10 in GPIOx_IFTP register, which means there are 6 sampling points;
- FLPRD0=1 in GPIOx_IFL register, sampling period

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

The configuration results are as follows:

- The sampling width is: $T_{SW}=6\times T_{SP}=6\times 2\times FLPRD0\times T_{CK_AHB}=6\times 2\times T_{CK_AHB}$;
- 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 sampling 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 13-8. Input filtering clock cycle***, the interference (A) is shorter than the total sampling window, so it will be filtered.

Figure 13-8. Input filtering clock cycle



13.4. Register definition

GPIOA base address: 0x5802 0000

GPIOB base address: 0x5802 0400

GPIOC base address: 0x5802 0800

GPIOD base address: 0x5802 0C00

GPIOE base address: 0x5802 1000

GPIOF base address: 0x5802 1400

GPIOG base address: 0x5802 1800

GPIOH base address: 0x5802 1C00

GPIOJ base address: 0x5802 2400

GPIOK base address: 0x5802 2800

13.4.1. Port control register (GPIOx_CTL, x=A...H, J, K)

Address offset: 0x00

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

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

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16
CTL15[1:0]	CTL14[1:0]	CTL13[1:0]	CTL12[1:0]	CTL11[1:0]	CTL10[1:0]	CTL9[1:0]	CTL8[1:0]
rw	rw	rw	rw	rw	rw	rw	rw
15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
CTL7[1:0]	CTL6[1:0]	CTL5[1:0]	CTL4[1:0]	CTL3[1:0]	CTL2[1:0]	CTL1[1:0]	CTL0[1:0]
rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:30	CTL15[1:0]	Pin 15 configuration bits These bits are set and cleared by software. Refer to CTL0[1:0] description
29:28	CTL14[1:0]	Pin 14 configuration bits These bits are set and cleared by software. Refer to CTL0[1:0] description
27:26	CTL13[1:0]	Pin 13 configuration bits These bits are set and cleared by software. Refer to CTL0[1:0] description

25:24	CTL12[1:0]	Pin 12 configuration bits These bits are set and cleared by software. Refer to CTL0[1:0] description
23:22	CTL11[1:0]	Pin 11 configuration bits These bits are set and cleared by software. Refer to CTL0[1:0] description
21:20	CTL10[1:0]	Pin 10 configuration bits These bits are set and cleared by software. Refer to CTL0[1:0] description
19:18	CTL9[1:0]	Pin 9 configuration bits These bits are set and cleared by software. Refer to CTL0[1:0] description
17:16	CTL8[1:0]	Pin 8 configuration bits These bits are set and cleared by software. Refer to CTL0[1:0] description
15:14	CTL7[1:0]	Pin 7 configuration bits These bits are set and cleared by software. Refer to CTL0[1:0] description
13:12	CTL6[1:0]	Pin 6 configuration bits These bits are set and cleared by software. Refer to CTL0[1:0] description
11:10	CTL5[1:0]	Pin 5 configuration bits These bits are set and cleared by software. Refer to CTL0[1:0] description
9:8	CTL4[1:0]	Pin 4 configuration bits These bits are set and cleared by software. Refer to CTL0[1:0] description
7:6	CTL3[1:0]	Pin 3 configuration bits These bits are set and cleared by software. Refer to CTL0[1:0] description
5:4	CTL2[1:0]	Pin 2 configuration bits These bits are set and cleared by software. Refer to CTL0[1:0] description
3:2	CTL1[1:0]	Pin 1 configuration bits These bits are set and cleared by software. Refer to CTL0[1:0] description
1:0	CTL0[1:0]	Pin 0 configuration bits These bits are set and cleared by software.

00: Input mode
01: GPIO output mode
10: Alternate function mode
11: Analog mode(reset value)

13.4.2. Port output mode register (GPIOx_OMODE, x=A…H, J, K)

Address offset: 0x04
Reset value: 0x0000 0000

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

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

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

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

These bits are set and cleared by software.
Refer to OM0 description

8	OM8	Pin 8 output mode bit These bits are set and cleared by software. Refer to OM0 description
7	OM7	Pin 7 output mode bit These bits are set and cleared by software. Refer to OM0 description
6	OM6	Pin 6 output mode bit These bits are set and cleared by software. Refer to OM0 description
5	OM5	Pin 5 output mode bit These bits are set and cleared by software. Refer to OM0 description
4	OM4	Pin 4 output mode bit These bits are set and cleared by software. Refer to OM0 description
3	OM3	Pin 3 output mode bit These bits are set and cleared by software. Refer to OM0 description
2	OM2	Pin 2 output mode bit These bits are set and cleared by software. Refer to OM0 description
1	OM1	Pin 1 output mode bit These bits are set and cleared by software. Refer to OM0 description
0	OM0	Pin 0 output mode bit These bits are set and cleared by software. 0: Output push-pull mode (reset value) 1: Output open-drain mode

13.4.3. Port output speed register (GPIOx_OSPD, x=A...H, J, K)

Address offset: 0x08
Reset value: 0x0C00 0000 for port A; 0x0000 00C0 for port B; 0x0000 0000 for others.

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

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

OSPD15[1:0]	OSPD14[1:0]	OSPD13[1:0]	OSPD12[1:0]	OSPD11[1:0]	OSPD10[1:0]	OSPD9[1:0]	OSPD8[1:0]
rw	rw	rw	rw	rw	rw	rw	rw
15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
OSPD7[1:0]	OSPD6[1:0]	OSPD5[1:0]	OSPD4[1:0]	OSPD3[1:0]	OSPD2[1:0]	OSPD1[1:0]	OSPD0[1:0]
rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:30	OSPD15[1:0]	Pin 15 output max speed bits These bits are set and cleared by software. Refer to OSPD0[1:0] description
29:28	OSPD14[1:0]	Pin 14 output max speed bits These bits are set and cleared by software. Refer to OSPD0[1:0] description
27:26	OSPD13[1:0]	Pin 13 output max speed bits These bits are set and cleared by software. Refer to OSPD0[1:0] description
25:24	OSPD12[1:0]	Pin 12 output max speed bits These bits are set and cleared by software. Refer to OSPD0[1:0] description
23:22	OSPD11[1:0]	Pin 11 output max speed bits These bits are set and cleared by software. Refer to OSPD0[1:0] description
21:20	OSPD10[1:0]	Pin 10 output max speed bits These bits are set and cleared by software. Refer to OSPD0[1:0] description
19:18	OSPD9[1:0]	Pin 9 output max speed bits These bits are set and cleared by software. Refer to OSPD0[1:0] description
17:16	OSPD8[1:0]	Pin 8 output max speed bits These bits are set and cleared by software. Refer to OSPD0[1:0] description
15:14	OSPD7[1:0]	Pin 7 output max speed bits These bits are set and cleared by software. Refer to OSPD0[1:0] description
13:12	OSPD6[1:0]	Pin 6 output max speed bits These bits are set and cleared by software. Refer to OSPD0[1:0] description
11:10	OSPD5[1:0]	Pin 5 output max speed bits

		These bits are set and cleared by software. Refer to OSPD0[1:0] description
9:8	OSPD4[1:0]	Pin 4 output max speed bits These bits are set and cleared by software. Refer to OSPD0[1:0] description
7:6	OSPD3[1:0]	Pin 3 output max speed bits These bits are set and cleared by software. Refer to OSPD0[1:0] description
5:4	OSPD2[1:0]	Pin 2 output max speed bits These bits are set and cleared by software. Refer to OSPD0[1:0] description
3:2	OSPD1[1:0]	Pin 1 output max speed bits These bits are set and cleared by software. Refer to OSPD0[1:0] description
1:0	OSPD0[1:0]	Pin 0 output max speed bits These bits are set and cleared by software. 00: Output max speed 12M (reset value) 01: Output max speed 60M 10: Output max speed 85M 11: Output max speed 100/220M

13.4.4. Port pull-up/down register (GPIOx_PUD, x=A…H, J, K)

Address offset: 0x0C

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

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

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16
PUD15[1:0]	PUD14[1:0]	PUD13[1:0]	PUD12[1:0]	PUD11[1:0]	PUD10[1:0]	PUD9[1:0]	PUD8[1:0]
rw	rw	rw	rw	rw	rw	rw	rw
15 14	**13 12**	**11 10**	**9 8**	**7 6**	**5 4**	**3 2**	**1 0**
PUD7[1:0]	PUD6[1:0]	PUD5[1:0]	PUD4[1:0]	PUD3[1:0]	PUD2[1:0]	PUD1[1:0]	PUD0[1:0]
rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:30	PUD15[1:0]	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

13.4.5. Port input status register (GPIOx_ISTAT, x=A…H, J, K)

Address offset: 0x10

Reset value: 0x0000 XXXX

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

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

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

13.4.6. Port output control register (GPIOx_OCTL, x=A…H, J, K)

Address offset: 0x14

Reset value: 0x0000 0000

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

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

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

rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

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

13.4.7. Port bit operate register (GPIOx_BOP, x=A...H, J, K)

Address offset: 0x18

Reset value: 0x0000 0000

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

13.4.8. Port configuration lock register (GPIOx_LOCK, x=A...H, J, K)

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

13.4.9. Alternate function selected register 0 (GPIOx_AFSEL0, x=A...H, J, K)

Address offset: 0x20

Reset value: 0x0000 0000

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

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

Bits	Fields	Descriptions
31:28	SEL7[3:0]	Pin 7 alternate function selected These bits are set and cleared by software. Refer to SEL0[3:0] description
27:24	SEL6[3:0]	Pin 6 alternate function selected These bits are set and cleared by software. Refer to SEL0[3:0] description
23:20	SEL5[3:0]	Pin 5 alternate function selected These bits are set and cleared by software.

		Refer to SEL0[3:0] description
19:16	SEL4[3:0]	Pin 4 alternate function selected These bits are set and cleared by software. Refer to SEL0[3:0] description
15:12	SEL3[3:0]	Pin 3 alternate function selected These bits are set and cleared by software. Refer to SEL0[3:0] description
11:8	SEL2[3:0]	Pin 2 alternate function selected These bits are set and cleared by software. Refer to SEL0[3:0] description
7:4	SEL1[3:0]	Pin 1 alternate function selected These bits are set and cleared by software. Refer to SEL0[3:0] description
3:0	SEL0[3:0]	Pin 0 alternate function selected These bits are set and cleared by software. 0000: AF0 selected (reset value) 0001: AF1 selected 0010: AF2 selected 0011: AF3 selected 0100: AF4 selected 0101: AF5 selected 0110: AF6 selected 0111: AF7 selected 1000: AF8 selected 1001: AF9 selected 1010: AF10 selected 1011: AF11 selected 1100: AF12 selected 1101: AF13 selected 1110: AF14 selected 1111: AF15 selected

13.4.10. Alternate function selected register 1 (GPIOx_AFSEL1, x=A…H, J, K)

Address offset: 0x24

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SEL15[3:0]				SEL14[3:0]				SEL13[3:0]				SEL12[3:0]			

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

Bits	Fields	Descriptions
31:28	SEL15[3:0]	Pin 15 alternate function selected These bits are set and cleared by software. Refer to SEL8[3:0] description
27:24	SEL14[3:0]	Pin 14 alternate function selected These bits are set and cleared by software. Refer to SEL8[3:0] description
23:20	SEL13[3:0]	Pin 13 alternate function selected These bits are set and cleared by software. Refer to SEL8[3:0] description
19:16	SEL12[3:0]	Pin 12 alternate function selected These bits are set and cleared by software. Refer to SEL8[3:0] description
15:12	SEL11[3:0]	Pin 11 alternate function selected These bits are set and cleared by software. Refer to SEL8[3:0] description
11:8	SEL10[3:0]	Pin 10 alternate function selected These bits are set and cleared by software. Refer to SEL8[3:0] description
7:4	SEL9[3:0]	Pin 9 alternate function selected These bits are set and cleared by software. Refer to SEL8[3:0] description
3:0	SEL8[3:0]	Pin 8 alternate function selected These bits are set and cleared by software. 0000: AF0 selected (reset value) 0001: AF1 selected 0010: AF2 selected 0011: AF3 selected 0100: AF4 selected 0101: AF5 selected 0110: AF6 selected 0111: AF7 selected 1000: AF8 selected 1001: AF9 selected 1010: AF10 selected

1011: AF11 selected
1100: AF12 selected
1101: AF13 selected
1110: AF14 selected
1111: AF15 selected

13.4.11. Bit clear register (GPIOx_BC, x=A…H, J, K)

Address offset: 0x28
Reset value: 0x0000 0000

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

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

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

13.4.12. Port bit toggle register (GPIOx_TG, x=A…H, J, K)

Address offset: 0x2C
Reset value: 0x0000 0000

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

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

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

13.4.13. Input filtering register (GPIOx_IFL, x=A...H, J, K)

Address offset: 0x30

Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLPRD1								FLPRD0							
rw								rw							

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:8	FLPRD1	Filter sampling frequency 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 frequency for GPIO0 to GPIO7: 00: FLPRDx = CK_AHB 01: FLPRDx = CK_AHB / 2 02: FLPRDx = CK_AHB / 4 .... FF: FLPRDx = CK_AHB / 510

13.4.14. Input filtering type register (GPIOx_IFTP, x=A...H, J, K)

Address offset: 0x34

Reset value: 0xFFFF FFFF

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

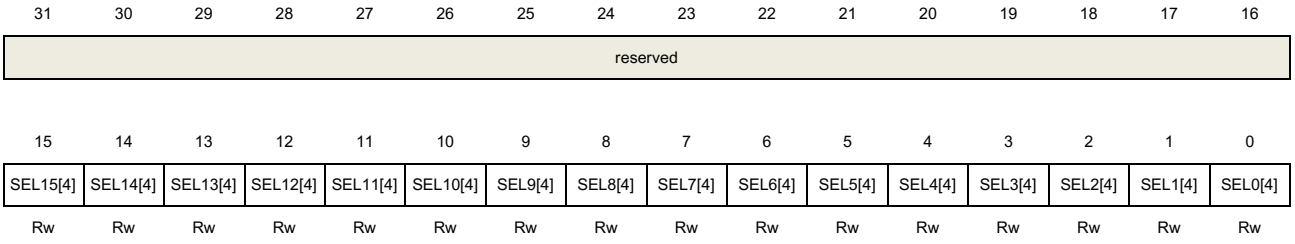
		Refer to IFTP0[1:0] description
9:8	IFTP4[1:0]	Pin 4 input filtering type bits These bits are set and cleared by software. Refer to IFTP0[1:0] description
7:6	IFTP3[1:0]	Pin 3 input filtering type bits These bits are set and cleared by software. Refer to IFTP0[1:0] description
5:4	IFTP2[1:0]	Pin 2 input 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)

13.4.15. Alternate function selected register 2 (GPIOx_AFSEL2, x=B,C,D,E)

Address offset: 0x38

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15	SEL15[4]	Pin15 alternate function selected BIT4 Refer to SEL0[4] description
14	SEL14[4]	Pin14 alternate function selected BIT4 Refer to SEL0[4] description

13	SEL13[4]	Pin13 alternate function selected BIT4 Refer to SEL0[4] description
12	SEL12[4]	Pin12 alternate function selected BIT4 Refer to SEL0[4] description
11	SEL11[4]	Pin11 alternate function selected BIT4 Refer to SEL0[4] description
10	SEL10[4]	Pin10 alternate function selected BIT4 Refer to SEL0[4] description
9	SEL9[4]	Pin9 alternate function selected BIT4 Refer to SEL0[4] description
8	SEL8[4]	Pin8 alternate function selected BIT4 Refer to SEL0[4] description
7	SEL7[4]	Pin7 alternate function selected BIT4 Refer to SEL0[4] description
6	SEL6[4]	Pin6 alternate function selected BIT4 Refer to SEL0[4] description
5	SEL5[4]	Pin5 alternate function selected BIT4 Refer to SEL0[4] description
4	SEL4[4]	Pin4 alternate function selected BIT4 Refer to SEL0[4] description
3	SEL3[4]	Pin3 alternate function selected BIT4 Refer to SEL0[4] description
2	SEL2[4]	Pin2 alternate function selected BIT4 Refer to SEL0[4] description
1	SEL1[4]	Pin1 alternate function selected BIT4 Refer to SEL0[4] description
0	SEL0[4]	Pin0 alternate function selected BIT4 This bit is set and cleared by software and only be accessed by GPIOB , C, D, E. Bit 4 of SEL0. 00000: AF0 selected (reset value) 00001: AF1 selected … 01111: AF15 selected 10000: AF16 selected 10001: AF17selected … 11111: AF31 selected

14. Trigger selection controller (TRIGSEL)

14.1. Overview

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

With TRIGSEL, there are up to 164 trigger input signals could be selected. Configure the corresponding register to select the different trigger signal for the specified trigger input of each peripheral.

14.2. Characteristics

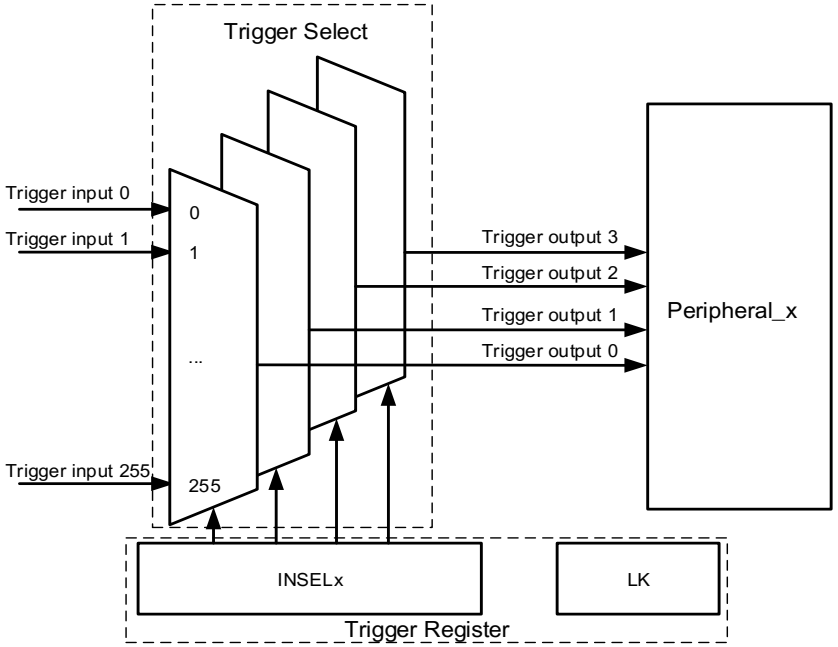
- Support up to 164 trigger input signals.
- Each peripheral has its corresponding register to select trigger input signal.
- Trigger input source could be external input signal or output of peripheral.
- Trigger selection output could be for external output or peripheral input.

14.3. 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 14-1. TRIGSEL main composition example*** shows the main composition of TRIGSEL.

Figure 14-1. TRIGSEL main composition example



14.4. Internal connect

The TRIGSEL allows software to select the trigger input for peripherals. The *Table 14-1. Trigger input bit fields selection* shows the trigger input register selection.

Table 14-1. Trigger input bit fields selection

fields	bits value	trigger input selection
INSELx	0x00	0
	0x01	1
	0x02	TRIGSEL_IN0
	0x03	TRIGSEL_IN1
	0x04	TRIGSEL_IN2
	0x05	TRIGSEL_IN3
	0x06	TRIGSEL_IN4
	0x07	TRIGSEL_IN5
	0x08	TRIGSEL_IN6
	0x09	TRIGSEL_IN7
	0x0a	TRIGSEL_IN8
	0x0b	TRIGSEL_IN9
	0x0c	TRIGSEL_IN10
	0x0d	TRIGSEL_IN11
	0x0e	TRIGSEL_IN12
	0x0f	TRIGSEL_IN13

fields	bits value	trigger input selection
	0x10	LXTAL_TRG
	0x11	TIMER0_TRGO0
	0x12	TIMER0_TRGO1
	0x13	TIMER0_CH0
	0x14	TIMER0_CH1
	0x15	TIMER0_CH2
	0x16	TIMER0_CH3
	0x17	TIMER0_MCH0
	0x18	TIMER0_MCH1
	0x19	TIMER0_MCH2
	0x1a	TIMER0_MCH3
	0x1b~0x20	Reserved
	0x21	TIMER0_BRKIN0
	0x22	TIMER0_BRKIN1
	0x23	TIMER0_BRKIN2
	0x24	TIMER0_ETI
	0x25	TIMER1_TRGO0
	0x26	TIMER1_CH0
	0x27	TIMER1_CH1
	0x28	TIMER1_CH2
	0x29	TIMER1_CH3
	0x2a	TIMER1_ETI
	0x2b	TIMER2_TRGO0
	0x2c	TIMER2_CH0
	0x2d	TIMER2_CH1
	0x2e	TIMER2_CH2
	0x2f	TIMER2_CH3
	0x30	TIMER2_ETI
	0x31	TIMER3_TRGO0
	0x32	TIMER3_CH0
	0x33	TIMER3_CH1
	0x34	TIMER3_CH2
	0x35	TIMER3_CH3
	0x36	TIMER3_ETI
	0x37	TIMER4_TRGO0
	0x38	TIMER4_CH0
	0x39	TIMER4_CH1
	0x3a	TIMER4_CH2
	0x3b	TIMER4_CH3
	0x3c	TIMER4_ETI
	0x3d	TIMER5_TRGO0

fields	bits value	trigger input selection
	0x3e	TIMER6_TRGO0
	0x3f	TIMER7_TRGO0
	0x40	TIMER7_TRGO1
	0x41	TIMER7_CH0
	0x42	TIMER7_CH1
	0x43	TIMER7_CH2
	0x44	TIMER7_CH3
	0x45	TIMER7_MCH0
	0x46	TIMER7_MCH1
	0x47	TIMER7_MCH2
	0x48	TIMER7_MCH3
	0x49~0x4e	Reserved
	0x4f	TIMER7_BRKIN0
	0x50	TIMER7_BRKIN1
	0x51	TIMER7_BRKIN2
	0x52	TIMER7_ETI
	0x53	TIMER14_TRGO0
	0x54	TIMER14_CH0
	0x55	TIMER14_CH1
	0x56	TIMER14_MCH0
	0x57~0x58	Reserved
	0x59	TIMER14_BRKIN0
	0x5a	TIMER15_CH0
	0x5b	TIMER15_MCH0
	0x5c~0x5d	Reserved
	0x5e	TIMER15_BRKIN0
	0x5f	TIMER16_CH0
	0x60	TIMER16_MCH0
	0x61~0x62	Reserved
	0x63	TIMER16_BRKIN0
	0x64	TIMER22_TRGO0
	0x65	TIMER22_CH0
	0x66	TIMER22_CH1
	0x67	TIMER22_CH2
	0x68	TIMER22_CH3
	0x69	TIMER22_ETI
	0x6a	TIMER23_TRGO0
	0x6b	TIMER23_CH0
	0x6c	TIMER23_CH1
	0x6d	TIMER23_CH2
	0x6e	TIMER23_CH3

fields	bits value	trigger input selection
	0x6f	TIMER23_ETI
	0x70	TIMER30_TRGO0
	0x71	TIMER30_CH0
	0x72	TIMER30_CH1
	0x73	TIMER30_CH2
	0x74	TIMER30_CH3
	0x75	TIMER30_ETI
	0x76	TIMER31_TRGO0
	0x77	TIMER31_CH0
	0x78	TIMER31_CH1
	0x79	TIMER31_CH2
	0x7a	TIMER31_CH3
	0x7b	TIMER31_ETI
	0x7c	TIMER40_TRGO0
	0x7d	TIMER40_CH0
	0x7e	TIMER40_CH1
	0x7f	TIMER40_MCH0
	0x80~0x81	Reserved
	0x82	TIMER40_BRKIN0
	0x83	TIMER41_TRGO0
	0x84	TIMER41_CH0
	0x85	TIMER41_CH1
	0x86	TIMER41_MCH0
	0x87~0x88	Reserved
	0x89	TIMER41_BRKIN0
	0x8a	TIMER42_TRGO0
	0x8b	TIMER42_CH0
	0x8c	TIMER42_CH1
	0x8d	TIMER42_MCH0
	0x8e~0x8f	Reserved
	0x90	TIMER42_BRKIN0
	0x91	TIMER43_TRGO0
	0x92	TIMER43_CH0
	0x93	TIMER43_CH1
	0x94	TIMER43_MCH0
	0x95~0x96	Reserved
	0x97	TIMER43_BRKIN0
	0x98	TIMER44_TRGO0
	0x99	TIMER44_CH0
	0x9a	TIMER44_CH1
	0x9b	TIMER44_MCH0

fields	bits value	trigger input selection
	0x9c~0x9d	Reserved
	0x9e	TIMER44_BRKIN0
	0x9f	TIMER50_TRGO0
	0xa0	TIMER51_TRGO0
	0xa1	RTC_Alarm
	0xa2	RTC_TPTS
	0xa3	ADC0_WD0_OUT
	0xa4	ADC0_WD1_OUT
	0xa5	ADC0_WD2_OUT
	0xa6	ADC1_WD0_OUT
	0xa7	ADC1_WD1_OUT
	0xa8	ADC1_WD2_OUT
	0xa9	ADC2_WD0_OUT
	0xaa	ADC2_WD1_OUT
	0xab	ADC2_WD2_OUT
	0xac	CMP0_OUT
	0xad	CMP1_OUT
	0xae	SAI0_FS0
	0xaf	SAI0_FS1
	0xb0	SAI2_FS0
	0xb1	SAI2_FS1
	0xb2	TIMER0_CH0BRKIN
	0xb3	TIMER0_CH1BRKIN
	0xb4	TIMER0_CH2BRKIN
	0xb5	TIMER0_CH3BRKIN
	0xb6	TIMER7_CH0BRKIN
	0xb7	TIMER7_CH1BRKIN
	0xb8	TIMER7_CH2BRKIN
	0xb9	TIMER7_CH3BRKIN
	0xba	ESC_SYNC0
	0xbb	ESC_SYNC1
	0xbc	Reserved
	0xbd	Reserved
	0xbe	RDCM_I_ENV_TH
	0xbf	RDCM_Q_ENV_TH
	0xc0	RDCM_MON_TH
	0xc1	RDCM_TRA_ERR_TH
	0xc2~0xff	Reserved

Table 14-2. TRIGSEL input and output mapping shows the connection relationship between TRIGSEL input and output. Through the INSELx[7:0] bits of TRIGSEL register, an

input trigger source can be selected for the output of TRIGSEL. Each TRIGSEL register can configure with up to 3 outputs, which are connected to the corresponding peripherals.

Table 14-2. TRIGSEL input and output mapping

Trigger Source	Trigger select	TRIGSEL Register	TRIGSEL output	Peripherals
0 1 TRIGSEL_IN0 TRIGSEL_IN1 TRIGSEL_IN2 TRIGSEL_IN3 TRIGSEL_IN4 TRIGSEL_IN5 TRIGSEL_IN6 TRIGSEL_IN7 TRIGSEL_IN8 TRIGSEL_IN9 TRIGSEL_IN10 TRIGSEL_IN11 TRIGSEL_IN12 TRIGSEL_IN13 LXTAL_TRG TIMER0_TRGO0 TIMER0_TRGO1 TIMER0_CH0 TIMER0_CH1 TIMER0_CH2 TIMER0_CH3 TIMER0_MCH0 TIMER0_MCH1 TIMER0_MCH2 TIMER0_MCH3 TIMER0_BRKIN0 TIMER0_BRKIN1 TIMER0_BRKIN2 TIMER0_ETI TIMER1_TRGO0 TIMER1_CH0 TIMER1_CH1 TIMER1_CH2 TIMER1_CH3	INSELx[7:0]	TRIGSEL_EXTOUT_0	Output0 Output1	TRIGSEL_OUT0 TRIGSEL_OUT1
		TRIGSEL_EXTOUT_1	Output0 Output1	TRIGSEL_OUT2 TRIGSEL_OUT3
		TRIGSEL_EXTOUT_2	Output0 Output1	TRIGSEL_OUT4 TRIGSEL_OUT5
		TRIGSEL_EXTOUT_3	Output0 Output1	TRIGSEL_OUT6 TRIGSEL_OUT7
		TRIGSEL_ADC0	Output0 Output1	ADC0_ROUTRG ADC0_INSTRG
		TRIGSEL_ADC1	Output0 Output1	ADC1_ROUTRG ADC1_INSTRG
		TRIGSEL_ADC2	Output0 Output1	ADC2_ROUTRG ADC2_INSTRG
		TRIGSEL_DAC0OUT0	Output0	DAC0_OUT0_EXTRG
		TRIGSEL_DAC0OUT1	Output0	DAC0_OUT1_EXTRG

Trigger Source	Trigger select	TRIGSEL Register	TRIGSEL output	Peripherals
TIMER1_ETI TIMER2_TRGO0 TIMER2_CH0 TIMER2_CH1 TIMER2_CH2 TIMER2_CH3 TIMER2_ETI TIMER3_TRGO0 TIMER3_CH0 TIMER3_CH1 TIMER3_CH2 TIMER3_CH3 TIMER3_ETI TIMER4_TRGO0 TIMER4_CH0 TIMER4_CH1 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_BRKIN0 TIMER7_BRKIN1 TIMER7_BRKIN2 TIMER7_ETI TIMER14_TRGO0 TIMER14_CH0 TIMER14_CH1 TIMER14_MCH0 TIMER14_BRKIN0		TRIGSEL_TIMER0BRKIN	Output0 Output1 Output2	TIMER0_BRKIN0 TIMER0_BRKIN1 TIMER0_BRKIN2
		TRIGSEL_TIMER7BRKIN	Output0 Output1 Output2	TIMER7_BRKIN0 TIMER7_BRKIN1 TIMER7_BRKIN2
		TRIGSEL_TIMER14BRKIN	Output0	TIMER14_BRKIN0
		TRIGSEL_TIMER15BRKIN	Output0	TIMER15_BRKIN0
		TRIGSEL_TIMER16BRKIN	Output0	TIMER16_BRKIN0
		TRIGSEL_TIMER40BRKIN	Output0	TIMER40_BRKIN0
		TRIGSEL_TIMER41BRKIN	Output0	TIMER41_BRKIN0
		TRIGSEL_TIMER42BRKIN	Output0	TIMER42_BRKIN0
		TRIGSEL_TIMER43BRKIN	Output0	TIMER43_BRKIN0
		TRIGSEL_TIMER44BRKIN	Output0	TIMER44_BRKIN0

Trigger Source	Trigger select	TRIGSEL Register	TRIGSEL output	Peripherals
TIMER15_CH0 TIMER15_MCH0 TIMER15_BRKIN0 TIMER16_CH0 TIMER16_MCH0 TIMER16_BRKIN0 TIMER22_TRGO0 TIMER22_CH0 TIMER22_CH1 TIMER22_CH2 TIMER22_CH3 TIMER22_ETI TIMER23_TRGO0 TIMER23_CH0 TIMER23_CH1 TIMER23_CH2 TIMER23_CH3 TIMER23_ETI TIMER30_TRGO0 TIMER30_CH0 TIMER30_CH1 TIMER30_CH2 TIMER30_CH3 TIMER30_ETI TIMER31_TRGO0 TIMER31_CH0 TIMER31_CH1 TIMER31_CH2 TIMER31_CH3 TIMER31_ETI TIMER40_TRGO0 TIMER40_CH0 TIMER40_CH1 TIMER40_MCH0 TIMER40_BRKIN0 TIMER41_TRGO0 TIMER41_CH0 TIMER41_CH1 TIMER41_MCH0 TIMER41_BRKIN0		TRIGSEL_CAN0	Output0	CAN0_EX_TIME_TICK
		TRIGSEL_CAN1	Output0	CAN1_EX_TIME_TICK
		TRIGSEL_CAN2	Output0	CAN2_EX_TIME_TICK
		TRIGSEL_LPDTS	Output0	LPDTS_TRG
		TRIGSEL_EDOUT	Output0	EDOUT_TRG
		TRIGSEL_HPDF	Output0	HPDF_ITRG
		TRIGSEL_TIMER0ETI	Output0	TIMER0_ETI
		TRIGSEL_TIMER1ETI	Output0	TIMER1_ETI
		TRIGSEL_TIMER2ETI	Output0	TIMER2_ETI
		TRIGSEL_TIMER3ETI	Output0	TIMER3_ETI

Trigger Source	Trigger select	TRIGSEL Register	TRIGSEL output	Peripherals
TIMER42_TRGO0 TIMER42_CH0 TIMER42_CH1 TIMER42_MCH0 TIMER42_BRKIN0 TIMER43_TRGO0 TIMER43_CH0 TIMER43_CH1 TIMER43_MCH0 TIMER43_BRKIN0 TIMER44_TRGO0 TIMER44_CH0 TIMER44_CH1 TIMER44_MCH0 TIMER44_BRKIN0 TIMER50_TRGO0 TIMER51_TRGO0 RTC_Alarm RTC_TPTS ADC0_WD0_OUT ADC0_WD1_OUT ADC0_WD2_OUT ADC1_WD0_OUT ADC1_WD1_OUT ADC1_WD2_OUT ADC2_WD0_OUT ADC2_WD1_OUT ADC2_WD2_OUT CMP0_OUT CMP1_OUT SAI0_FS0 SAI0_FS1 SAI2_FS0 SAI2_FS1 TIMER0_CH0BRKIN TIMER0_CH1BRKIN TIMER0_CH2BRKIN TIMER0_CH3BRKIN TIMER7_CH0BRKIN TIMER7_CH1BRKIN TIMER7_CH2BRKIN		TRIGSEL_TIMER4ETI	Output0	TIMER4_ETI
		TRIGSEL_TIMER7ETI	Output0	TIMER7_ETI
		TRIGSEL_TIMER22ETI	Output0	TIMER22_ETI
		TRIGSEL_TIMER23ETI	Output0	TIMER23_ETI
		TRIGSEL_TIMER30ETI	Output0	TIMER30_ETI
		TRIGSEL_TIMER31ETI	Output0	TIMER31_ETI
		TRIGSEL_TIMER0ITI14	Output0	TIMER0_ITI14
		TRIGSEL_TIMER1ITI14	Output0	TIMER1_ITI14
		TRIGSEL_TIMER2ITI14	Output0	TIMER2_ITI14
		TRIGSEL_TIMER3ITI14	Output0	TIMER3_ITI14

Trigger Source	Trigger select	TRIGSEL Register	TRIGSEL output	Peripherals
TIMER7_CH3BRKIN ESC_SYNC0 ESC_SYNC1 RDCM_I_ENV_TH RDCM_Q_ENV_TH RDCM_MON_TH RDCM_TRA_ERR_TH		TRIGSEL_TIMER4ITI14	Output0	TIMER4_ITI14
		TRIGSEL_TIMER7ITI14	Output0	TIMER7_ITI14
		TRIGSEL_TIMER14ITI14	Output0	TIMER14_ITI14
		TRIGSEL_TIMER22ITI14	Output0	TIMER22_ITI14
		TRIGSEL_TIMER23ITI14	Output0	TIMER23_ITI14
		TRIGSEL_TIMER30ITI14	Output0	TIMER30_ITI14
		TRIGSEL_TIMER31ITI14	Output0	TIMER31_ITI14
		TRIGSEL_TIMER40ITI14	Output0	TIMER40_ITI14
		TRIGSEL_TIMER41ITI14	Output0	TIMER41_ITI14
		TRIGSEL_TIMER42ITI14	Output0	TIMER42_ITI14

Trigger Source	Trigger select	TRIGSEL Register	TRIGSEL output	Peripherals
		TRIGSEL_TIMER43ITI14	Output0	TIMER43_ITI14
		TRIGSEL_TIMER44ITI14	Output0	TIMER44_ITI14
		TRIGSEL_RDCM	Output0 Output1 Output2	RDCM_NCO RDCM_TS RDCM_MAINCAL
		TRIGSEL_TIMER0CHBRKIN_0	Output0 Output1	TIMER0_CH0BRKIN TIMER0_CH1BRKIN
		TRIGSEL_TIMER0CHBRKIN_1	Output0 Output1	TIMER0_CH2BRKIN TIMER0_CH3BRKIN
		TRIGSEL_TIMER7CHBRKIN_0	Output0 Output1	TIMER7_CH0BRKIN TIMER7_CH1BRKIN
		TRIGSEL_TIMER7CHBRKIN_1	Output0 Output1	TIMER7_CH2BRKIN TIMER7_CH3BRKIN
		TRIGSEL_ENDAT	Output0	ENDAT
		TRIGSEL_AFMT	Output0	AFMT
		TRIGSEL_TFMT	Output0	TFMT

Trigger Source	Trigger select	TRIGSEL Register	TRIGSEL output	Peripherals
		TRIGSEL_BISS	Output0	BISS
		TRIGSEL_HDSL	Output0	HDSL

Note :

- TIMERx_BRKINx can only select BRKINx or CHyBRKIN of any TIMER as the trigger source, which means there are only 22 trigger sources available.
- TIMERx_CHyBRKIN can only select BRKINx or CHyBRKIN of any TIMER as the trigger source, which means there are only 22 trigger sources available.
- TIMERx_ITI14 cannot select CMPx_OUT ,LXTAL_TRG, TIMERx_BRKINx, TIMERx_CHyBRKIN and their own signals as trigger input.
- GPIO cannot select ESC_SYNC0 and ESC_SYNC1 as trigger input.
- EDOUT cannot select ESC_SYNC0, ESC_SYNC1, TIMERx_BRKINx, TIMERx_CHyBRKIN and their own signals as trigger input.
- Only RDCM, EDIM (ENDAT, AFMT, TFMT, BISS, HDSL), HPDF, ADC, DAC, CAN, TIMERx_ITI14 can select RDCM signals (RDCM_I_ENV_TH, RDCM_Q_ENV_TH, RDCM_MON_TH, RDCM_TRA_ERR_TH) as trigger input.
- Other output can select all input as trigger source except TIMERx_BRKINx and TIMERx_CHyBRKIN. When violate value is selected for these outputs, the output will be selected as 0.
- When illegal data is selected for these outputs, the output will be selected as 0.
- 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.

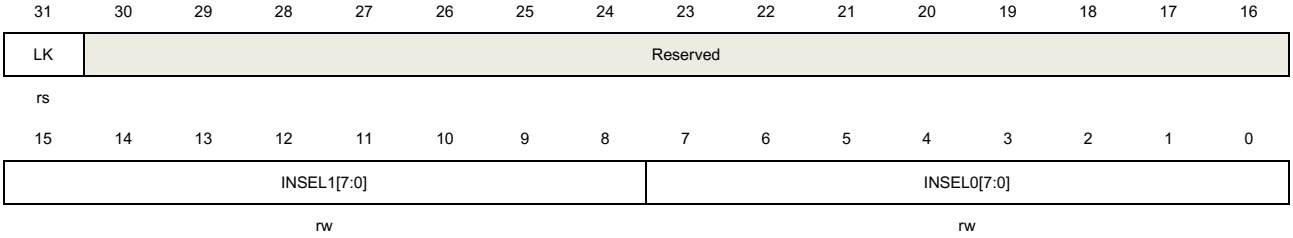
14.5. Register definition

TRIGSEL base address: 0x4001 8400

14.5.1. Trigger selection for EXTOUT register 0 (TRIGSEL_EXTOUT_0)

Address offset: 0x00
Reset value: 0x0000 0000

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

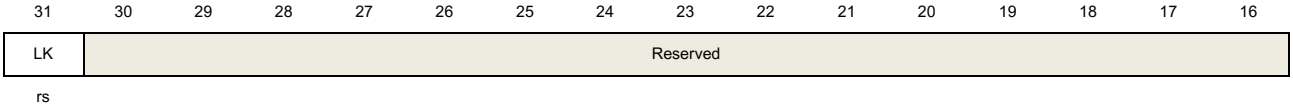


Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_EXTOUT_0 register. 0: TRIGSEL_EXTOUT_0 register write is enabled. 1: TRIGSEL_EXTOUT_0 register write is disabled.
30:16	Reserved	Must be kept at reset value.
15:8	INSEL1[7:0]	Trigger input source selection for output1 These bits are used to select trigger input signal connected to output1. The output can be as the source of TRIGSEL_OUT1 (external output1 signal). For the detailed configuration, please refer to ***Table 14-1. 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 14-1. Trigger input bit fields selection***.

14.5.2. Trigger selection for EXTOUT register 1 (TRIGSEL_EXTOUT_1)

Address offset: 0x04
Reset value: 0x0000 0000

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



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

Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_EXTOUT_1 register. 0: TRIGSEL_EXTOUT_1 register write is enabled. 1: TRIGSEL_EXTOUT_1 register write is disabled.
30:16	Reserved	Must be kept at reset value.
15:8	INSEL1[7:0]	Trigger input source selection for output1 These bits are used to select trigger input signal connected to output1. The output can be as the source of TRIGSEL_OUT3 (external output3 signal). For the detailed configuration, please refer to ***Table 14-1. 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 14-1. Trigger input bit fields selection***.

14.5.3. 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 can be as the source of TRIGSEL_OUT5 (external output5 signal). For the detailed configuration, please refer to ***Table 14-1. 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 14-1. Trigger input bit fields selection***.

14.5.4. Trigger selection for EXTOUT register 3 (TRIGSEL_EXTOUT_3)

Address offset: 0x0C

Reset value: 0x0000 0000

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

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

Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_EXTOUT_3 register. 0: TRIGSEL_EXTOUT_3 register write is enabled. 1: TRIGSEL_EXTOUT_3 register write is disabled.
30:16	Reserved	Must be kept at reset value.
15:8	INSEL1[7:0]	Trigger input source selection for output1 These bits are used to select trigger input signal connected to output1. The output can be as the source of TRIGSEL_OUT7 (external output7 signal). For the detailed configuration, please refer to ***Table 14-1. 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 14-1. Trigger input bit fields selection***.

14.5.5. Trigger selection for ADC0 register (TRIGSEL_ADC0)

Address offset: 0x10

Reset value: 0x0000 1113

This register 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_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	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 ADC0_INSTRG(ADC0 inserted sequence) trigger input. For the detailed configuration, please refer to ***Table 14-1. 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 ADC0_ROUTRG(ADC0 routine sequence) trigger input. For the detailed configuration, please refer to ***Table 14-1. Trigger input bit fields selection***.

14.5.6. Trigger selection for ADC1 register (TRIGSEL_ADC1)

Address offset: 0x14

Reset value: 0x0000 1113

This register 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

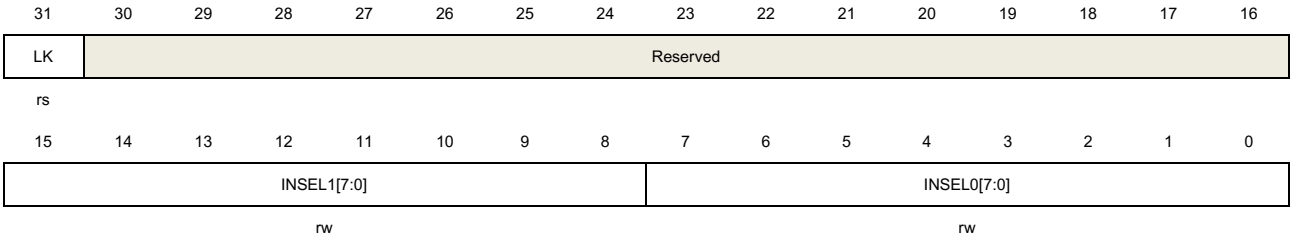
		disables write access to TRIGSEL_ADC1 register. 0: TRIGSEL_ADC1 register write is enabled. 1: TRIGSEL_ADC1 register write is disabled.
30:16	Reserved	Must be kept at reset value.
15:8	INSEL1[7:0]	Trigger input source selection for output1 These bits are used to select trigger input signal connected to output1. The output is used as the source of ADC1_INSTRG(ADC1 inserted sequence) trigger input. For the detailed configuration, please refer to ***Table 14-1. 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 ADC1_ROUTRG(ADC1 routine sequence) trigger input. For the detailed configuration, please refer to ***Table 14-1. Trigger input bit fields selection***.

14.5.7. Trigger selection for ADC2 register (TRIGSEL_ADC2)

Address offset: 0x18

Reset value: 0x0000 1113

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



Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_ADC2 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	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 ADC2_INSTRG(ADC2 inserted sequence) trigger input. For the detailed configuration, please refer to ***Table 14-1. 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 ADC2_ROUTRG(ADC2 routine sequence) trigger input. For the detailed configuration, please refer to ***Table 14-1. Trigger input bit fields selection***.

14.5.8. Trigger selection for DAC0_OUT0 register (TRIGSEL_DAC0OUT0)

Address offset: 0x1C

Reset value: 0x0000 0025

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

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

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

Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_DAC0OUT0 register. 0: TRIGSEL_DAC0OUT0 register write is enabled. 1: TRIGSEL_DAC0OUT0 register write is disabled.
30:8	Reserved	Must be kept at reset value.
7:0	INSEL0[7:0]	Trigger input source selection for output0 These bits are used to select trigger input signal connected to output0. The output is used as the source of DAC0_OUT0_EXTRIG (DAC0_OUT0 external trigger) input. For the detailed configuration, please refer to ***Table 14-1. Trigger input bit fields selection***.

14.5.9. Trigger selection for DAC0_OUT1 register (TRIGSEL_DAC0OUT1)

Address offset: 0x20

Reset value: 0x0000 0025

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

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

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

Reserved	INSEL0[7:0]
	rw

Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_DAC0OUT1 register. 0: TRIGSEL_DAC0OUT1 register write is enabled. 1: TRIGSEL_DAC0OUT1 register write is disabled.
30:8	Reserved	Must be kept at reset value.
7:0	INSEL0[7:0]	Trigger input source selection for output0 These bits are used to select trigger input signal connected to output0. The output is used as the source of DAC0_OUT1_EXTRIG (DAC0_OUT1 external trigger) input. For the detailed configuration, please refer to ***Table 14-1. Trigger input bit fields selection***.

14.5.10. Trigger selection for TIMER0_BRKIN register (TRIGSEL_TIMER0BRKIN)

Address offset: 0x24

Reset value: 0x0023 2221

This register 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 TIMER0_BRKIN2 trigger input. For the detailed configuration, please refer to ***Table 14-1. 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 TIMER0_BRKIN1 trigger input. For the detailed configuration, please refer to ***Table 14-1. 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 TIMER0_BRKIN0 trigger input. For the detailed configuration, please refer to ***Table 14-1. Trigger input bit fields selection***.

14.5.11. Trigger selection for TIMER7_BRKIN register (TRIGSEL_TIMER7BRKIN)

Address offset: 0x28

Reset value: 0x0051 504F

This register 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 TIMER7_BRKIN2 trigger input. For the detailed configuration, please refer to ***Table 14-1. 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 TIMER7_BRKIN1 trigger input. For the detailed configuration, please refer to ***Table 14-1. 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 TIMER7_BRKIN0 trigger input. For the detailed configuration, please refer to ***Table 14-1. Trigger input bit fields selection***.

14.5.12. Trigger selection for TIMER14_BRKIN register (TRIGSEL_TIMER14BRKIN)

Address offset: 0x2C

Reset value: 0x0000 0059

This register 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 TIMER14_BRKIN0 trigger input. For the detailed configuration, please refer to ***Table 14-1. Trigger input bit fields selection***.

14.5.13. Trigger selection for TIMER15_BRKIN register (TRIGSEL_TIMER15BRKIN)

Address offset: 0x30

Reset value: 0x0000 005E

This register 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							

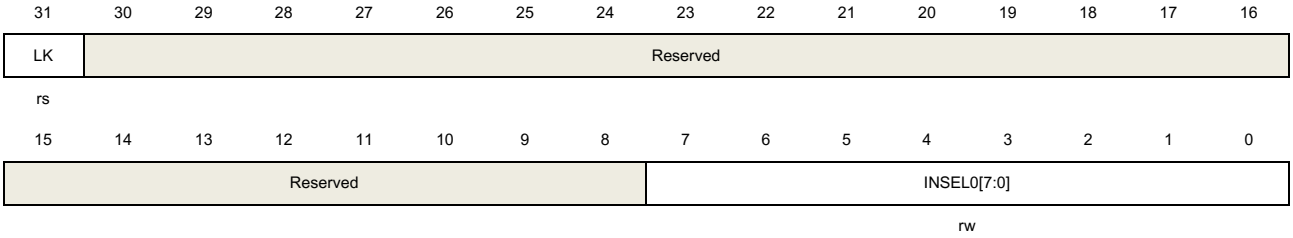
Bits	Fields	Descriptions

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 TIMER15_BRKIN0 trigger input. For the detailed configuration, please refer to *Table 14-1. Trigger input bit fields selection*.

14.5.14. Trigger selection for TIMER16_BRKIN register (TRIGSEL_TIMER16BRKIN)

Address offset: 0x34
Reset value: 0x0000 0063

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



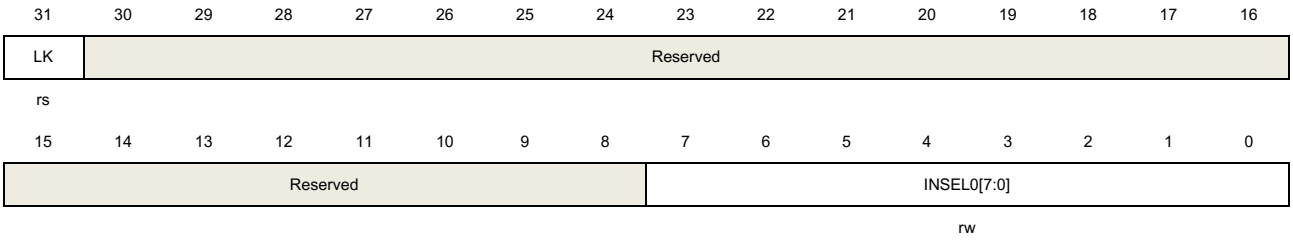
Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER16BRKIN register. 0: TRIGSEL_TIMER16BRKIN register write is enabled. 1: TRIGSEL_TIMER16BRKIN register write is disabled.
30:8	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 TIMER16_BRKIN0 trigger input. For the detailed configuration, please refer to *Table 14-1. Trigger input bit fields selection*.

14.5.15. Trigger selection for TIMER40_BRKIN register (TRIGSEL_TIMER40BRKIN)

Address offset: 0x38

Reset value: 0x0000 0082

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



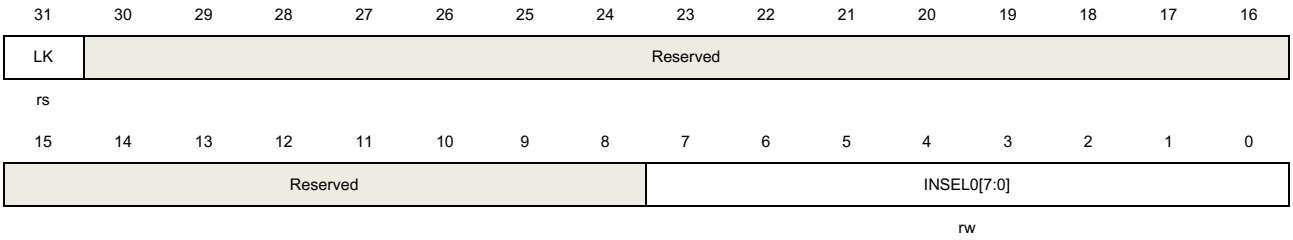
Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER40BRKIN register. 0: TRIGSEL_TIMER40BRKIN register write is enabled. 1: TRIGSEL_TIMER40BRKIN register write is disabled.
30:8	Reserved	Must be kept at reset value.
7:0	INSEL0[7:0]	Trigger input source selection for output0 These bits are used to select trigger input signal connected to output0. The output is used as the source of TIMER40_BRKIN0 trigger input. For the detailed configuration, please refer to ***Table 14-1. Trigger input bit fields selection***.

14.5.16. Trigger selection for TIMER41_BRKIN register (TRIGSEL_TIMER41BRKIN)

Address offset: 0x3C

Reset value: 0x0000 0089

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



Bits	Fields	Descriptions

31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER41BRKIN register. 0: TRIGSEL_TIMER41BRKIN register write is enabled. 1: TRIGSEL_TIMER41BRKIN register write is disabled.
30:8	Reserved	Must be kept at reset value.
7:0	INSEL0[7:0]	Trigger input source selection for output0 These bits are used to select trigger input signal connected to output0. The output is used as the source of TIMER41_BRKIN0 trigger input. For the detailed configuration, please refer to ***Table 14-1. Trigger input bit fields selection***.

14.5.17. Trigger selection for TIMER42_BRKIN register (TRIGSEL_TIMER42BRKIN)

Address offset: 0x40
Reset value: 0x0000 0090

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

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

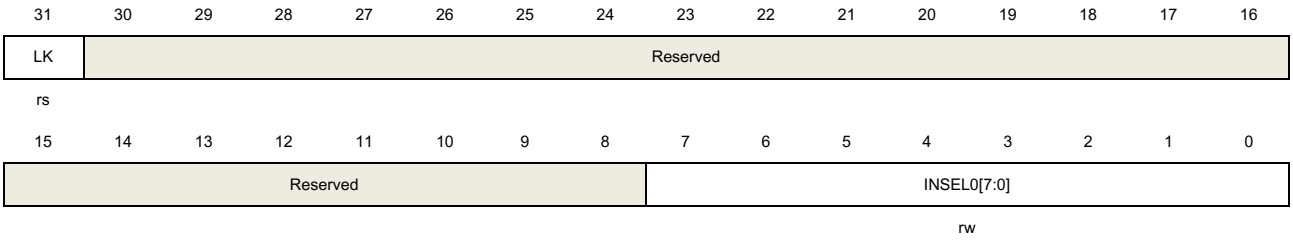
Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER42BRKIN register. 0: TRIGSEL_TIMER42BRKIN register write is enabled. 1: TRIGSEL_TIMER42BRKIN register write is disabled.
30:8	Reserved	Must be kept at reset value.
7:0	INSEL0[7:0]	Trigger input source selection for output0 These bits are used to select trigger input signal connected to output0. The output is used as the source of TIMER42_BRKIN0 trigger input. For the detailed configuration, please refer to ***Table 14-1. Trigger input bit fields selection***.

14.5.18. Trigger selection for TIMER43_BRKIN register (TRIGSEL_TIMER43BRKIN)

Address offset: 0x44

Reset value: 0x0000 0097

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



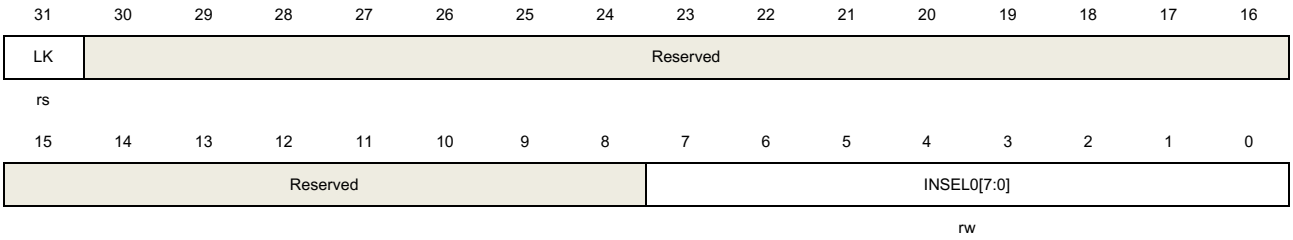
Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER43BRKIN register. 0: TRIGSEL_TIMER43BRKIN register write is enabled. 1: TRIGSEL_TIMER43BRKIN register write is disabled.
30:8	Reserved	Must be kept at reset value.
7:0	INSEL0[7:0]	Trigger input source selection for output0 These bits are used to select trigger input signal connected to output0. The output is used as the source of TIMER43_BRKIN0 trigger input. For the detailed configuration, please refer to ***Table 14-1. Trigger input bit fields selection***.

14.5.19. Trigger selection for TIMER44_BRKIN register (TRIGSEL_TIMER44BRKIN)

Address offset: 0x48

Reset value: 0x0000 009e

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



Bits	Fields	Descriptions

31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER44BRKIN register. 0: TRIGSEL_TIMER44BRKIN register write is enabled. 1: TRIGSEL_TIMER44BRKIN register write is disabled.
30:8	Reserved	Must be kept at reset value.
7:0	INSEL0[7:0]	Trigger input source selection for output0 These bits are used to select trigger input signal connected to output0. The output is used as the source of TIMER44_BRKIN0 trigger input. For the detailed configuration, please refer to ***Table 14-1. Trigger input bit fields selection***.

14.5.20. Trigger selection for CAN0 register (TRIGSEL_CAN0)

Address offset: 0x4C
Reset value: 0x0000 003d

This register 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															

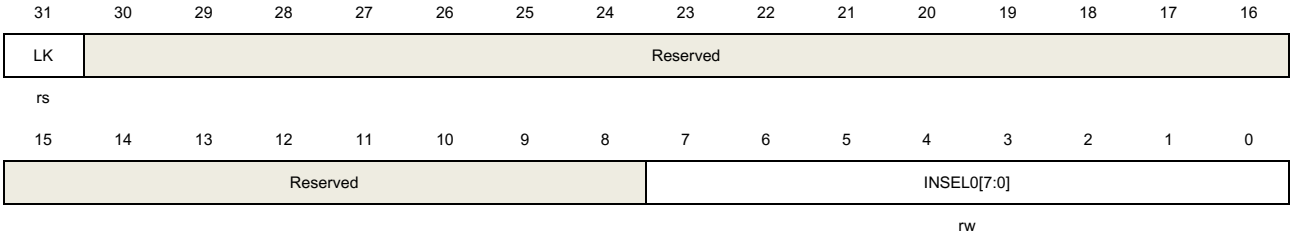
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								INSEL0[7:0]							
								rw							

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 14-1. Trigger input bit fields selection***.

14.5.21. Trigger selection for CAN1 register (TRIGSEL_CAN1)

Address offset: 0x50
Reset value: 0x0000 003d

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



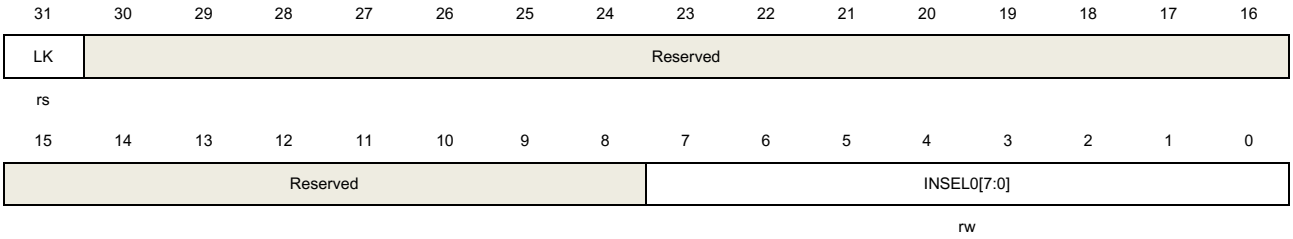
Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_CAN1 register. 0: TRIGSEL_CAN1 register write is enabled. 1: TRIGSEL_CAN1 register write is disabled.
30:8	Reserved	Must be kept at 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 14-1. Trigger input bit fields selection***.

14.5.22. Trigger selection for CAN2 register (TRIGSEL_CAN2)

Address offset: 0x54

Reset value: 0x0000 003d

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



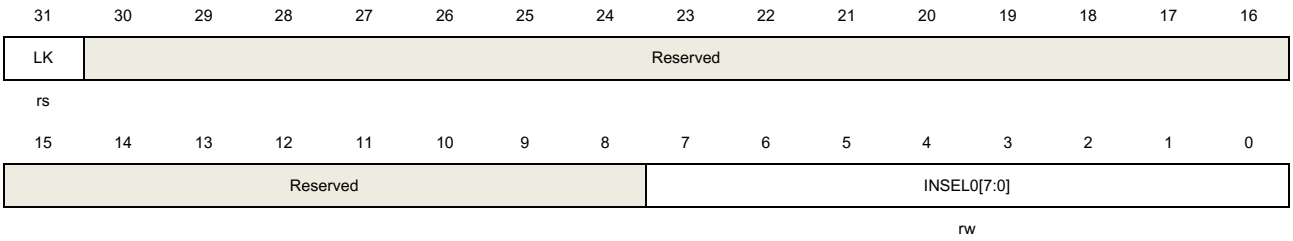
Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_CAN2 register. 0: TRIGSEL_CAN2 register write is enabled. 1: TRIGSEL_CAN2 register write is disabled.
30:8	Reserved	Must be kept at 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 14-1. Trigger input bit fields selection*.

14.5.23. Trigger selection for LPDTS register (TRIGSEL_LPDTS)

Address offset: 0x58
Reset value: 0x0000 0000

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

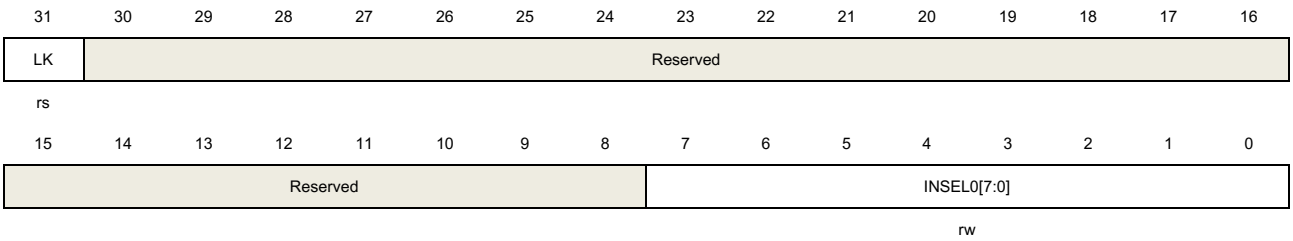


Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_LPDTS register. 0: TRIGSEL_LPDTS register write is enabled. 1: TRIGSEL_LPDTS register write is disabled.
30:8	Reserved	Must be kept 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 LPDTS_TRG trigger input. For the detailed configuration, please refer to *Table 14-1. Trigger input bit fields selection*.

14.5.24. Trigger selection for TIMER0_ETI register (TRIGSEL_TIMER0ETI)

Address offset: 0x5C
Reset value: 0x0000 0024

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



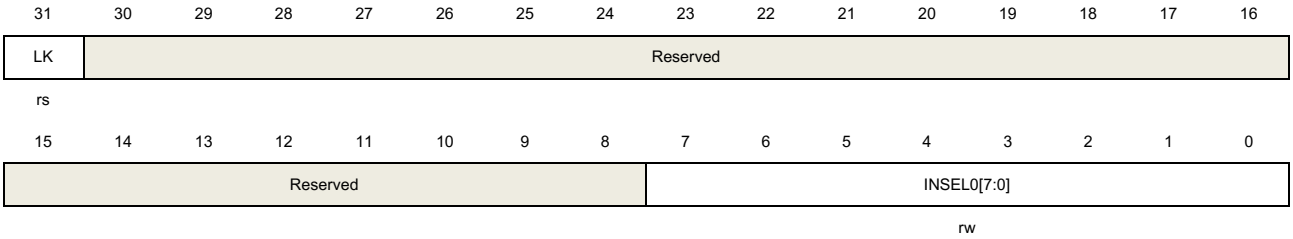
Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER0ETI register. 0: TRIGSEL_TIMER0ETI register write is enabled. 1: TRIGSEL_TIMER0ETI register write is disabled.
30:8	Reserved	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 14-1. Trigger input bit fields selection*.

14.5.25. Trigger selection for TIMER1_ETI register (TRIGSEL_TIMER1ETI)

Address offset: 0x60

Reset value: 0x0000 002a

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



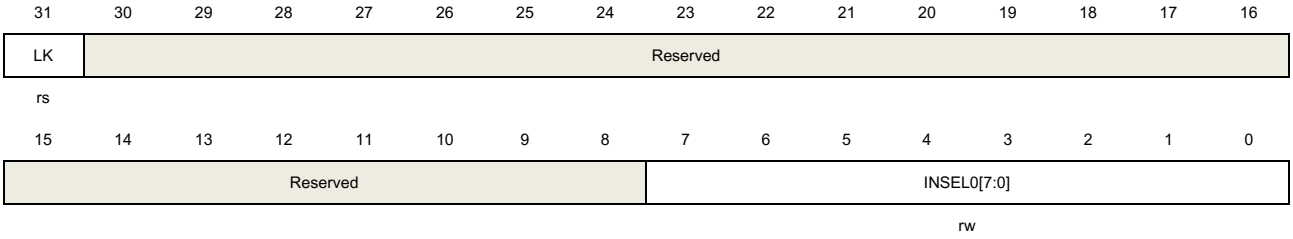
Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER1ETI register. 0: TRIGSEL_TIMER1ETI register write is enabled. 1: TRIGSEL_TIMER1ETI register write is disabled.
30:8	Reserved	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 14-1. Trigger input bit fields selection*.

14.5.26. Trigger selection for TIMER2_ETI register (TRIGSEL_TIMER2ETI)

Address offset: 0x64

Reset value: 0x0000 0030

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



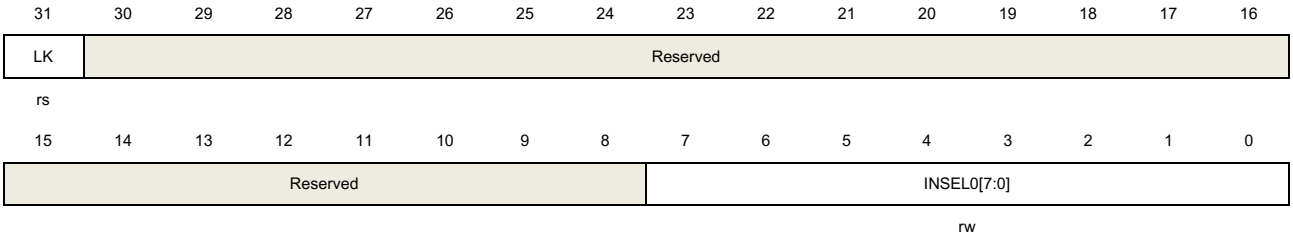
Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER2ETI register. 0: TRIGSEL_TIMER2ETI register write is enabled. 1: TRIGSEL_TIMER2ETI register write is disabled.
30:8	Reserved	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 14-1. Trigger input bit fields selection***.

14.5.27. Trigger selection for TIMER3_ETI register (TRIGSEL_TIMER3ETI)

Address offset: 0x68

Reset value: 0x0000 0036

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_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

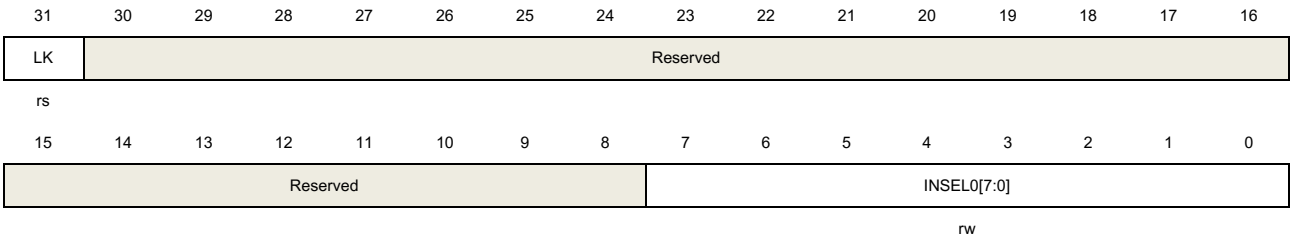
These bits are used to select trigger input signal connected to output0. The output is used as the source of TIMER3_ETI trigger input. For the detailed configuration, please refer to ***Table 14-1. Trigger input bit fields selection***.

14.5.28. Trigger selection for TIMER4_ETI register (TRIGSEL_TIMER4ETI)

Address offset: 0x6C

Reset value: 0x0000 003c

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



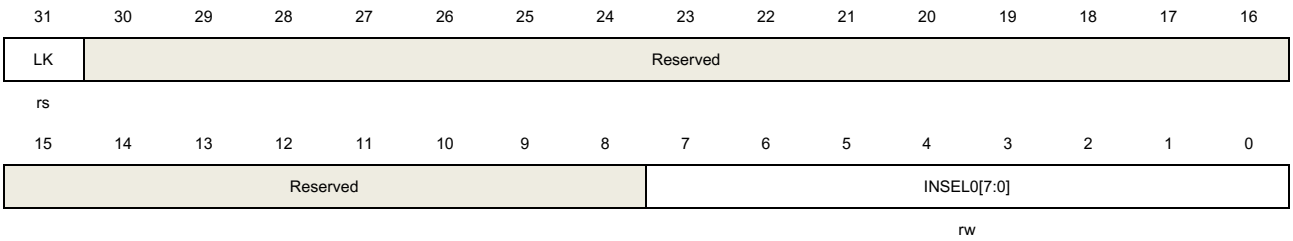
Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER4ETI register. 0: TRIGSEL_TIMER4ETI register write is enabled. 1: TRIGSEL_TIMER4ETI register write is disabled.
30:8	Reserved	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 14-1. Trigger input bit fields selection***.

14.5.29. Trigger selection for TIMER7_ETI register (TRIGSEL_TIMER7ETI)

Address offset: 0x70

Reset value: 0x0000 0052

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



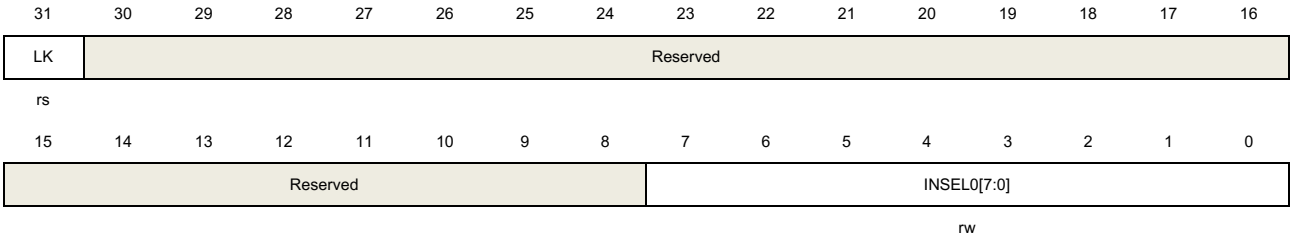
Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER7ETI register. 0: TRIGSEL_TIMER7ETI register write is enabled. 1: TRIGSEL_TIMER7ETI register write is disabled.
30:8	Reserved	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 14-1. Trigger input bit fields selection*.

14.5.30. Trigger selection for TIMER22_ETI register (TRIGSEL_TIMER22ETI)

Address offset: 0x74

Reset value: 0x0000 0069

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



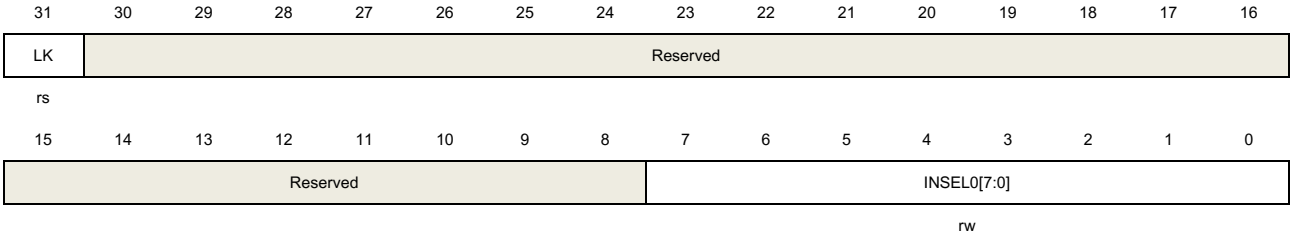
Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER22ETI register. 0: TRIGSEL_TIMER22ETI register write is enabled. 1: TRIGSEL_TIMER22ETI register write is disabled.
30:8	Reserved	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 TIMER22_ETI trigger input. For the detailed configuration, please refer to *Table 14-1. Trigger input bit fields selection*.

14.5.31. Trigger selection for TIMER23_ETI register (TRIGSEL_TIMER23ETI)

Address offset: 0x78

Reset value: 0x0000 006f

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



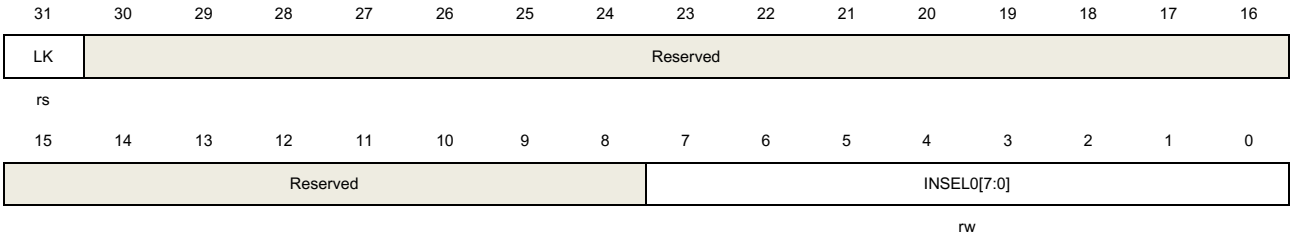
Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER23ETI register. 0: TRIGSEL_TIMER23ETI register write is enabled. 1: TRIGSEL_TIMER23ETI register write is disabled.
30:8	Reserved	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 TIMER23_ETI trigger input. For the detailed configuration, please refer to ***Table 14-1. Trigger input bit fields selection***.

14.5.32. Trigger selection for TIMER30_ETI register (TRIGSEL_TIMER30ETI)

Address offset: 0x7C

Reset value: 0x0000 0075

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



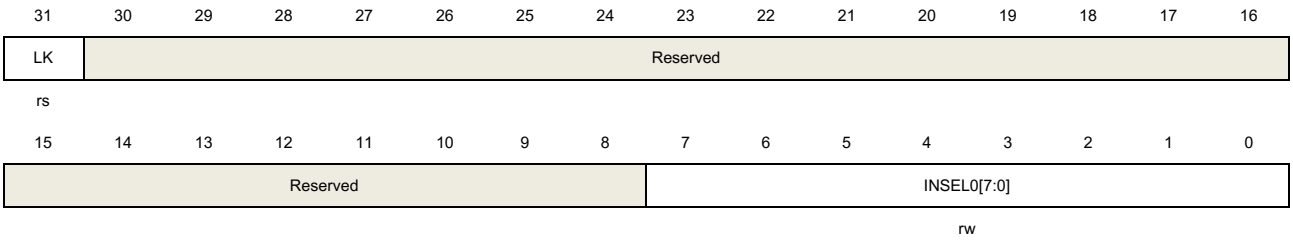
Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER30ETI register. 0: TRIGSEL_TIMER30ETI register write is enabled. 1: TRIGSEL_TIMER30ETI register write is disabled.
30:8	Reserved	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 TIMER30_ETI trigger input. For the detailed configuration, please refer to ***Table 14-1. Trigger input bit fields selection***.

14.5.33. Trigger selection for TIMER31_ETI register (TRIGSEL_TIMER31ETI)

Address offset: 0x80
Reset value: 0x0000 007b

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

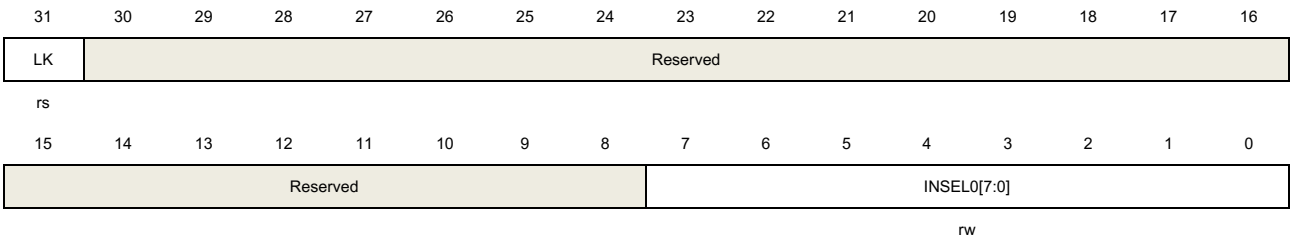


Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER31ETI register. 0: TRIGSEL_TIMER31ETI register write is enabled. 1: TRIGSEL_TIMER31ETI register write is disabled.
30:8	Reserved	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 TIMER31_ETI trigger input. For the detailed configuration, please refer to ***Table 14-1. Trigger input bit fields selection***.

14.5.34. Trigger selection for EDOUT register (TRIGSEL_EDOUT)

Address offset: 0x84
Reset value: 0x0000 0000

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



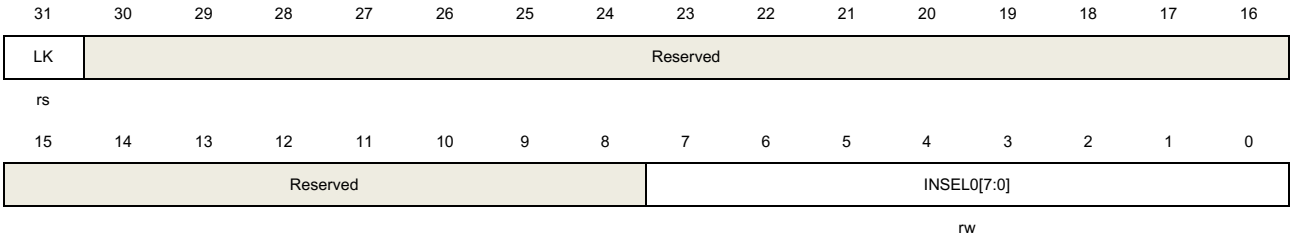
Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_EDOUT register. 0: TRIGSEL_EDOUT register write is enabled. 1: TRIGSEL_EDOUT register write is disabled.
30:8	Reserved	Must be kept 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 EDOUT_TRG trigger input. For the detailed configuration, please refer to *Table 14-1. Trigger input bit fields selection*

14.5.35. Trigger selection for HPDF register (TRIGSEL_HPDF)

Address offset: 0x88

Reset value: 0x0000 0000

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



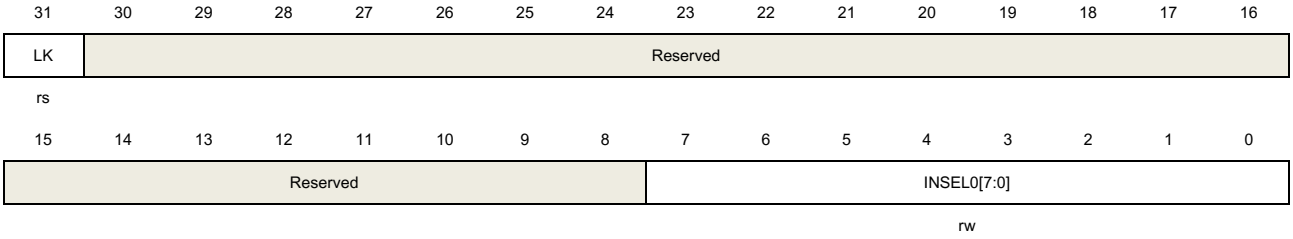
Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_HPDF register. 0: TRIGSEL_HPDF register write is enabled. 1: TRIGSEL_HPDF register write is disabled.
30:8	Reserved	Must be kept at 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 14-1. Trigger input bit fields selection*.

14.5.36. Trigger selection for TIMER0_ITI14 register (TRIGSEL_TIMER0ITI14)

Address offset: 0x8C

Reset value: 0x0000 0000

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



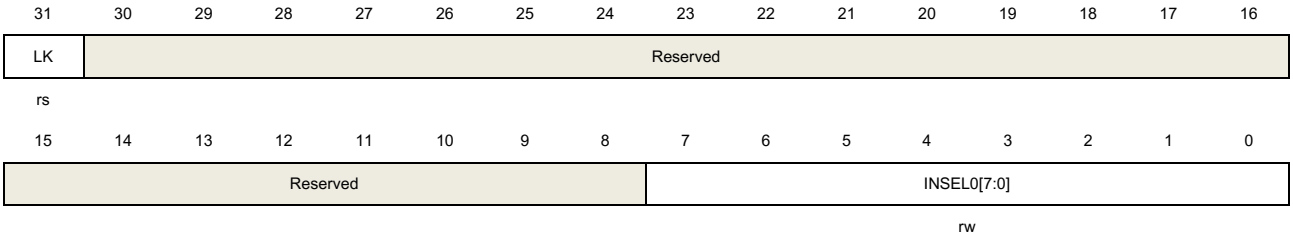
Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER0ITI14 register. 0: TRIGSEL_TIMER0ITI14 register write is enabled. 1: TRIGSEL_TIMER0ITI14 register write is disabled.
30:8	Reserved	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 14-1. Trigger input bit fields selection***.

14.5.37. Trigger selection for TIMER1_ITI14 register (TRIGSEL_TIMER1ITI14)

Address offset: 0x90

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_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 14-1. Trigger input bit fields selection*.

14.5.38. Trigger selection for TIMER2_ITI14 register (TRIGSEL_TIMER2ITI14)

Address offset: 0x94

Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
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 14-1. Trigger input bit fields selection*.

14.5.39. Trigger selection for TIMER3_ITI14 register (TRIGSEL_TIMER3ITI14)

Address offset: 0x98

Reset value: 0x0000 0000

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

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

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

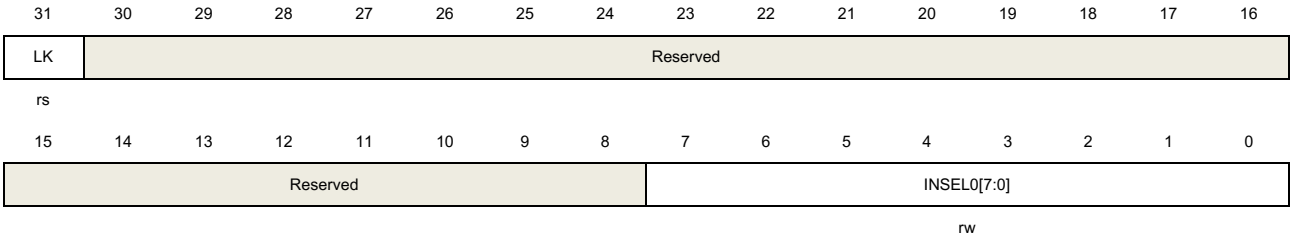
Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER3ITI14 register. 0: TRIGSEL_TIMER3ITI14 register write is enabled. 1: TRIGSEL_TIMER3ITI14 register write is disabled.
30:8	Reserved	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 14-1. Trigger input bit fields selection*.

14.5.40. Trigger selection for TIMER4_ITI14 register (TRIGSEL_TIMER4ITI14)

Address offset: 0x9C

Reset value: 0x0000 0000

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



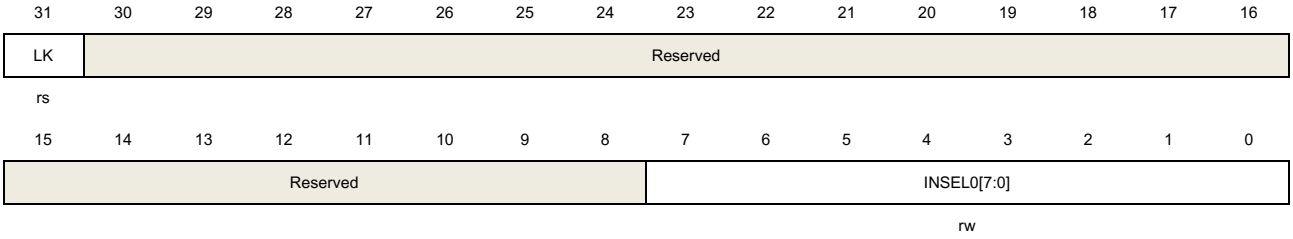
Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER4ITI14 register. 0: TRIGSEL_TIMER4ITI14 register write is enabled. 1: TRIGSEL_TIMER4ITI14 register write is disabled.
30:8	Reserved	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 14-1. Trigger input bit fields selection*.

14.5.41. Trigger selection for TIMER7_ITI14 register (TRIGSEL_TIMER7ITI14)

Address offset: 0xA0

Reset value: 0x0000 0000

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



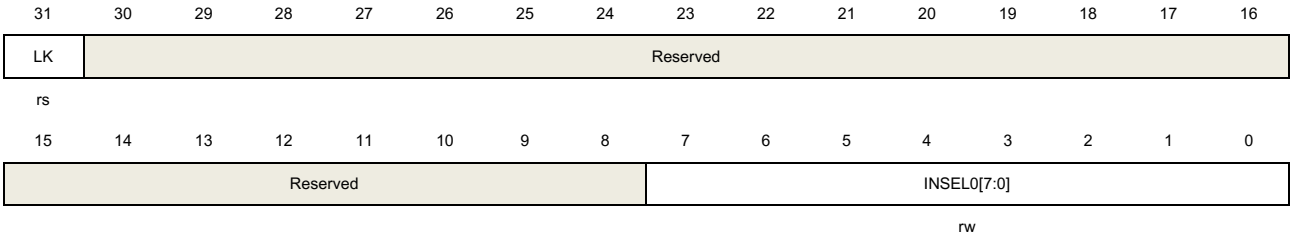
Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER7ITI14 register. 0: TRIGSEL_TIMER7ITI14 register write is enabled. 1: TRIGSEL_TIMER7ITI14 register write is disabled.
30:8	Reserved	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 14-1. Trigger input bit fields selection***.

14.5.42. Trigger selection for TIMER14_ITI14 register (TRIGSEL_TIMER14ITI14)

Address offset: 0xA4

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_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 14-1. Trigger input bit fields selection*.

14.5.43. Trigger selection for TIMER22_ITI14 register (TRIGSEL_TIMER22ITI14)

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
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_TIMER22ITI14 register. 0: TRIGSEL_TIMER22ITI14 register write is enabled. 1: TRIGSEL_TIMER22ITI14 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 TIMER22_ITI14 trigger input. For the detailed configuration, please refer to *Table 14-1. Trigger input bit fields selection*.

14.5.44. Trigger selection for TIMER23_ITI14 register (TRIGSEL_TIMER23ITI14)

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
LK	Reserved														
rs															

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

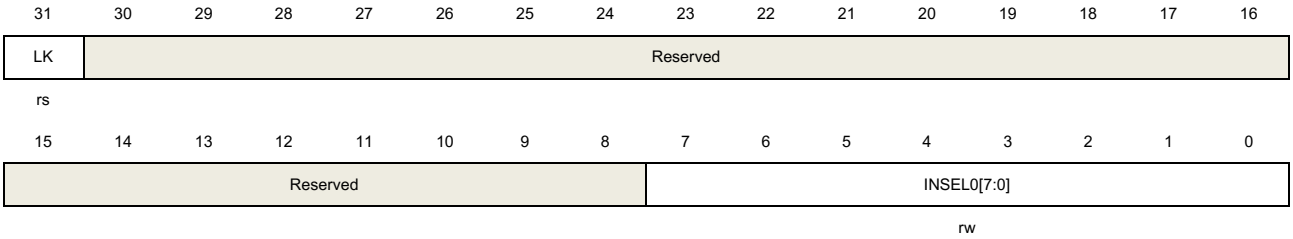
Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER23ITI14 register. 0: TRIGSEL_TIMER23ITI14 register write is enabled. 1: TRIGSEL_TIMER23ITI14 register write is disabled.
30:8	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 TIMER23_ITI14 trigger input. For the detailed configuration, please refer to *Table 14-1. Trigger input bit fields selection*.

14.5.45. Trigger selection for TIMER30_ITI14 register (TRIGSEL_TIMER30ITI14)

Address offset: 0xB0

Reset value: 0x0000 0000

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



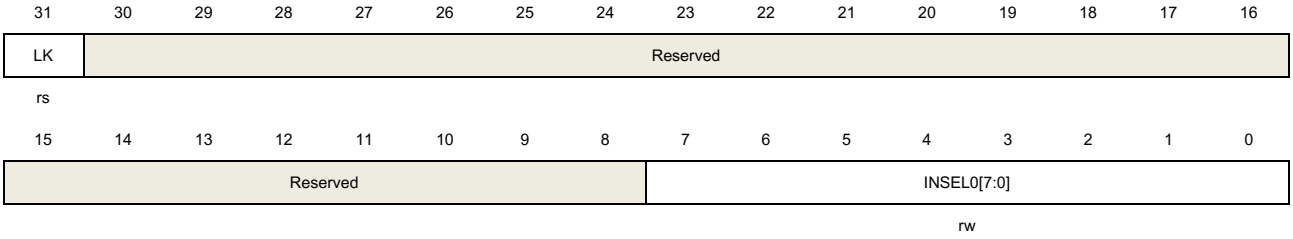
Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER30ITI14 register. 0: TRIGSEL_TIMER30ITI14 register write is enabled. 1: TRIGSEL_TIMER30ITI14 register write is disabled.
30:8	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 TIMER30_ITI14 trigger input. For the detailed configuration, please refer to *Table 14-1. Trigger input bit fields selection*.

14.5.46. Trigger selection for TIMER31_ITI14 register (TRIGSEL_TIMER31ITI14)

Address offset: 0xB4

Reset value: 0x0000 0000

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



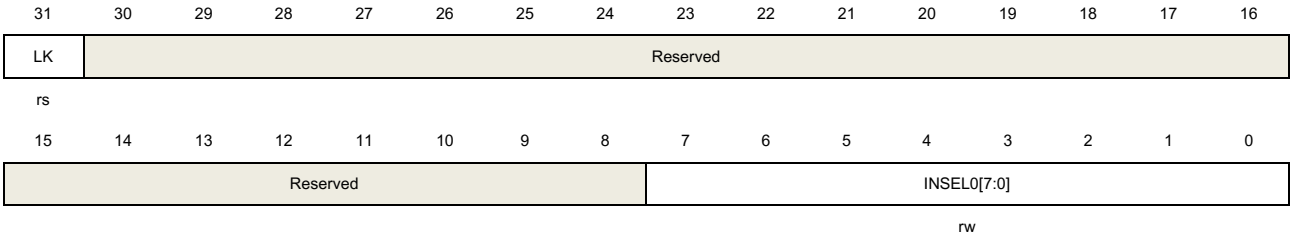
Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER31ITI14 register. 0: TRIGSEL_TIMER31ITI14 register write is enabled. 1: TRIGSEL_TIMER31ITI14 register write is disabled.
30:8	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 TIMER31_ITI14 trigger input. For the detailed configuration, please refer to ***Table 14-1. Trigger input bit fields selection***.

14.5.47. Trigger selection for TIMER40_ITI14 register (TRIGSEL_TIMER40ITI14)

Address offset: 0xB8

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_TIMER40ITI14 register. 0: TRIGSEL_TIMER40ITI14 register write is enabled. 1: TRIGSEL_TIMER40ITI14 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 TIMER40_ITI14 trigger input. For the detailed configuration, please refer to *Table 14-1. Trigger input bit fields selection*.

14.5.48. Trigger selection for TIMER41_ITI14 register (TRIGSEL_TIMER41ITI14)

Address offset: 0xBC
Reset value: 0x0000 0000

This register 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_TIMER41ITI14 register. 0: TRIGSEL_TIMER41ITI14 register write is enabled. 1: TRIGSEL_TIMER41ITI14 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 TIMER41_ITI14 trigger input. For the detailed configuration, please refer to *Table 14-1. Trigger input bit fields selection*.

14.5.49. Trigger selection for TIMER42_ITI14 register (TRIGSEL_TIMER42ITI14)

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
LK	Reserved														
rs															

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

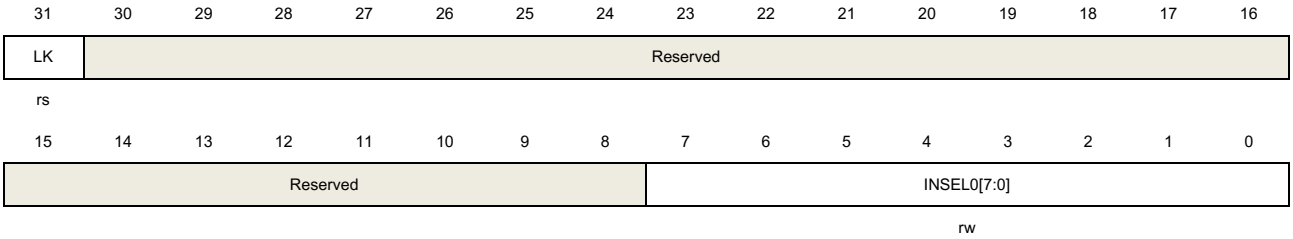
Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER42ITI14 register. 0: TRIGSEL_TIMER42ITI14 register write is enabled. 1: TRIGSEL_TIMER42ITI14 register write is disabled.
30:8	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 TIMER42_ITI14 trigger input. For the detailed configuration, please refer to *Table 14-1. Trigger input bit fields selection*.

14.5.50. Trigger selection for TIMER43_ITI14 register (TRIGSEL_TIMER43ITI14)

Address offset: 0xC4

Reset value: 0x0000 0000

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



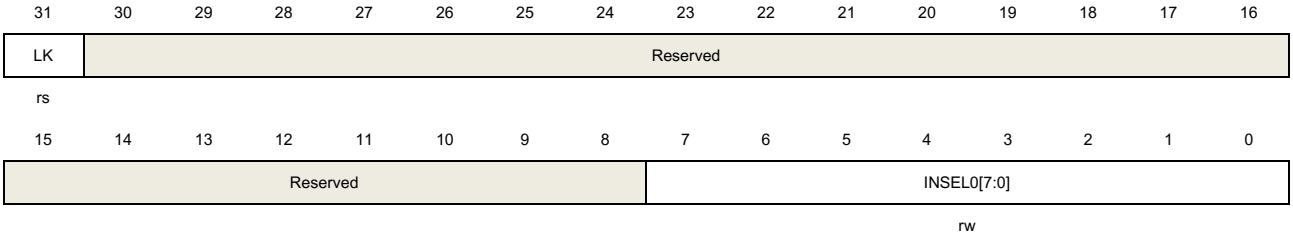
Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER43ITI14 register. 0: TRIGSEL_TIMER43ITI14 register write is enabled. 1: TRIGSEL_TIMER43ITI14 register write is disabled.
30:8	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 TIMER43_ITI14 trigger input. For the detailed configuration, please refer to *Table 14-1. Trigger input bit fields selection*.

14.5.51. Trigger selection for TIMER44_ITI14 register (TRIGSEL_TIMER44ITI14)

Address offset: 0xC8

Reset value: 0x0000 0000

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



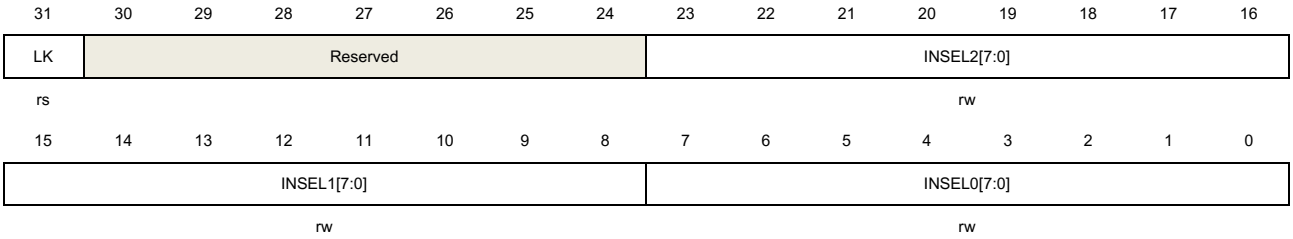
Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER44ITI14 register. 0: TRIGSEL_TIMER44ITI14 register write is enabled. 1: TRIGSEL_TIMER44ITI14 register write is disabled.
30:8	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 TIMER44_ITI14 trigger input. For the detailed configuration, please refer to ***Table 14-1. Trigger input bit fields selection***.

14.5.52. Trigger selection for RDCM register (TRIGSEL_RDCM)

Address offset: 0xD4

Reset value: 0x0000 0000

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



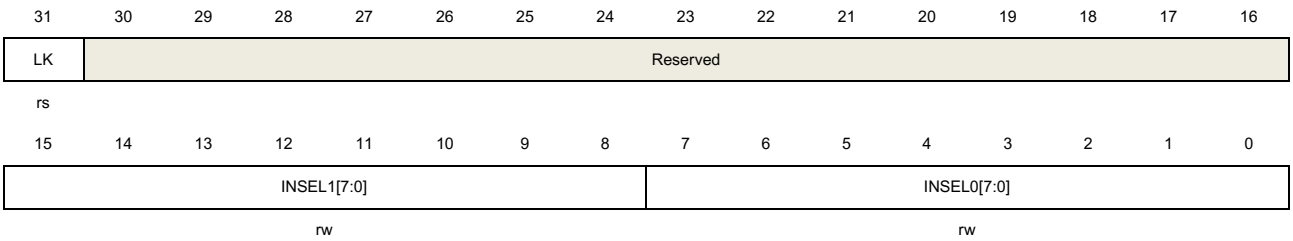
Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_RDCM register. 0: TRIGSEL_RDCM register write is enabled. 1: TRIGSEL_RDCM register write is disabled.
30:24	Reserved	Must be kept at 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 RDCM_MAINCAL (RDCM main chain calculation) trigger input. For the detailed configuration, please refer to ***Table 14-1. 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 RDCM_TS (RDCM timestamp) trigger input. For the detailed configuration, please refer to ***Table 14-1. 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 RDCM_NCO (RDCM NCO calculation) trigger input. For the detailed configuration, please refer to ***Table 14-1. Trigger input bit fields selection***.

14.5.53. Trigger selection for TIMER0_CHBRKIN register 0 (TRIGSEL_TIMER0CHBRKIN_0)

Address offset: 0xD8
Reset value: 0x0000 B3B2

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



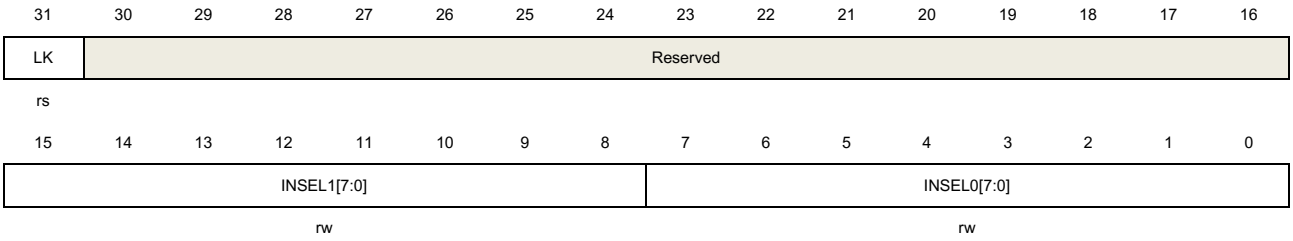
Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER0CHBRKIN_0 register. 0: TRIGSEL_TIMER0CHBRKIN_0 register write is enabled. 1: TRIGSEL_TIMER0CHBRKIN_0 register write is disabled.
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 is used as the source of TIMER0 CH1BRKIN trigger input. For the detailed configuration, please refer to ***Table 14-1. 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 TIMER0 CH0BRKIN trigger input. For the detailed configuration, please refer to *Table 14-1. Trigger input bit fields selection*.

14.5.54. Trigger selection for TIMER0_CHBRKIN register 1 (TRIGSEL_TIMER0CHBRKIN_1)

Address offset: 0xDC
Reset value: 0x0000 B5B4

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_TIMER0CHBRKIN_1 register. 0: TRIGSEL_TIMER0CHBRKIN_1 register write is enabled. 1: TRIGSEL_TIMER0CHBRKIN_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 is used as the source of TIMER0 CH3BRKIN trigger input. For the detailed configuration, please refer to *Table 14-1. 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 TIMER0 CH2BRKIN trigger input. For the detailed configuration, please refer to *Table 14-1. Trigger input bit fields selection*.

14.5.55. Trigger selection for TIMER7_CHBRKIN register 0 (TRIGSEL_TIMER7CHBRKIN_0)

Address offset: 0xE0
Reset value: 0x0000 B7B6

This register 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_TIMER7CHBRKIN_0 register. 0: TRIGSEL_TIMER7CHBRKIN_0 register write is enabled. 1: TRIGSEL_TIMER7CHBRKIN_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 is used as the source of TIMER7 CH1BRKIN trigger input. For the detailed configuration, please refer to ***Table 14-1. 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 TIMER7 CH0BRKIN trigger input. For the detailed configuration, please refer to ***Table 14-1. Trigger input bit fields selection***.

14.5.56. Trigger selection for TIMER7_CHBRKIN register 1 (TRIGSEL_TIMER7CHBRKIN_1)

Address offset: 0xE4

Reset value: 0x0000 B9B8

This register 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							

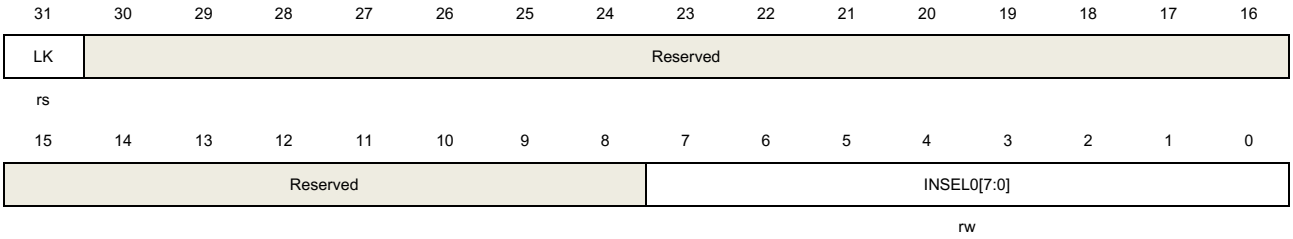
Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it

		disables write access to TRIGSEL_TIMER7CHBRKIN_1 register. 0: TRIGSEL_TIMER7CHBRKIN_1 register write is enabled. 1: TRIGSEL_TIMER7CHBRKIN_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 is used as the source of TIMER7 CH3BRKIN trigger input. For the detailed configuration, please refer to ***Table 14-1. 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 TIMER7 CH2BRKIN trigger input. For the detailed configuration, please refer to ***Table 14-1. Trigger input bit fields selection***.

14.5.57. Trigger selection for ENDAT register (TRIGSEL_ENDAT)

Address offset: 0xE8
Reset value: 0x0000 0000

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



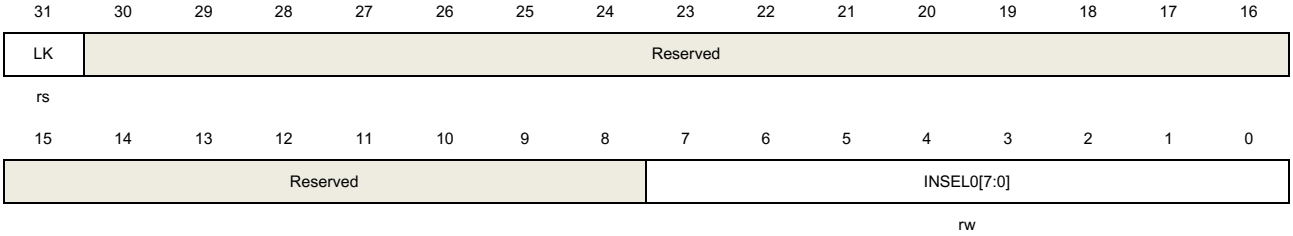
Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_ENDAT register. 0: TRIGSEL_ENDAT register write is enabled. 1: TRIGSEL_ENDAT register write is disabled.
30:8	Reserved	Must be kept 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 ENDAT trigger input. For the detailed configuration, please refer to ***Table 14-1. Trigger input bit fields selection***.

14.5.58. Trigger selection for AFMT register (TRIGSEL_AFMT)

Address offset: 0xEC

Reset value: 0x0000 0000

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



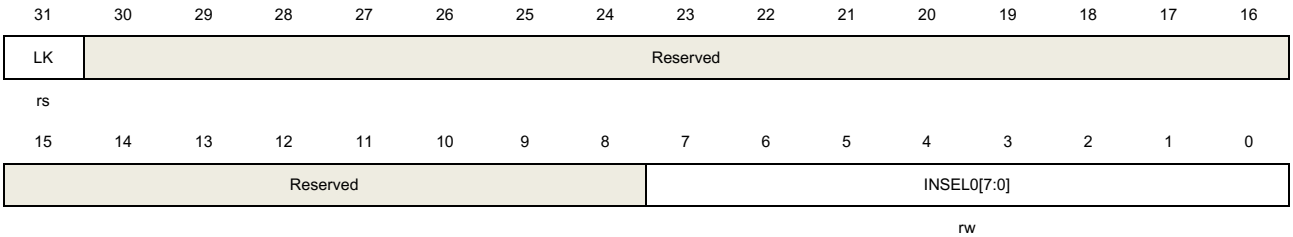
Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_AFMT register. 0: TRIGSEL_AFMT register write is enabled. 1: TRIGSEL_AFMT register write is disabled.
30:8	Reserved	Must be kept at 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 AFMT trigger input. For the detailed configuration, please refer to ***Table 14-1. Trigger input bit fields selection***.

14.5.59. Trigger selection for TFMT register (TRIGSEL_TFMT)

Address offset: 0xF0

Reset value: 0x0000 0000

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



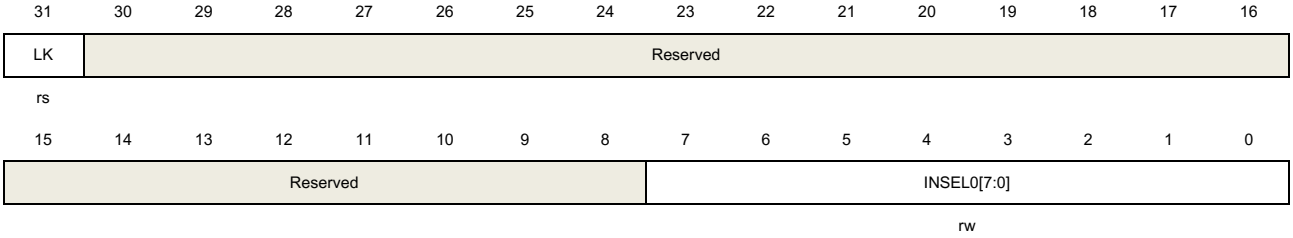
Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TFMT register. 0: TRIGSEL_TFMT register write is enabled.

		1: TRIGSEL_TFMT 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 TFMT trigger input. For the detailed configuration, please refer to ***Table 14-1. Trigger input bit fields selection***.

14.5.60. Trigger selection for BISS register (TRIGSEL_BISS)

Address offset: 0xF4
Reset value: 0x0000 0000

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

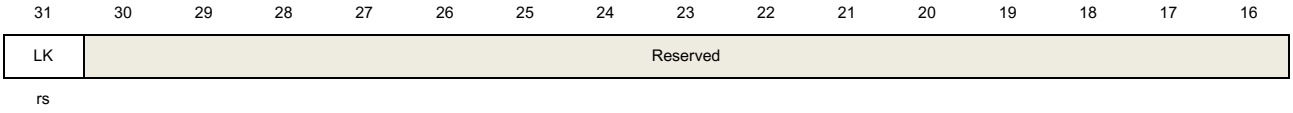


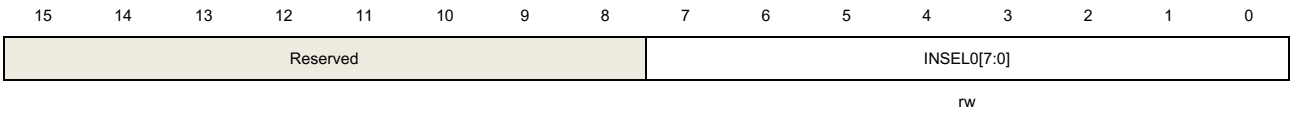
Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_BISS register. 0: TRIGSEL_BISS register write is enabled. 1: TRIGSEL_BISS register write is disabled.
30:8	Reserved	Must be kept at 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 BISS trigger input. For the detailed configuration, please refer to ***Table 14-1. Trigger input bit fields selection***.

14.5.61. Trigger selection for HDSL register (TRIGSEL_HDSL)

Address offset: 0xF8
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_HDSL register. 0: TRIGSEL_HDSL register write is enabled. 1: TRIGSEL_HDSL 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 HDSL trigger input. For the detailed configuration, please refer to ***Table 14-1. Trigger input bit fields selection***.

15. TIMER (TIMERx)

Table 15-1. Timers (TIMERx) are divided into five sorts

TIMER		TIMER0/7	TIMER1/2/3/4/22/23/30/31	TIMER14/40/41/42/43/44	TIMER15/16	TIMER5/6/50/51
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(TIMER2/3/30/31) 32-bit(TIMER1/4/22/23)	16-bit	16-bit	32-bit (TIMER5/6) 64-bit(TIMER50/51)
Count mode		UP, DOWN, Center-aligned	UP, DOWN, Center-aligned	UP, DOWN, Center-aligned	UP ONLY	UP ONLY
Repetition		●	×	●	●	×
Channel Capture/ Compare		8	4	3	2	0
Composite PWM mode		●	●	●	×	×
Asymmetric PWM mode		●	×	×	×	×
Output match pulse select		●	●	●	×	×
Complementary & Dead-time		●	×	●	●	×
Break function	BREAK0	●	×	●	●	×
	BREAK1	●	×	×	×	×
	Channel BREAK	●	×	×	×	×
Locked break function		●	×	●	●	×
Single Pulse		●	●	●	●	●
Delayable single pulse		●	●	●	×	×
Quadrature decoder		●	●	×	×	×
Non-quadrature decoder		●	●	×	×	×
Master-slave management		●	●	●	×	×
Inter		● (1)	● (2)	● (3)	×	TRGO TO DAC

TIMER	TIMER0/7	TIMER1/2/3/4/22/23/30/31	TIMER14/40/41/42/43/44	TIMER15/16	TIMER5/6/50/51
Connection					
DMA	●	●	●	●	● (4)
Debug Mode	●	●	●	●	●

	TIMERx	ITI0	ITI1	ITI2	ITI3	ITI4	ITI5	ITI6	ITI7	ITI8	ITI9	ITI10	ITI11	ITI12	ITI13	ITI14
(1)	TIMER0	TIMER14_TRGO0	TIMER1_TRGO0	TIMER2_TRGO0	TIMER3_TRGO0	-	-	-	-	-	-	-	-	TIMER22_TRGO0	TIMER23_TRGO0	
	TIMER7	TIMER0_TRGO0	TIMER1_TRGO0	TIMER3_TRGO0	TIMER4_TRGO0	-	-	-	-	-	-	-	-	TIMER22_TRGO0	TIMER23_TRGO0	
(2)	TIMER1	TIMER0_TRGO0	TIMER7_TRGO0	TIMER2_TRGO0	TIMER3_TRGO0	ENET0/1_PPS	USBHS0_SOF	-	-	-	-	-	-	TIMER22_TRGO0	TIMER23_TRGO0	
	TIMER2	TIMER0_TRGO0	TIMER1_TRGO0	TIMER14_TRGO0	TIMER3_TRGO0	ENET0/1_PPS	-	-	-	-	-	-	-	TIMER22_TRGO0	TIMER23_TRGO0	
	TIMER3	TIMER0_TRGO0	TIMER1_TRGO0	TIMER2_TRGO0	TIMER7_TRGO0	-	-	-	-	-	-	-	-	TIMER22_TRGO0	TIMER23_TRGO0	
	TIMER4	TIMER0_TRGO0	TIMER7_TRGO0	TIMER2_TRGO0	TIMER3_TRGO0	-	-		USBHS1_SOF	-	-	-	-	TIMER22_TRGO0	TIMER23_TRGO0	
	TIMER22	TIMER0_TRGO0	TIMER1_TRGO0	TIMER2_TRGO0	TIMER3_TRGO0	TIMER4_TRGO0	TIMER7_TRGO0	-	-	-	TIMER14_TRGO0	TIMER15_CH0	TIMER16_CH0	-	TIMER23_TRGO0	
	TIMER23	TIMER0_TRGO0	TIMER1_TRGO0	TIMER2_TRGO0	TIMER3_TRGO0	TIMER4_TRGO0	TIMER7_TRGO0	-	-	-	TIMER14_TRGO0	TIMER15_CH0	TIMER16_CH0	TIMER22_TRGO0	-	from TRIGSEL
	TIMER30	TIMER0_TRGO0	TIMER1_TRGO0	TIMER14_TRGO0	TIMER3_TRGO0	ENET0/1_PPS	USBHS1_SOF	-	-	-	-	-	-	TIMER22_TRGO0	TIMER23_TRGO0	
	TIMER31	TIMER0_TRGO0	TIMER1_TRGO0	TIMER14_TRGO0	TIMER3_TRGO0	ENET0/1_PPS	-	-	-	-	-	-	-	TIMER22_TRGO0	TIMER23_TRGO0	
(3)	TIMER14	TIMER0_TRGO0	TIMER2_TRGO0	TIMER15_CH0	TIMER16_CH0	-	-	-	-	-	-	-	-	-	-	
	TIMER40	TIMER0_TRGO0	TIMER2_TRGO0	TIMER15_CH0	TIMER16_CH0	-	-	-	-	-	-	-	-	-	-	
	TIMER41	TIMER0_TRGO0	TIMER2_TRGO0	TIMER15_CH0	TIMER16_CH0	-	-	-	-	-	-	-	-	-	-	
	TIMER42	TIMER0_TRGO0	TIMER2_TRGO0	TIMER15_CH0	TIMER16_CH0	-	-	-	-	-	-	-	-	-	-	
	TIMER43	TIMER0_TRGO0	TIMER2_TRGO0	TIMER15_CH0	TIMER16_CH0	-	-	-	-	-	-	-	-	-	-	
	TIMER44	TIMER0_TRGO0	TIMER2_TRGO0	TIMER15_CH0	TIMER16_CH0	-	-	-	-	-	-	-	-	-	-	
(4)	Only update events will generate a DMA request. TIMER5/6/50/51 do not have DMAS bit (DMA request source selection).															

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

15.1.1. Overview

The advanced timer module (TIMER0/7) 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.

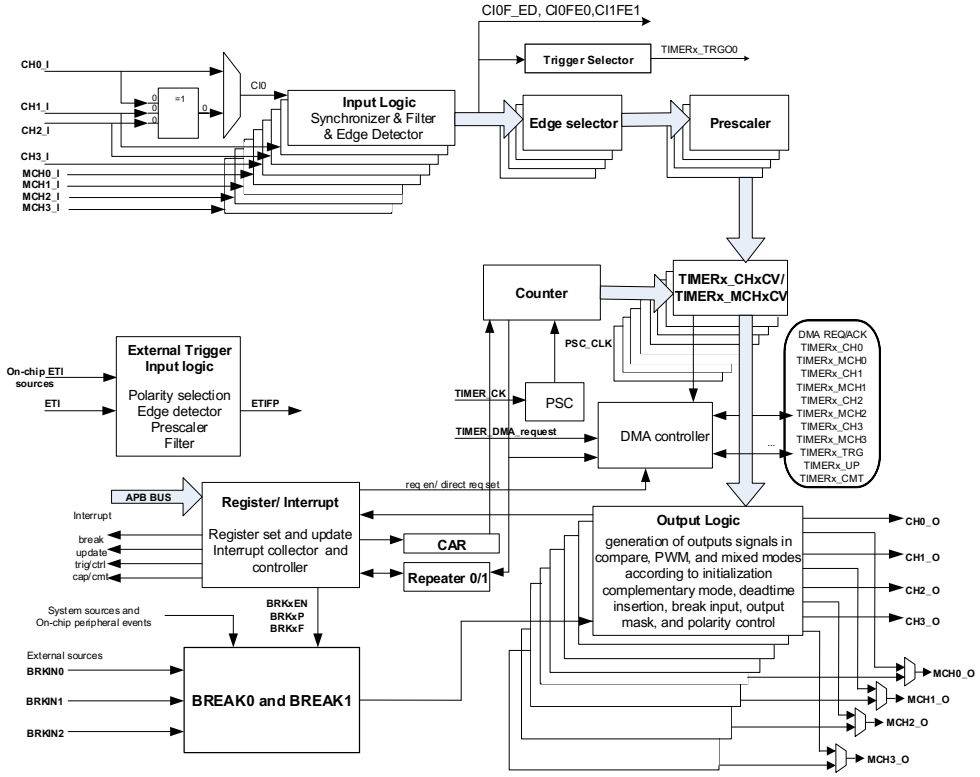
15.1.2. Characteristics

- 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, BREAK1 and channel 0/1/2/3 BREAK.
- Interrupt output or DMA request: update event, trigger event, compare/capture event and break input.
- Daisy chaining of timer module allows a single timer to start multiple timers.
- Timer synchronization allows the selected timers to start counting on the same clock cycle.
- Timer master-slave management.

15.1.3. Block diagram

Figure 15-1. Advanced timer block diagram provides details of the internal configuration of the advanced timer, and *Table 15-2. Advanced timer channel description* introduces the input and output of the channels.

Figure 15-1. Advanced timer block diagram



Table

Table 15-2. 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)		

15.1.4. 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...7,9…15) in SYSCFG_TIMERxCFG(x=0,7) registers.

- TSCFGy[4:0] (y=0..7,9…15) = 5'b00000 in SYSCFG_TIMERxCFG(x=0,7) registers.

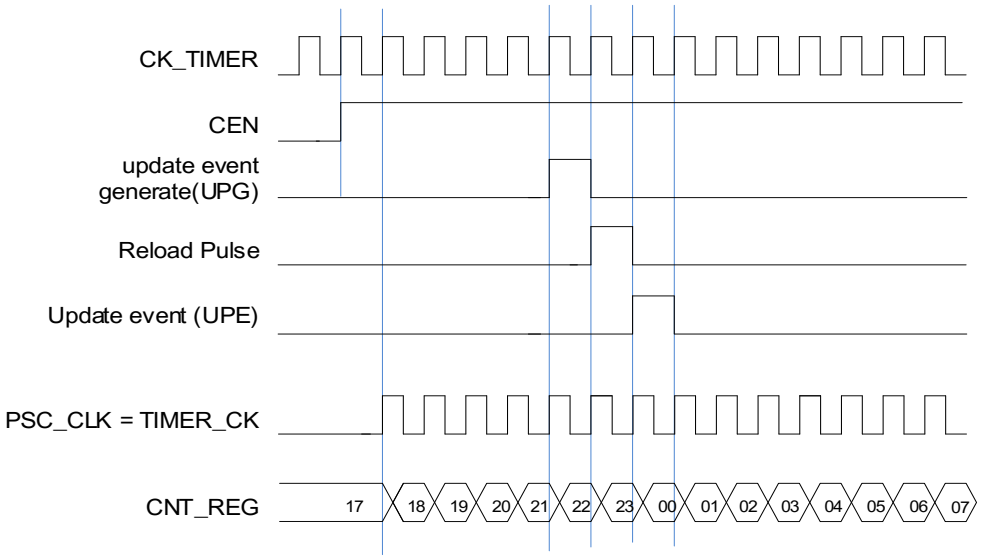
Internal clock CK_TIMER is selected as timer clock source which is from module RCU.

The default clock source is the CK_TIMER for driving the counter prescaler when TSCFGy[4:0] (y=0..7,9…15) = 5'b00000 in SYSCFG_TIMERxCFG(x=0,7) registers. When the CEN is set, the CK_TIMER will be divided by PSC value to generate PSC_CLK.

In this mode, the TIMER_CK which drives counter's prescaler to count is equal to CK_TIMER which is from RCU module.

If TSCFGy[4:0] (y=0..2,6,9..14) in SYSCFG_TIMERxCFG(x=0,7) registers are setting to a nonzero value, the prescaler is clocked by other clock sources selected in the TSCFGy[4:0] (y=0..2,6,9..14) bit-field, 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 15-2. Normal mode, internal clock divided by 1



■ TSCFG6[4:0] are setting to a nonzero value (external clock mode 0). External input pin is selected as timer clock source.

The TIMER_CK, which drives counter's prescaler to count, can be triggered by the event of rising or falling edge on the external pin TIMERx_CHn/ TIMERx_MCHn(n=0...3). 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 ITI0/1/2/3/12/13/14. This mode can be selected by setting TSCFG6[4:0] to 0x1~0x4, 0x11, 0x12 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

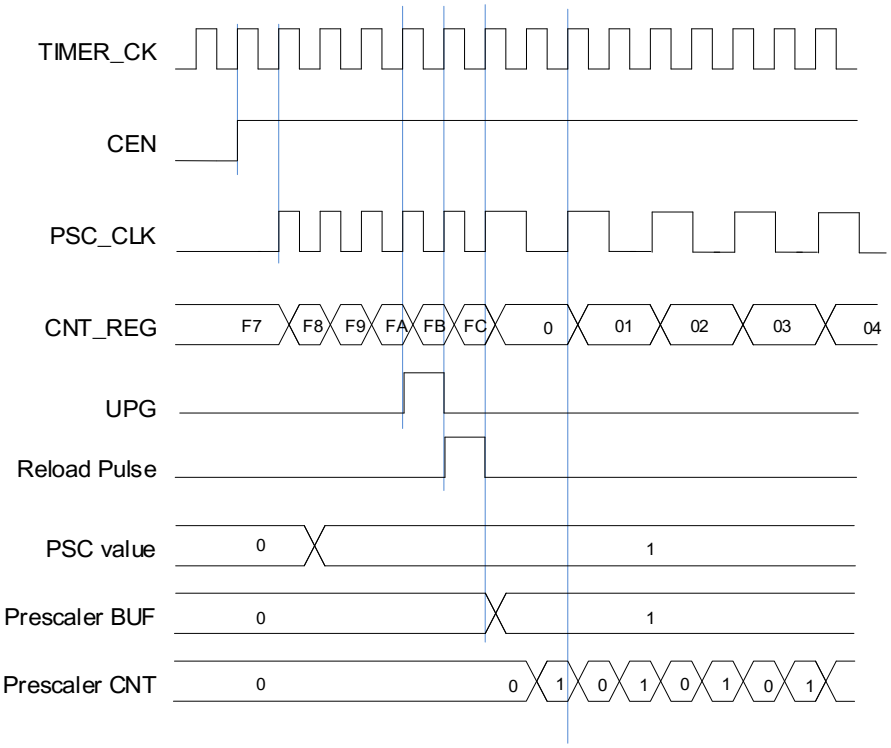
source is setting the TSCFG6[4:0] to 0x8. Note that the ETI signal is derived from the ETI pin sampled by a digital filter. When the ETI signal is selected as the clock source, the trigger controller including the edge detection circuitry will generate a clock pulse on each ETI signal rising edge to 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)*** 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 15-3. 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 15-4. Timing diagram of up counting mode, PSC=0/2 and *Figure 15-5. 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 15-4. Timing diagram of up counting mode, PSC=0/2

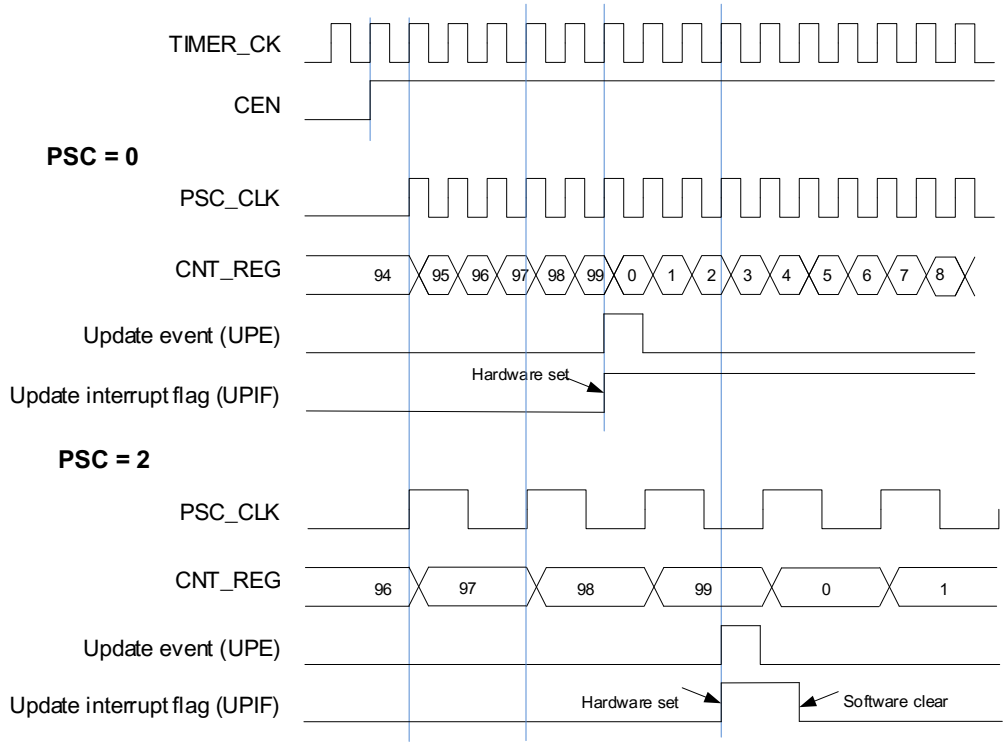
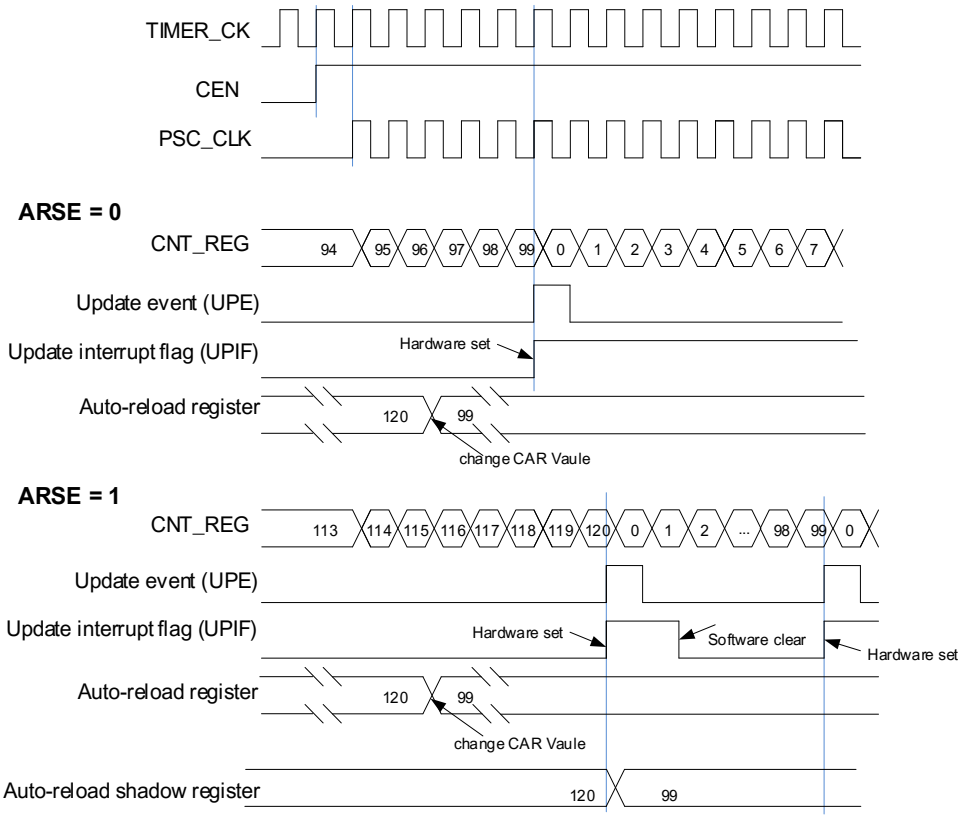


Figure 15-5. Timing diagram of up counting mode, change TIMERx_CAR ongoing



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

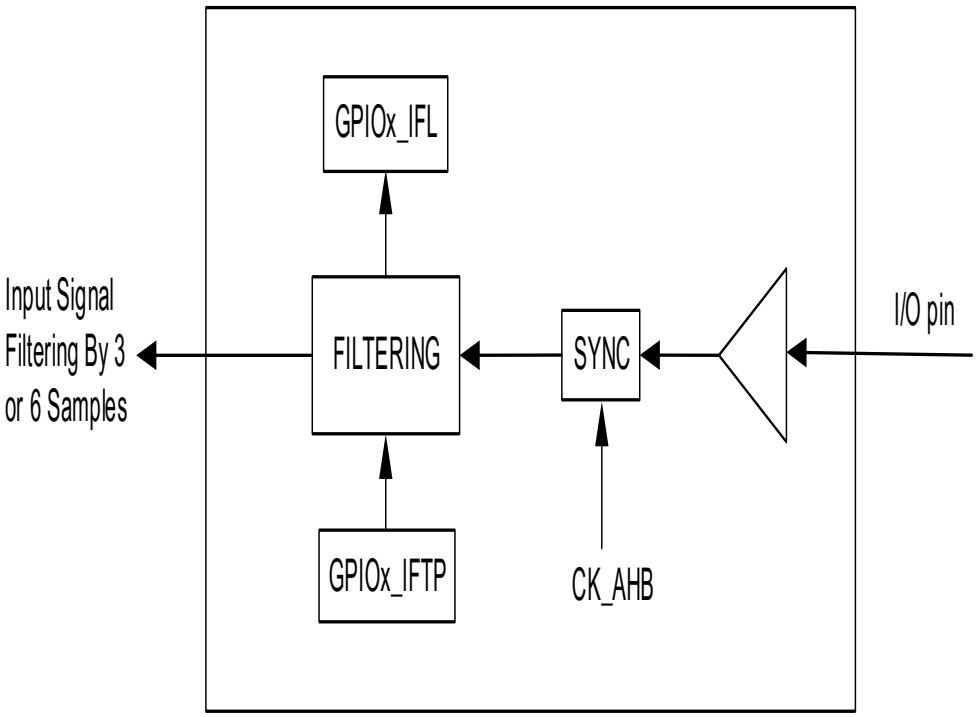
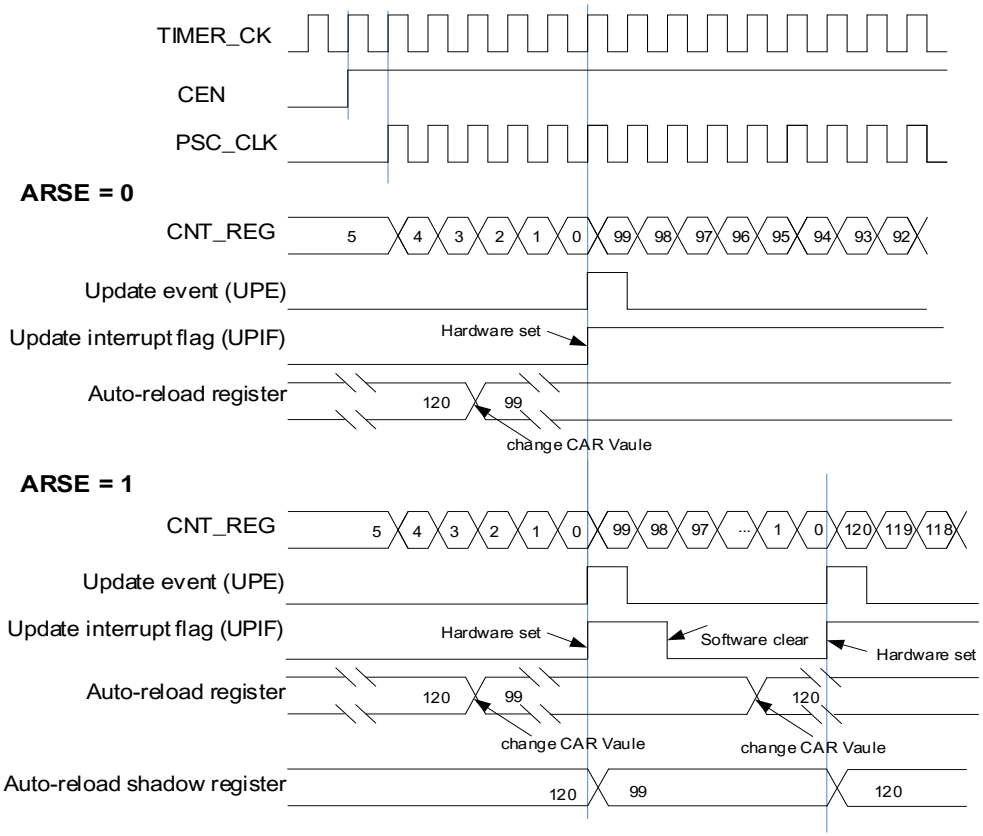


Figure 15-7. Timing diagram of down counting mode, change TIMERx_CAR ongoing



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.

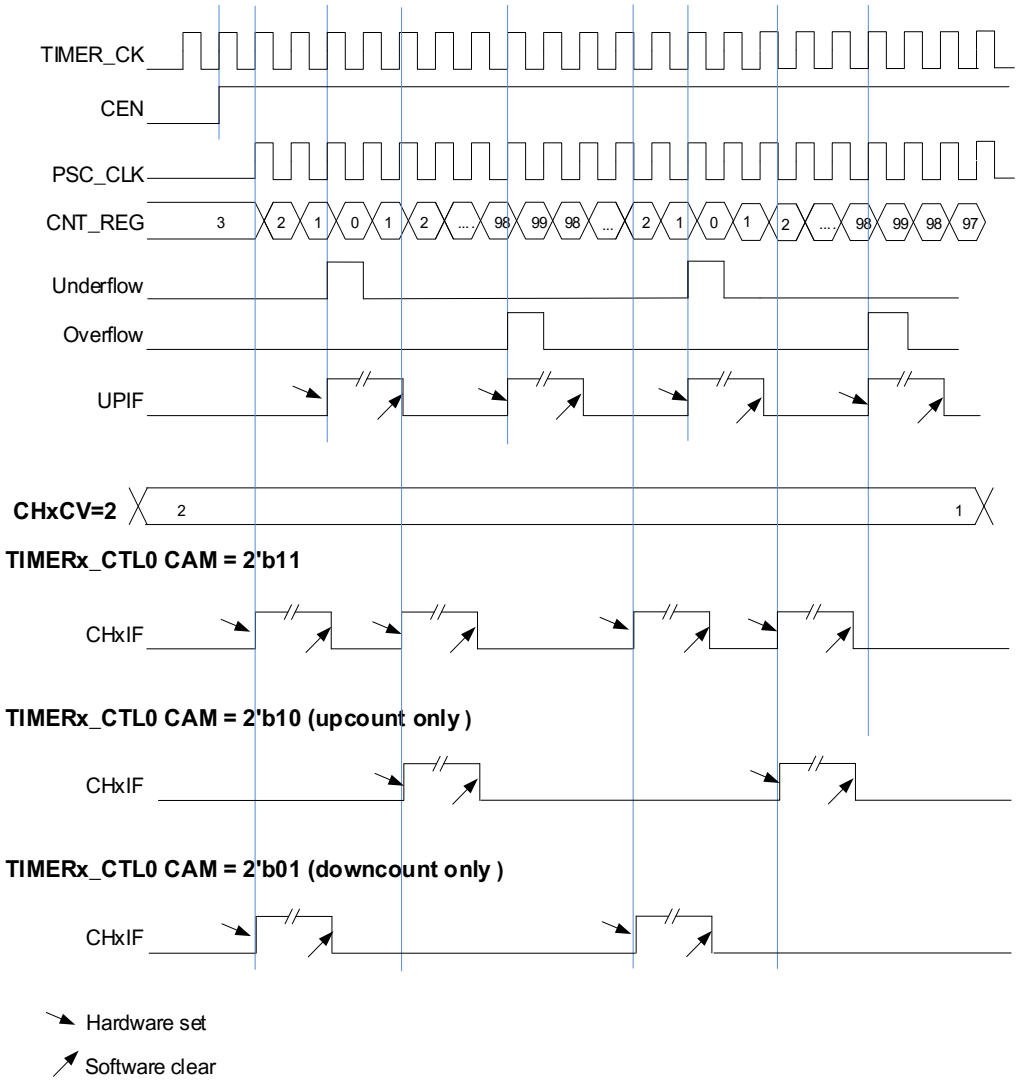
The UPIF bit in the TIMERx_INTF register will be set to 1 either when an underflow event or an overflow event occurs. While the ChxIF bit is associated with the value of CAM[1:0] in TIMERx_CTL0. The details refer to ***Figure 15-8. Timing diagram of center-aligned counting mode***.

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 15-8. Timing diagram of center-aligned counting mode shows some examples of the counter behavior when TIMERx_CAR=0x99. TIMERx_PSC=0x0.

Figure 15-8. 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 CREPSEL 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 15-9. Repetition counter timing diagram of center-aligned counting mode

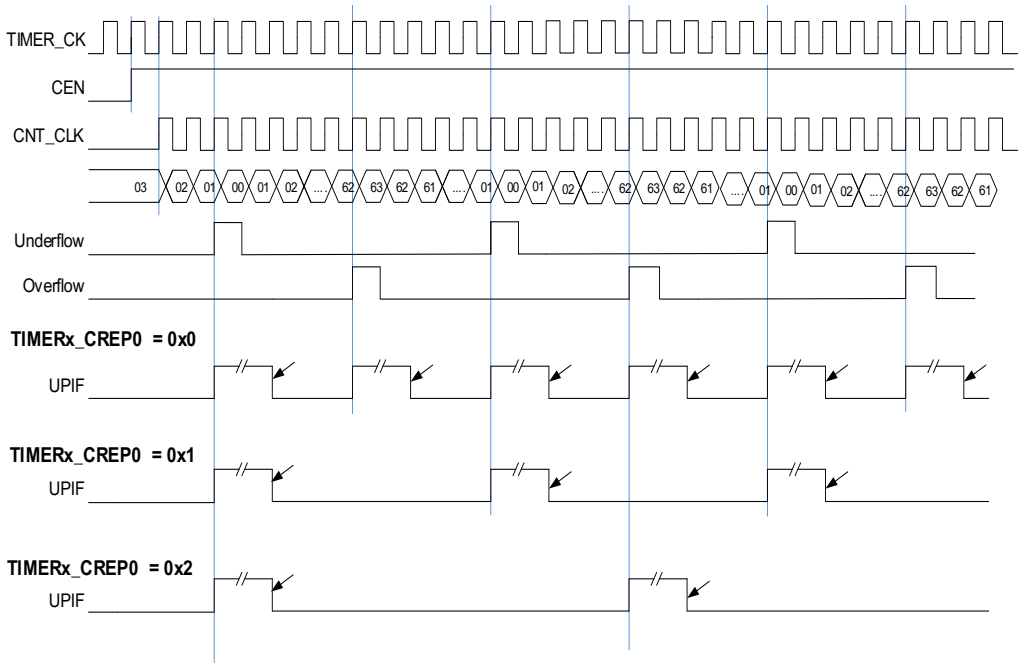


Figure 15-10. Repetition counter timing diagram of up counting mode

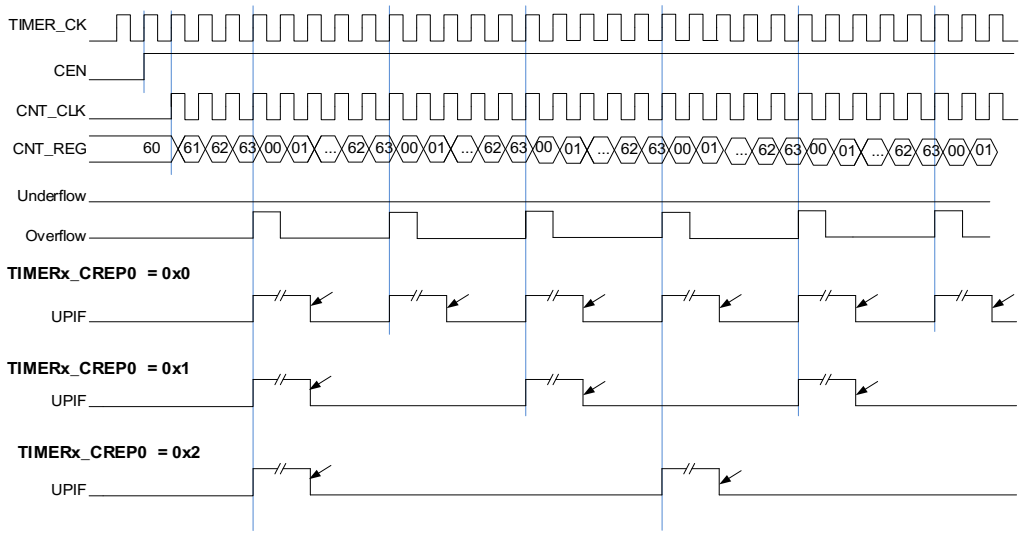
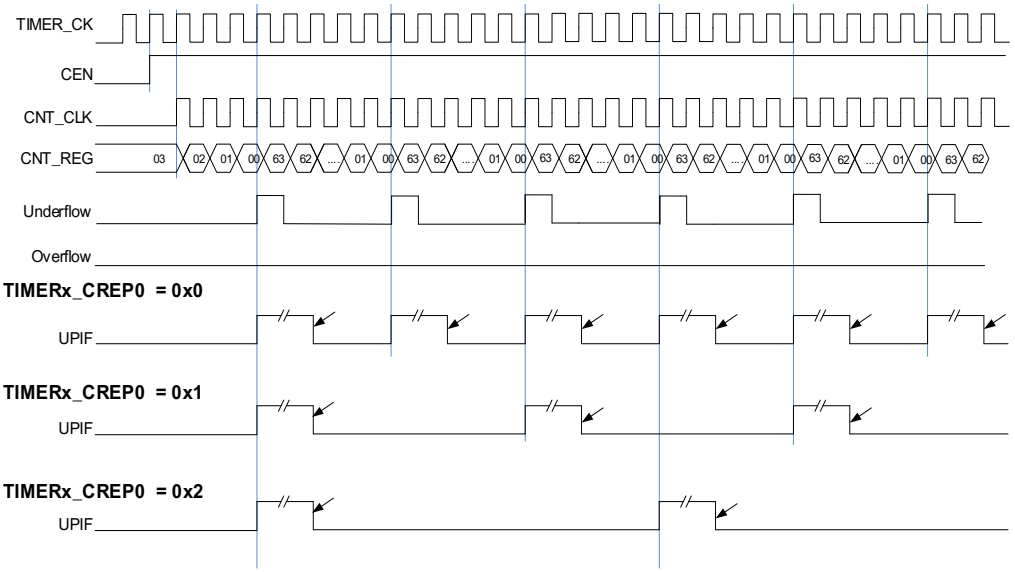


Figure 15-11. Repetition counter timing diagram of down counting mode



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 15-12. Input capture logic for channel 0

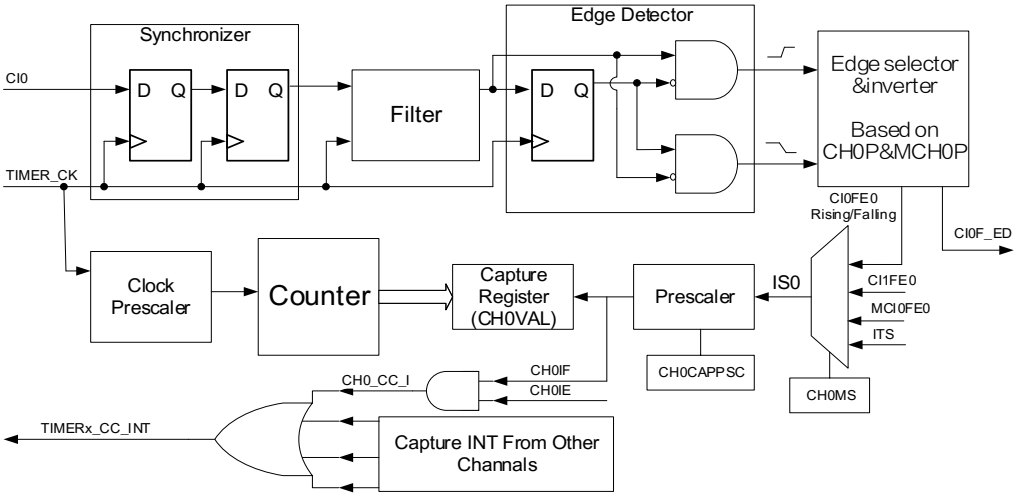
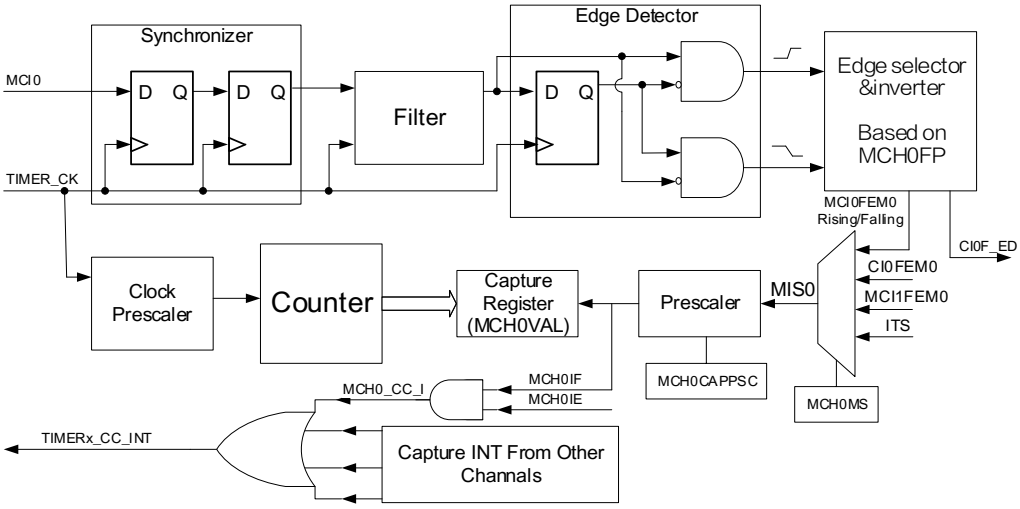


Figure 15-13. Input capture logic for multi mode channel 0



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 15-14. Output compare logic (when MCHxMSEL = 2'b00, x=0, 1, 2, 3) and *Figure 15-15. Output compare logic (when MCHxMSEL = 2'b11, x=0,1,2,3)* show the logic circuit of output compare mode.

Figure 15-14. Output compare logic (when MCHxMSEL = 2'b00, x=0, 1, 2, 3)

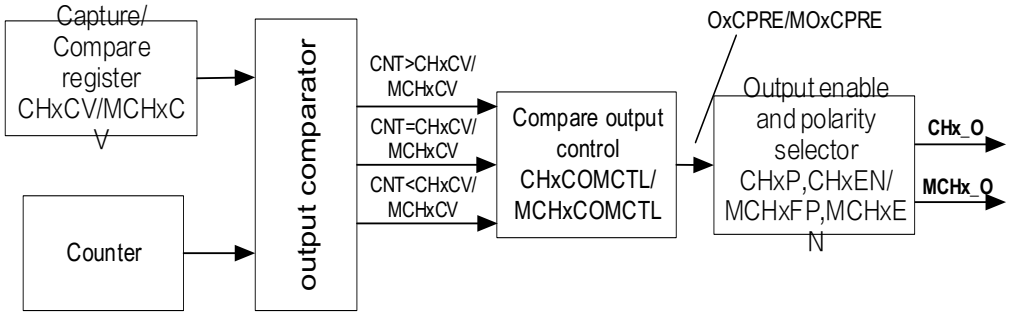
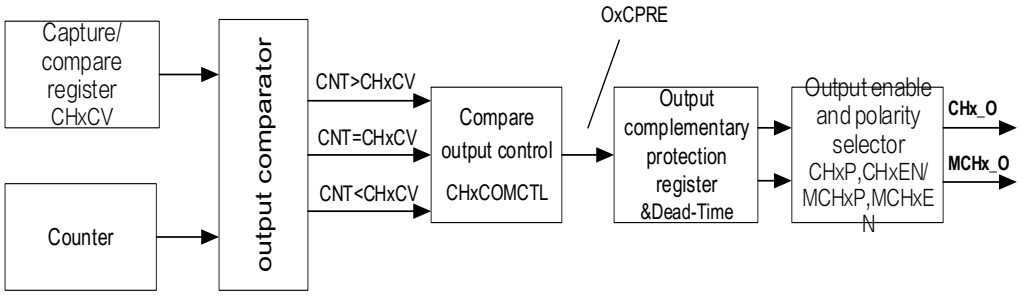


Figure 15-15. Output compare logic (when MCHxMSEL = 2'b11, x=0,1,2,3)



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 15-14. 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 15-15. 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_CCHP register. Please refer to ***Outputs Complementary*** for more details.

In output compare mode, the TIMERx can generate timed pulses with programmable position, polarity, duration and frequency. When the counter matches the value in the TIMERx_CHxCV/ TIMERx_MCHxCV register of an output compare channel, the channel (n) output can be set, cleared, or toggled based on CHxCOMCTL/ MCHxCOMCTL. When the counter reaches the value in the TIMERx_CHxCV/ TIMERx_MCHxCV register, the ChxIF/ MCHxIF bit will be set and the channel (n) interrupt is generated if ChxIE/ MCHxIE = 1. And the DMA request will be asserted, if CHxDEN/ MCHxDEN =1.

So, the process can be divided into several steps as below:

Step1: Clock Configuration. Such as clock source, clock prescaler and so on.

Step2: Compare mode configuration.

- Set the shadow enable mode by CHxCOMSEN/ MCHxCOMSEN.
- Set the output mode (set/clear/toggle) by CHxCOMCTL/ MCHxCOMCTL.
- Select the active polarity by CHxP/MCHxP/ MCHxFP.
- Enable the output by ChxEN/ MCHxEN.

Step3: Interrupt/DMA request enable configuration by ChxIE/ MCHxIE /CHxDEN/ MCHxDEN.

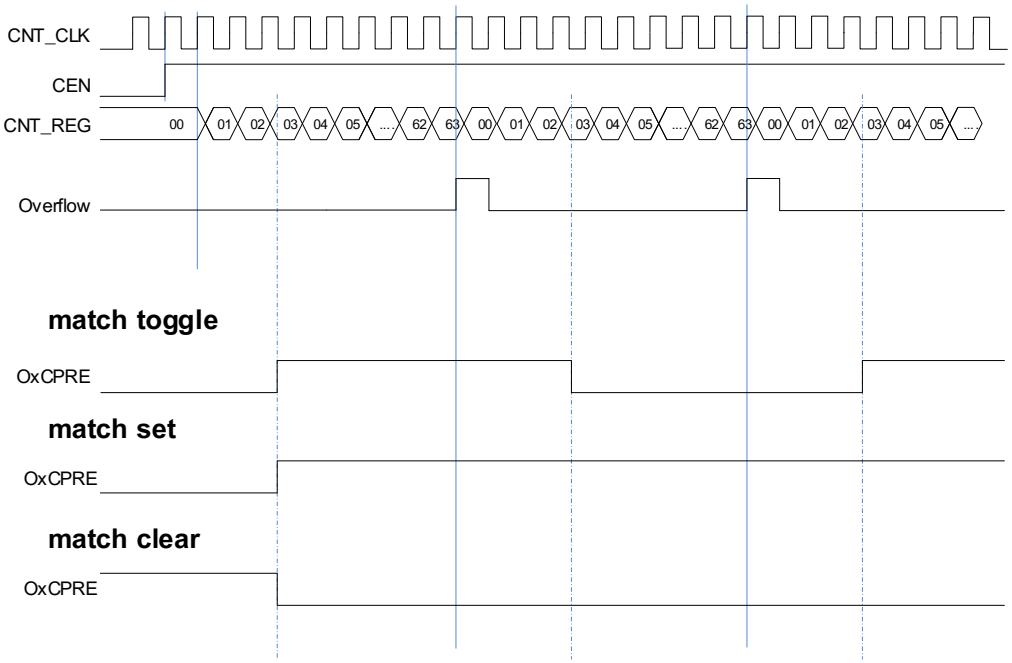
Step4: Compare output timing configuration by TIMERx_CAR and TIMERx_CHxCV/ TIMERx_MCHxCV.

The TIMERx_CHxCV/ TIMERx_MCHxCV can be changed ongoing to meet the expected waveform.

Step5: Start the counter by configuring CEN to 1.

Figure 15-16. Output-compare in three modes shows the three compare modes: toggle/set/clear. CARL=0x63, CHxVAL=0x3.

Figure 15-16. 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 15-17. 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 15-18. 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 15-17. Timing diagram of EAPWM

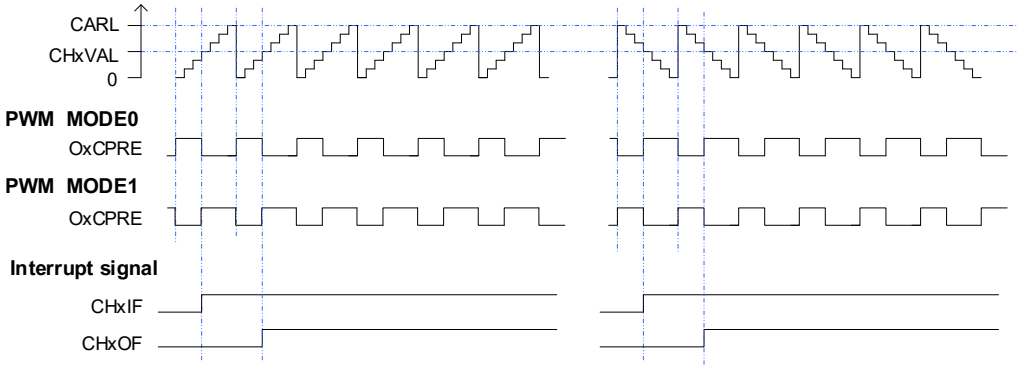
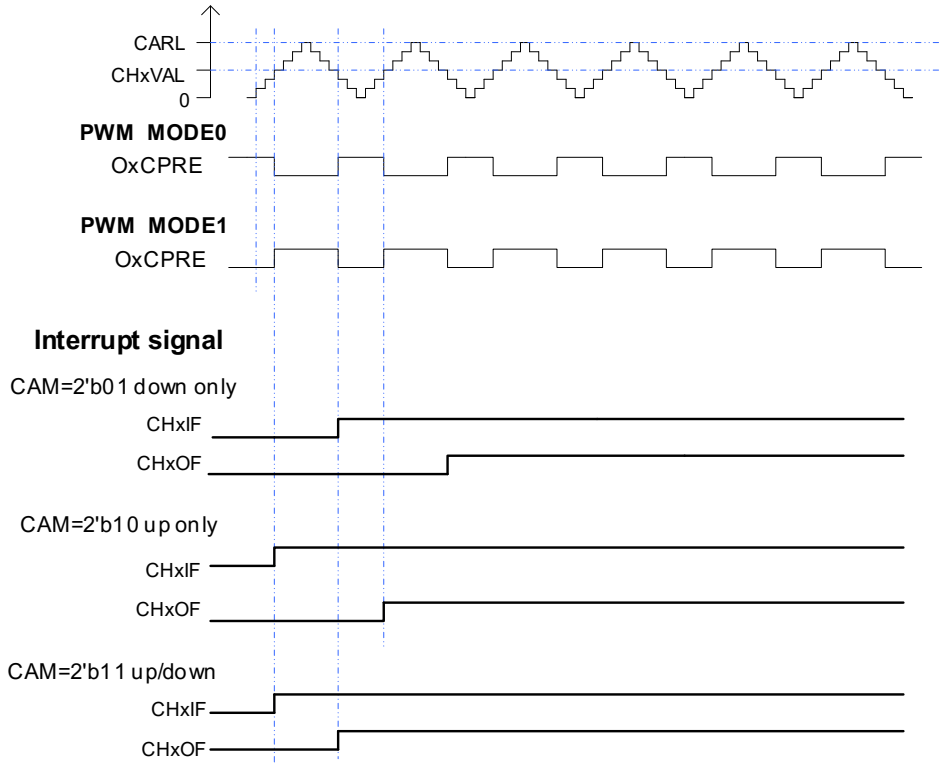


Figure 15-18. Timing diagram of CAPWM



Composite PWM mode

In the Composite PWM mode (CHxCPWMEN = 1'b1, CHxMS[2:0] = 3'b000 and CHxCOMCTL = 4'b0110 or 4'b0111), the PWM signal output in channel x (x=0..3) is composited by CHxVAL and CHxCOMVAL_ADD bits.

If CHxCOMCTL = 4'b0110 (PWM mode 0) and DIR = 1'b0 (up counting mode), or CHxCOMCTL = 4'b0111 (PWM mode 1) and DIR = 1'b1 (Down counting mode), the channel x output is forced low when the counter matches the value of CHxVAL. It is forced high when the counter matches the value of CHxCOMVAL_ADD.

If CHxCOMCTL = 4'b0111 (PWM mode 1) and DIR = 1'b0 (up counting mode), or

CHxCOMCTL = 4'b0110 (PWM mode 0) and DIR = 1'b1 (down counting mode) the channel x output is forced high when the counter matches the value of CHxVAL. It is forced low when the counter matches the value of CHxCOMVAL_ADD.

When CHxVAL or CHxCOMVAL_ADD = 0 / CARL, the channel x output is specially processed, and by setting the CHxPERFOREN bit in the TIMERx_CHCTL2 register, the OxCPRE output is forced to either high or low level (determined according to the selected composite PWM mode).

The PWM period is determined by (CARL + 0x0001) and the PWM pulse width is determined by the following table.

Table 15-3.The Composite PWM pulse width

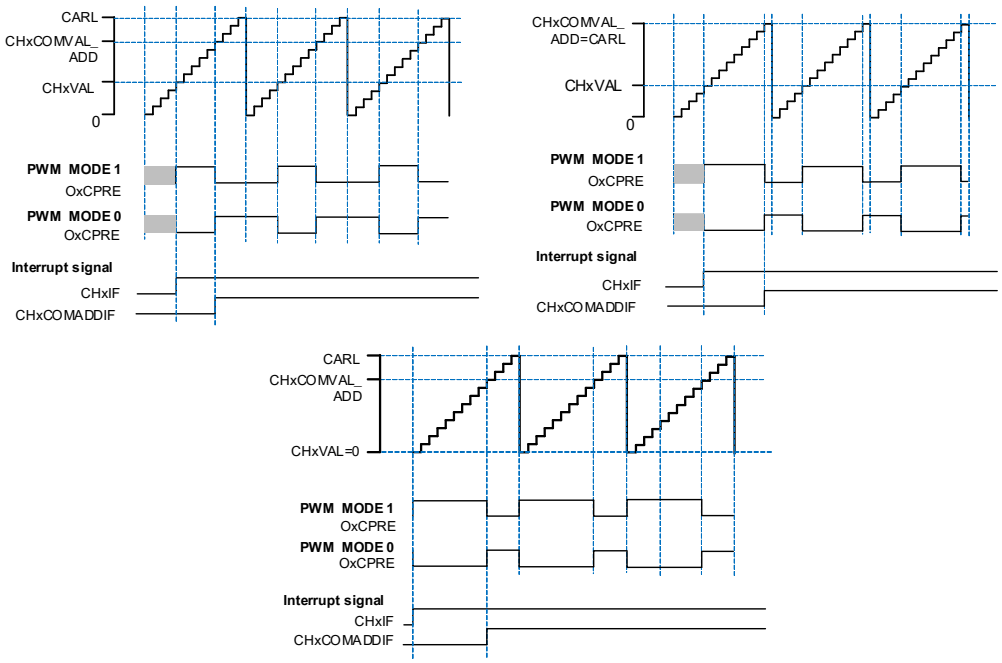
Condition	Mode	PWM pulse width
CHxVAL < CHxCOMVAL_ADD <= CARL	PWM mode 0	(CARL + 0x0001) + (CHxVAL - CHxCOMVAL_ADD)
	PWM mode 1	(CHxCOMVAL_ADD - CHxVAL)
CHxCOMVAL_ADD < CHxVAL <= CARL	PWM mode 0	(CHxVAL - CHxCOMVAL_ADD)
	PWM mode 1	(CARL + 0x0001) + (CHxCOMVAL_ADD - CHxVAL)
(CHxVAL = CHxCOMVAL_ADD <= CARL) or (CHxVAL > CARL > CHxCOMVAL_ADD)	PWM mode 0 (up counting) or PWM mode 1 (down counting)	100%
	PWM mode 0 (down counting) or PWM mode 1 (up counting)	0%
CHxCOMVAL_ADD > CARL > CHxVAL	PWM mode 0(up counting) or PWM mode 1(down counting)	0%
	PWM mode 0(down counting) or PWM mode 1(up counting)	100%
(CHxVAL>CARL) and (CHxCOMVAL_ADD > CARL)	-	The output of CHx_O is keeping

When the counter reaches the value of CHxVAL, the ChxIF bit is set and the channel x interrupt is generated if ChxIE = 1, and the DMA request will be asserted, if CHxDEN=1. When the counter reaches the value of CHxCOMVAL_ADD, the CHxCOMADDIF bit is set (this flag just used in composite PWM mode, when CHxCPWMEN=1, and Asymmetric PWM mode) and the channel x additional compare interrupt is generated if CHxCOMADDIE = 1 (Only interrupt is generated, no DMA request is generated).

According to the relationship among CHxVAL, CHxCOMVAL_ADD and CARL, it can be divided into four situations:

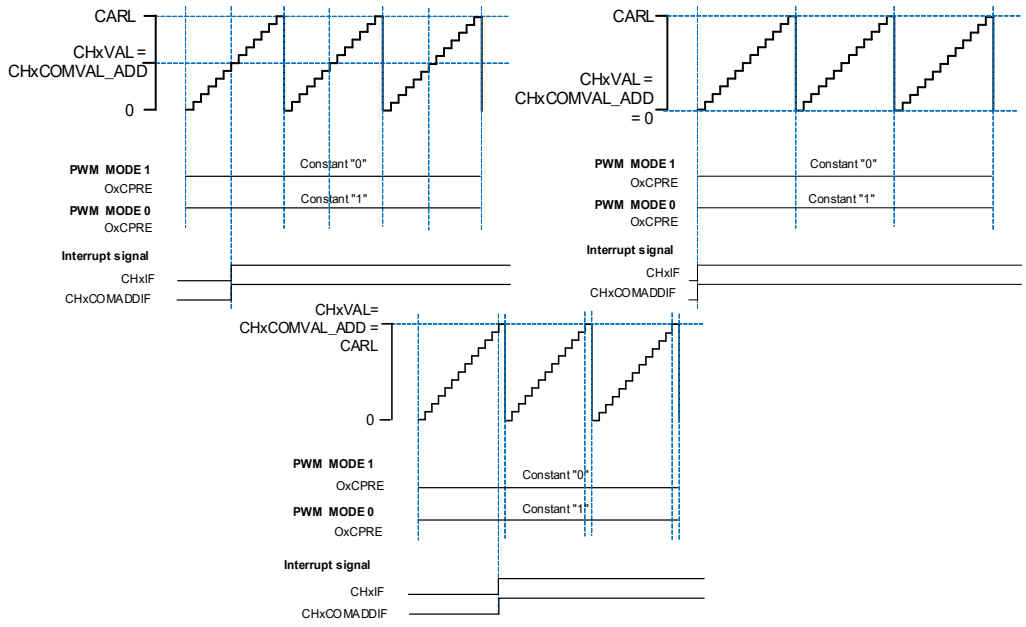
1) CHxVAL < CHxCOMVAL_ADD, and the values of CHxVAL and CHxCOMVAL_ADD between 0 and CARL.

Figure 15-19. Channel x output PWM with (CHxVAL < CHxCOMVAL_ADD)



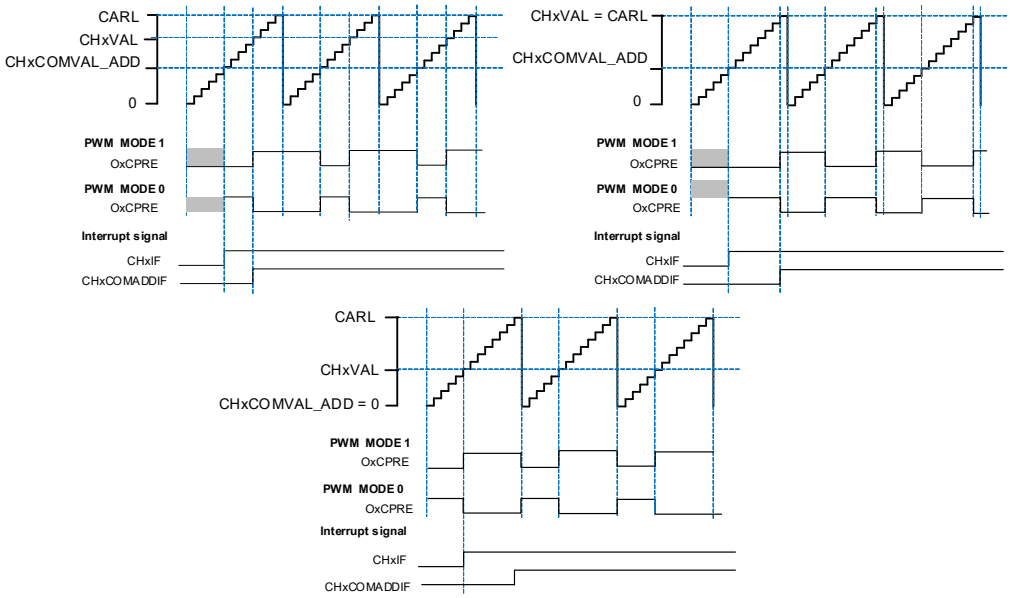
2) CHxVAL = CHxCOMVAL_ADD, and the value of CHxVAL and CHxCOMVAL_ADD between 0 and CARL.

Figure 15-20. Channel x output PWM with (CHxVAL = CHxCOMVAL_ADD)



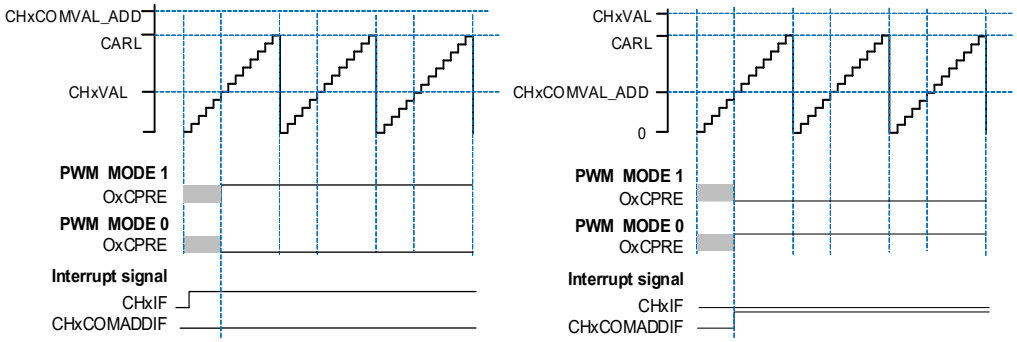
3) CHxVAL > CHxCOMVAL_ADD, and the value of CHxVAL and CHxCOMVAL_ADD between 0 and CARL.

Figure 15-21. Channel x output PWM with (CHxVAL > CHxCOMVAL_ADD)



4) One of the value of CHxVAL and CHxCOMVAL_ADD exceeds CARL.

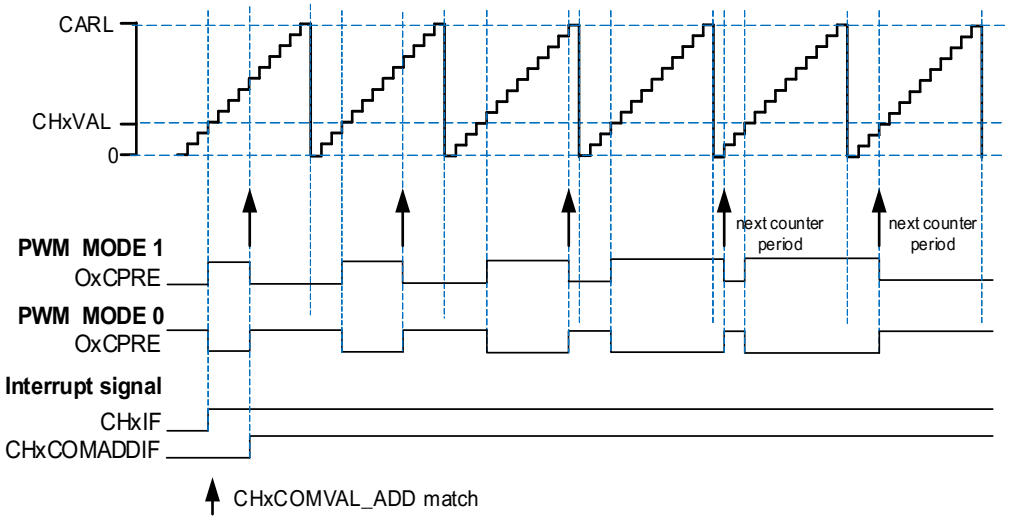
Figure 15-22. Channel x output PWM with CHxVAL or CHxCOMVAL_ADD exceeds CARL



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 15-23. Channel x output PWM duty cycle changing with CHxCOMVAL_ADD*** shows the PWM output and interrupts waveform.

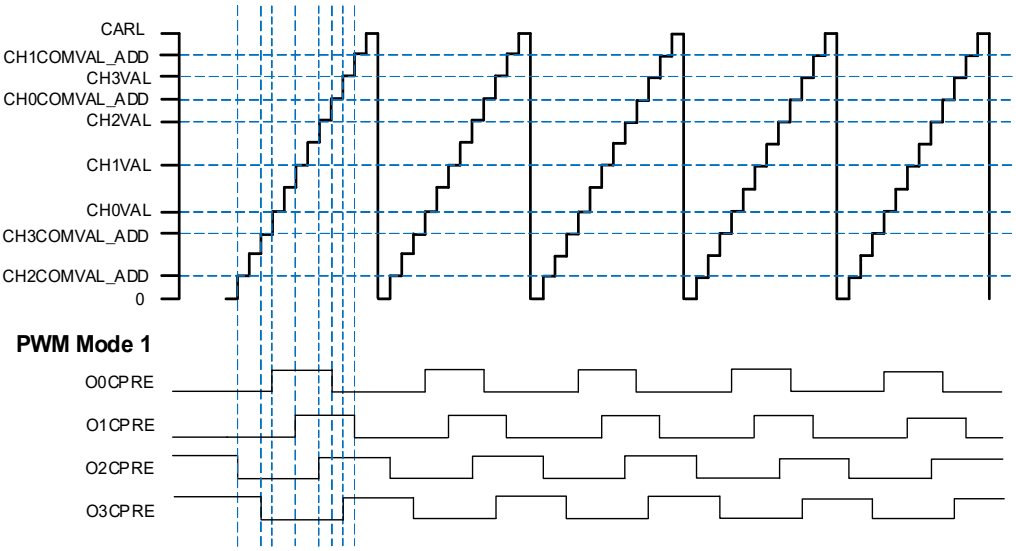
In some cases, the CHxCOMVAL_ADD match can happen on the next counter period (the value of CHxCOMVAL_ADD was written after the counter reaches the value of CHxVAL, and the value of CHxCOMVAL_ADD was less than or equal to the CHxVAL).

Figure 15-23. 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 15-24. Four Channels outputs in Composite PWM mode



Asymmetric PWM mode

Asymmetric PWM mode allows two center-aligned PWM signals to be generated with a programmable phase shift. In the Asymmetric PWM mode(CHxCOMCTL[3:0]=4'b1010/4'b1011 and CHxMS[2:0] = 3'b000), the PWM signal output in channel x (x=0..3) is composited by CHxVAL and CHxCOMVAL_ADD bits. The Asymmetric PWM mode is only available in the center-aligned counting mode.

If CHxCOMCTL = 4'b1010 (Asymmetric PWM mode 0) and DIR = 1'b0 (up counting mode),

the channel x output is forced low when the counter matches the value of CHxVAL.

If CHxCOMCTL = 4'b1010 (Asymmetric PWM mode 0) and DIR = 1'b1 (down counting mode), the channel x output is forced high when the counter matches the value of CHxCOMVAL_ADD.

If CHxCOMCTL = 4'b1011 (Asymmetric PWM mode 1) and DIR = 1'b0 (up counting mode), the channel x output is forced high when the counter matches the value of CHxVAL.

If CHxCOMCTL = 4'b1011 (Asymmetric PWM mode 1) and DIR = 1'b1 (down counting mode), the channel x output is forced low when the counter matches the value of CHxCOMVAL_ADD.

Compared to Composite PWM mode, Asymmetric PWM mode is only valid with center-aligned mode, and the CHxVAL is only valid when counting up and CHxCOMVAL_ADD is only valid when counting down. When CHxVAL or CHxCOMVAL_ADD = 0 / CARL, the channel x output is specially processed, and by setting the CHxPERFOREN bit in the TIMERx_CHCTL2 register, the OxCPRE output is forced to either high or low level (determined according to the selected asymmetric PWM mode).

When CHxVAL =0, the channel x output level is low in Asymmetric PWM mode 0. When CHxVAL =0, the channel x output level is high in Asymmetric PWM mode 1.

When CHxCOMVAL_ADD =0, the channel x output level is high in Asymmetric PWM mode 0. When CHxCOMVAL_ADD =0, the channel x output level is low in Asymmetric PWM mode 1.

When CHxVAL = CARL, the channel x output level is low in Asymmetric PWM mode 0. When CHxVAL = CARL, the channel x output level is high in Asymmetric PWM mode 1.

When CHxCOMVAL_ADD = CARL, the channel x output level is high in Asymmetric PWM mode 0. When CHxCOMVAL_ADD = CARL, the channel x output level is low in Asymmetric PWM mode 1.

The PWM period is determined by 2CARL and the PWM pulse width is determined by the following table ***Table 15-4.The Asymmetric PWM pulse*** **width**.

Table 15-4.The Asymmetric PWM pulse width

Condition	Mode	PWM pulse width
CHxVAL < CHxCOMVAL_ADD <= CARL or CHxCOMVAL_ADD < CHxVAL <= CARL or CHxVAL = CHxCOMVAL_ADD	Asymmetric PWM mode 0	CHxCOMVAL_ADD + CHxVAL
	Asymmetric PWM mode 1	2CARL - CHxCOMVAL_ADD - CHxVAL
CHxVAL > CARL > CHxVAL_ADD	Asymmetric PWM mode 0	100%
	Asymmetric PWM mode 1	0%
CHxCOMVAL_ADD > CARL > CHxVAL	Asymmetric PWM mode 0	0%
	Asymmetric PWM mode 1	100%
(CHxVAL>CARL)and (CHxCOMVAL_ADD > CARL)	-	The output of CHx_O is keeping

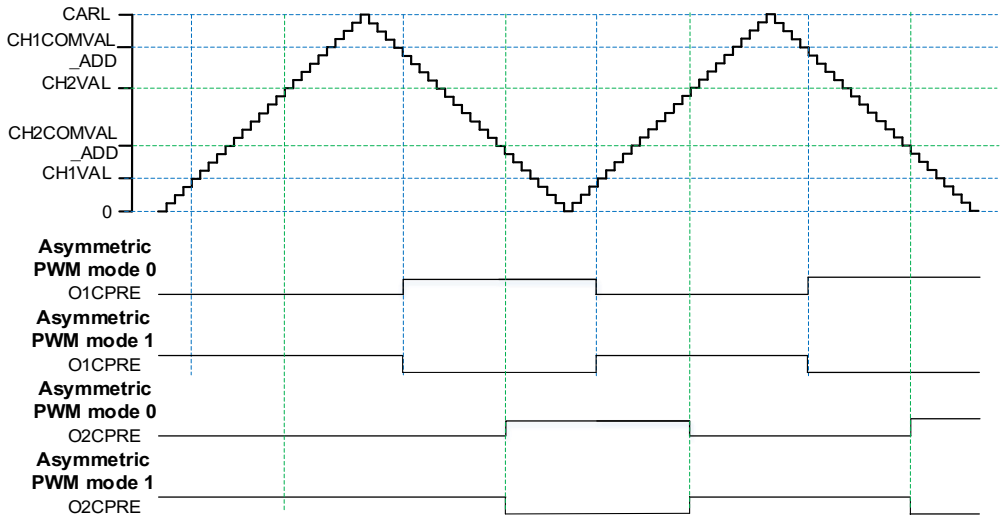
When the counter reaches the value of CHxVAL, the ChxIF bit is set and the channel x interrupt is generated if ChxIE = 1, and the DMA request will be asserted, if CHxDEN=1. When the counter reaches the value of CHxCOMVAL_ADD, the CHxCOMADDIF bit is set (this flag just used in Composite PWM mode and Asymmetric PWM mode, when CHxCOMCTL[3:2]= 2'b10) and the channel x additional compare interrupt is generated if CHxCOMADDIE = 1 (Only interrupt is generated, no DMA request is generated).

According to the relationship among CHxVAL, CHxCOMVAL_ADD and CARL, it can be divided into three situations:

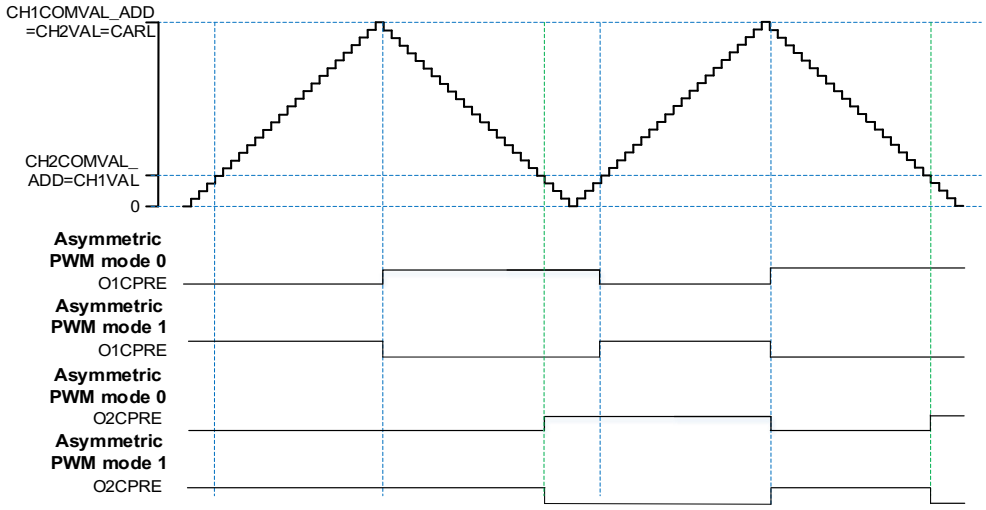
1) CHxVAL < CHxCOMVAL_ADD, and the values of CHxVAL and CHxCOMVAL_ADD between 0 and CARL.

Figure 15-25. Channel x output PWM with(CHxVAL < CHxCOMVAL_ADD / CHxCOMVAL_ADD < CHxVAL)

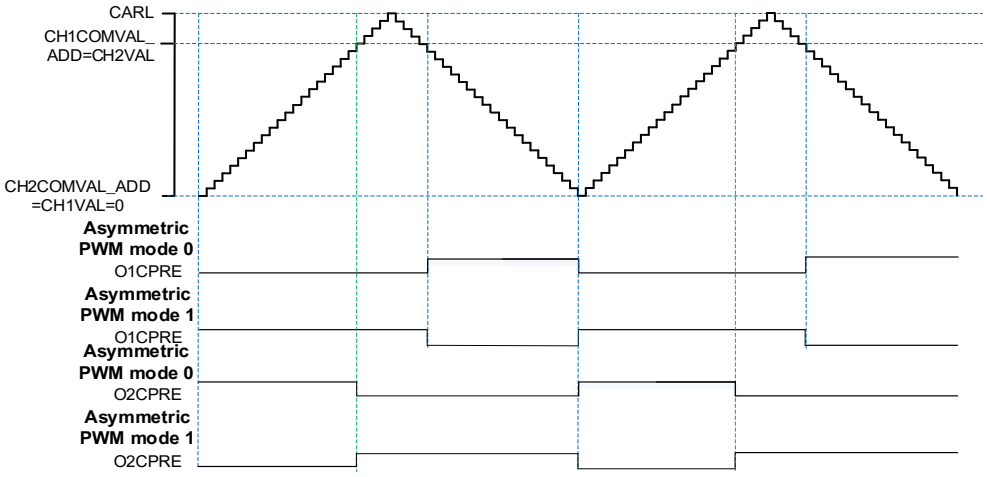
8 0 < CH1VAL < CH1COMVAL_ADD < CARL / 0 < CH2COMVAL_ADD < CH2VAL < CARL



9 0 < CH1VAL < CH1COMVAL_ADD = CARL / 0 < CH2COMVAL_ADD < CH2VAL = CARL, and CH1VAL = CH2COMVAL_ADD



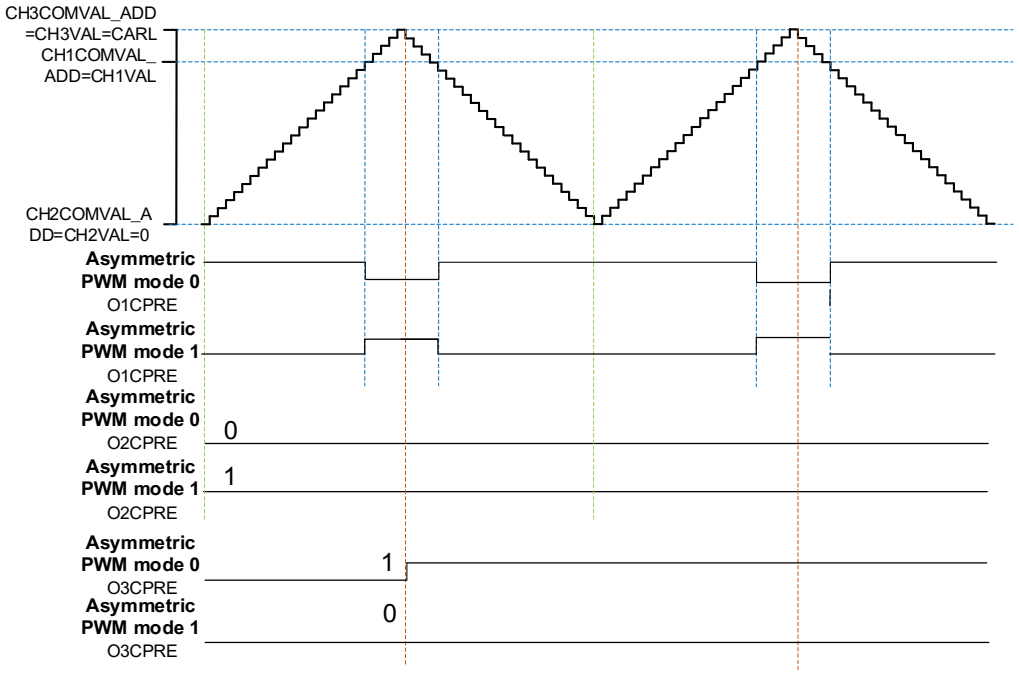
10 0 = CH1VAL < CH1COMVAL_ADD < CARL / 0 = CH2COMVAL_ADD < CH2VAL < CARL, and CH1COMVAL_ADD = CH2VAL



2) CHxVAL = CHxCOMVAL_ADD, and the value of CHxVAL and CHxCOMVAL_ADD between 0 and CARL.

Figure 15-26. Channel x output PWM with(CHxVAL = CHxCOMVAL_ADD)

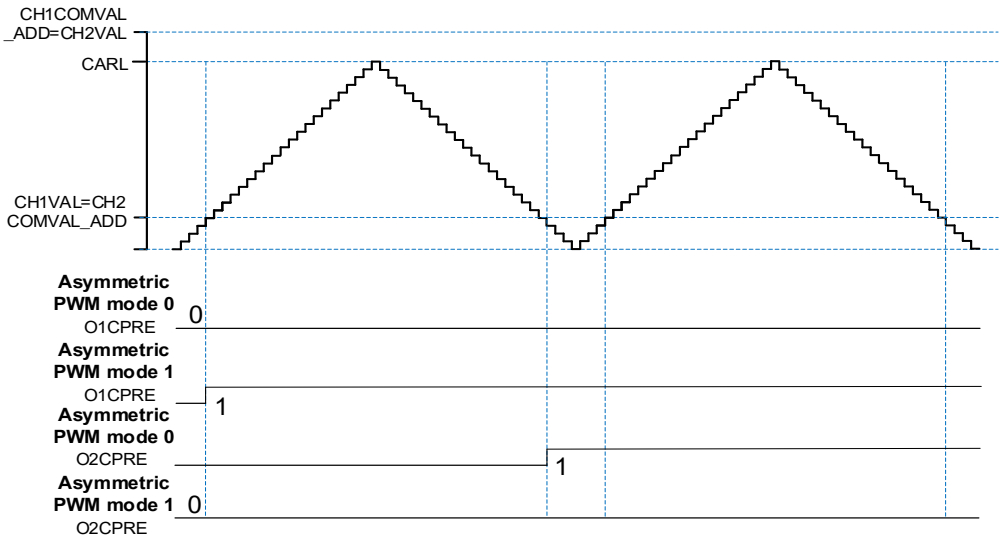
0 < CH1VAL = CH1COMVAL_ADD < CARL / CH2VAL = CH2COMVAL_ADD = 0 / CH3VAL = CH3COMVAL_ADD = CARL



3) One of the value of CHxVAL and CHxCOMVAL_ADD exceeds CARL.

Figure 15-27. Channel x output PWM with CHxVAL or CHxCOMVAL_ADD exceeds CARL

0 < CH1VAL < CARL < CH1COMVAL_ADD / 0 < CH2COMVAL_ADD < CARL < CH2VAL



Output match pulse select

Basing on that CHx_O (x=0..3) outputs are configured by CHxCOMCTL[3:0] (x=0..3) bits when the match events occur, the output signal is configured by CHxOMPSEL[1:0] (x=0..3) bit to be normal or a pulse.

When the match events occur, the CHxOMPSEL[1:0] (x=0..3) bits are used to select the output of OxCPRE which drives CHx_O:

- CHxOMPSEL = 2'b00, the OxCPRE signal is output normally with the configuration of CHxCOMCTL[3:0] bits;
- CHxOMPSEL = 2'b01, only the counter is counting up, the OxCPRE signal is output a pulse when the match events occur, and the pulse width is one CK_TIMER clock cycle.
- CHxOMPSEL = 2'b10, only the counter is counting down, the OxCPRE signal is output a pulse when the match events occur, and the pulse width is one CK_TIMER clock cycle.
- CHxOMPSEL = 2'b11, both the counter is counting up and counting down, the OxCPRE signal is output a pulse when the match events occur, and the pulse width is one CK_TIMER clock cycle.

Figure 15-28. CHx_O output with a pulse in edge-aligned mode (CHxOMPSEL!=2'b00)

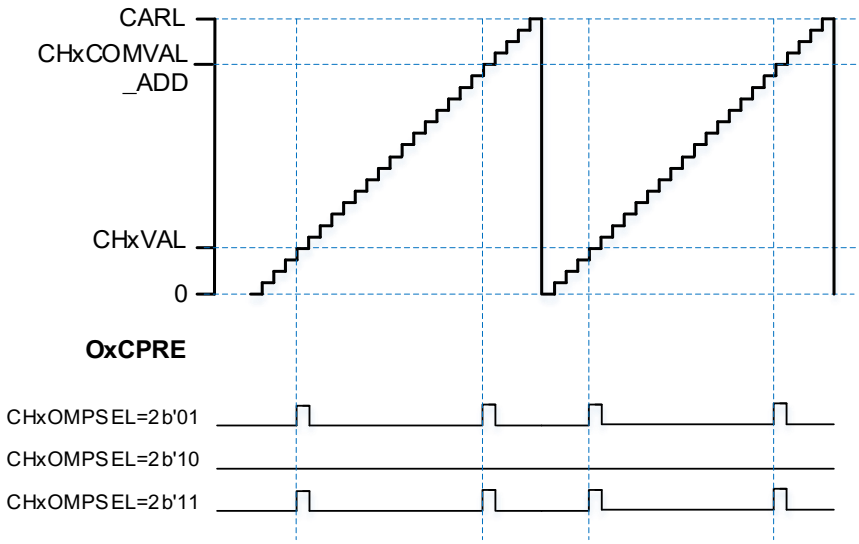
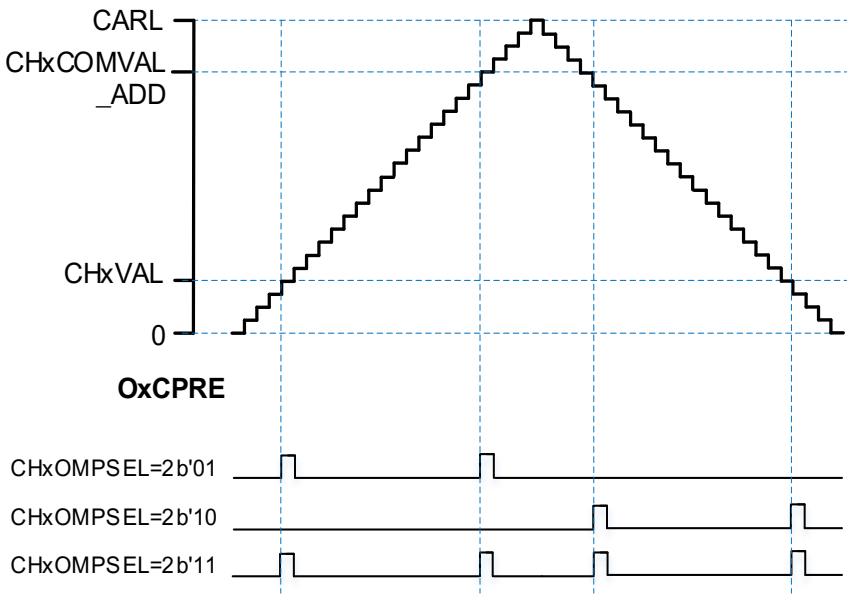


Figure 15-29. CHx_O output with a pulse in center-aligned mode (CHxOMPSEL!=2'b00)



Channel output prepare signal

As is shown in ***Figure 15-14. Output compare logic (when MCHxMSEL = 2'b00, x=0, 1, 2, 3)*** and ***Figure 15-15. 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.

The Composite PWM mode 0/ Composite PWM mode 1 output is one output type of OxCPRE which is setup by configuring the CHxCOMCTL field to 0x06/0x07 and CHxCPWMEN = 1'b1. In the two modes, the OxCPRE signal level is changed according to the counting direction and the relationship between the counter value and the TIMERx_CHxCV / TIMERx_CHxCOMVAL_ADD content. Refer to the definition of relative bit for more details.

The Asymmetric PWM mode 0/ Asymmetric PWM mode 1 output is one output type of OxCPRE which is setup by configuring the CHxCOMCTL field to 0x0a/0x0b. In the two modes, the OxCPRE signal level is changed according to the counting direction and the relationship between the counter value and the TIMERx_CHxCV / TIMERx_CHxCOMVAL_ADD content. Refer to the definition of relative bit for more details.

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

Configure the CHxCOMCEN bit to 1 in the TIMERx_CHCTL0 register, the OxCPRE signal can be forced to 0 when the ETIFP signal derived from the external ETI pin is set to a high level. The OxCPRE signal will not return to its active level until the next update event occurs (OCPRECIVM = 1) or underflow/overflow event occurs (OCPRECIVM = 0).

Outputs Complementary

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&CHPOENx, ROS and IOS bits in the TIMERx_CCHP register(when CHx_O and MCHx_O channels has separated deadtime value and break function, please refer to ***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 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 15-5. Complementary outputs controlled by parameters (MCHxMSEL =2'b11)***.

Table 15-5. Complementary outputs controlled by parameters (MCHxMSEL =2'b11)

Complementary Parameters					Output Status	
POEN &CHPOENx	ROS	IOS	ChxEN	MCHxEN	CHx_O	MCHx_O
0	0/1	0	0	0	CHx_O / CHx_ON = LOW CHx_O / CHx_ON output disable(1).	
				1	CHx_O/ CHx_ON output "off-state" (2): the CHx_O/ CHx_ON output inactive level firstly: CHx_O = CHxP, CHx_ON = CHxNP; If the clock for deadtime generator is present, after a deadtime: CHx_O = ISOx, CHx_ON = ISOxN. (3)	
			1	0		
				1		
		1	x	x	CHx_O/ CHx_ON output "off-state": the CHx_O/ CHx_ON output inactive level firstly: CHx_O = CHxP, CHx_ON = CHxNP; If the clock for deadtime generator is present, after a deadtime: CHx_O = ISOx, CHx_ON = ISOxN.	
1	0	0/1	0	0	CHx_O/MCHx_O = LOW CHx_O/MCHx_O output disable.	
				1	CHx_O = LOW CHx_O output disable.	MCHx_O=OxCPRE⊕(4)MCHxP MCHx_O output enable.
			1	0	CHx_O=OxCPRE⊕CHxP CHx_O output enable.	MCHx_O = LOW MCHx_O output disable.
				1	CHx_O=OxCPRE⊕CHxP CHx_O output enable.	MCHx_O = (!OxCPRE)(5)⊕MCHxP. MCHx_O output enable.
	1		0	0	CHx_O = CHxP CHx_O output "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		

				1	CHx_O=OxCPRE⊕CHxP CHx_O output enable	MCHx_O = (! OxCPRE)⊕MCHxP MCHx_O output enable.

Note:

(1) output disable: the CHx_O / CHx_ON are disconnected to corresponding pins, the pin is floating with GPIO pull up/down setting which will be Hi-Z if no pull.

(2) "off-state": CHx_O / CHx_ON output with inactive state (e.g., CHx_O = 0⊕CHxP = CHxP).

(3) See Break mode section for more details.

(4) ⊕: Xor calculate.

(5) (! OxCPRE)：the complementary output of the OxCPRE signal.

Dead time insertion

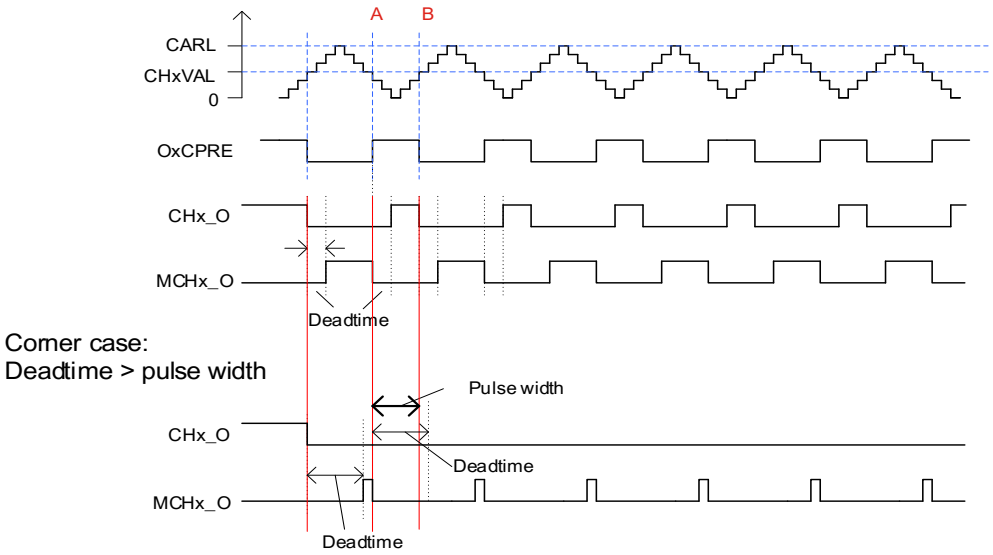
The dead time insertion is enabled when MCHxMSEL=2'b11 and both ChxEN and MCHxEN are configured to 1'b1, it is also necessary to configure POEN&CHPOENx to 1. The field named DTCFG defines the dead time delay that can be used for all channels. Refer to the ***Complementary channel protection register (TIMERx_CCHP)*** for details about the delay time.

The dead time delay insertion ensures that two complementary signals are not active at the same time.

When the channel x match event (TIMERx_CNT = CHxVAL) occurs, OxCPRE will be toggled in PWM mode 0. At point A in ***Figure 15-30. 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 657pprox. 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 15-30. Complementary output with dead time insertion***).

Figure 15-30. 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 CHx_ON will not be inserted into the dead-time.

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_CCHP register. The break input polarities are configured by the BRK0P and BRK1P bits in TIMERx_CCHP 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_CCHP register, ISOx and ISOxN bits in the TIMERx_CTL1 register.

The break event is the result of logic Ored of all sources. The break 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®-M7 LOCKUP_LOCK event or SRAM parity 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 15-31. BREAK0 function logic diagram* and *Figure 15-32. 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)* and *Trigger selection for TIMER7 BRKIN register (TRIGSEL_TIMER7BRKIN)*.

Figure 15-31. BREAK0 function logic diagram

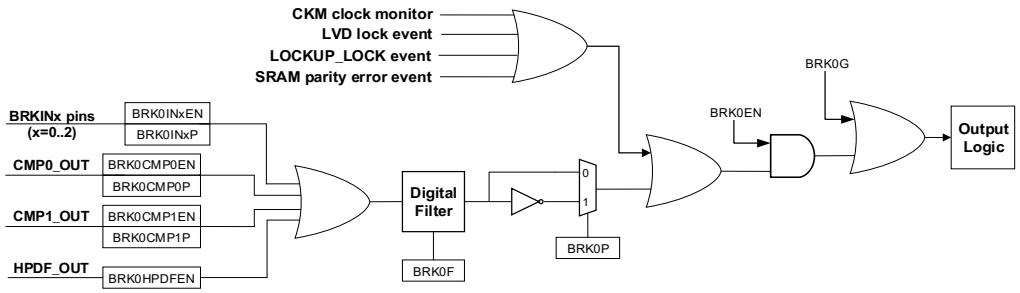
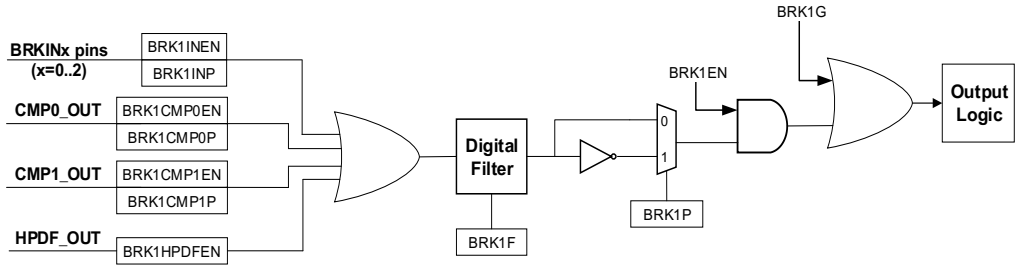


Figure 15-32. BREAK1 function logic diagram

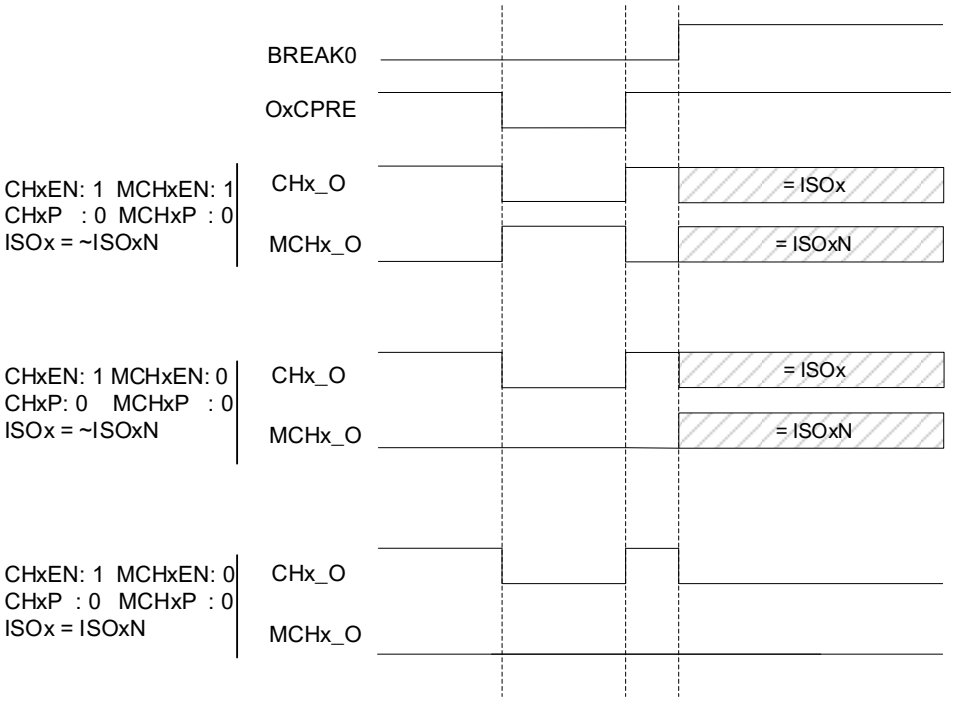


BREAK0 can be used to handle the faults of system sources, on-chip peripheral events and external sources. When a BREAK0 event occurs, the outputs are force at an inactive level, or at a predefined level (either active or inactive) after a deadtime duration. BREAK1 only can be used to handle the faults 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 15-33. 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 15-33. Output behavior of the channel in response to BREAK0 (the break input

high active and IOS=1)



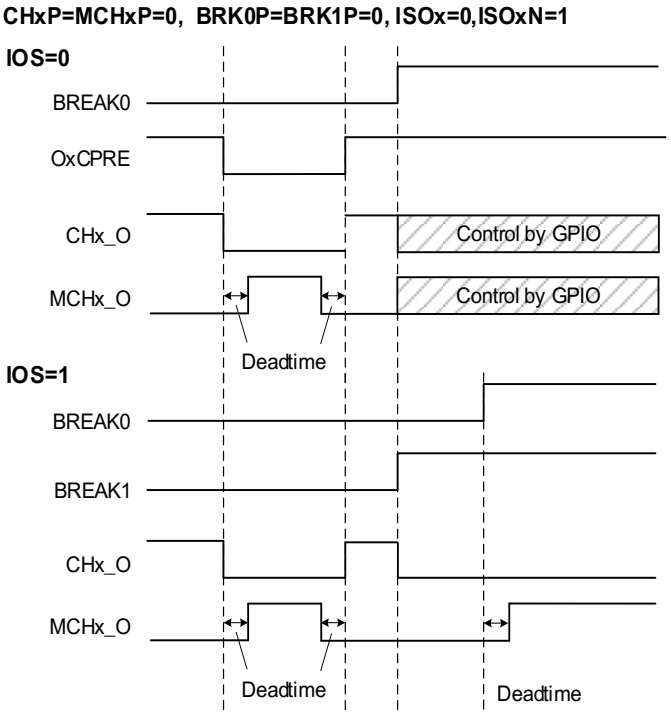
BREAK0 function has a higher priority than BREAK1 function. BREAK1 function only can be used when the IOS =1 and ROS =1.

Table 15-6. 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 BRK0IF/BRK1IF bit in the TIMERx_INTF register will be set. If BRKIE is 1, an interrupt will be generated.

Figure 15-34. 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 BEKENCHx (x=0...3) bit in the TIMERx_CTL2 register to realize the independent control of break function for each pair of channels. When the BEKENCHx(x=0...3) bit is "0" and a break event occurs, the corresponding channels CHx_O and MCHx_O will not be changed and the outputs is keeping.

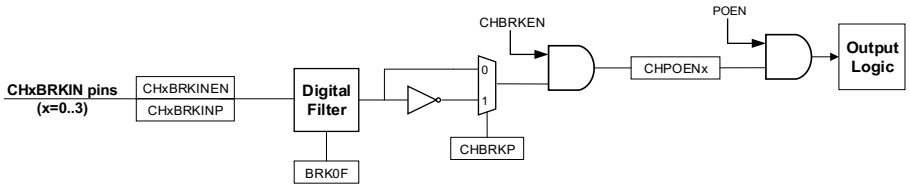
To enhance the flexibility of the break function, for the advanced timers, each pair of channels has a separate break input CHxBRKIN (x=0...3). When the corresponding CHxBRKIN (x=0...3) input is at an active level, it can asynchronously clear the corresponding CHPOENx (x=0...3) to 0, achieving individual break input control for the output of each pair of channels. The channel BREAK can be enabled by setting the CHBRKEN bit in the TIMERx_CCHP register. Additionally, the polarity of channel BREAK is configured by setting the CHBRKP bit in the TIMERx_CCHP register.

Furthermore, if the channel x BREAK is enabled, CHx_O and MCHx_O (x=0...3) are controlled not only by POEN but also by CHPOENx (x=0...3). The output of CHx and MCHx (x=0...3) after being broken is controlled by the following bits: the IOS and ROS bits in the TIMERx_FCCHPx (x=0...3) registers, and the ISOx and ISOxN bits in the TIMERx_CTL1 register. Each channel BREAK input has its own enable bit CHxBRKINEN (x=0...3) and polarity control bit CHxBRKINP (x=0...3), and shares the same filter configuration field BRK0F[3:0] with BREAK0. Please refer to ***Figure 15-35. Channel BREAK function logic diagram***.

When the channel x (x=0...3) BREAK input is active, the corresponding channel BREAK flag

bit CHxBRKIF (x=0...3) is set to 1. If the channel x (x=0...3) BREAK input is active for multiple periods, flag bit CHxBRKMIF (x=0...3) will be set to 1. The period counts can be determined by configuring the TIMERx_CHBRKPER register.

Figure 15-35. Channel BREAK function logic diagram



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_CCHP 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_CCHP 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 is 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 15-7. Break function input pins locked/ released conditions

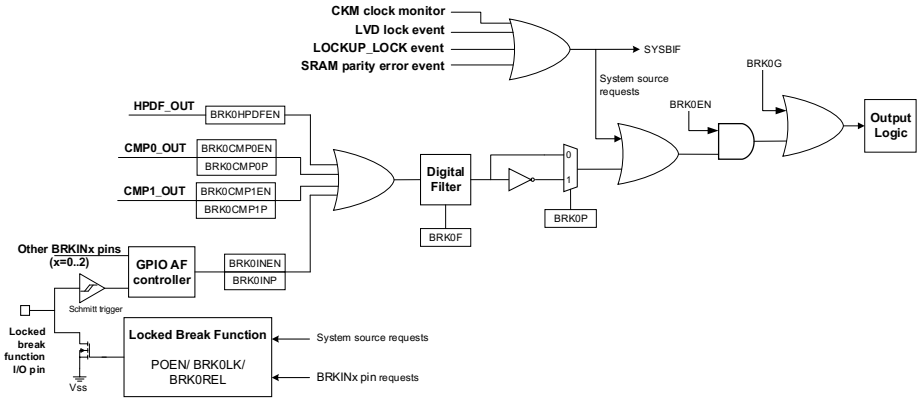
POEN	BRK0LK/ BRK1LK	BRK0REL/ BRK1REL	Break input pin state
0	1	0	Locked
	1	1	Released

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 15-36. 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_CCHP register is active; When the FCCHPyEN=1, the ROS、IOS and DTCFG[7:0] bits in TIMERx_FCCHP0 register is active.

Quadrature decoder

The quadrature decoder function uses two quadrature inputs 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 15-8. Counting direction in different quadrature decoder signals*. The CI0FE0 and CI1FE1 are the signals of the CI0 and CI1 after the filtering and polarity selection. The quadrature decoder can be regarded as an external clock with a direction selection. This means that the counter counts continuously from 0 to the counter-reload value. Therefore, users must configure the TIMERx_CAR register before the counter starts to count.

Table 15-8. Counting direction in different quadrature decoder signals

Counting mode	Level	CI0FE0		CI1FE1	
		Rising	Falling	Rising	Falling
Quadrature decoder mode 0 TSCFG0[4:0]!= 5'b00000	CI1FE1=1	Down	Up	-	-
	CI1FE1=0	Up	Down	-	-
Quadrature decoder mode 1 TSCFG1[4:0]!= 5'b00000	CI0FE0=1	-	-	Up	Down
	CI0FE0=0	-	-	Down	Up
Quadrature decoder mode 2 TSCFG2[4:0]!= 5'b00000	CI1FE1=1	Down	Up	X	X
	CI1FE1=0	Up	Down	X	X
	CI0FE0=1	X	X	Up	Down
	CI0FE0=0	X	X	Down	Up
Quadrature decoder mode 3 TSCFG13[4:0]!= 5'b00000	CI1FE1=1	Down	Up	-	-
	CI1FE1=0	-	-	-	-
Quadrature decoder mode 4 TSCFG14[4:0]!= 5'b00000	CI0FE0=1	-	-	Up	Down
	CI0FE0=0	-	-	-	-

Note: "-" means "no counting"; "X" means impossible. "0" means "low level", "1" means "high level".

Figure 15-37. Example of counter operation in decoder interface mode

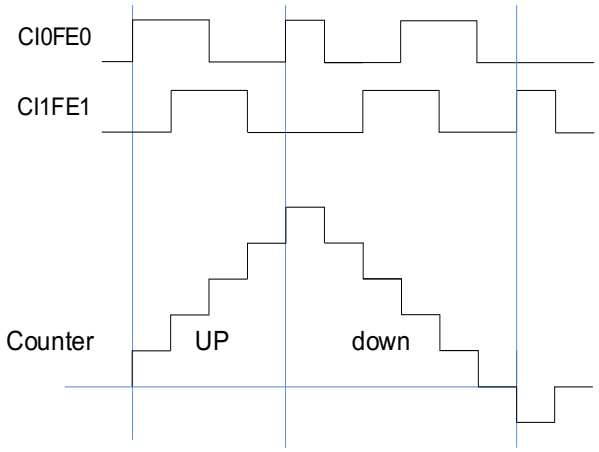
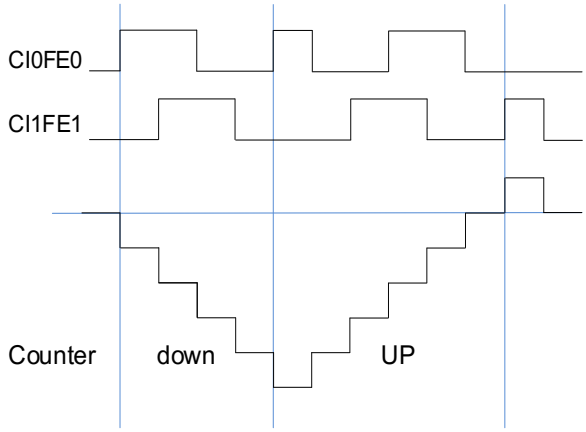


Figure 15-38. Example of decoder interface mode with CI0FE0 polarity inverted



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.

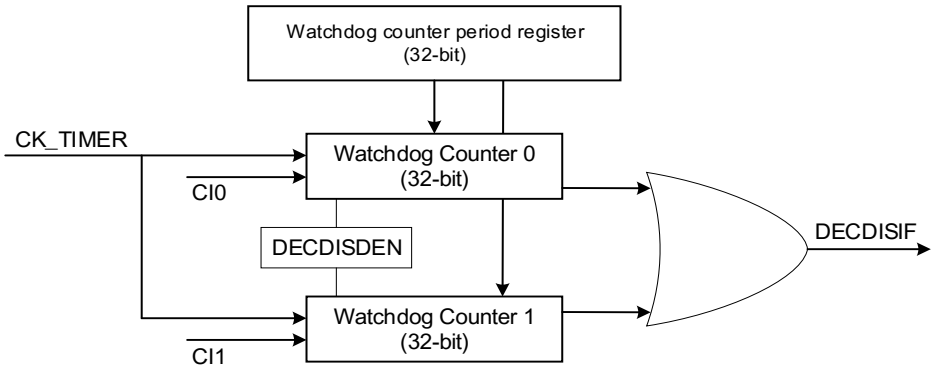
When DECJDEN=1, if the level transitions of the two quadrature signals CI0 and CI1 occur simultaneously, the interrupt flag DECJIF is set, if DECJIE=1, the corresponding interrupt is generated.

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 15-39. 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 15-39. Quadrature decoder signal disconnection detection block diagram



Non-quadrature decoder

The non-quadrature decoder function has four modes: non-quadrature decoder mode 0~3, which can be selected by setting TSCFGy[4:0] (y=9, 10, 11, 12) != 5'b00000.There are two input sources in these four modes: CI0 and CI1, and the CI0FE0 and CI1FE1 are the signals of the CI0 and CI1 after the filtering and polarity selection.

When the non-quadrature decoder mode 0 and non-quadrature decoder mode 1 are enabled, the CI0FE0 is used as the count direction signal and the CI1FE1 signal is used as the count pulse.

The CH0P is used to configure the count direction selection: When CH0P =0, the counter will count up when the CI0FE0 signal is high and will count down when the CI0FE0 signal is low; When CH0P =1, the counter will count down when the CI0FE0 signal is high and will count up when the CI0FE0 signal is low.

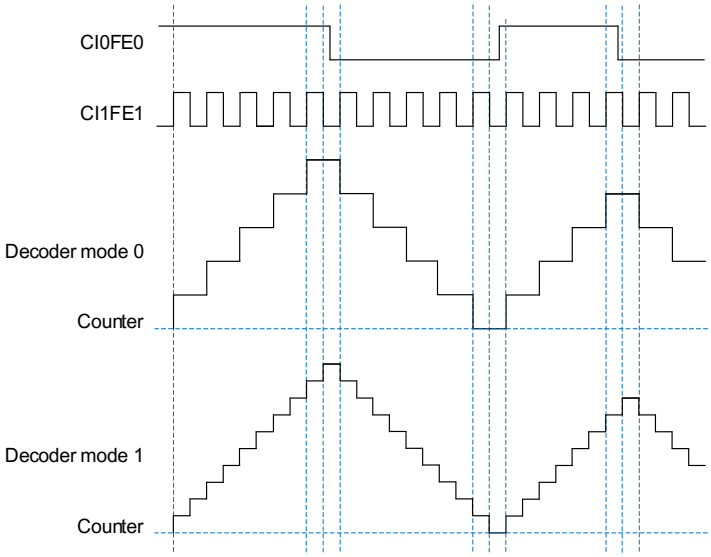
The CH1P is used to configure the count pulse edge selection: In non-quadrature decoder mode 0, the counter will count on both rising and falling edges of the CI1FE1 signal. In non-quadrature decoder mode 1, when CH1P=0, the counter will count on the rising edge of the CI1FE1 signal; When CH1P=1, the counter will count on the falling edge of the CI1FE1 signal. The more details for non-quadrature decoder mode 1 is shown in *Table 15-9.* **the counter operation in non-quadrature decoder mode 1** and ***Figure 15-40. Example of counter operation in non-quadrature decoder mode 0 / 1 with CH1P=0.***

Table 15-9. the counter operation in non-quadrature 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 15-40. Example of counter operation in non-quadrature decoder mode 0 / 1 with

CH1P=0



The non-quadrature decoder mode 2/3 uses two inputs CI0FE0 and CI1FE1 respectively to interact with each other to generate the counter value, and the counting direction DIR bit modified by hardware automatically.

When in non-quadrature decoder mode 2, the counter will count on both rising and falling edges of CI0FE0 and CI1FE1 signals. And the counting direction determined by the CH0P and CH1P bits.

When in non-quadrature decoder mode 3, the counter will count on rising or falling edge of CI0FE0 and CI1FE1 signals. When CHxP=0, the counter will counter on the high level or the falling edge or the CixFEx signal; When CHxP=1, the counter will counter on the low level or the rising edge or the CixFEx signal. The more details for non-quadrature decoder mode 2/3 is shown in *Figure 15-41. Example of counter operation in non-quadrature decoder mode 2 / 3 (CH0P / CH1P=0)* and *Table 15-10. the counter operation in non-quadrature decoder mode 2 / 3.*

Figure 15-41. Example of counter operation in non-quadrature decoder mode 2 / 3

(CH0P / CH1P=0)

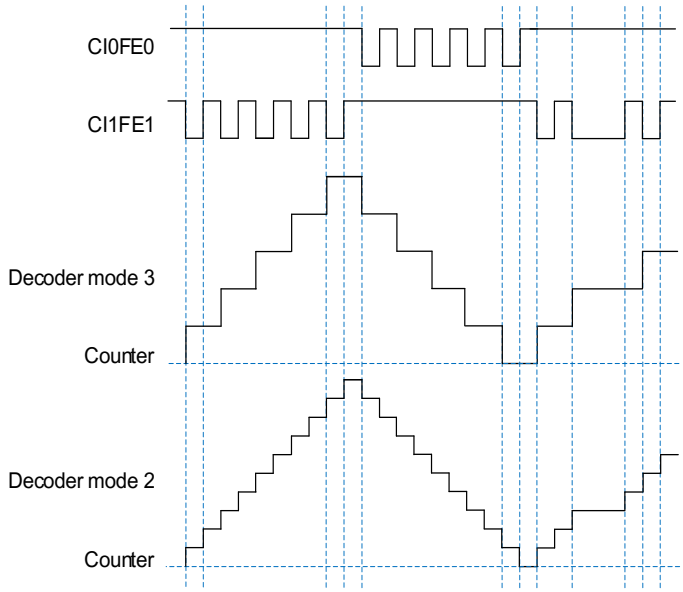


Table 15-10. the counter operation in non-quadrature decoder mode 2 / 3

Counting mode	Polarity	Level	CI0FE0		CI1FE1	
			Rising	Falling	Rising	Falling
non-quadrature decoder mode 2 TSCFG11[4:0] != 5'b00000	CHxP=0 (x=0 or 1)	CI1FE1=1	Down	Down	x	x
		CI1FE1=0	-	-	x	x
		CI0FE0=1	x	x	Up	Up
		CI0FE0=0	x	x	-	-
	CHxP=1 (x=0 or 1)	CI1FE1=1	-	-	x	x
		CI1FE1=0	Down	Down	x	x
		CI0FE0=1	x	x	-	-
		CI0FE0=0	x	x	Up	Up
non-quadrature decoder mode 3 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	-

Quadrature decoder and non-quadrature 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 non-quadrature decoder mode 0~3.

Hall sensor function

Hall sensor is generally used to control BLDC motor; the advanced timer supports this function.

Figure 15-42. 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 15-42. Hall sensor is used for BLDC motor

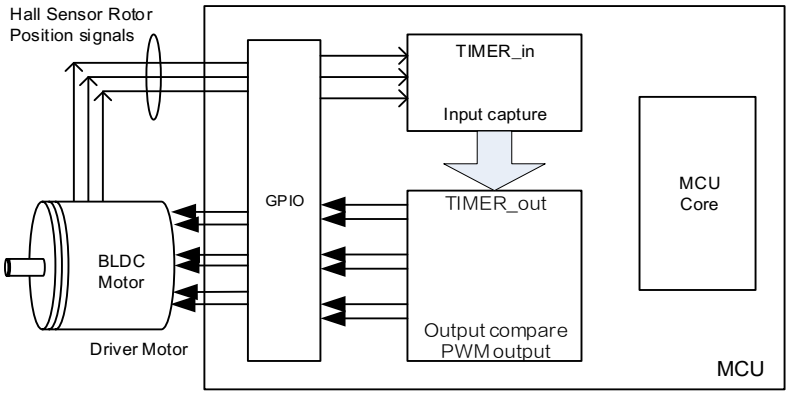
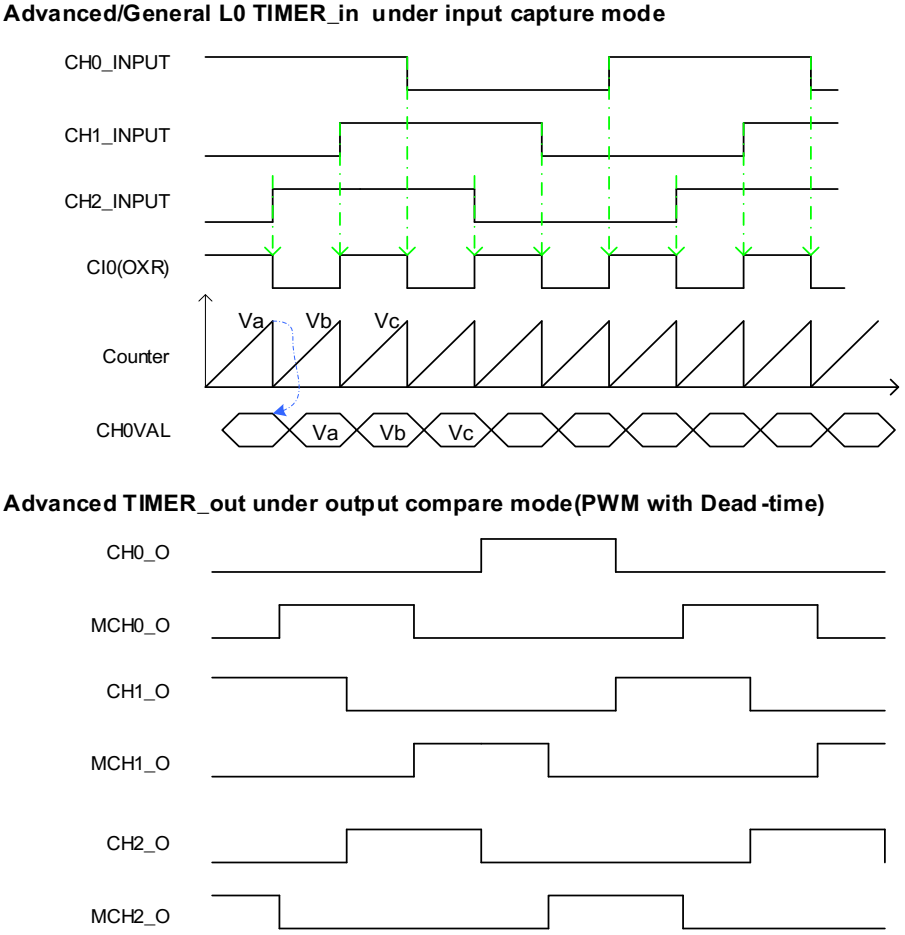


Figure 15-43. Hall sensor timing between two timers

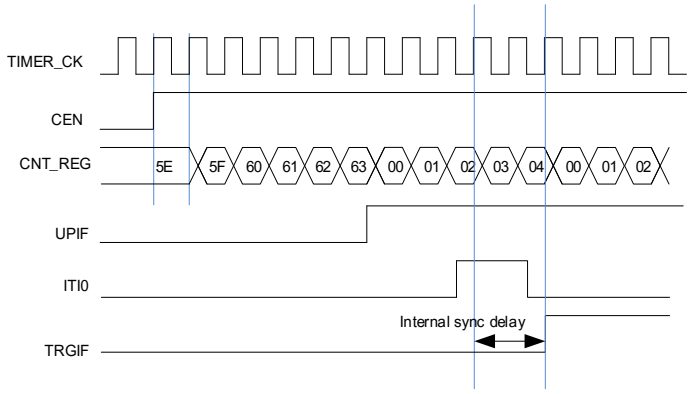
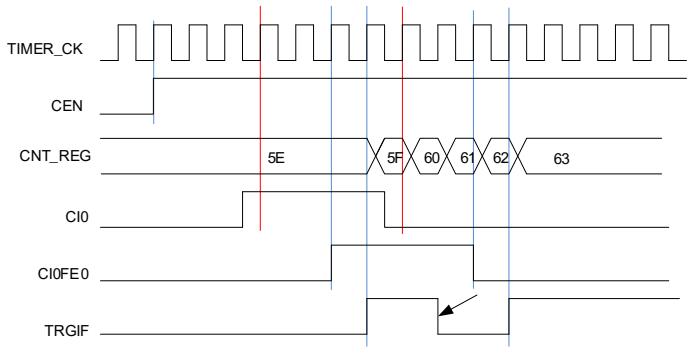


Master-slave management

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..7) in SYSCFG_TIMERxCFG(x=0,7).

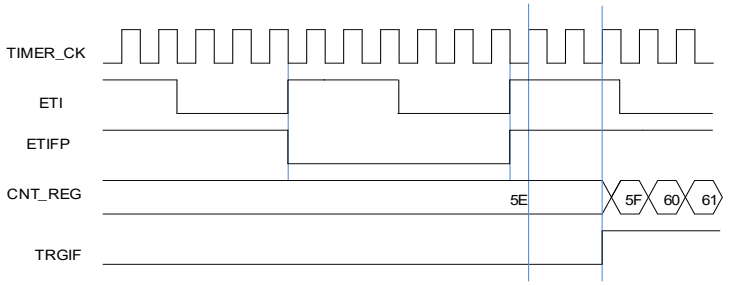
Table 15-11. Examples of slave mode

	Mode Selection	Source Selection	Polarity Selection	Filter and Prescaler
LIST	TSCFGy[4:0] y=3: restart mode y=4: pause mode y=5: event mode y=6: external clock mode 0 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	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 output of external	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
		01000: ETIFP(1) 01001: CI2FE2 01010: CI3FE3 01011: MCI0FEM0 01100: MCI1FEM1 01101: MCI2FEM2 01110: MCI3FEM3 10001: ITI12 10010: ITI13 10011: ITI14	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 15-44. 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.
	Figure 15-45. Pause mode 			

	Mode Selection	Source Selection	Polarity Selection	Filter and Prescaler
Exam3	**Event mode** The counter will start to count when a rising edge of trigger input comes.	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 15-46. Event mode



(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)* for more details.

Single pulse mode

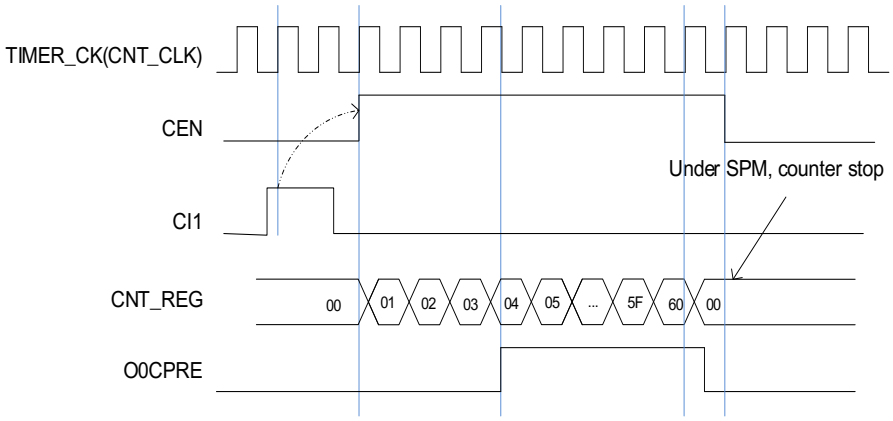
Single pulse mode is opposite to the repetitive mode, which can be enabled by setting SPM in TIMERx_CTL0. If SPM is set, the counter will be cleared and stopped automatically when the next update event occurs. In order to get a pulse waveform, the TIMERx is configured to PWM mode or compare mode by CHxCOMCTL 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) and asymmetric PWM mode.

Figure 15-47. 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) (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.

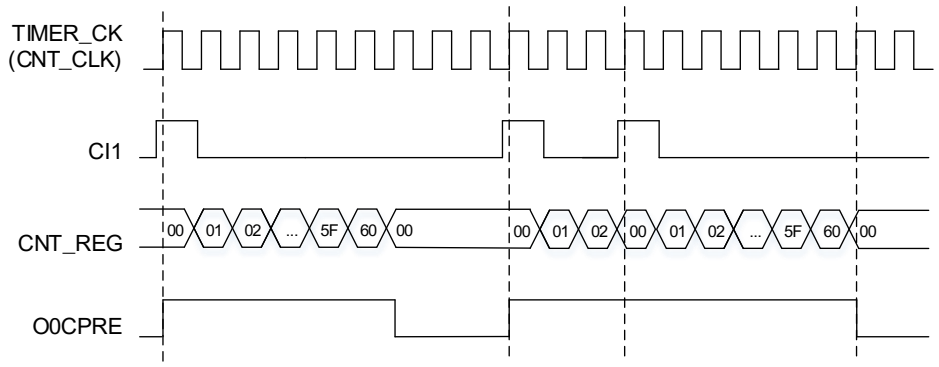
In delayable mode 1. The behavior of OxCPRE/ MOxCPRE is performed as in PWM mode 1. When counting up, the OxCPRE/ MOxCPRE is inactive, when a trigger event occurs, the OxCPRE/ MOxCPRE is active. The OxCPRE/ MOxCPRE is inactive again at the next update event; When counting down, the OxCPRE/ MOxCPRE is active. When a trigger event occurs, the OxCPRE/ MOxCPRE is inactive. The OxCPRE/ MOxCPRE is active again at the next update event.

Note:

1) The center-aligned counting mode cannot be used in this mode and the CAM[1:0] = 2'b00(in TIMERx_CTL0 register);
2) 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 15-48. delayable single pulse mode with TIMERx_CHxCV=0x00, TIMERx_CAR=0x60



Timers interconnection

The timers can be internally connected for timer chaining or synchronization. This can be implemented by configuring one timer to operate in the master mode while configuring another timer to be in the slave mode. The following figures show several examples of trigger selection for the master mode and slave mode.

Some interconnection examples:

■ TIMER2 as the prescaler for TIMER0

TIMER2 is configured as a prescaler for TIMER0, steps are shown as follows:

1. Configure TIMER2 in master mode and select its update event (UPE) as trigger output (MMC0=4'b0010 in the TIMER2_CTL1 register). Then TIMER2 drives a periodic signal on each counter overflow.
2. Configure TIMER2 period (TIMER2_CAR register).
3. Configure TIMER0 in external clock mode 0 and select the TIMER2 as TIMER0 input trigger source (TSCFG6[4:0] = 5'b00011 in the SYSCFG_TIMER0CFG1 register).
4. Start TIMER0 by writing '1' to the CEN bit (TIMER0_CTL0 register).
5. Start TIMER2 by writing '1' to the CEN bit (TIMER2_CTL0 register).

■ Start TIMER0 with TIMER2's enable/update signal

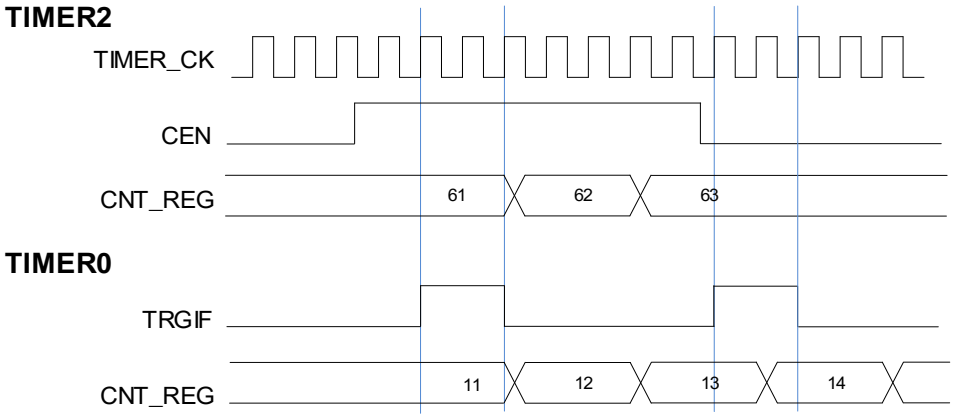
First, enable TIMER0 with the enable signal of TIMER2. Refer to ***Figure 15-49. Trigger mode of TIMER0 controlled by enable signal of TIMER2***. TIMER0 starts counting from its current value with the divided internal clock after being triggered by TIMER2 enable signal output.

When TIMER0 receives the trigger signal, its CEN bit is set automatically and the counter counts until TIMER0 is disabled. Both clock frequency of the counters is divided by 3 from TIMER_CK ($f_{PSC_CLK} = f_{TIMER_CK}/3$). Steps are shown as follows:

1. Configure TIMER2 in master mode to send its enable signal as trigger output (MMC0=4'b0001 in the TIMER2_CTL1 register).

2. Configure TIMER0 in event mode and select the TIMER2 as TIMER0 input trigger source (TSCFG5[4:0] = 5'b00011 in the_SYSCFG_TIMER0CFG0 register).
3. Start TIMER2 by writing 1 to the CEN bit (TIMER2_CTL0 register).

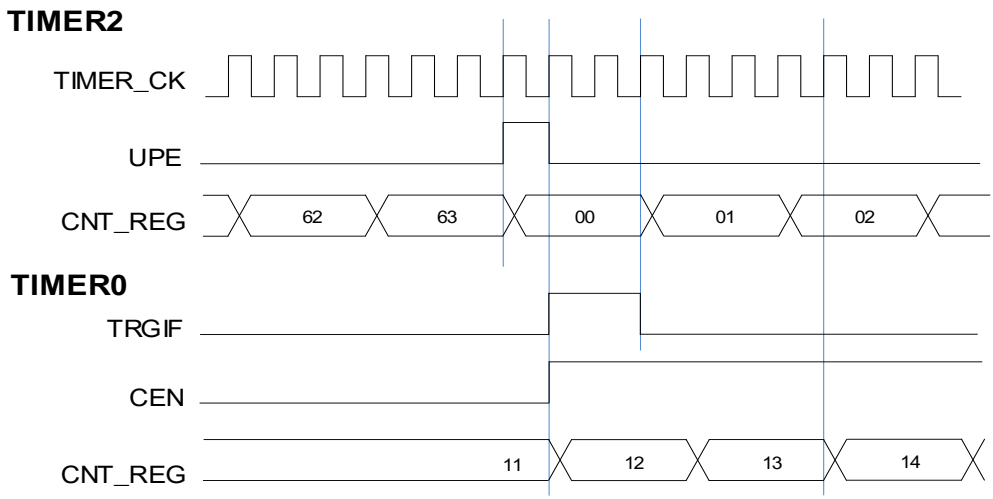
Figure 15-49. Trigger mode of TIMER0 controlled by enable signal of TIMER2



In this example, the update event can also be used as trigger source instead of enable signal. Refer to ***Figure 15-50. Trigger mode of TIMER0 controlled by update signal of TIMER2***. Steps are shown as follows:

1. Configure TIMER2 in master mode to send its update event (UPE) as trigger output (MMC0=4'b0010 in the TIMER2_CTL1 register).
2. Configure the TIMER2 period (TIMER2_CARL registers).
3. Configure TIMER0 in event mode and select the TIMER2 as TIMER0 input trigger source (TSCFG5[4:0] = 5'b00011 in the_SYSCFG_TIMER0CFG0 register).
4. Start TIMER2 by writing '1' to the CEN bit (TIMER2_CTL0 register).

Figure 15-50. Trigger mode of TIMER0 controlled by update signal of TIMER2



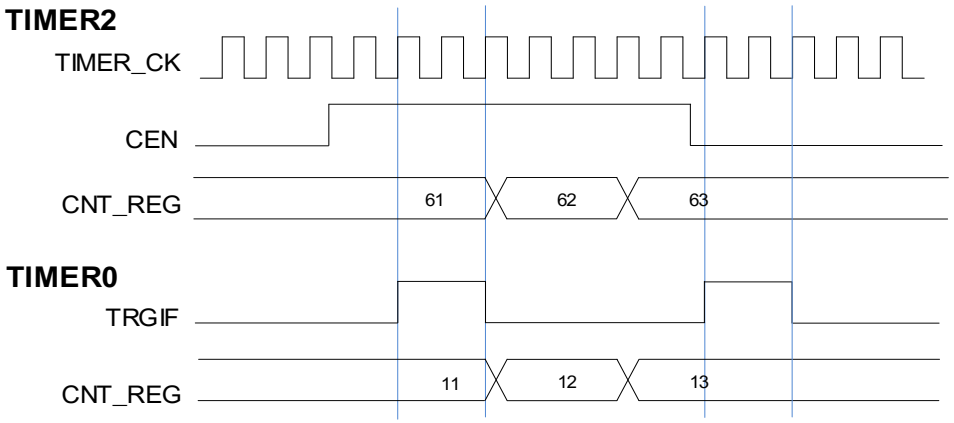
■ Enable TIMER0 to count with the enable/O0CPRE signal of TIMER2.

In this example, TIMER0 is enabled with the enable signal of TIMER2. Refer to ***Figure 15-51. Pause mode of TIMER0 controlled by enable signal of TIMER2***. TIMER0 counts with the

divided internal clock only when TIMER2 is enabled. Both clock frequency of the counters is divided by 3 from TIMER_CK ($f_{PSC_CLK} = f_{TIMER_CK}/3$). Steps are shown as follows:

1. Configure TIMER2 in master mode and output enable signal as trigger output (MMC0=4'b0001 in the TIMER2_CTL1 register).
2. Configure TIMER0 in pause mode and select the TIMER2 as TIMER0 input trigger source (TSCFG4[4:0] = 5'b00011 in the_SYSCFG_TIMER0CFG0 register).
3. Enable TIMER0 by writing '1' to the CEN bit (TIMER0_CTL0 register).
4. Start TIMER2 by writing '1' to the CEN bit (TIMER2_CTL0 register).
5. Stop TIMER2 by writing '0' to the CEN bit (TIMER2_CTL0 register).

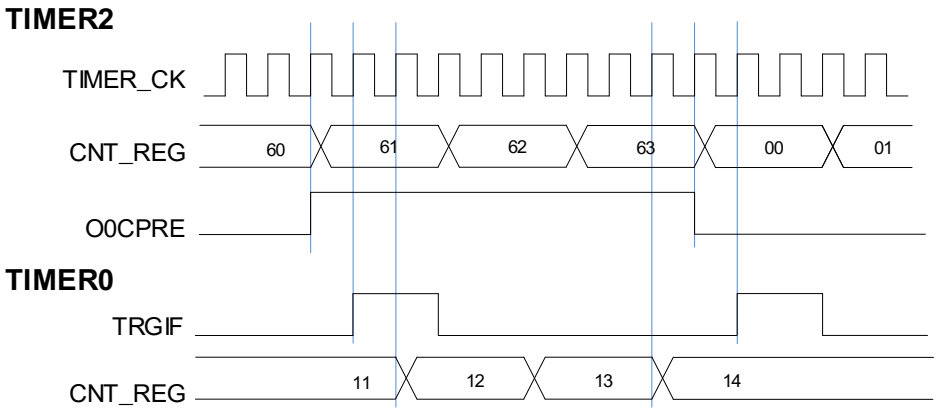
Figure 15-51. Pause mode of TIMER0 controlled by enable signal of TIMER2



In this example, O0CPRE can also be used as trigger source instead of enable signal output. Steps are shown as follows:

1. Configure TIMER2 in master mode and O0CPRE as trigger output (MMS=3'b100 in the TIMER2_CTL1 register).
2. Configure the TIMER2 O0CPRE waveform (TIMER2_CHCTL0 register).
3. Configure TIMER0 in pause mode and select the TIMER2 as TIMER0 input trigger source (TSCFG4[4:0] = 5'b00011 in the_SYSCFG_TIMER0CFG0 register).
4. Enable TIMER0 by writing '1' to the CEN bit (TIMER0_CTL0 register).
5. Start TIMER2 by writing '1' to the CEN bit (TIMER2_CTL0 register).

Figure 15-52. Pause mode of TIMER0 controlled by O0CPRE signal of TIMER2



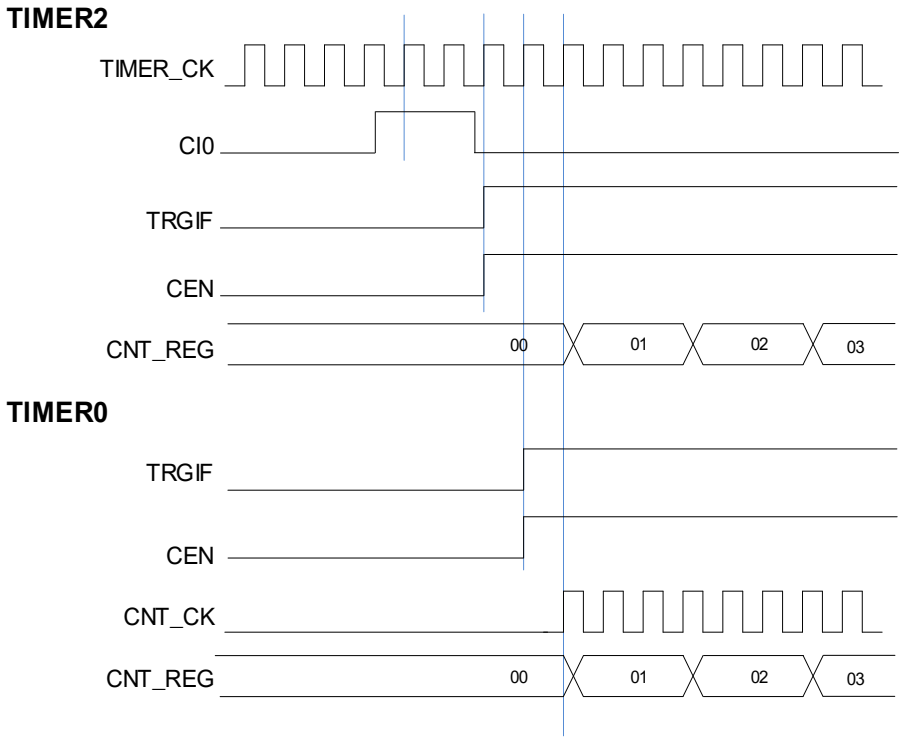
■ Using an external trigger to start two timers synchronously.

The start of TIMER0 is triggered by the enable signal of TIMER2, and TIMER2 is triggered by its CI0 input rising edge. To ensure that two timers start synchronously, TIMER2 must be configured in master/slave mode. Steps are shown as follows:

1. Configure TIMER2 in event mode and select the CI0F_ED as TIMER2 input trigger source (TSCFG5[4:0] = 5'b00101 in the_SYSCFG_TIMER2CFG0 register).
2. Configure TIMER2 in master/slave mode by writing MSM=1 (TIMER2_SMCFG register).
3. Configure TIMER0 in event mode and select the TIMER2 as TIMER0 input trigger source (TSCFG5[4:0] = 5'b00011 in the_SYSCFG_TIMER0CFG0 register).

When the CI0 signal of TIMER2 generates a rising edge, two timer counters start counting synchronously with the internal clock and both TRGIF flags are set.

Figure 15-53. Trigger TIMER0 and TIMER2 by the CI0 signal of TIMER2



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.

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®-M7 is halted, and the TIMERx_HOLD configuration bit in DBG register is set to 1, the TIMERx counter stops.

15.1.5. Registers definition (TIMERx, x=0, 7)

TIMER0 base address: 0x4001 0000

TIMER7 base address: 0x4001 0400

Control register 0 (TIMERx_CTL0)

Address offset: 0x00
Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				UPIFBUEN	Reserved	CKDIV[1:0]		ARSE	CAM[1:0]		DIR	SPM	UPS	UPDIS	CEN
				rw		rw		rw	rw		rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
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 UPG bit is set or the slave mode controller generates a hardware reset event.
0	CEN	Counter enable 0: Counter disable 1: Counter enable The CEN bit must be set by software when timer works in external clock mode,

pause mode or decoder mode. While in event mode, the hardware can set the CEN bit automatically.

Control register 1 (TIMERx_CTL1)

Address offset: 0x04

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCUC[2:1]		Reserved				MMC0[3]	Reserved		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&CHPOENx bit is reset, MCH0_O is set low. 1: When POEN&CHPOENx bit is reset, MCH0_O is set high. This bit can be modified only when PROT[1:0] bits in TIMERx_CCHP register is 00.
8	ISO0	Idle state of channel 0 output 0: When POEN&CHPOENx bit is reset, CH0_O is set low. 1: When POEN&CHPOENx bit is reset, CH0_O is set high. The CH0_O output changes after a dead time if MCH0_O is implemented. This bit can be modified only when PROT[1:0] bits in TIMERx_CCHP register is 00.
7	TI0S	Channel 0 trigger input selection 0: The TIMERx_CH0 pin input is selected as channel 0 trigger input. 1: The result of combinational XOR of TIMERx_CH0, TIMERx_CH1 and TIMERx_CH2 pins is selected as channel 0 trigger input.
6:4	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. 0011: 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 non-quadrature decoder mode 0~3. 1001~1111: 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 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, CHxCOMCTL, CHxP, MCHxP and MCHxCOMCTL bits) are disabled. 1: The shadow registers (for ChxEN, MCHxEN, CHxCOMCTL, CHxP, MCHxP and MCHxCOMCTL bits) are enabled. After these bits have been written, they are updated when commutation event comes. When a channel does not have a complementary output, this bit has no effect.

Slave mode configuration register (TIMERx_SMCFG)

Address offset: 0x08

Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	SMC1	ETPSC[1:0]		ETFC[3:0]				MSM	Reserved						
rw	rw	rw		rw				rw							

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	ETP	External trigger polarity This bit specifies the polarity of ETI signal. 0: ETI is active at high level or rising edge. 1: ETI is active at low level or falling edge.
14	SMC1	Part of slave mode controller is used to enable external clock mode 1 In external clock mode 1, the counter is clocked by any active edge of the ETIFP signal. 0: External clock mode 1 disabled 1: External clock mode 1 enabled It is possible to simultaneously use external clock mode 1 with the restart mode, pause mode or event mode. But the TSCFGy[4:0](y=3,4,5,7) bits must not be 5'b01000 in this case. The external clock input will be ETIFP if external clock mode 0 and external clock mode 1 are enabled at the same time. **Note:** External clock mode 0 enable is in TSCFG6[4:0] bit-field in SYSCFG_TIMERxCFG1 register.

13:12	ETPSC[1:0]	External trigger prescaler The frequency of external trigger signal ETIFP must not be higher than 1/4 of TIMER_CK frequency. When the frequency of external trigger signal is high, the prescaler can be enabled to reduce ETIFP frequency. 00: Prescaler disabled 01: ETIFP frequency divided by 2 10: ETIFP frequency divided by 4 11: ETIFP frequency divided by 8
11:8	ETFC[3:0]	External trigger filter control An event counter is used in the digital filter, in which a transition on the output occurs after N input events. This bit-field specifies the frequency used to sample ETIFP signal and the length of the digital filter applied to ETIFP. 0000: Filter disabled. F_{SAMP}= f_{DTS}, N=1. 0001: f_{SAMP}= f_{CK_TIMER}, N=2. 0010: f_{SAMP}= f_{CK_TIMER}, N=4. 0011: f_{SAMP}= f_{CK_TIMER}, N=8. 0100: f_{SAMP}=f_{DTS}/2, N=6. 0101: f_{SAMP}=f_{DTS}/2, N=8. 0110: f_{SAMP}=f_{DTS}/4, N=6. 0111: f_{SAMP}=f_{DTS}/4, N=8. 1000: f_{SAMP}=f_{DTS}/8, N=6. 1001: f_{SAMP}=f_{DTS}/8, N=8. 1010: f_{SAMP}=f_{DTS}/16, N=5. 1011: f_{SAMP}=f_{DTS}/16, N=6. 1100: f_{SAMP}=f_{DTS}/16, N=8. 1101: f_{SAMP}=f_{DTS}/32, N=5. 1110: f_{SAMP}=f_{DTS}/32, N=6. 1111: f_{SAMP}=f_{DTS}/32, N=8.
7	MSM	Master-slave mode This bit can be used to synchronize the selected timers to begin counting at the same time. The TRGI is used as the start event, and through TRGO, timers are connected. 0: Master-slave mode disabled 1: Master-slave mode enabled
6:0	Reserved	Must be kept at reset value.

DMA and interrupt enable register (TIMERx_DMAINTEN)

Address offset: 0x0C
Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH3COMADDIE	CH2COMADDIE	CH1COMADDIE	CH0COMADDIE	MCH3DEN	MCH2DEN	MCH1DEN	MCH0DEN	MCH3IE	MCH2IE	MCH1IE	MCH0IE	Reserved		DECDISIE	DECJIE
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	TRGDEN	CMTDEN	CH3DEN	CH2DEN	CH1DEN	CH0DEN	UPDEN	BRKIE	TRGIE	CMTIE	CH3IE	CH2IE	CH1IE	CH0IE	UPIE
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31	CH3COMADDIE	Channel 3 additional compare interrupt enable 0: Disabled 1: Enabled **Note:** This bit just used in composite PWM mode and asymmetric PWM mode.
30	CH2COMADDIE	Channel 2 additional compare interrupt enable 0: Disabled 1: Enabled **Note:** This bit just used in composite PWM mode and asymmetric PWM mode.
29	CH1COMADDIE	Channel 1 additional compare interrupt enable 0: Disabled 1: Enabled **Note:** This bit just used in composite PWM mode and asymmetric PWM mode.
28	CH0COMADDIE	Channel 0 additional compare interrupt enable 0: Disabled 1: Enabled **Note:** This bit just used in composite PWM mode and asymmetric 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] = 2b'00).
26	MCH2DEN	Multi mode channel 2 capture/compare DMA request enable 0: Disabled 1: Enabled **Note:** This bit just used for channel input and output independent mode (when MCH2MSEL[1:0] = 2b'00).
25	MCH1DEN	Multi mode channel 1 capture/compare DMA request enable 0: Disabled 1: Enabled **Note:** This bit just used for channel input and output independent mode (when MCH1SEL[1:0] = 2b'00).

24	MCH0DEN	Multi mode channel 0 capture/compare DMA request enable 0: Disabled 1: Enabled **Note:** This bit just used for channel input and output independent mode (when MCH0SEL[1:0] = 2b'00).
23	MCH3IE	Multi mode channel 3 capture/compare interrupt enable 0: Disabled 1: Enabled **Note:** This bit just used for channel input and output independent mode (when MCH3MSEL[1:0] = 2b'00).
22	MCH2IE	Multi mode channel 2 capture/compare interrupt enable 0: Disabled 1: Enabled **Note:** This bit just used for channel input and output independent mode (when MCH2MSEL[1:0] = 2b'00).
21	MCH1IE	Multi mode channel 1 capture/compare interrupt enable 0: Disabled 1: Enabled **Note:** This bit just used for channel input and output independent mode (when MCH1SEL[1:0] = 2b'00).
20	MCH0IE	Multi mode channel 0 capture/compare interrupt enable 0: Disabled 1: Enabled **Note:** This bit just used for channel input and output independent mode (when MCH0SEL[1:0] = 2b'00).
19:18	Reserved	Must be kept at reset value.
17	DECDISIE	Quadrature decoder signal disconnection interrupt enable 0: Disabled 1: Enabled **Note:** This bit just used for quadrature decoder signal disconnection detection is enabled (when DECDISDEN =1).
16	DECJIE	Quadrature decoder signal jump (the two signals jump at the same time) interrupt enable 0: Disabled 1: Enabled **Note:** This bit just used for quadrature decoder signal jump detection is enabled (when DECJDEN =1).
15	Reserved	Must be kept at reset value.
14	TRGDEN	Trigger DMA request enable

		0: Disabled 1: Enabled
13	CMTDEN	Commutation DMA request enable 0: Disabled 1: Enabled
12	CH3DEN	Channel 3 capture/compare DMA request enable 0: Disabled 1: Enabled
11	CH2DEN	Channel 2 capture/compare DMA request enable 0: Disabled 1: Enabled
10	CH1DEN	Channel 1 capture/compare DMA request enable 0: Disabled 1: Enabled
9	CH0DEN	Channel 0 capture/compare DMA request enable 0: Disabled 1: Enabled
8	UPDEN	Update DMA request enable 0: Disabled 1: Enabled
7	BRKIE	Break interrupt enable 0: Disabled 1: Enabled
6	TRGIE	Trigger interrupt enable 0: Disabled 1: Enabled
5	CMTIE	Commutation interrupt enable 0: Disabled 1: Enabled
4	CH3IE	Channel 3 capture/compare interrupt enable 0: Disabled 1: Enabled
3	CH2IE	Channel 2 capture/compare interrupt enable 0: Disabled 1: Enabled
2	CH1IE	Channel 1 capture/compare interrupt enable 0: Disabled

		1: Enabled
1	CH0IE	Channel 0 capture/compare interrupt enable 0: Disabled 1: Enabled
0	UPIE	Update interrupt enable 0: Disabled 1: Enabled

Interrupt flag register (TIMERx_INTF)

Address offset: 0x10

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH3COMADDIF	CH2COMADDIF	CH1COMADDIF	CH0COMADDIF	MCH3OF	MCH2OF	MCH1OF	MCH0OF	MCH3IF	MCH2IF	MCH1IF	MCH0IF	Reserved		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
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
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

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 and asymmetric 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
19:18	Reserved	Must be kept at reset value.
17	DECDISIF	Quadrature decoder signal disconnection interrupt flag 0: No quadrature decoder signal disconnection interrupt occurred 1: Quadrature decoder signal disconnection interrupt occurred **Note:** This bit just used for quadrature decoder signal disconnection detection is enabled (when DECDISDEN =1).
16	DECJIF	Quadrature decoder signal jump (the two signals jump at the same time) interrupt flag 0: No quadrature decoder signal jump interrupt occurred 1: Quadrature decoder signal jump interrupt occurred **Note:** This bit just used for quadrature decoder signal jump detection is enabled (when DECJDEN =1).
15:14	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 and DMA 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 and asymmetric 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

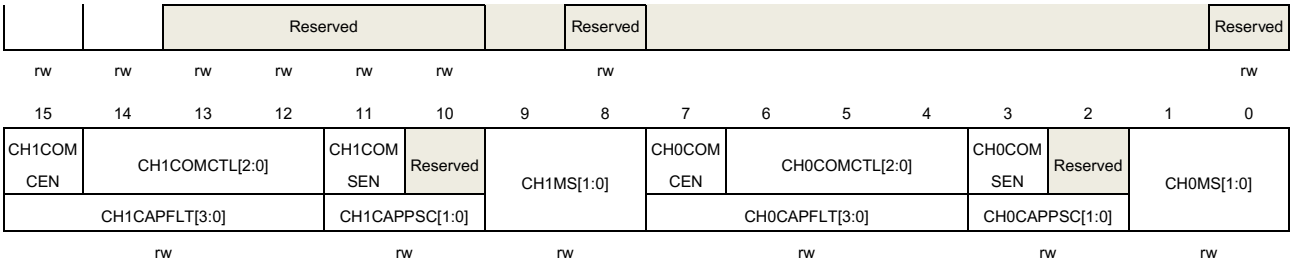
0 UPG Update event generation
This bit can be set by software, and automatically cleared by hardware. When this bit is set, the counter is cleared if the center-aligned or up counting mode is selected, while in down counting mode it takes the auto-reload value. The prescaler counter is cleared at the 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]	CH1COM ADDSEN	CH0COM ADDSEN	CH1ADDU PS	CH0ADDU PS	Reserved	CH1COM CTL[3]	Reserved							CH0COM CTL[3]



Output compare mode:

Bits	Fields	Descriptions
31	CH1MS[2]	Channel 1 I/O mode selection Refer to CH1MS[1:0]description
30	CH0MS[2]	Channel 0 I/O mode selection Refer to CH0MS[1:0] description
29	CH1COMADDSEN	Channel 1 additional compare output shadow enable Refer to CH0COMADDSEN description.
28	CH0COMADDSEN	Channel 0 additional compare output shadow enable When this bit is set, the shadow register of TIMERx_CH0COMV_ADD register which updates at each update event will be enabled. 0: Channel 0 additional compare output shadow disabled 1: Channel 0 additional compare output shadow enabled The PWM mode can be used without validating the shadow register only in single pulse mode (SPM bit in TIMERx_CTL0 register is set). This bit cannot be modified when PROT[1:0] bit-field in TIMERx_CCHP register is 11 and CH0MS bit-field is 000.
27	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	Reserved	Must be kept at reset value.
9:8	CH1MS[1:0]	Channel 1 mode selection This bit-field specifies the direction of the channel and the input signal selection. The CH1MS[2:0] bit-field is writable only when the channel is not active (When MCH1MSEL[1:0] = 2b'00, the CH1EN bit in TIMERx_CHCTL2 register is reset; when MCH1MSEL[1:0] = 2b'11, 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) 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] = 2b'11, This bit-field controls the behavior of O0CPRE which drives CH0_O and MCH0_O. The active level of O0CPRE is high, while the active level of CH0_O and MCH0_O depends on CH0P and MCH0P bits. 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: Asymmetric PWM mode 0. Only for Center aligned mode, when counting up, the channel output is inactive when the counter is matched with the value of TIMERx_CH0CV, when counting down, the channel output is active when the counter is matched with the value of TIMERx_CH0CV_ADD.

1011: Asymmetric PWM mode 1. Only for Center aligned mode, when counting up, the channel output is active when the counter is matched with the value of TIMERx_CH0CV, when counting down, the channel output is inactive when the counter is matched with the value of TIMERx_CH0CV_ADD.

1100~1111: Reserved.

Note: In the composite PWM mode (CH0CPWMEN = 1'b1 and CH0MS = 3'b000) or asymmetric PWM mode, the PWM signal output in channel 0 is composited by TIMERx_CH0CV and TIMERx_CH0COMV_ADD. Please refer to ***Composite PWM mode*** and ***Asymmetric PWM mode*** for more details.

If configured in PWM mode (PWM mode 0/1、Composite PWM mode0/1、Asymmetric PWM mode0/1), 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_CCHP register is 11 and CH0MS bit-field is 000 (compare mode).

3	CH0COMSEN	Channel 0 compare output shadow enable

		When this bit is set, the shadow register of TIMERx_CH0CV register which updates at each update event will be enabled. 0: Channel 0 output compare shadow disabled 1: Channel 0 output compare shadow enabled The PWM mode can be used without validating the shadow register only in single pulse mode (SPM bit in TIMERx_CTL0 register is set). This bit cannot be modified when PROT[1:0] bit-field in TIMERx_CCHP register is 11 and CH0MS bit-field is 000.
2	Reserved	Must be kept at reset value.
1:0	CH0MS[1:0]	Channel 0 I/O mode selection This bit-field specifies the work mode of the channel and the input signal selection. The CH0MS[2:0] bit-field is writable only when the channel is not active (When MCH0MSEL[1:0] = 2b'00, the CH1EN bit in TIMERx_CHCTL2 register is reset; when MCH0MSEL[1:0] = 2b'11, 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=0,7) 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]	CH3COM ADDSEN	CH2COM ADDSEN	CH3ADDU PS	CH2ADDU PS	Reserved	CH3COM CTL[3]								CH2COM CTL[3]
		Reserved					Reserved								Reserved
rw	rw	rw	rw	rw	rw		rw								rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3COM CEN	CH3COMCTL[2:0]			CH3COM SEN	Reserved	CH3MS[1:0]		CH2COM CEN	CH2COMCTL[2:0]			CH2COM SEN	Reserved	CH2MS[1:0]	
CH3CAPFLT[3:0]				CH3CAPPSC[1:0]				CH2CAPFLT[3:0]				CH2CAPPSC[1:0]			
rw				rw		rw		rw				rw		rw	

Output compare mode:

Bits	Fields	Descriptions
31	CH3MS[2]	Channel 3 I/O mode selection Refer to CH3MS[1:0]description.
30	CH2MS[2]	Channel 2 I/O mode selection Refer to CH2MS[1:0] description.
29	CH3COMADDSEN	Channel 3 additional compare output shadow enable Refer to CH2COMADDSEN description.
28	CH2COMADDSEN	Channel 2 additional compare output shadow enable When this bit is set, the shadow register of TIMERx_CH2COMV_ADD register which updates at each update event will be enabled. 0: Channel 2 additional compare shadow disabled 1: Channel 2 additional compare shadow enabled The PWM mode can be used without validating the shadow register only in single pulse mode (SPM bit in TIMERx_CTL0 register is set). This bit cannot be modified when PROT[1:0] bit-field in TIMERx_CCHP register is 11 and CH2MS bit-field is 000.
27	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	Reserved	Must be kept at reset value.
9:8	CH3MS[1:0]	Channel 3 I/O mode selection This bit-field specifies the direction of the channel and the input signal selection. The CH3MS[2:0] bit-field is writable only when the channel is not active (When MCH3MSEL[1:0] = 2b'00, the CH3EN bit in TIMERx_CHCTL2 register is reset; when MCH3MSEL[1:0] = 2b'11, 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(x=0,7) 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] = 2b'11, This bit-field controls the behavior of O2CPRE which drives CH2_O and MCH2_O. The active level of O2CPRE is high, while the active level of CH2_O and MCH2_O depends on CH2P and MCH2P bits. 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: Asymmetric PWM mode 0. Only for Center aligned mode, when counting up, the channel output is inactive when the counter is matched with the value of TIMERx_CH2CV, when counting down, the channel output is active when the counter is matched with the value of TIMERx_CH2CV_ADD.

1011: Asymmetric PWM mode 1. Only for Center aligned mode, when counting up, the channel output is active when the counter is matched with the value of TIMERx_CH2CV, when counting down, the channel output is inactive when the counter is matched with the value of TIMERx_CH2CV_ADD.

1100~1111: Reserved.

Note: In the composite PWM mode (CH2CPWMEN = 1'b1 and CH2MS = 3'b000) or asymmetric PWM mode, the PWM signal output in channel 2 is composited by TIMERx_CH2CV and TIMERx_CH2COMV_ADD. Please refer to ***Composite PWM mode*** and ***Asymmetric 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_CCHP register is 11 and CH2MS bit-field is 000(compare mode).

3 CH2COMSEN Channel 2 compare output shadow enable

When this bit is set, the shadow register of TIMERx_CH2CV register, which updates at each update event will be enabled.

0: Channel 2 output compare shadow disabled

1: Channel 2 output compare shadow enabled

The PWM mode can be used without validating the shadow register only in single pulse mode (SPM bit in TIMERx_CTL0 register is set).

This bit cannot be modified when PROT[1:0] bit-field in TIMERx_CCHP register is

		11 and CH2MS bit-field is 000.
2	Reserved	Must be kept at reset value.
1:0	CH2MS[1:0]	Channel 2 I/O mode selection This bit-field specifies the work mode of the channel and the input signal selection. The CH2MS[2:0] bit-field is writable only when the channel is not active (When MCH2MSEL[1:0] = 2b'00, the CH2EN bit in TIMERx_CHCTL2 register is reset; when MCH2MSEL[1:0] = 2b'11, 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(x=0,7) 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.
31:16	Reserved	Must be kept at reset value.
15:12	CH3CAPFLT[3:0]	Channel 3 input capture filter control Refer to CH2CAPFLT description.
11:10	CH3CAPPSC[1:0]	Channel 3 input capture prescaler Refer to CH2CAPPSC description.
9:8	CH3MS[1:0]	Channel 3 mode selection Same as output compare mode.
7:4	CH2CAPFLT[3:0]	Channel 2 input capture filter control An event counter is used in the digital filter, in which a transition on the output occurs after N input events. This bit-field specifies the frequency used to sample CI2 input signal and the length of the digital filter applied to CI2. 0000: Filter disabled, $f_{SAMP}=f_{DTS}$, N=1. 0001: $f_{SAMP}=f_{CK_TIMER}$, N=2. 0010: $f_{SAMP}= f_{CK_TIMER}$, N=4. 0011: $f_{SAMP}= f_{CK_TIMER}$, N=8. 0100: $f_{SAMP}=f_{DTS}/2$, N=6.

0101: $f_{SAMP}=f_{DTS}/2$, N=8.
0110: $f_{SAMP}=f_{DTS}/4$, N=6.
0111: $f_{SAMP}=f_{DTS}/4$, N=8.
1000: $f_{SAMP}=f_{DTS}/8$, N=6.
1001: $f_{SAMP}=f_{DTS}/8$, N=8.
1010: $f_{SAMP}=f_{DTS}/16$, N=5.
1011: $f_{SAMP}=f_{DTS}/16$, N=6.
1100: $f_{SAMP}=f_{DTS}/16$, N=8.
1101: $f_{SAMP}=f_{DTS}/32$, N=5.
1110: $f_{SAMP}=f_{DTS}/32$, N=6.
1111: $f_{SAMP}=f_{DTS}/32$, N=8.

3:2	CH2CAPPSC[1:0]	Channel 2 input capture prescaler This bit-field specifies the factor of the prescaler on channel 2 input. The prescaler is reset when CH2EN bit in TIMERx_CHCTL2 register is cleared. 00: Prescaler disabled, capture is done on each channel input edge. 01: Capture is done every 2 channel input edges. 10: Capture is done every 4 channel input edges. 11: Capture is done every 8 channel input edges.
1:0	CH2MS[1:0]	Channel 2 mode selection Same as output compare mode.

Channel control register 2 (TIMERx_CHCTL2)

Address offset: 0x20
Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH3PERFOREN	CH2PERFOREN	CH1PERFOREN	CH0PERFOREN	Reserved											
rw	rw	rw	rw												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
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	CH3PERFOREN	Channel 3 in the composite PWM mode or asymmetric PWM mode, the period point match moment O3CPRE level setting Refer to CH0PERFOREN description.
30	CH2PERFOREN	Channel 2 in the composite PWM mode or asymmetric PWM mode, the period point match moment O2CPRE level setting Refer to CH0PERFOREN description.

29	CH1PERFOREN	Channel 1 in the composite PWM mode or asymmetric PWM mode, the period point match moment O1CPRE level setting Refer to CH0PERFOREN description.
28	CH0PERFOREN	Channel 0 in the composite PWM mode or asymmetric PWM mode, the period point match moment O0CPRE level setting 0: disable 1: In the composite PWM0 mode or asymmetric PWM0, the level will be forced to high at the time of period point matching; In the composite PWM1 mode or asymmetric PWM1 mode, the level at the matching time of the period point is forced to low. **Note:** In the central alignment mode, the period point is the counter underflow time; In the edge-aligned mode, the period point is the counter flow moment.
27:16	Reserved	Must be kept at reset value.
15	MCH3P	Multi mode channel 3 capture/compare polarity Refer to MCH0P description.
14	MCH3EN	Multi mode channel 3 capture/compare enable Refer to MCH0EN description.
13	CH3P	Channel 3 capture/compare polarity Refer to CH0P description.
12	CH3EN	Channel 3 capture/compare enable Refer to CH0EN description.
11	MCH2P	Multi mode channel 2 output polarity Refer to MCH0P description.
10	MCH2EN	Multi mode channel 2 output enable Refer to MCH0EN description.
9	CH2P	Channel 2 capture/compare polarity Refer to CH0P description.
8	CH2EN	Channel 2 capture/compare enable Refer to CH0EN description.
7	MCH1P	Multi mode channel 1 output polarity Refer to MCH0P description.
6	MCH1EN	Multi mode channel 1 output enable Refer to MCH0EN description.
5	CH1P	Channel 1 capture/compare polarity Refer to CH0P description.
4	CH1EN	Channel 1 capture/compare enable

		Refer to CH0EN description.
3	MCH0P	Multi mode channel 0 output polarity When Multi mode channel 0 is configured in output mode, and the MCH0MSEL[1:0] = 2b'11, this bit specifies the MCH0_O output signal polarity. 0: Multi mode channel 0 output active high 1: Multi mode channel 0 output active low When CH0 is configured in input mode, in conjunction with CH0P, this bit is used to define the polarity of CH0. This bit cannot be modified when PROT[1:0] bit-field in TIMERx_CCHP register is 11 or 10.
2	MCH0EN	Multi mode channel 0 capture/compare enable When multi mode channel 0 is configured in output mode, setting this bit enables MCH0_O signal in active state. When multi mode channel 0 is configured in input mode, setting this bit enables the capture event in multi mode channel 0. 0: Multi mode channel 0 disabled 1: Multi mode channel 0 enabled
1	CH0P	Channel 0 capture/compare polarity When channel 0 is configured in output mode, this bit specifies the output signal polarity. 0: Channel 0 active high 1: Channel 0 active low When channel 0 is configured in input mode, these bits 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_CCHP register is 11 or 10.
0	CH0EN	Channel 0 capture/compare enable When channel 0 is configured in output mode, setting this bit enables CH0_O signal in active state. When channel 0 is configured in input mode, setting this bit enables the capture event in channel 0. 0: Channel 0 disabled 1: Channel 0 enabled

Counter register (TIMERx_CNT)

Address offset: 0x24

Reset value: 0x0000 0000

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

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

Prescaler register (TIMERx_PSC)

Address offset: 0x28

Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	PSC[15:0]	Prescaler value of the counter clock The PSC clock is divided by (PSC+1) to generate the counter clock. The value of this bit-field will be loaded to the corresponding shadow register at every update event.

Counter auto reload register (TIMERx_CAR)

Address offset: 0x2C

Reset value: 0x0000 FFFF

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

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

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

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CARL[15:0]	Counter auto reload value This bit-field specifies the auto reload value of the counter.

Counter repetition register 0 (TIMERx_CREP0)

Address offset: 0x30

Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								CREP0[7:0]							
								rw							

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	CREP0[7:0]	Counter repetition value 0 This bit-field specifies the update event generation rate. Each time the repetition counter counts down to zero, an update event will be generated. The update rate of the shadow registers is also affected by this bit-field when these shadow registers are enabled. **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
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH0VAL[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH0VAL[15:0]	Capture/compare value of channel 0 When channel 0 is configured in input mode, this bit-field indicates the counter value at the last capture event. And this bit-field is read-only. When channel 0 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event.

Channel 1 capture/compare value register (TIMERx_CH1CV)

Address offset: 0x38

Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1VAL[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH1VAL[15:0]	Capture/compare value of channel 1 When channel 1 is configured in input mode, this bit-field indicates the counter value at the last capture event. And this bit-field is read-only. When channel 1 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event.

Channel 2 capture/compare value register (TIMERx_CH2CV)

Address offset: 0x3C

Reset value: 0x0000 0000

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

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

Reserved

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH2VAL[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH2VAL[15:0]	Capture/compare value of channel 2 When channel 2 is configured in input mode, this bit-field indicates the counter value at the last capture event. And this bit-field is read-only. When channel 2 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event.

Channel 3 capture/compare value register (TIMERx_CH3CV)

Address offset: 0x40

Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3VAL[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH3VAL[15:0]	Capture/compare value of channel 3 When channel 3 is configured in input mode, this bit-field indicates the counter value at the last capture event. And this bit-field is read-only. When channel 3 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event.

Complementary channel protection register (TIMERx_CCHP)

Address offset: 0x44

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CHBRKP	CHBRKEN	BRK1LK	BRK0LK	BRK1REL	BRK0REL	BRK1P	BRK1EN	BRK1F[3:0]				BRK0F[3:0]			
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
POEN	OAEN	BRK0P	BRK0EN	ROS	IOS	PROT[1:0]		DTCFG[7:0]							
rw	rw	rw	rw	rw	rw	rw		rw							

Bits	Fields	Descriptions
31	CHBRKP	Channel BREAK input signal polarity This bit specifies the polarity of the channel BREAK input signal. 0: Channel BREAK input active low 1: Channel BREAK input active high This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP register is 00.
30	CHBRKEN	Channel BREAK input signal enable This bit can be set to enable the channel BREAK input signal. 0: Channel BREAK input disabled 1: Channel BREAK input enabled This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP register is 00.
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_CCHP 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_CCHP 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_CCHP 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_CCHP register is 00.

19:16 BRK0F[3:0]

BREAK0 and channel BREAK input signal filter

An event counter is used in the digital filter, in which a transition on the output occurs after N input events. This bit-field specifies the frequency used to sample BREAK0

and channel BREAK input signal and the length of the digital filter applied to BREAK0 and channel BREAK.
0000: Filter disabled. BREAK0 and channel BREAK act asynchronously, N=1
0001: $f_{SAMP} = f_{CK_TIMER}$, N=2
0010: $f_{SAMP} = f_{CK_TIMER}$, N=4
0011: $f_{SAMP} = f_{CK_TIMER}$, N=8
0100: $f_{SAMP} = f_{DTS}/2$, N=6
0101: $f_{SAMP} = f_{DTS}/2$, N=8
0110: $f_{SAMP} = f_{DTS}/4$, N=6
0111: $f_{SAMP} = f_{DTS}/4$, N=8
1000: $f_{SAMP} = f_{DTS}/8$, N=6
1001: $f_{SAMP} = f_{DTS}/8$, N=8
1010: $f_{SAMP} = f_{DTS}/16$, N=5
1011: $f_{SAMP} = f_{DTS}/16$, N=6
1100: $f_{SAMP} = f_{DTS}/16$, N=8
1101: $f_{SAMP} = f_{DTS}/32$, N=5
1110: $f_{SAMP} = f_{DTS}/32$, N=6
1111: $f_{SAMP} = f_{DTS}/32$, N=8
This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP 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 and CHPOENx(x=0…3) bit can be set automatically by hardware.
0: POEN and CHPOENx(x=0…3) cannot be set by hardware.
1: POEN and CHPOENx(x=0…3) 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_CCHP 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_CCHP register is 00.

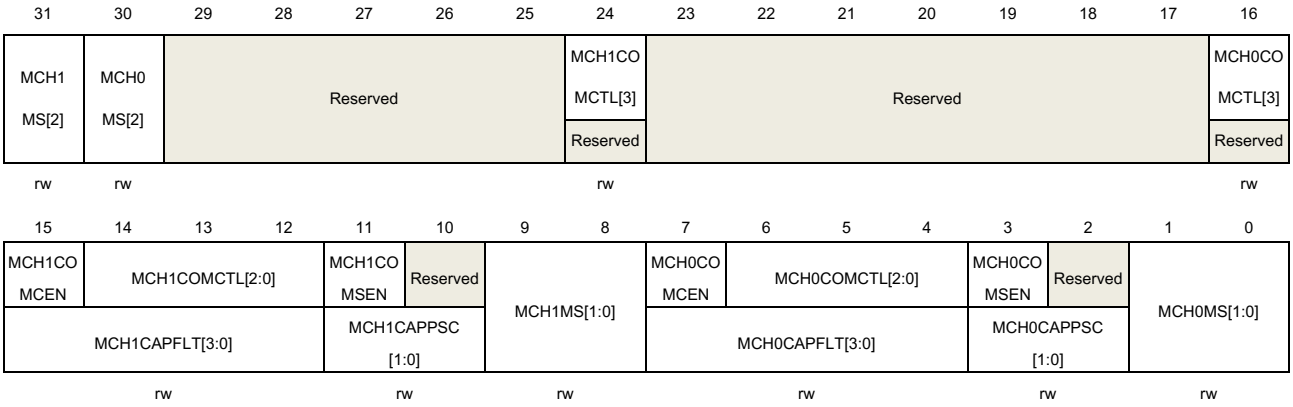
12	BRK0EN	BREAK0 input signal enable This bit can be set to enable the BREAK0 input signal 0: BREAK0 input disabled 1: BREAK0 input enabled This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP register is 00.
11	ROS	Run mode "off-state" enable When POEN&CHPOENx bit is set (Run mode), this bit can be set to enable the "off-state" for the channels which has been configured in output mode. Please refer to ***Table 15-5. Complementary outputs controlled by parameters (MCHxMSEL =2'b11)***. 0: "off-state" disabled. If the ChxEN or MCHxEN bit is reset, the corresponding channel is output disabled. 1: "off-state" enabled. If the ChxEN or MCHxEN bit is reset, the corresponding channel is "off-state". This bit cannot be modified when PROT [1:0] bit-field in TIMERx_CCHP register is 10 or 11.
10	IOS	Idle mode "off-state" enable When POEN&CHPOENx bit is reset (Idle mode), this bit can be set to enable the "off-state" for the channels which has been configured in output mode. Please refer to ***Table 15-5. Complementary outputs controlled by parameters (MCHxMSEL =2'b11)***. 0: "off-state" disabled. If the ChxEN/MCHxEN bits are both reset, the channels are output disabled. 1: "off-state" enabled. No matter the ChxEN/MCHxEN bits, the channels are "off-state". This bit cannot be modified when PROT [1:0] bit-field in TIMERx_CCHP register is 10 or 11.
9:8	PROT[1:0]	Complementary register protect control This bit-field specifies the write protection property of registers. 00: Protect disabled. No write protection. 01: PROT mode 0. The ISOx/ISOxN bits in TIMERx_CTL1 register, the CHBRKEN/CHBRKP/BRK0EN/ BRK0P/BRK1EN/ BRK1P/OAEN/DTCFG bits in TIMERx_CCHP 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_CCHP register and the ROS/IOS bits in TIMERx_FCCHPx (x = 0..3) register are writing protected. 11: PROT mode 2. In addition to the registers in PROT mode 1, the CHxCOMCTL/ CHxCOMSEN/ CHxCOMADDSEN/ MCHxCOMCTL/ MCHxCOMSEN bits in TIMERx_CHCTL0/1 and TIMERx_MCHCTL0/1 registers (if the related channel is

		configured in output) are writing protected. This bit-field can be written only once after the system reset. Once the TIMERx_CCHP register has been written, this bit-field will be writing protected.
7:0	DTCFG[7:0]	Dead time configuration 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] x t_{DT}, t_{DT} = t_{DTS}. DTCFG[7:5] = 3'b10x: DT value = (64+DTCFG[5:0]) x t_{DT}, t_{DT} = t_{DTS}*2. DTCFG[7:5] = 3'b110: DT value = (32+DTCFG[4:0]) x t_{DT}, t_{DT} = t_{DTS}*8. DTCFG[7:5] = 3'b111: DT value = (32+DTCFG[4:0]) x t_{DT}, t_{DT} = t_{DTS}*16. This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP register is 00.

Multi mode channel control register 0 (TIMERx_MCHCTL0)

Address offset: 0x48
Reset value: 0x0000 0000

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



Output compare mode:

Bits	Fields	Descriptions
31	MCH1MS[2]	Multi mode channel 1 I/O mode selection Refer to MCH1MS[1:0]description.
30	MCH0MS[2]	Multi mode channel 0 I/O mode selection Refer to MCH0MS[1:0] description.
29:25	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	Reserved	Must be kept at reset value.
9:8	MCH1MS[1:0]	Multi mode channel 1 I/O mode selection This bit-field specifies the direction of the channel and the input signal selection. This bit-field is writable only when the channel is not active (the MCH1EN bit in TIMERx_CHCTL2 register is reset) 000: Multi mode channel 1 is 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) 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] = 2b'00, 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] = 2b'11, the CH0COMCTL[2:0] bit-field controls the behavior of O0CPRE which drives CH0_O and MCH0_O, while the active level of CH0_O and MCH0_O depends on CH0P and MCH0P bits. 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 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_CCHP register is 11 and MCH0MS bit-field is 00(compare mode).

3 MCH0COMSEN Multi mode channel 0 output compare shadow enable
When this bit is set, the shadow register of TIMERx_MCH0CV register which updates at each update event will be enabled.
0: Multi mode channel 0 output compare shadow disabled
1: Multi mode channel 0 output compare shadow enabled
The PWM mode can be used without validating the shadow register only in single pulse mode (SPM bit in TIMERx_CTL0 register is set).
This bit cannot be modified when PROT[1:0] bit-field in TIMERx_CCHP register is 11 and MCH0MS bit-field is 00.

2	Reserved	Must be kept at reset value.
1:0	MCH0MS[1:0]	Multi mode channel 0 I/O mode selection This bit-field specifies the work mode of the channel and the input signal selection. This bit-field is writable only when the channel is not active (MCH0EN bit in TIMERx_CHCTL2 register is reset). 000: Multi mode channel 0 is 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) 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.

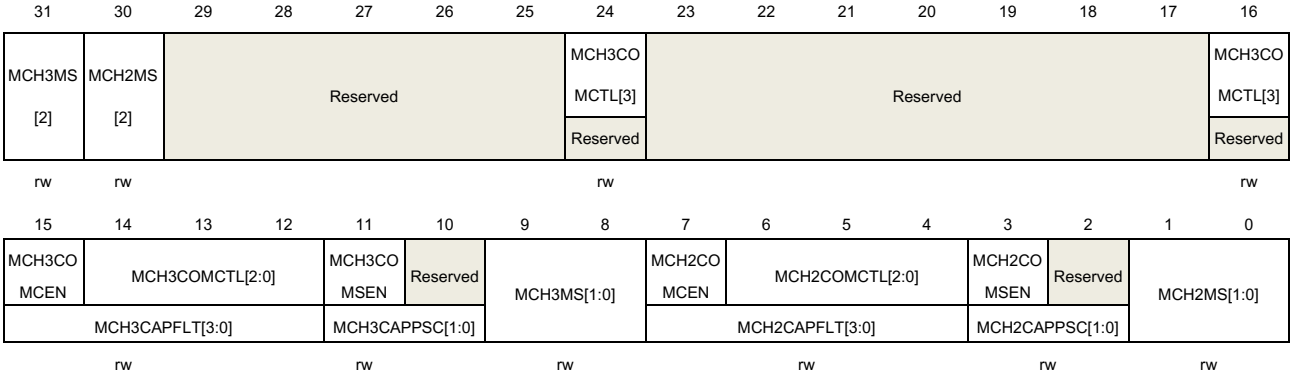
1001: $f_{SAMP}=f_{DTS}/8$, N=8.
1010: $f_{SAMP}=f_{DTS}/16$, N=5.
1011: $f_{SAMP}=f_{DTS}/16$, N=6.
1100: $f_{SAMP}=f_{DTS}/16$, N=8.
1101: $f_{SAMP}=f_{DTS}/32$, N=5.
1110: $f_{SAMP}=f_{DTS}/32$, N=6.
1111: $f_{SAMP}=f_{DTS}/32$, N=8.

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).



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	Reserved	Must be kept at reset value.
9:8	MCH3MS[1:0]	Multi mode channel 3 I/O mode selection This bit-field specifies the direction of the channel and the input signal selection. This bit-field is writable only when the channel is not active (MCH3EN bit in TIMERx_CHCTL2 register is reset). 000: Multi mode channel 3 is 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) 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] = 2b'00, 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] = 2b'11, the CH2COMCTL[2:0] bit-field controls the behavior of O2CPRE which drives CH2_O and MCH2_O, while the active level of CH2_O and MCH2_O depends on CH2P and MCH2P bits. 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_CCHP register is 11 and CH2NMS bit-field is 00(compare mode).

3	MCH2COMSEN	Multi mode channel 2 output compare shadow enable When this bit is set, the shadow register of TIMERx_MCH2CV register, which updates at each update event will be enabled. 0: Multi mode channel 2 output compare shadow disabled 1: Multi mode channel 2 output compare shadow enabled The PWM mode can be used without validating the shadow register only in single

		pulse mode (SPM bit in TIMERx_CTL0 register is set). This bit cannot be modified when PROT[1:0] bit-field in TIMERx_CCHP register is 11 and CH2NMS bit-field is 00.
2	Reserved	Must be kept at reset value.
1:0	MCH2MS[1:0]	Multi mode channel 2 I/O mode selection This bit-field specifies the work mode of the channel and the input signal selection. This bit-field is writable only when the channel is not active (MCH2EN bit in TIMERx_CHCTL2 register is reset). 000: Multi mode channel 2 is 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) 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 word(32-bit).

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								MCH3FP[1:0]		MCH2FP[1:0]		MCH1FP[1:0]		MCH0FP[1:0]	
								rw		rw		rw		rw	

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:6	MCH3FP[1:0]	Multi mode channel 3 capture/compare free polarity Refer to MCH0FP[1:0] description.
5:4	MCH2FP[1:0]	Multi mode channel 2 capture/compare free polarity Refer to MCH0FP[1:0] description.
3:2	MCH1FP[1:0]	Multi mode channel 1 capture/compare free polarity

		Refer to MCH0FP[1:0] description.
1:0	MCH0FP[1:0]	Multi mode channel 0 capture/compare free polarity When multi mode channel 0 is configured in output mode, and the MCH0MSEL[1:0] = 2b'00, these bits 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_CCHP register is 11 or 10.

Multi mode channel 0 capture/compare value register (TIMERx_MCH0CV)

Address offset: 0x54

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	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.

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).

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCH1VAL[15:0]															
rw															

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).

31	30	29	28	27	26	25	24	23	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) and asymmetric PWM mode.

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) and asymmetric PWM mode.

Channel 2 additional compare value register (TIMERx_CH2COMV_ADD)

Address offset: 0x6C

Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH2COMVAL_ADD[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH2COMVAL_ADD[15:0]	Additional compare value of channel 2 When channel 2 is configured in output mode, this bit-field contains value to be

compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event.

Note: This register just used in composite PWM mode(when CH0CPWMEN=1) and asymmetric PWM mode.

Channel 3 additional compare value register (TIMERx_CH3COMV_ADD)

Address offset: 0x70

Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3COMVAL_ADD[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH3COMVAL_ADD[15:0]	Additional compare value of channel 3 When channel 3 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event. **Note:** This register just used in composite PWM mode(when CH0CPWMEN=1) and asymmetric PWM mode.

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

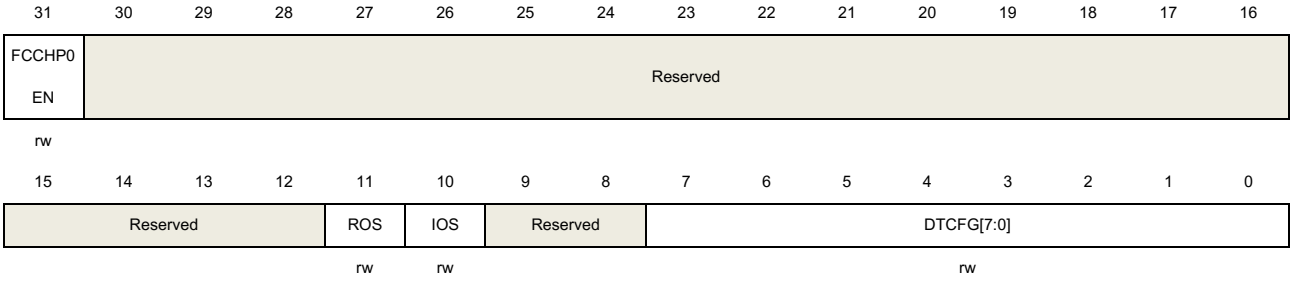
Free complementary channel protection register 0 (TIMERx_FCCHP0)

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.



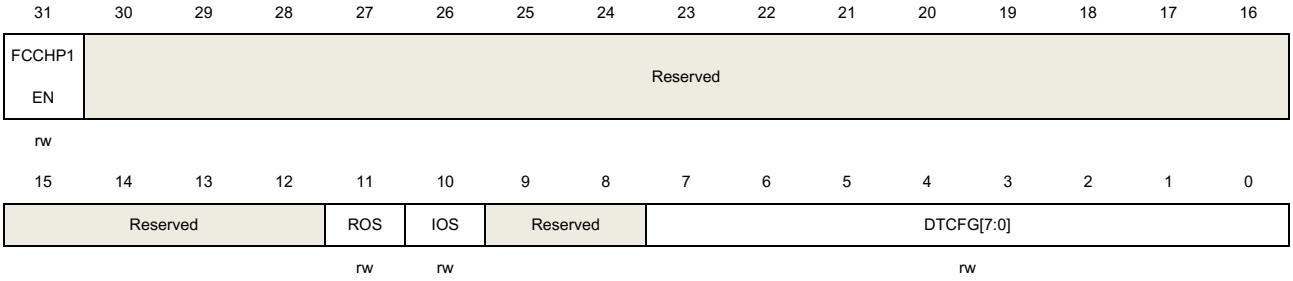
Bits	Fields	Descriptions
31	FCCHP0EN	Free complementary channel protection register 0 enable 0: the ROS、IOS and DTCFG[7:0] bits in TIMERx_CCHP register is active 1: the ROS、IOS and DTCFG[7:0] bits in TIMERx_FCCHP0 register is active This bit can be modified only when PROT [1:0] bit-field in TIMERx_CCHP register is 00.
30:12	Reserved	Must be kept at reset value.
11	ROS	Run mode "off-state" enable When POEN&CHPOEN0 bit is set (Run mode), this bit can be set to enable the "off-state" for the channels which has been configured in output mode. 0: "off-state" disabled. If the CH0EN or MCH0EN bit is reset, the corresponding channel is output disabled. 1: "off-state" enabled. If the CH0EN or MCH0EN bit is reset, the corresponding channel is "off-state". This bit cannot be modified when PROT [1:0] bit-field in TIMERx_CCHP register is 10 or 11.
10	IOS	Idle mode "off-state" enable When POEN&CHPOEN0 bit is reset (Idle mode), this bit can be set to enable the "off-state" for the channels which has been configured in output mode. 0: "off-state" disabled. If the CH0EN/MCH0EN bits are both reset, the channels are output disabled. 1: "off-state" enabled. No matter the CH0EN/MCH0EN bits, the channels are "off-state". This bit cannot be modified when PROT [1:0] bit-field in TIMERx_CCHP register is 10 or 11.

9:8	Reserved	Must be kept at reset value.
7:0	DTCFG[7:0]	Dead time configure This bit-field controls the value of the dead-time, which is inserted before the output transitions. The relationship between DTCFG value and the duration of dead-time is as follow: DTCFG [7:5] =3'b0xx: Dtvalue = DTCFG [7:0]x t_{DT}, t_{DT}=t_{DTS}. DTCFG [7:5] =3'b10x: Dtvalue = (64+DTCFG [5:0])xt_{DT}, t_{DT} =t_{DTS}*2. DTCFG [7:5] =3'b110: Dtvalue = (32+DTCFG [4:0])xt_{DT}, t_{DT}=t_{DTS}*8. DTCFG [7:5] =3'b111: Dtvalue = (32+DTCFG [4:0])xt_{DT}, t_{DT} =t_{DTS}*16. This bit can be modified only when PROT [1:0] bit-field in TIMERx_CCHP register is 00.

Free complementary channel protection register 1 (TIMERx_FCCHP1)

Address offset: 0x80
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).
This register is used to configure the outputs of CH1_O/MCH1_O.



Bits	Fields	Descriptions
31	FCCHP1EN	Free complementary channel protection register 1 enable 0: the ROS、IOS and DTCFG[7:0] bits in TIMERx_CCHP register is active 1: the ROS、IOS and DTCFG[7:0] bits in TIMERx_FCCHP1 register is active This bit can be modified only when PROT [1:0] bit-field in TIMERx_CCHP register is 00.
30:12	Reserved	Must be kept at reset value.
11	ROS	Run mode "off-state" enable When POEN&CHPOEN1 bit is set (Run mode), this bit can be set to enable the "off-state" for the channels which has been configured in output mode. 0: "off-state" disabled. If the CH1EN or MCH1EN bit is reset, the corresponding channel is output disabled. 1: "off-state" enabled. If the CH1EN or MCH1EN bit is reset, the corresponding channel is "off-state". This bit cannot be modified when PROT [1:0] bit-field in TIMERx_CCHP register is

		10 or 11.
10	IOS	Idle mode "off-state" enable When POEN&CHPOEN1 bit is reset (Idle mode), this bit can be set to enable the "off-state" for the channels which has been configured in output mode. 0: "off-state" disabled. If the CH1EN/MCH1EN bits are both reset, the channels are output disabled. 1: "off-state" enabled. No matter the CH1EN/MCH1EN bits, the channels are "off-state". This bit cannot be modified when PROT [1:0] bit-field in TIMERx_CCHP register is 10 or 11.
9:8	Reserved	Must be kept at reset value.
7:0	DTCFG[7:0]	Dead time configure This bit-field controls the value of the dead-time, which is inserted before the output transitions. The relationship between DTCFG value and the duration of dead-time is as follow: DTCFG [7:5] =3'b0xx: Dtvalue = DTCFG [7:0]x t_{DT}, t_{DT}=t_{DTS}. DTCFG [7:5] =3'b10x: Dtvalue = (64+DTCFG [5:0])xt_{DT}, t_{DT} =t_{DTS}*2. DTCFG [7:5] =3'b110: Dtvalue = (32+DTCFG [4:0])xt_{DT}, t_{DT}=t_{DTS}*8. DTCFG [7:5] =3'b111: Dtvalue = (32+DTCFG [4:0])xt_{DT}, t_{DT} =t_{DTS}*16. This bit can be modified only when PROT [1:0] bit-field in TIMERx_CCHP register is 00.

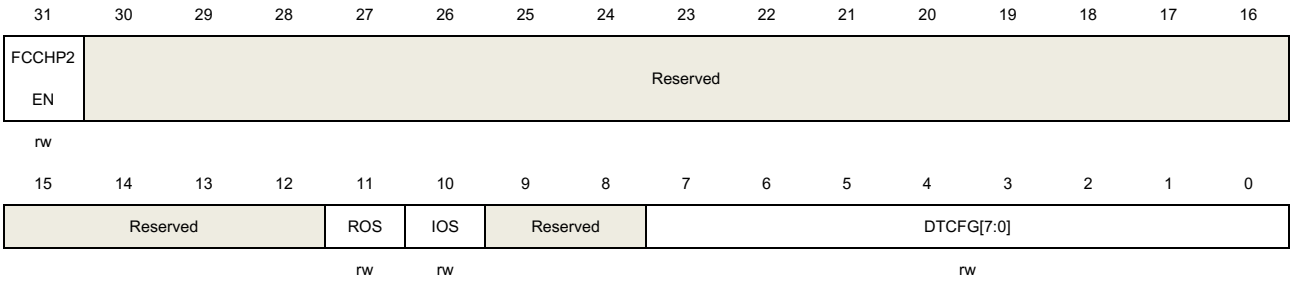
Free complementary channel protection register 2 (TIMERx_FCCHP2)

Address offset: 0x84

Reset value: 0x0000 0000

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

This register is used to configure the outputs of CH2_O/MCH2_O.



Bits	Fields	Descriptions
31	FCCHP2EN	Free complementary channel protection register 2 enable 0: the ROS、IOS and DTCFG[7:0] bits in TIMERx_CCHP register is active 1: the ROS、IOS and DTCFG[7:0] bits in TIMERx_FCCHP2 register is active This bit can be modified only when PROT [1:0] bit-field in TIMERx_CCHP register

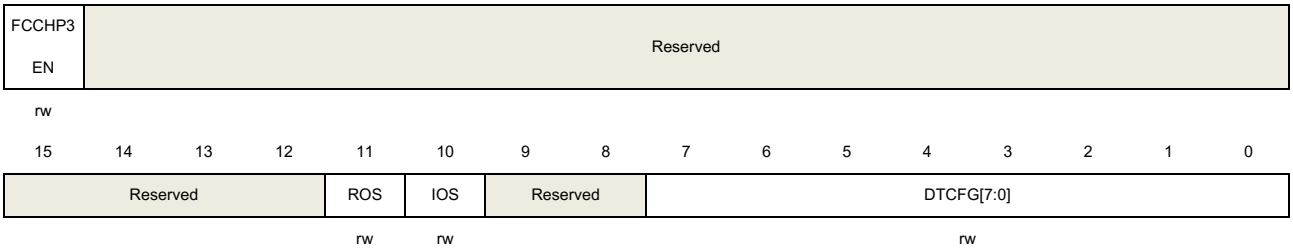
		is 00.
30:12	Reserved	Must be kept at reset value.
11	ROS	Run mode "off-state" enable When POEN&CHPOEN2 bit is set (Run mode), this bit can be set to enable the "off-state" for the channels which has been configured in output mode. 0: "off-state" disabled. If the CH2EN or MCH2EN bit is reset, the corresponding channel is output disabled. 1: "off-state" enabled. If the CH2EN or MCH2EN bit is reset, the corresponding channel is "off-state". This bit cannot be modified when PROT [1:0] bit-field in TIMERx_CCHP register is 10 or 11.
10	IOS	Idle mode "off-state" enable When POEN&CHPOEN2 bit is reset (Idle mode), this bit can be set to enable the "off-state" for the channels which has been configured in output mode. 0: "off-state" disabled. If the CH2EN/MCH2EN bits are both reset, the channels are output disabled. 1: "off-state" enabled. No matter the CH2EN/MCH2EN bits, the channels are "off-state". This bit cannot be modified when PROT [1:0] bit-field in TIMERx_CCHP register is 10 or 11.
9:8	Reserved	Must be kept at reset value.
7:0	DTCFG[7:0]	Dead time configure This bit-field controls the value of the dead-time, which is inserted before the output transitions. The relationship between DTCFG value and the duration of dead-time is as follow: DTCFG [7:5] =3'b0xx: Dtvalue = DTCFG [7:0]x t_{DT}, t_{DT}=t_{DTS}. DTCFG [7:5] =3'b10x: Dtvalue = (64+DTCFG [5:0])xt_{DT}, t_{DT} =t_{DTS}*2. DTCFG [7:5] =3'b110: Dtvalue = (32+DTCFG [4:0])xt_{DT}, t_{DT}=t_{DTS}*8. DTCFG [7:5] =3'b111: Dtvalue = (32+DTCFG [4:0])xt_{DT}, t_{DT} =t_{DTS}*16. This bit can be modified only when PROT [1:0] bit-field in TIMERx_CCHP register is 00.

Free complementary channel protection register 3 (TIMERx_FCCHP3)

Address offset: 0x88
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).
This register is used to configure the outputs of CH3_O/MCH3_O.

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



Bits	Fields	Descriptions
31	FCCHP3EN	Free complementary channel protection register 0 enable 0: the ROS、IOS and DTCFG[7:0] bits in TIMERx_CCHP register is active 1: the ROS、IOS and DTCFG[7:0] bits in TIMERx_FCCHP3 register is active This bit can be modified only when PROT [1:0] bit-field in TIMERx_CCHP register is 00.
30:12	Reserved	Must be kept at reset value.
11	ROS	Run mode "off-state" enable When POEN&CHPOEN3 bit is set (Run mode), this bit can be set to enable the "off-state" for the channels which has been configured in output mode. 0: "off-state" disabled. If the CH3EN or MCH3EN bit is reset, the corresponding channel is output disabled. 1: "off-state" enabled. If the CH3EN or MCH3EN bit is reset, the corresponding channel is "off-state". This bit cannot be modified when PROT [1:0] bit-field in TIMERx_CCHP register is 10 or 11.
10	IOS	Idle mode "off-state" enable When POEN&CHPOEN3 bit is reset (Idle mode), this bit can be set to enable the "off-state" for the channels which has been configured in output mode. 0: "off-state" disabled. If the CH3EN/MCH3EN bits are both reset, the channels are output disabled. 1: "off-state" enabled. No matter the CH3EN/MCH3EN bits, the channels are "off-state". This bit cannot be modified when PROT [1:0] bit-field in TIMERx_CCHP register is 10 or 11.
9:8	Reserved	Must be kept at reset value.
7:0	DTCFG[7:0]	Dead time configure This bit-field controls the value of the dead-time, which is inserted before the output transitions. The relationship between DTCFG value and the duration of dead-time is as follow: DTCFG [7:5] =3'b0xx: Dtvalue = DTCFG [7:0]x t_{DT}, t_{DT}=t_{DTS}. DTCFG [7:5] =3'b10x: Dtvalue = (64+DTCFG [5:0])xt_{DT}, t_{DT} =t_{DTS}*2. DTCFG [7:5] =3'b110: Dtvalue = (32+DTCFG [4:0])xt_{DT}, t_{DT}=t_{DTS}*8. DTCFG [7:5] =3'b111: Dtvalue = (32+DTCFG [4:0])xt_{DT}, t_{DT} =t_{DTS}*16.

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

Alternate function control register 0 (TIMER0_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					BRK0CMP1P	BRK0CMP0P	Reserved						BRK0IN2P	BRK0IN1P	BRK0IN0P
					rw	rw							rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					BRK0CMP1EN	BRK0CMP0EN	BRK0HPDFEN	Reserved					BRK0IN2EN	BRK0IN1EN	BRK0IN0EN
					rw	rw	rw						rw	rw	rw

Bits	Fields	Descriptions
31:27	Reserved	Must be kept at reset value.
26	BRK0CMP1P	BREAK0 CMP1 input polarity This bit is used to configure the CMP1 input polarity, and the specific polarity is determined by this bit and the BRK0P bit. 0: CMP1 input signal will not be inverted (BRK0P =0, the input signal is active low; BRK0P =1, the input signal is active high) 1: CMP1 input signal will be inverted (BRK0P=0, the input signal is active high; BRK0P =1, the input signal is active low) This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP 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_CCHP 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_CCHP 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_CCHP 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_CCHP register is 00.
15:11	Reserved	Must be kept at reset value.
10	BRK0CMP1EN	BREAK0 CMP1 enable 0: CMP1 input disabled 1: CMP1 input enabled This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP 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_CCHP 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_CCHP register is

		00.
7:3	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_CCHP 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_CCHP 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_CCHP register is 00.

Alternate function control register 1 (TIMER0_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					BRK1CMP1P	BRK1CMP0P	Reserved						BRK1IN2P	BRK1IN1P	BRK1IN0P
					rw	rw							rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					BRK1CMP1EN	BRK1CMP0EN	BRK1HPDFEN	Reserved					BRK1IN2EN	BRK1IN1EN	BRK1IN0EN
					rw	rw	rw						rw	rw	rw

Bits	Fields	Descriptions
31:27	Reserved	Must be kept at reset value.
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_CCHP 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_CCHP register is 00.
24:19	Reserved	Must be kept at reset value.
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_CCHP 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_CCHP 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_CCHP register is 00.

15:11	Reserved	Must be kept at reset value.
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_CCHP 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_CCHP register is 00.
8	BRK1HPDFEN	BREAK1 HPDF input (HPDF_BREAK[1]) enable 0: HPDF input disabled 1: HPDF input enabled This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP register is 00.
7:3	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_CCHP 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_CCHP 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_CCHP register is 00.

Alternate function control register 0 (TIMER7_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					BRK0CMP1P	BRK0CMP0P	Reserved						BRK0IN2P	BRK0IN1P	BRK0IN0P
					rw	rw							rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					BRK0CMP1EN	BRK0CMP0EN	BRK0HPDFEN	Reserved					BRK0IN2EN	BRK0IN1EN	BRK0IN0EN
					rw	rw	rw						rw	rw	rw

Bits	Fields	Descriptions
31:27	Reserved	Must be kept at reset value.
26	BRK0CMP1P	BREAK0 CMP1 input polarity This bit is used to configure the CMP1 input polarity, and the specific polarity is determined by this bit and the BRK0P bit. 0: CMP1 input signal will not be inverted (BRK0P =0, the input signal is active low; BRK0P =1, the input signal is active high) 1: CMP1 input signal will be inverted (BRK0P=0, the input signal is active high; BRK0P =1, the input signal is active low) This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP 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_CCHP 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_CCHP 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_CCHP 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_CCHP register is 00.
15:11	Reserved	Must be kept at reset value.
10	BRK0CMP1EN	BREAK0 CMP1 enable 0: CMP1 input disabled 1: CMP1 input enabled This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP 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_CCHP register is 00.
8	BRK0HPDFEN	BREAK0 HPDF input (HPDF_BREAK[2]) enable 0: HPDF input disabled 1: HPDF input enabled This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP register is 00.
7:3	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_CCHP 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_CCHP 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_CCHP register is 00.

Alternate function control register 1 (TIMER7_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					BRK1CMP1P	BRK1CMP0P	Reserved						BRK1IN2P	BRK1IN1P	BRK1IN0P
					rw	rw							rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					BRK1CMP1EN	BRK1CMP0EN	BRK1HPDFEN	Reserved					BRK1IN2EN	BRK1IN1EN	BRK1IN0EN
					rw	rw	rw						rw	rw	rw

Bits	Fields	Descriptions
31:27	Reserved	Must be kept at reset value.
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_CCHP 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_CCHP register is 00.
24:19	Reserved	Must be kept at reset value.
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_CCHP 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_CCHP 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_CCHP register is 00.
15:11	Reserved	Must be kept at reset value.
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_CCHP 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_CCHP register is

		00.
8	BRK1HPDFEN	BREAK1 HPDF input (HPDF_BREAK[3]) enable 0: HPDF input disabled 1: HPDF input enabled This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP register is 00.
7:3	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_CCHP 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_CCHP 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_CCHP 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 747pprox.747 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).

Channel break control register (TIMERx_CHBRKCTL)

Address offset: 0xAC
Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								CHPOEN3	CHPOEN2	CHPOEN1	CHPOEN0	Reserved			
								rw	rw	rw	rw				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3BRKINEN	CH2BRKINEN	CH1BRKINEN	CH0BRKINEN	CH3BRKIE	CH2BRKIE	CH1BRKIE	CH0BRKIE	CH3BRKINP	CH2BRKINP	CH1BRKINP	CH0BRKINP	CH3BRKMIE	CH2BRKMIE	CH1BRKMIE	CH0BRKMIE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23	CHPOEN3	Channel primary output enable 3 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 channel 3 BREAK input is

		active. When one of channels is configured in output mode, setting this bit enables the channel outputs (CH3_O and MCH3_O) if the corresponding enable bits (CH3EN, MCH3EN) have been set. 0: CH3/MCH3 outputs are disabled or forced to idle state. 1: CH3/MCH3 outputs are enabled.
22	CHPOEN2	Channel primary output enable 2 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 channel 2 BREAK input is active. When one of channels is configured in output mode, setting this bit enables the channel outputs (CH2_O and MCH2_O) if the corresponding enable bits (CH2EN, MCH2EN) have been set. 0: CH2/MCH2 outputs are disabled or forced to idle state. 1: CH2/MCH2 outputs are enabled.
21	CHPOEN1	Channel primary output enable 1 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 channel 1 BREAK input is active. When one of channels is configured in output mode, setting this bit enables the channel outputs (CH1_O and MCH1_O) if the corresponding enable bits (CH1EN, MCH1EN) have been set. 0: CH1/MCH1 outputs are disabled or forced to idle state. 1: CH1/MCH1 outputs are enabled.
20	CHPOEN0	Channel primary output enable 0 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 channel 0 BREAK input is active. When one of channels is configured in output mode, setting this bit enables the channel outputs (CH0_O and MCH0_O) if the corresponding enable bits (CH0EN, MCH0EN) have been set. 0: CH0/MCH0 outputs are disabled or forced to idle state. 1: CH0/MCH0 outputs are enabled.
19:16	Reserved	Must be kept at reset value.
15	CH3BRKINEN	Channel 3 BREAK input signal enable This bit can be set to enable the channel 3 BREAK input signal. 0: Channel 3 BREAK input disabled 1: Channel 3 BREAK input enabled
14	CH2BRKINEN	Channel 2 BREAK input signal enable This bit can be set to enable the channel 2 BREAK input signal. 0: Channel 2 BREAK input disabled 1: Channel 2 BREAK input enabled
13	CH1BRKINEN	Channel 1 BREAK input signal enable This bit can be set to enable the channel 1 BREAK input signal.

0: Channel 1 BREAK input disabled
1: Channel 1 BREAK input enabled

12 CH0BRKINEN Channel 0 BREAK input signal enable
This bit can be set to enable the channel 0 BREAK input signal.
0: Channel 0 BREAK input disabled
1: Channel 0 BREAK input enabled

11 CH3BRKIE Channel 3 BREAK interrupt enable
0: Disabled
1: Enabled

10 CH2BRKIE Channel 2 BREAK interrupt enable
0: Disabled
1: Enabled

9 CH1BRKIE Channel 1 BREAK interrupt enable
0: Disabled
1: Enabled

8 CH0BRKIE Channel 0 BREAK interrupt enable
0: Disabled
1: Enabled

7 CH3BRKINP Channel 3 BREAK input signal polarity
This bit specifies the polarity of the channel 3 BREAK input signal.
0: Channel 3 BREAK input signal will not be inverted
1: Channel 3 BREAK input signal will be inverted

6 CH2BRKINP Channel 2 BREAK input signal polarity
This bit specifies the polarity of the channel 2 BREAK input signal.
0: Channel 2 BREAK input signal will not be inverted
1: Channel 2 BREAK input signal will be inverted

5 CH1BRKINP Channel 1 BREAK input signal polarity
This bit specifies the polarity of the channel 1 BREAK input signal.
0: Channel 1 BREAK input signal will not be inverted
1: Channel 1 BREAK input signal will be inverted

4 CH0BRKINP Channel 0 BREAK input signal polarity
This bit specifies the polarity of the channel 0 BREAK input signal.
0: Channel 0 BREAK input signal will not be inverted
1: Channel 0 BREAK input signal will be inverted

3 CH3BRKMIE Channel 3 BREAK multi-period interrupt enable
0: Disabled
1: Enabled

2 CH2BRKMIE Channel 2 BREAK multi-period interrupt enable

0: Disabled
1: Enabled

1	CH1BRKMIE	Channel 1 BREAK multi-period interrupt enable 0: Disabled 1: Enabled
0	CH0BRKMIE	Channel 0 BREAK multi-period interrupt enable 0: Disabled 1: Enabled

Channel break period register (TIMERx_CHBRKPER)

Address offset: 0xB0
Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	PERCNT[15:0]	Channel BREAK period count In Edge-aligned, this field represents PERCNT PWM periods. In Center-aligned, this field represents PERCNT/2 PWM periods.

Channel break interrupt flag register (TIMERx_CHBRKINTF)

Address offset: 0xB4
Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								CH3BRKIF	CH2BRKIF	CH1BRKIF	CH0BRKIF	CH3BRKMIF	CH2BRKMIF	CH1BRKMIF	CH0BRKMIF
								rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

Bits	Fields	Descriptions

31:8	Reserved	Must be kept at reset value.
7	CH3BRKIF	Channel 3 BREAK interrupt flag This flag is set by hardware as soon as the channel 3 BREAK input is active, and cleared by software if the channel 3 BREAK input is not at active level. 0: No active level on channel 3 BREAK inputs has been detected. 1: An active level on channel 3 BREAK inputs has been detected. **Note:** When CH3BRKIE =1 and BRKIE =1, if this flag is set, an interrupt is generated.
6	CH2BRKIF	Channel 2 BREAK interrupt flag This flag is set by hardware as soon as the channel 2 BREAK input is active, and cleared by software if the channel 2 BREAK input is not at active level. 0: No active level on channel 2 BREAK inputs has been detected. 1: An active level on channel 2 BREAK inputs has been detected. **Note:** When CH2BRKIE =1 and BRKIE =1, if this flag is set, an interrupt is generated.
5	CH1BRKIF	Channel 1 BREAK interrupt flag This flag is set by hardware as soon as the channel 1 BREAK input is active, and cleared by software if the channel 1 BREAK input is not at active level. 0: No active level on channel 1 BREAK inputs has been detected. 1: An active level on channel 1 BREAK inputs has been detected. **Note:** When CH1BRKIE =1 and BRKIE =1, if this flag is set, an interrupt is generated.
4	CH0BRKIF	Channel 0 BREAK interrupt flag This flag is set by hardware as soon as the channel 0 BREAK input is active, and cleared by software if the channel 0 BREAK input is not at active level. 0: No active level on channel 0 BREAK inputs has been detected. 1: An active level on channel 0 BREAK inputs has been detected. **Note:** When CH0BRKIE =1 and BRKIE =1, if this flag is set, an interrupt is generated.
3	CH3BRKMIF	Channel 3 BREAK multi-period interrupt flag This flag is set by hardware when the channel 3 BREAK input is active for multiple consecutive periods (determined by the TIMERx_CHBRKPER register), and cleared by software if the channel 3 BREAK input is not at active level. 0: No active level on channel 3 BREAK inputs for multiple consecutive periods has been detected. 1: An active level on channel 3 BREAK inputs for multiple consecutive periods has been detected. **Note:** When CH3BRKMIE =1 and BRKIE =1, if this flag is set, an interrupt is generated.
2	CH2BRKMIF	Channel 2 BREAK multi-period interrupt flag This flag is set by hardware when the channel 2 BREAK input is active for multiple

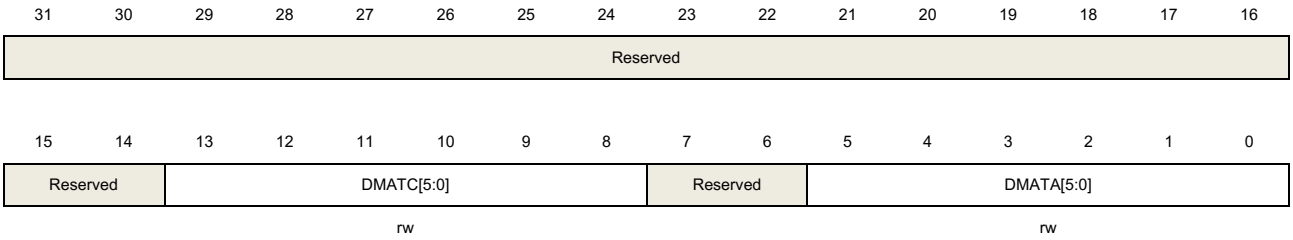
consecutive periods (determined by the TIMERx_CHBRKPER register), and cleared by software if the channel 2 BREAK input is not at active level.
0: No active level on channel 2 BREAK inputs for multiple consecutive periods has been detected.
1: An active level on channel 2 BREAK inputs for multiple consecutive periods has been detected.
Note: When CH2BRKMIE =1 and BRKIE =1, if this flag is set, an interrupt is generated.

1	CH1BRKMIF	Channel 1 BREAK multi-period interrupt flag This flag is set by hardware when the channel 1 BREAK input is active for multiple consecutive periods (determined by the TIMERx_CHBRKPER register), and cleared by software if the channel 1 BREAK input is not at active level. 0: No active level on channel 1 BREAK inputs for multiple consecutive periods has been detected. 1: An active level on channel 1 BREAK inputs for multiple consecutive periods has been detected. **Note:** When CH1BRKMIE =1 and BRKIE =1, if this flag is set, an interrupt is generated.
0	CH0BRKMIF	Channel 0 BREAK multi-period interrupt flag This flag is set by hardware when the channel 0 BREAK input is active for multiple consecutive periods (determined by the TIMERx_CHBRKPER register), and cleared by software if the channel 0 BREAK input is not at active level. 0: No active level on channel 0 BREAK inputs for multiple consecutive periods has been detected. 1: An active level on channel 1 BREAK inputs for multiple consecutive periods has been detected. **Note:** When CH0BRKMIE =1 and BRKIE =1, if this flag is set, an interrupt is generated.

DMA configuration register (TIMERx_DMACFG)

Address offset: 0xE0
Reset value: 0x0000 0000

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



Bits	Fields	Descriptions

31:14	Reserved	Must be kept at reset value.
13:8	DMATC[5:0]	DMA transfer count This field defines the times of accessing(R/W) the TIMERx_DMATB register by DMA. 6'b000000: transfer 1 time 6'b000001: transfer 2 times … 6'b100101: transfer 38 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'b101101: TIMERx_CHBRKINTF 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															

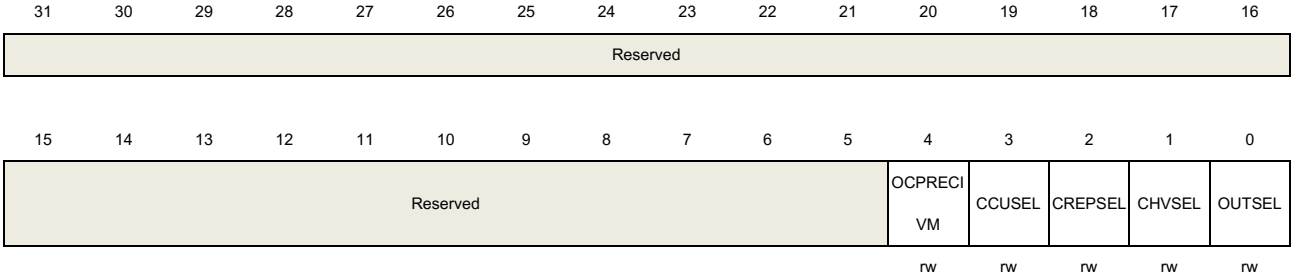
Bits	Fields	Descriptions
31:0	DMATB[31:0]	DMA transfer buffer When a read or write operation is assigned to this register, the register located at the address ranges from (start address) to (start address + transfer count * 4) will be accessed. The transfer count is calculated by hardware, and ranges 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
31:5	Reserved	Must be kept at reset value.
4	OCPRECIVM	Output prepare signal clear invalid mode select This bit determines when the output prepare signal clear is set to invalid by the hardware. 0：The next counter overflow or underflow event 1：The next counter update event
3	CCUSEL	Commutation control shadow register update select This bit is valid only when the CCUC[2:0] bit-field are set to 100, 101 and 110. 0: The shadow registers update when the counter generates an overflow/ underflow event. 1: The shadow registers update when the counter generates an overflow/ underflow event and the repetition counter value is zero.
2	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.

15.2. General level0 timer (TIMERx, x=1,2,3,4,22,23,30,31)

15.2.1. Overview

The general level0 timer module (TIMER1/2/3/4/22/23/30/31) 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.

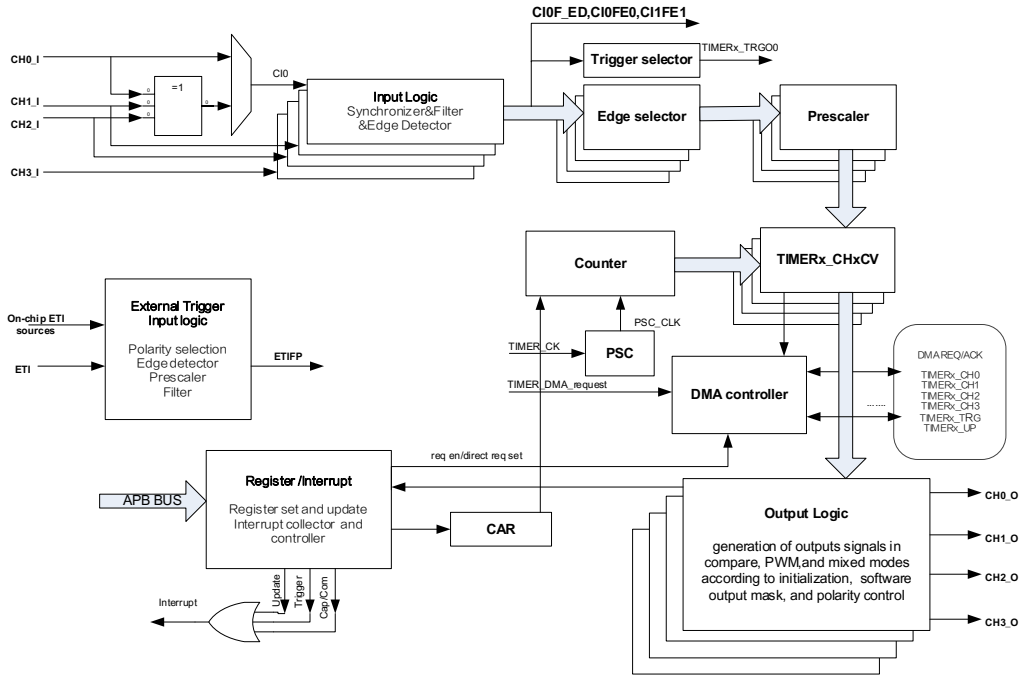
15.2.2. Characteristics

- Total channel num: 4.
- Counter width: 16 bits(TIMER2/3/30/31) or 32 bits(TIMER1/4/22/23).
- 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.

15.2.3. Block diagram

Figure 15-54. General Level 0 timer block diagram provides details on the internal configuration of the general level 0 timer.

Figure 15-54. General Level 0 timer block diagram



15.2.4. 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..7,9…15) in SYSCFG_TIMERxCFG(x=1..4,22,23,30,31) registers.

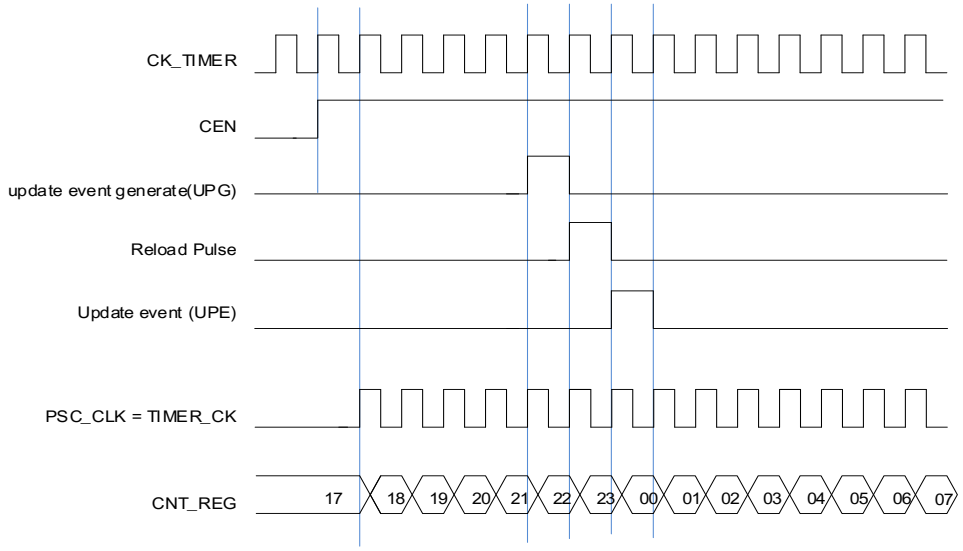
- TSCFGy[4:0] (y=0..7,9…15) in SYSCFG_TIMERxCFG(x=1..4,22,23,30,31) registers. Internal clock CK_TIMER is selected as timer clock source which is from module RCU.

The default clock source is the CK_TIMER for driving the counter prescaler when TSCFGy[4:0] (y=0..7,9…15) = 5'b00000 in SYSCFG_TIMERxCFG(x=1..4,22,23,30,31) registers. When the CEN is set, the CK_TIMER will be divided by PSC value to generate PSC_CLK.

In this mode, the TIMER_CK which drives counter's prescaler to count is equal to CK_TIMER which is from RCU module.

If TSCFGy[4:0] (y=0..2,6,9..14) in SYSCFG_TIMERxCFG(x=1..4,22,23,30,31) registers are setting to an available value, the prescaler is clocked by other clock sources selected in the TSCFGy[4:0] (y=0..2,6,9..14) bit-field, 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 15-55. Normal mode, internal clock divided by 1



■ TSCFG6[4:0] are setting to a nonzero value (external clock mode 0). External input pin is selected as timer clock source.

The TIMER_CK, which drives counter's prescaler to count, can be triggered by the event of rising or falling edge on the external pin TIMERx_CH0/TIMERx_CH1. This mode can be selected by setting TSCFG6[4:0] to 0x5~0x7.

And, the counter prescaler can also be driven by rising edge on the internal trigger input pin ITI0~ITI14. This mode can be selected by setting TSCFG6[4:0] to 0x1~0x4, 0x9~0x14.

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

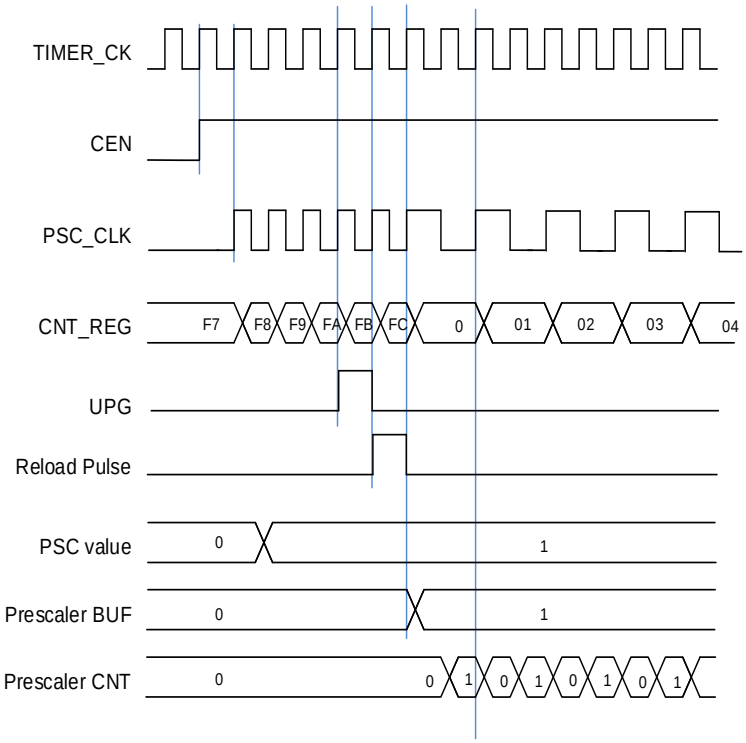
The TIMER_CK, which drives counter's prescaler to count, can be triggered by the event of rising or falling edge on the external pin ETI. This mode can be selected by setting the SMC1 bit in the TIMERx_SMCFG register to 1. The other way to select the ETI signal as the clock source is setting the TSCFG6[4:0] 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)*** 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 15-56. 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 15-57. Timing chart of up counting mode, PSC=0/2 and *Figure 15-58. 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 15-57. Timing chart of up counting mode, PSC=0/2

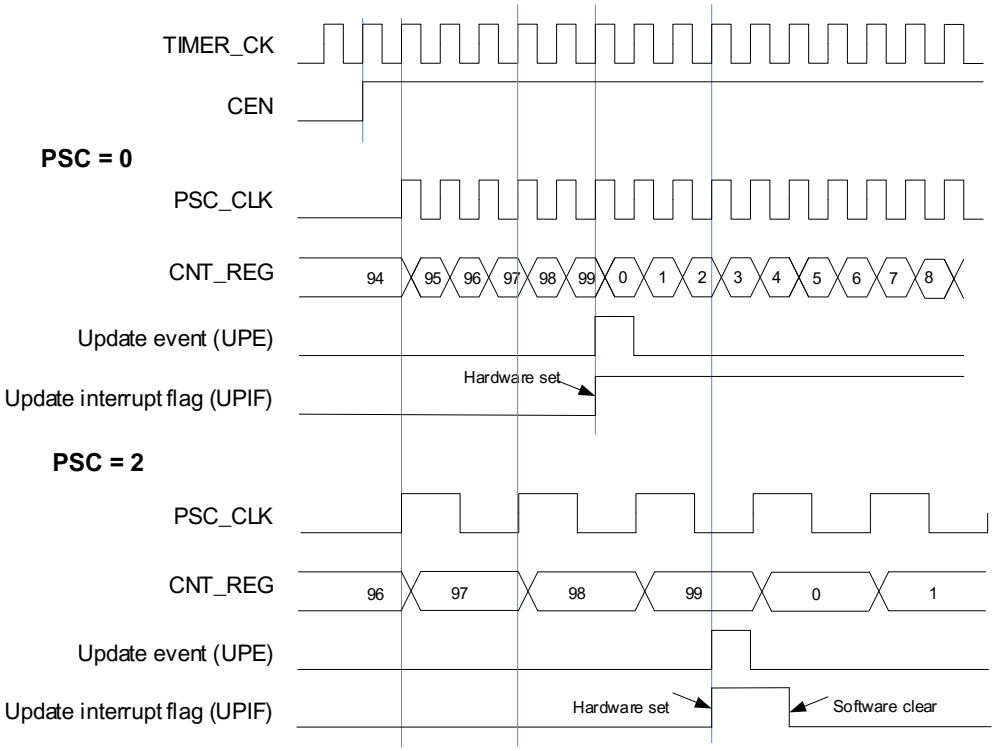
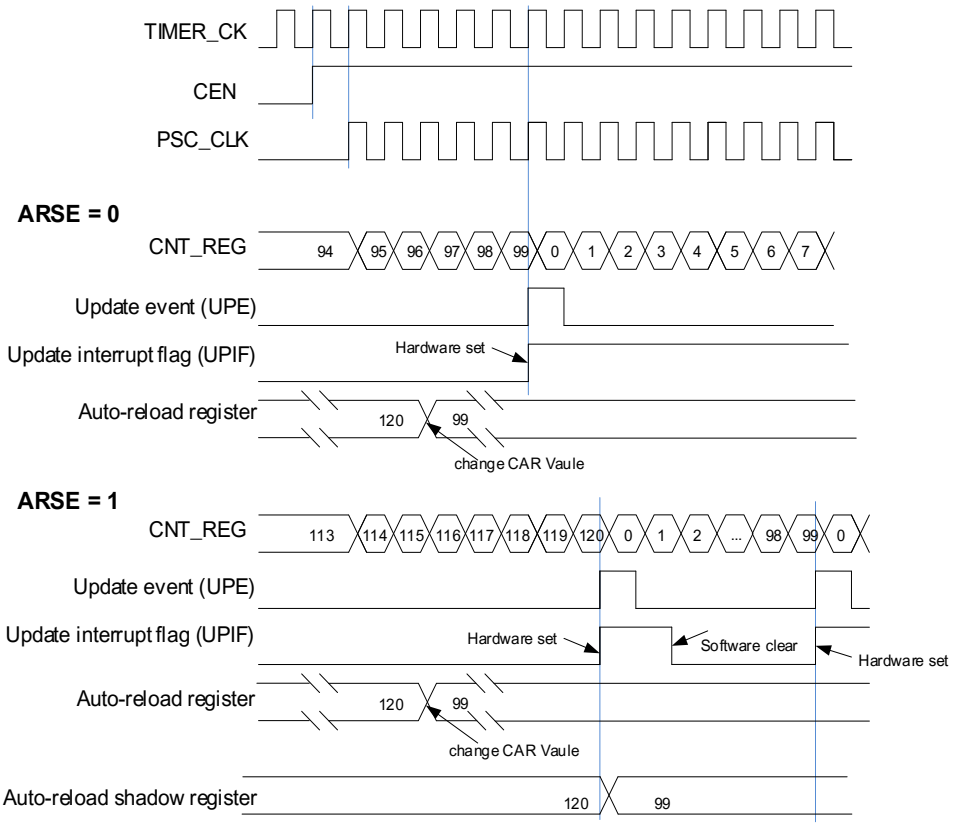


Figure 15-58. Timing chart of up counting, change TIMERx_CAR ongoing



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 31-6 and *Figure 31-7* show some examples of the counter behavior for different clock frequencies when TIMERx_CAR = 0x99.

Figure 15-59. Timing chart of down counting mode, PSC=0/2

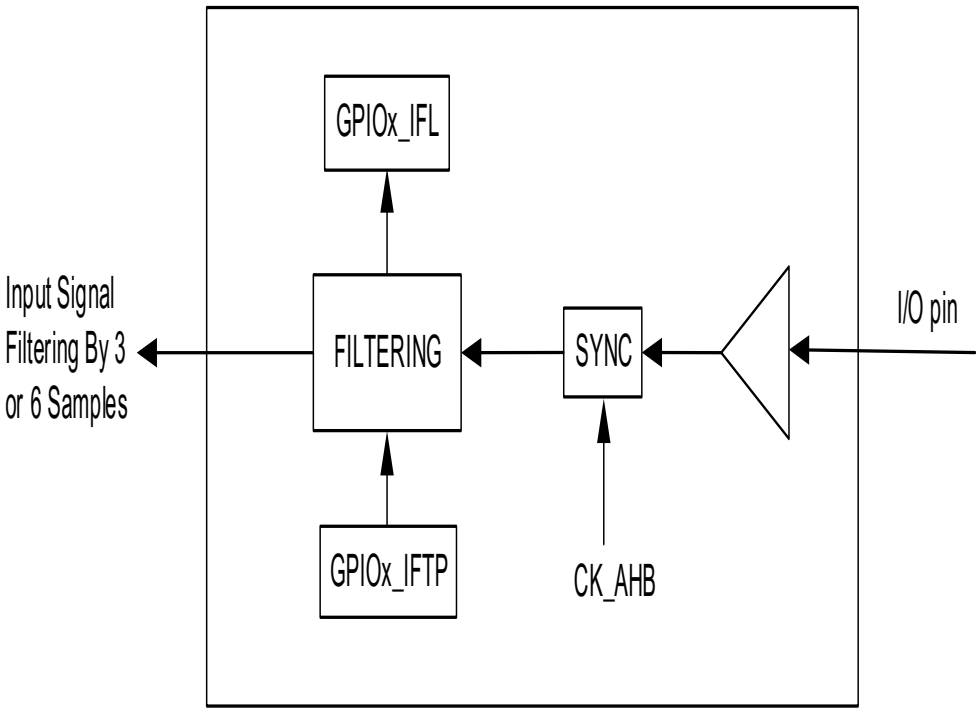
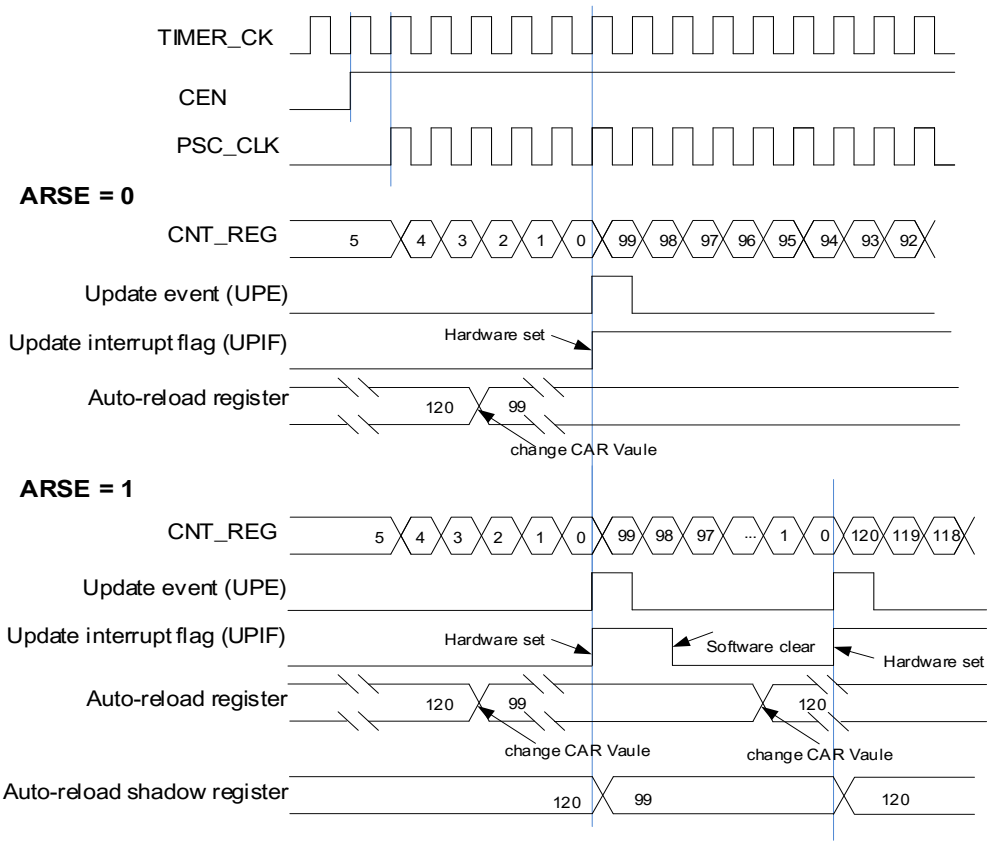


Figure 15-60. Timing chart of down counting mode, change TIMERx_CAR ongoing



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.

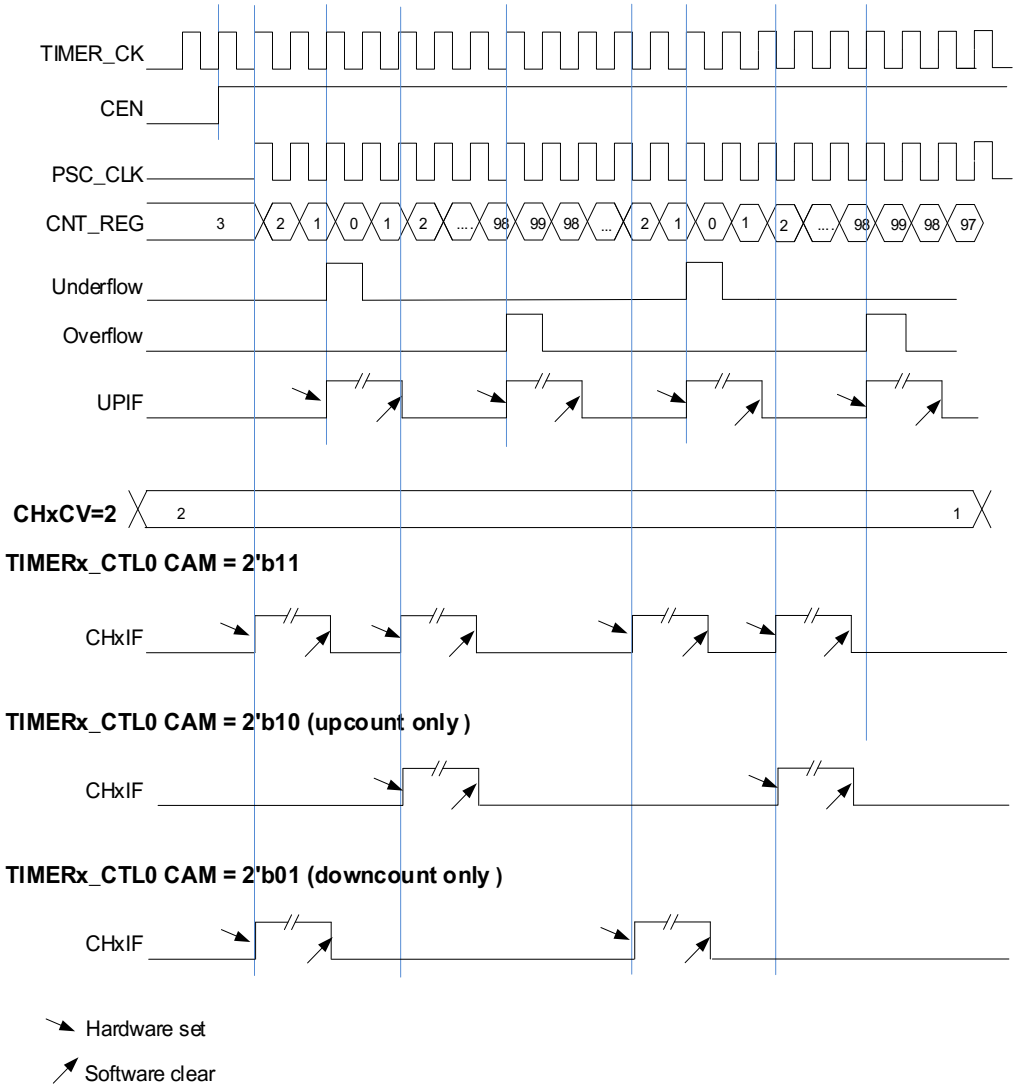
The UPIF bit in the TIMERx_INTF register will be set to 1 either when an underflow event or an overflow event occurs. While the ChxIF bit is associated with the value of CAM in TIMERx_CTL0. The details refer to ***Figure 15-61. Timing chart of center-aligned counting mode***.

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 15-61. Timing chart of center-aligned counting mode* shows the example of the counter behavior when TIMERx_CAR=0x99, TIMERx_PSC=0x0.

Figure 15-61. Timing chart of center-aligned counting mode



Capture/compare channels

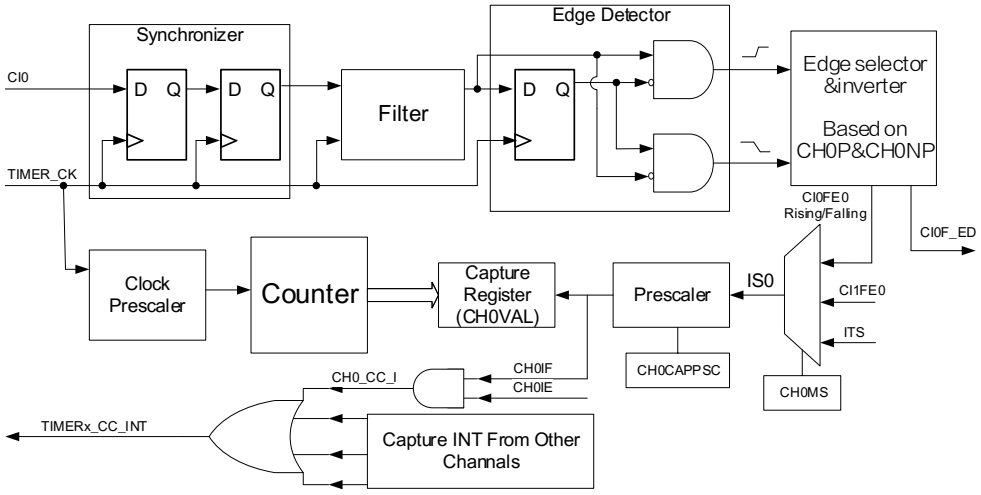
The general level0 timer has four independent channels which can be used as capture inputs or compare match outputs. Each channel is built around a channel capture compare register including an input stage, channel controller and an output stage.

■ Input capture mode

Input capture mode allows the channel to perform measurements such as pulse timing, frequency, period, duty cycle and so on. The input stage consists of a digital filter, a channel polarity selection, edge detection and a channel prescaler. When a selected edge occurs on the channel input, the current value of the counter is captured into the TIMERx_CHxCV register, at the same time the ChxIF bit is set and the channel interrupt is generated if it is

enabled when ChxIE=1.

Figure 15-62. Input capture logic



The input signals of channelx (Cix) can be the TIMERx_CHx signal or the XOR signal of the TIMERx_CH0, TIMERx_CH1 and TIMERx_CH2 signals (just for CI0). First, the input signal of channel (Cix) is synchronized to TIMER_CK signal, and then sampled by a digital filter to generate a filtered input signal. Then through the edge detector, the rising or falling edge is detected by configuring CHxP bit. The input capture signal can also be selected from the input signal of other channel or the internal trigger signal by configuring CHxMS bits. The IC prescaler makes several input events generate one effective capture event. On the capture event, TIMERx_CHxCV will store the value of counter.

So, the process can be divided into several steps as below:

Step1: Filter configuration (CHxCAPFLT in TIMERx_CHCTL0).
Based on the input signal and quality of requested signal, configure compatible CHxCAPFLT.

Step2: Edge selection (CHxP and CHxNP bits in TIMERx_CHCTL2).
Rising edge or falling edge, choose one by configuring CHxP and CHxNP bits.

Step3: Capture source selection (CHxMS in TIMERx_CHCTL0)
As soon as selecting one input capture source by CHxMS, the channel must be set to input mode (CHxMS! =0x0) and TIMERx_CHxCV cannot be written any more.

Step4: Interrupt enable (ChxIE and CHxDEN in TIMERx_DMAINTEN)
Enable the related interrupt to get the interrupt and DMA request.

Step5: Capture enable (ChxEN in TIMERx_CHCTL2)

Result: When the wanted input signal is captured, TIMERx_CHxCV will be set by counter's value and ChxIF is asserted. If the ChxIF is 1, the ChxOF will also be asserted. The interrupt and DMA request will be asserted or not based on the configuration of ChxIE and CHxDEN in TIMERx_DMAINTEN.

Direct generation: A DMA request or interrupt is generated by setting CHxG directly.

The input capture mode can be also used for pulse width measurement from signals on the TIMERx_CHx pins. For example, PWM signal connects to CI0 input. Select CI0 as channel 0 capture signals by setting CH0MS to 3'b001 in the channel control register (TIMERx_CHCTL0) and set capture on rising edge. Select CI0 as channel 1 capture signal by setting CH1MS to 3'b010 in the channel control register (TIMERx_CHCTL0) and set capture on falling edge. The counter is set to restart mode and is restarted on channel 0 rising edge. Then the TIMERX_CH0CV can measure the PWM period and the TIMERx_CH1CV can measure the PWM duty cycle.

- **Output compare mode**

Figure 15-63. Output compare logic (x=0,1,2,3)

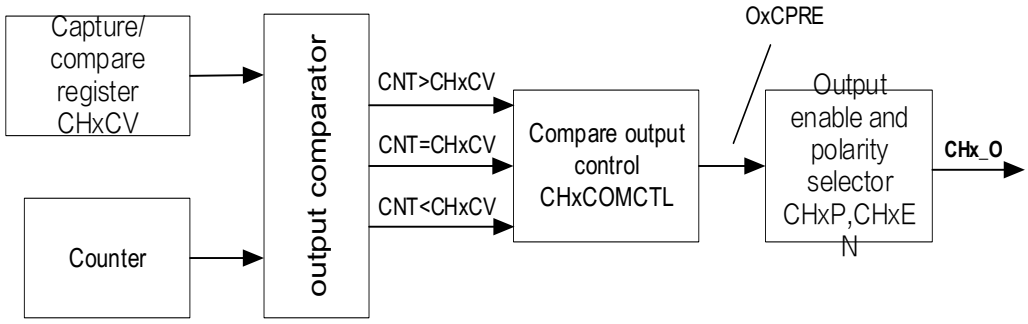


Figure 15-63. Output compare logic (x=0,1,2,3) shows the logic circuit of output compare mode. The relationship between the channel output signal CHx_O and the OxCPRE signal (more details refer to *Channel output prepare signal*) is described as below: The active level of O0CPRE is high, the output level of CH0_O depends on OxCPRE signal, CHxP bit and CH0P bit (please refer to the TIMERx_CHCTL2 register for more details).For example, configure CHxP=0 (the active level of CHx_O is high, the same as OxCPRE), ChxEN=1 (the output of CHx_O is enabled),

If the output of OxCPRE is active(high) level, the output of CHx_O is active(high) level;

If the output of OxCPRE is inactive(low) level, the output of CHx_O is active(low) level.

In output compare mode, the TIMERx can generate timed pulses with programmable position, polarity, duration and frequency. When the counter matches the value in the TIMERx_CHxCV register of an output compare channel, the channel (n) output can be set, cleared, or toggled based on CHxCOMCTL. When the counter reaches the value in the TIMERx_CHxCV register, the ChxIF bit will be set and the channel (n) interrupt is generated if ChxIE = 1. And the DMA request will be asserted, if CHxDEN=1.

So, the process can be divided into several steps as below:

Step1: Clock configuration. Such as clock source, clock prescaler and so on.

Step2: Compare mode configuration.

- Set the shadow enable mode by CHxCOMSEN.
- Set the output mode (set/clear/toggle) by CHxCOMCTL.
- Select the active polarity by CHxP.
- Enable the output by ChxEN.

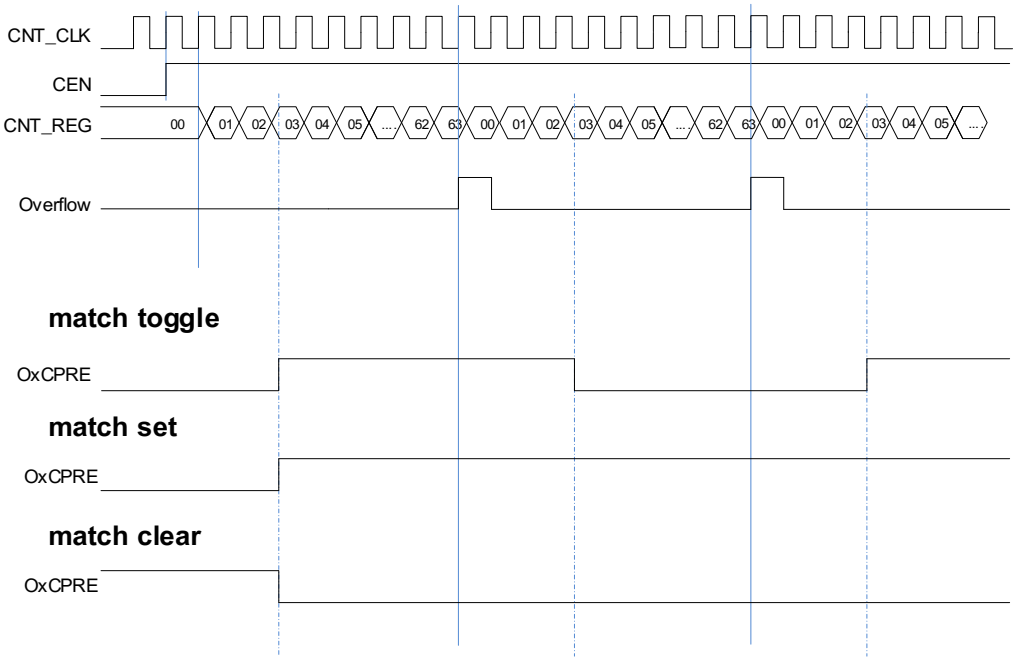
Step3: Interrupt/DMA-request enables configuration by ChxIE/CHxDEN.

Step4: Compare output timing configuration by TIMERx_CAR and TIMERx_CHxCV.
The TIMERx_CHxCV can be changed onging to meet the expected waveform.

Step5: Start the counter by configuring CEN to 1.

Figure 15-64. Output-compare under three modes shows the three compare modes toggle/set/clear. CAR=0x63, CHxVAL=0x3

Figure 15-64. 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 15-65. Timing chart of EAPWM* shows the EAPWM output and interrupts waveform.

The CAPWM period is determined by 2*TIMERx_CAR, and duty cycle is determined by 2*TIMERx_CHxCV. *Figure 15-66. 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 15-65. Timing chart of EAPWM

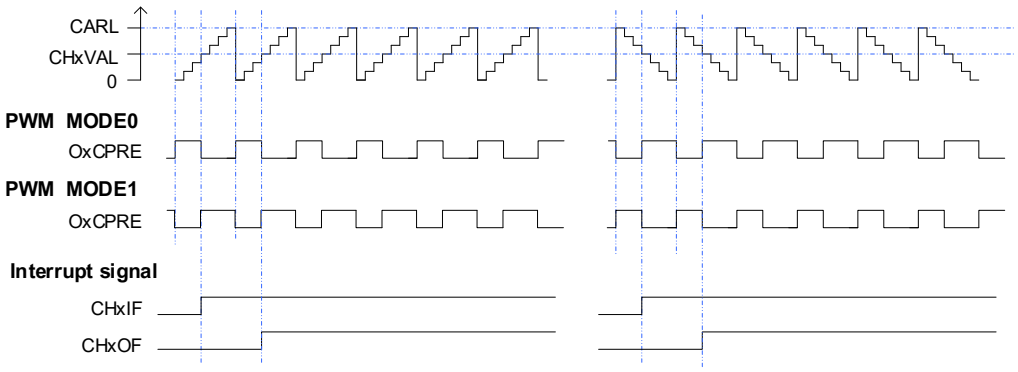
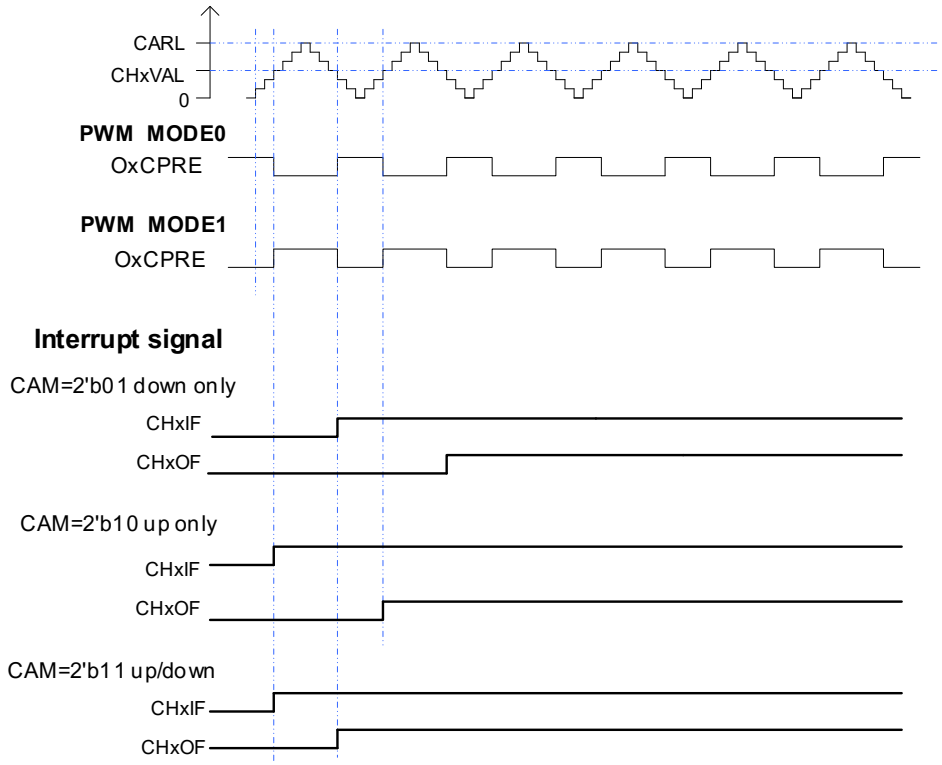


Figure 15-66. Timing chart of CAPWM



Composite PWM mode

In the Composite PWM mode (CHxCPWMEN = 1'b1, CHxMS[2:0] = 3'b000 and CHxCOMCTL = 4'b0110 or 4'b0111), the PWM signal output in channel x (x=0..3) is composited by CHxVAL and CHxCOMVAL_ADD bits.

If CHxCOMCTL = 4'b0110 (PWM mode 0) and DIR = 1'b0 (up counting mode), or CHxCOMCTL = 4'b0111 (PWM mode 1) and DIR = 1'b1 (Down counting mode), the channel x output is forced low when the counter matches the value of CHxVAL. It is forced high when the counter matches the value of CHxCOMVAL_ADD.

If CHxCOMCTL = 4'b0111 (PWM mode 1) and DIR = 1'b0 (up counting mode), or CHxCOMCTL = 4'b0110 (PWM mode 0) and DIR = 1'b1 (down counting mode) the channel x output is forced high when the counter matches the value of CHxVAL. It is forced low when the counter matches the value of CHxCOMVAL_ADD.

When CHxVAL or CHxCOMVAL_ADD = 0 / CARL, the channel x output is specially processed, and by setting the CHxPERFOREN bit in the TIMERx_CHCTL2 register, the OxCPRE output is forced to either high or low level (determined according to the selected composite PWM mode).

The PWM period is determined by (CARL + 0x0001) and the PWM pulse width is determined by the following table.

Table 15-12.The Composite PWM pulse width

Condition	Mode	PWM pulse width
CHxVAL < CHxCOMVAL_ADD <= CARL	PWM mode 0	(CARL + 0x0001) + (CHxVAL - CHxCOMVAL_ADD)
	PWM mode 1	(CHxCOMVAL_ADD - CHxVAL)
CHxCOMVAL_ADD < CHxVAL <= CARL	PWM mode 0	(CHxVAL - CHxCOMVAL_ADD)
	PWM mode 1	(CARL + 0x0001) + (CHxCOMVAL_ADD - CHxVAL)
(CHxVAL = CHxCOMVAL_ADD <= CARL) or (CHxVAL > CARL > CHxCOMVAL_ADD)	PWM mode 0 (up counting) or PWM mode 1 (down counting)	100%
	PWM mode 0 (down counting) or PWM mode 1 (up counting)	0%
CHxCOMVAL_ADD > CARL > CHxVAL	PWM mode 0(up counting) or PWM mode 1(down counting)	0%
	PWM mode 0(down counting) or PWM mode 1(up counting)	100%
(CHxVAL>CARL) and (CHxCOMVAL_ADD > CARL)	-	The output of CHx_O is keeping

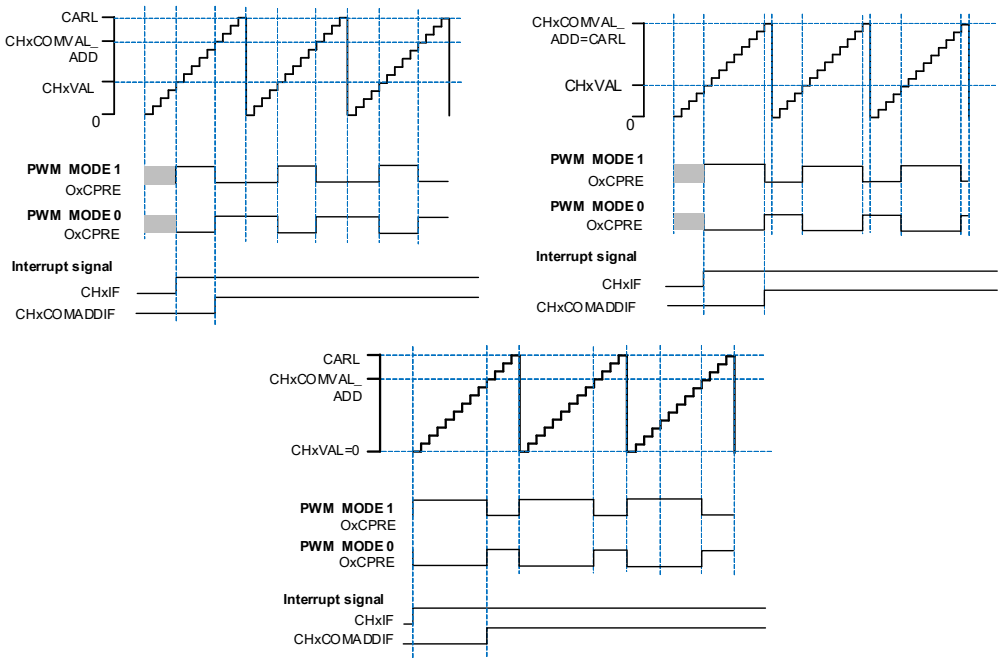
When the counter reaches the value of CHxVAL, the ChxIF bit is set and the channel x interrupt is generated if ChxIE = 1, and the DMA request will be asserted, if CHxDEN=1. When the counter reaches the value of CHxCOMVAL_ADD, the CHxCOMADDIF bit is set (this flag just used in composite PWM mode, when CHxCPWMEN=1) and the channel x additional compare interrupt is generated if CHxCOMADDIE = 1 (Only interrupt is generated,

no DMA request is generated).

According to the relationship among CHxVAL, CHxCOMVAL_ADD and CARL, it can be divided into four situations:

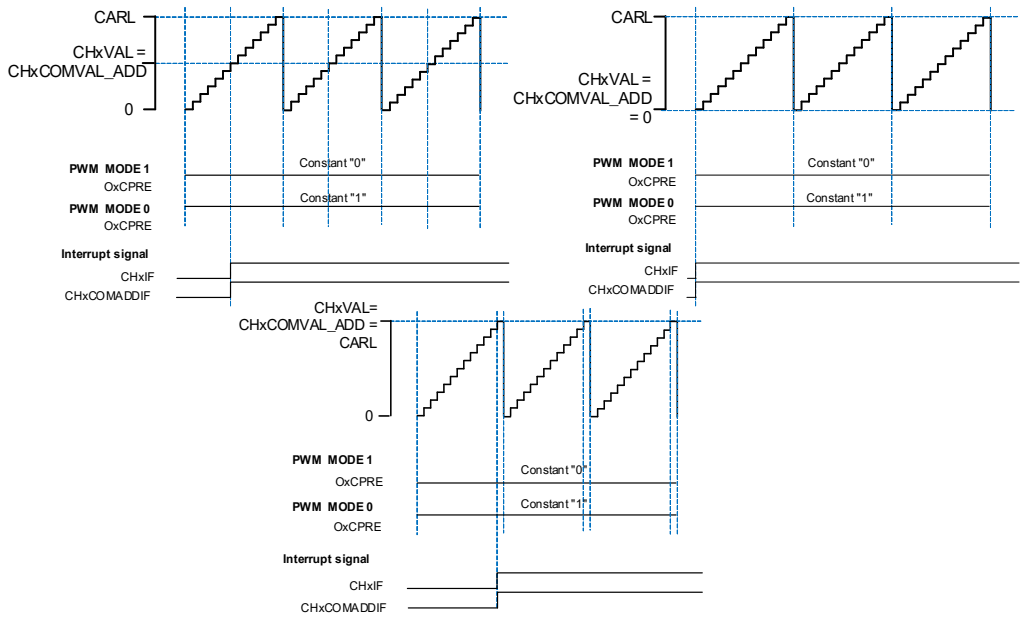
5) CHxVAL < CHxCOMVAL_ADD, and the values of CHxVAL and CHxCOMVAL_ADD between 0 and CARL.

Figure 15-67. Channel x output PWM with (CHxVAL < CHxCOMVAL_ADD)



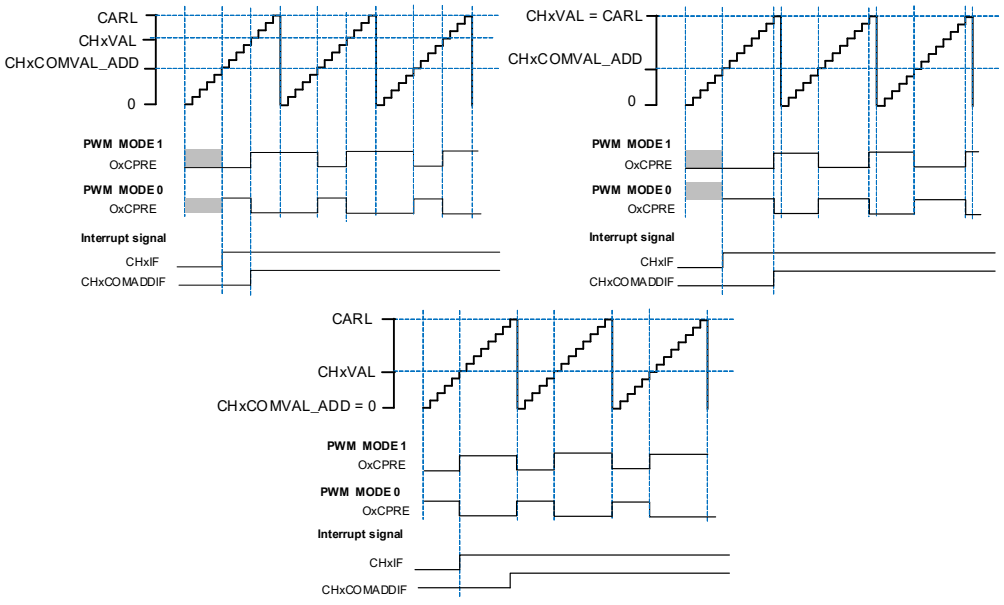
6) CHxVAL = CHxCOMVAL_ADD, and the value of CHxVAL and CHxCOMVAL_ADD between 0 and CARL.

Figure 15-68. Channel x output PWM with (CHxVAL = CHxCOMVAL_ADD)



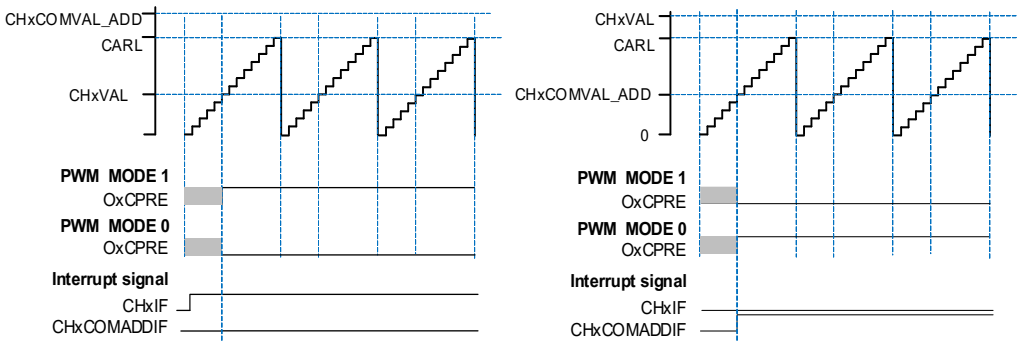
7) CHxVAL > CHxCOMVAL_ADD, and the value of CHxVAL and CHxCOMVAL_ADD between 0 and CARL.

Figure 15-69. Channel x output PWM with (CHxVAL > CHxCOMVAL_ADD)



8) One of the value of CHxVAL and CHxCOMVAL_ADD exceeds CARL.

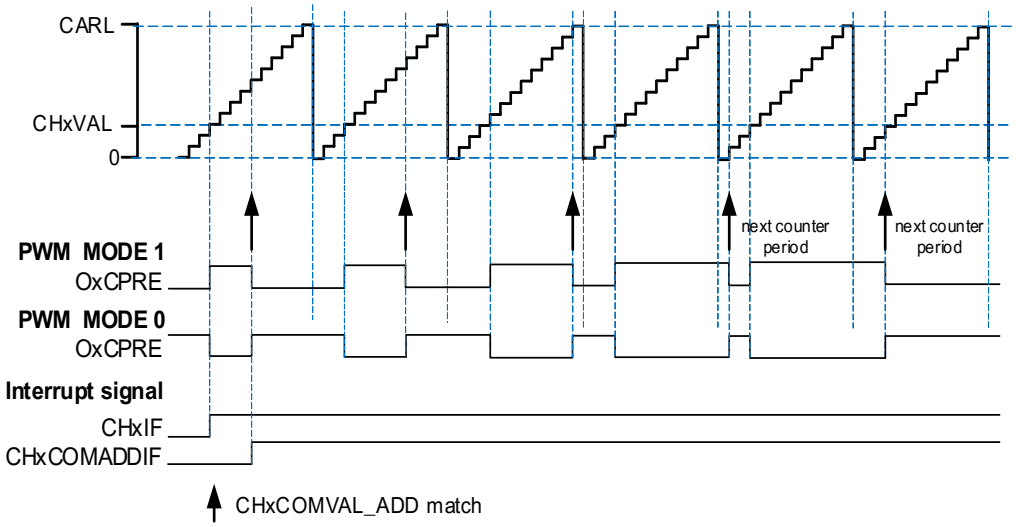
Figure 15-70. Channel x output PWM with CHxVAL or CHxCOMVAL_ADD exceeds CARL



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 15-71. Channel x output PWM duty cycle changing with CHxCOMVAL_ADD* shows the PWM output and interrupts waveform.

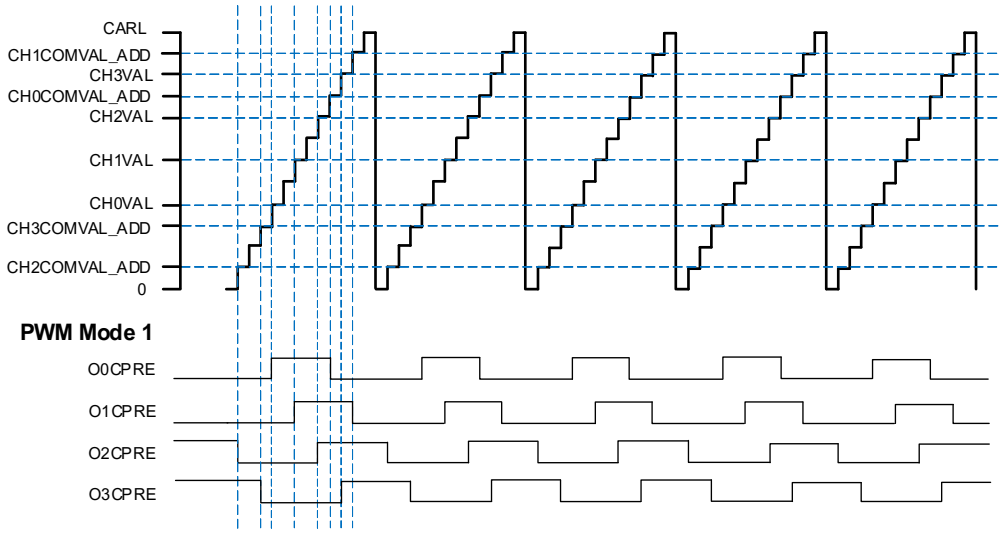
In some cases, the CHxCOMVAL_ADD match can happen on the next counter period (the value of CHxCOMVAL_ADD was written after the counter reaches the value of CHxVAL, and the value of CHxCOMVAL_ADD was less than or equal to the CHxVAL).

Figure 15-71. 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 15-72. 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 15-73. CHx_O output with a pulse in edge-aligned mode (CHxOMPSEL!=2'b00)

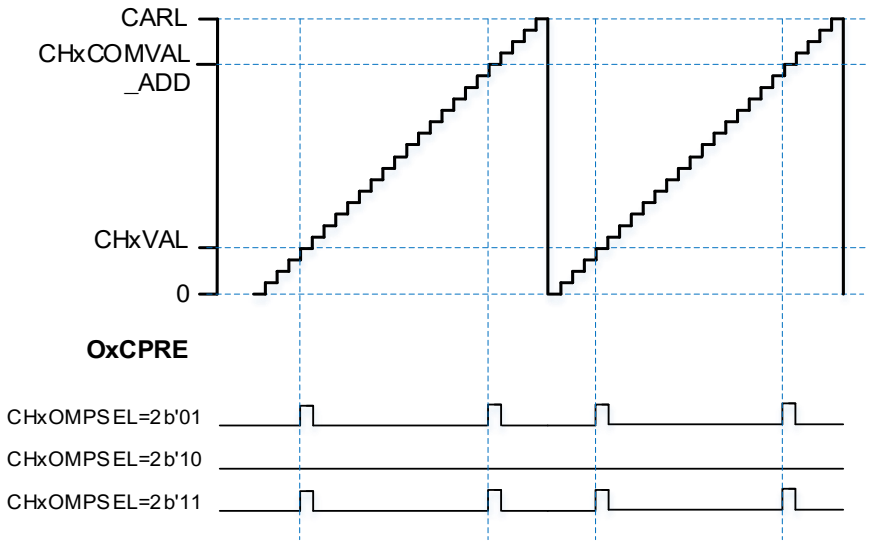
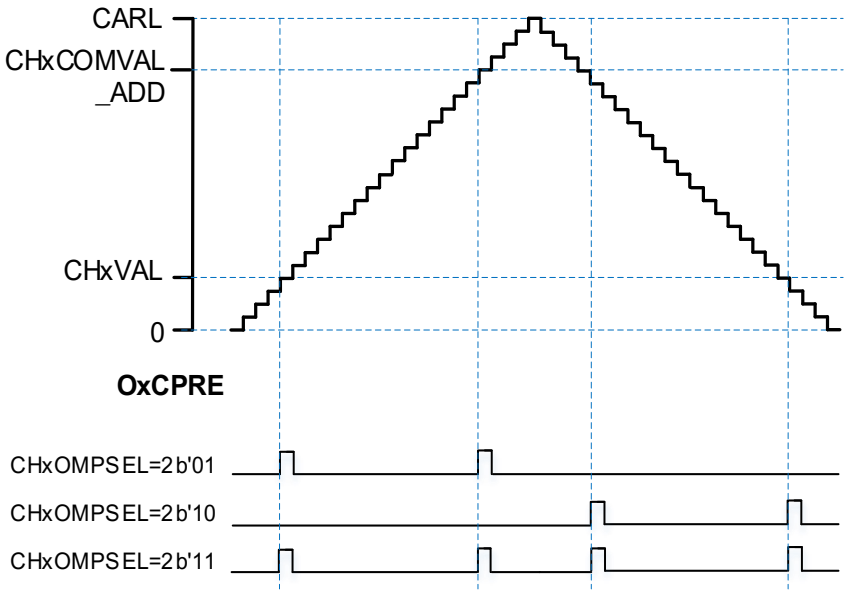


Figure 15-74. CHx_O output with a pulse in center-aligned mode (CHxOMPSEL!=2'b00)



Channel output prepare signal

As is shown in *Figure 15-63. 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 (OCPRECIVM = 1) or underflow/overflow event occurs (OCPRECIVM = 0).

Quadrature decoder

Please refer to ***Advanced timer (TIMERx, x=0, 7) Quadrature decoder***.

Non-quadrature decoder

Please refer to ***Advanced timer (TIMERx, x=0, 7) Non-quadrature decoder***.

Quadrature decoder and non-quadrature decoder clock output

Please refer to ***Advanced timer (TIMERx, x=0, 7) Quadrature decoder and non-quadrature decoder clock output***.

Hall sensor function

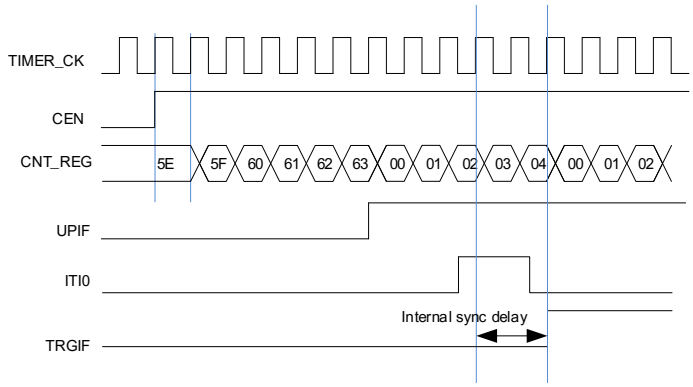
Refer to ***Advanced timer (TIMERx, x=0, 7)Hall sensor function***.

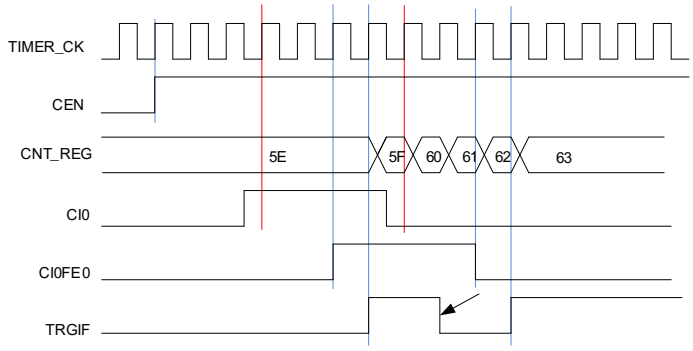
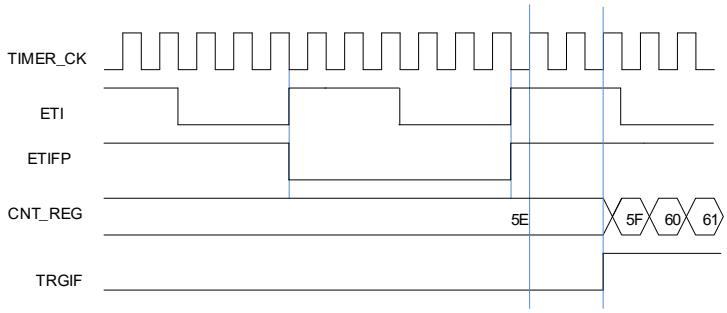
Master-slave management

The TIMERx can be synchronized with a trigger in several modes including restart mode, pause mode and event mode and so on, which is selected by the TSCFGy[4:0] (y=3..7) in SYSCFG_TIMERxCFG(x=1..4,22,23,30,31).

Table 15-13. Examples of slave mode

	Mode Selection	Source Selection	Polarity Selection	Filter and Prescaler
LIST	TSCFGy[4:0] y=3: restart mode y=4: pause mode y=5: event mode y=6: external clock mode 0	TSCFGy[4:0] 00000:Mode disable 00001: ITI0 00010: ITI1 00011: ITI2	If CI0FE0 or CI1FE1 is selected as the trigger source, configure the CHxP and CHxNP for the 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

<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>y=7: restart + event mode</td><td>00100: ITI3
00101: CI0F_ED
00110: CI0FE0
00111: CI1FE1
01000: ETIFP(1)
01001: ITI4
01010: ITI5
01100: ITI7
01110: ITI9
01111: ITI10
10000: ITI11
10001: ITI12
10010: ITI13
10011: ITI14</td><td>If ETIFP (the filtered output of external trigger input ETI) is selected as the trigger source, configure the ETP for polarity selection and inversion.</td><td>prescaler can be used. For the ETIFP, filter can be used by configuring ETFC and prescaler can be used by configuring ETPSC.</td></tr>
<tr><td rowspan="2">Exam1</td><td>Restart mode
The counter will be cleared and restart when a rising edge of trigger input comes.</td><td>TSCFG3[4:0] 5'b00001, ITI0 is selected.</td><td>For ITI0, no polarity selector can be used.</td><td>For the ITI0, no filter and prescaler can be used.</td></tr>
<tr><td colspan="4">Figure 15-75. Restart mode

</td></tr>
<tr><td>Exam2</td><td>Pause mode
The counter will be paused when the trigger input is low, and it will start when the trigger input is high.</td><td>TSCFG4[4:0] =5'b00110, CI0FE0 is selected.</td><td>TI0S = 0 (Non-xor) [CH0NP=0, CH0P=0] CI0FE0 does not invert. The capture event will occur on the rising edge only.</td><td>Filter is bypassed in this example.</td></tr>
</table>

<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 15-76. 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 15-77. Event mode

</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 TIMER1_ETI register (TRIGSEL_TIMER1ETI)* 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.

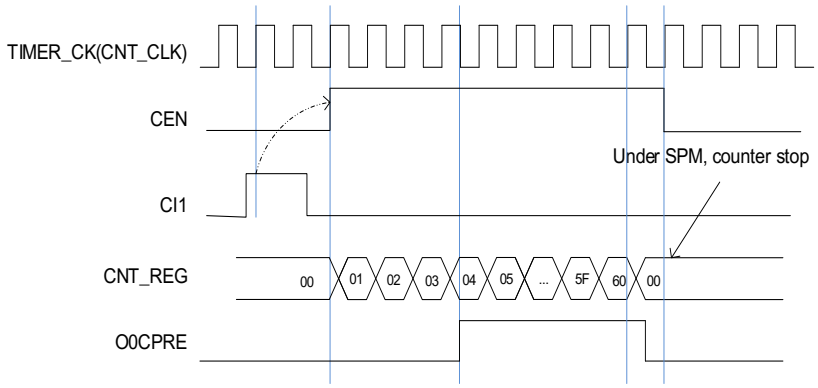
Once the timer is set to the single pulse mode, it is not necessary to configure the timer enable bit CEN in the TIMERx_CTL0 register to 1 to enable the counter. Setting the CEN bit to 1 or a trigger signal edge can generate a pulse and then keep the CEN bit at a high state until the update event occurs 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 15-78. 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..4,22,23,30,31);
- The CHxCOMCTL[3:0] bit-field is setting to 4'b1000 (delayable single pulse mode 0) or 4'b1001 (delayable single pulse mode 1).

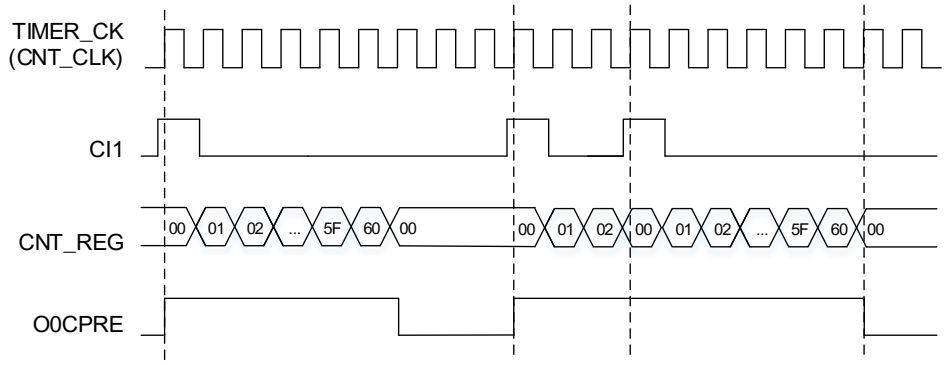
In delayable SPM mode 0. The behavior of OxCPREis performed as in PWM mode 0. When counting up, the OxCPRE is active. When a trigger event occurs, the OxCPRE is inactive. The OxCPRE is active again at the next update event; When counting down, the OxCPRE is inactive, when a trigger event occurs, the 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.

Note:

1) The center-aligned counting mode cannot be used in this mode and the CAM[1:0] = 2'b00(in TIMERx_CTL0 register);
2) 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 15-79. delayable single pulse mode TIMERx_CHxCV=0x00, TIMERx_CAR=0x60



Timers interconnection

Please refer to *Advanced timer (TIMERx, x=0, 7)Timers interconnection*.

Timer DMA mode

Timer DMA mode is the function that configures timer's register by DMA module. The relative registers are TIMERx_DMACFG and TIMERx_DMATB. Corresponding DMA request bit should be asserted to enable DMA request for internal interrupt event. TIMERx will send a request to DMA when the interrupt event occurs. DMA is configured to M2P (memory to peripheral) mode and the address of TIMERx_DMATB is configured to PADDR (peripheral base address), then DMA will access the TIMERx_DMATB. In fact, TIMERx_DMATB register is only a buffer, timer will map the TIMERx_DMATB to an internal register, appointed by the field of DMATA in TIMERx_DMACFG. If the field of DMATC in TIMERx_DMACFG is 0 (1 transfer), the timer sends only one DMA request. While if TIMERx_DMATC is not 0, such as 3 (4 transfers), then timer will send 3 more requests to DMA, and DMA will access timer's registers DMATA+0x4, DMATA+0x8 and DMATA+0xC at the next 3 accesses to TIMERx_DMATB. In a word, one-time DMA internal interrupt event asserts, (DMATC+1) times request will be sent by TIMERx.

If one more DMA request event occurs, TIMERx will repeat the process above.

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®-M7 is halted, and the TIMERx_HOLD configuration bit in DBG register set to 1, the TIMERx counter stops.

15.2.5. Registers definition (TIMERx, x=1,2,3,4,22,23,30,31)

TIMER1 base address: 0x4000 0000

TIMER2 base address: 0x4000 0400

TIMER3 base address: 0x4000 0800

TIMER4 base address: 0x4000 0C00

TIMER22 base address: 0x4000 E000

TIMER23 base address: 0x4000 E400

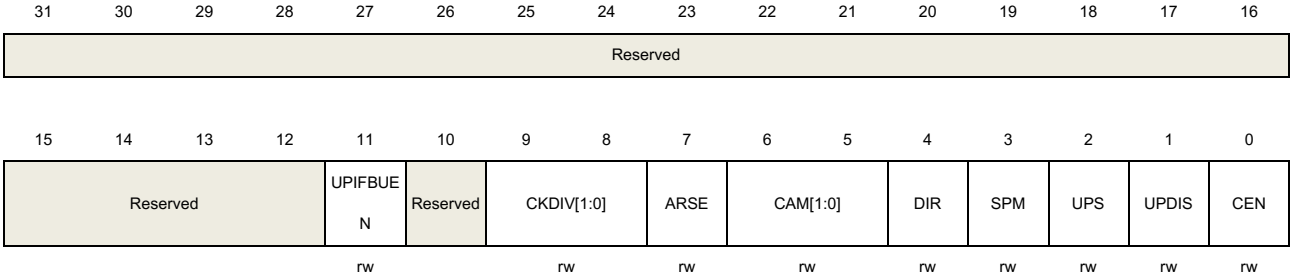
TIMER30 base address: 0x4000 E800

TIMER31 base address: 0x4000 EC00

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
31:12	Reserved	Must be kept at reset value.
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 UPG bit is set or the slave mode controller generates a hardware reset event.

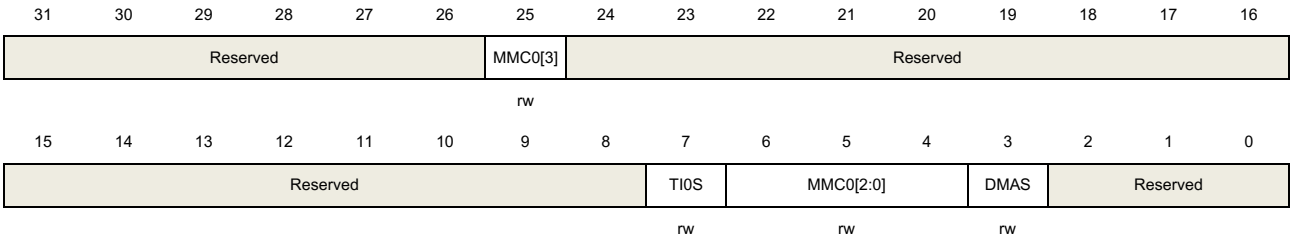
0	CEN	Counter enable 0: Counter disable 1: Counter enable The CEN bit must be set by software when timer works in external clock mode, pause mode or decoder mode. While in event mode, the hardware can set the CEN bit automatically.

Control register 1 (TIMERx_CTL1)

Address offset: 0x04

Reset value: 0x0000 0000

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



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 TRGO0 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. 0011: 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 non-quadrature decoder mode 0~3. 1001~1111: Reserved.
3	DMAS	DMA request source selection 0: DMA request of CHx is sent when capture/compare event occurs. 1: DMA request of channel CHx is sent when update event occurs.
2:0	Reserved	Must be kept at reset value.

Slave mode configuration register (TIMERx_SMCFG)

Address offset: 0x08

Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	SMC1	ETPSC[1:0]		ETFC[3:0]				MSM	Reserved						
rw	rw	rw		rw				rw							

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	ETP	External trigger polarity This bit specifies the polarity of ETI signal.

		0: ETI is active at high level or rising edge. 1: ETI is active at low level or falling edge.
14	SMC1	Part of slave mode controller is used to enable external clock mode 1 In external clock mode 1, the counter is clocked by any active edge of the ETIFP signal. 0: External clock mode 1 disabled 1: External clock mode 1 enabled It is possible to simultaneously use external clock mode 1 with the restart mode, pause mode or event mode. But the TSCFGy[4:0](y=3,4,5) bits must not be 5'b01000 in this case. The external clock input will be ETIFP if external clock mode 0 and external clock mode 1 are enabled at the same time. **Note:** External clock mode 0 enable is in TSCFG6[4:0] bit-field in SYSCFG_TIMERxCFG1 register.
13:12	ETPSC[1:0]	External trigger prescaler The frequency of external trigger signal ETIFP must not be higher than 1/4 of TIMER_CK frequency. When the frequency of external trigger signal is high, the prescaler can be enabled to reduce ETIFP frequency. 00: Prescaler disabled 01: ETIFP frequency divided by 2 10: ETIFP frequency divided by 4 11: ETIFP frequency divided by 8
11:8	ETFC[3:0]	External trigger filter control An event counter is used in the digital filter, in which a transition on the output occurs after N input events. This bit-field specifies the frequency used to sample ETIFP signal and the length of the digital filter applied to ETIFP. 0000: Filter disabled. F_{SAMP}= f_{DTS}, N=1. 0001: f_{SAMP}= f_{CK_TIMER}, N=2. 0010: f_{SAMP}= f_{CK_TIMER}, N=4. 0011: f_{SAMP}= f_{CK_TIMER}, N=8. 0100: f_{SAMP}=f_{DTS}/2, N=6. 0101: f_{SAMP}=f_{DTS}/2, N=8. 0110: f_{SAMP}=f_{DTS}/4, N=6. 0111: f_{SAMP}=f_{DTS}/4, N=8. 1000: f_{SAMP}=f_{DTS}/8, N=6. 1001: f_{SAMP}=f_{DTS}/8, N=8. 1010: f_{SAMP}=f_{DTS}/16, N=5. 1011: f_{SAMP}=f_{DTS}/16, N=6. 1100: f_{SAMP}=f_{DTS}/16, N=8. 1101: f_{SAMP}=f_{DTS}/32, N=5. 1110: f_{SAMP}=f_{DTS}/32, N=6. 1111: f_{SAMP}=f_{DTS}/32, N=8.

7	MSM	Master-slave mode This bit can be used to synchronize the selected timers to begin counting at the same time. The TRGI is used as the start event, and through TRGO, timers are connected. 0: Master-slave mode disabled 1: Master-slave mode enabled
6:0	Reserved	Must be kept at reset value.

DMA and interrupt enable register (TIMERx_DMAINTEN)

Address offset: 0x0C

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH3COMADDIE	CH2COMADDIE	CH1COMADDIE	CH0COMADDIE	Reserved										DECDISIE	DECJIE
rw	rw	rw	rw											rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	TRGDEN	Reserved	CH3DEN	CH2DEN	CH1DEN	CH0DEN	UPDEN	Reserved	TRGIE	Reserved	CH3IE	CH2IE	CH1IE	CH0IE	UPIE
	rw		rw	rw	rw	rw	rw		rw		rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31	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.
27:18	Reserved	Must be kept at reset value.
17	DECDISIE	Quadrature decoder signal disconnection interrupt enable

		0: Disabled 1: Enabled **Note:** This bit just used for quadrature decoder signal disconnection detection is enabled (when DECDISDEN =1).
16	DECJIE	Quadrature decoder signal jump (the two signals jump at the same time) interrupt enable 0: Disabled 1: Enabled **Note:** This bit just used for quadrature decoder signal jump detection is enabled (when DECJDEN =1).
15	Reserved	Must be kept at reset value.
14	TRGDEN	Trigger DMA request enable 0: Disabled 1: Enabled
13	Reserved	Must be kept at reset value.
12	CH3DEN	Channel 3 capture/compare DMA request enable 0: Disabled 1: Enabled
11	CH2DEN	Channel 2 capture/compare DMA request enable 0: Disabled 1: Enabled
10	CH1DEN	Channel 1 capture/compare DMA request enable 0: Disabled 1: Enabled
9	CH0DEN	Channel 0 capture/compare DMA request enable 0: Disabled 1: Enabled
8	UPDEN	Update DMA request enable 0: Disabled 1: Enabled
7	Reserved	Must be kept at reset value.
6	TRGIE	Trigger interrupt enable 0: Disabled 1: Enabled
5	Reserved	Must be kept at reset value.
4	CH3IE	Channel 3 capture/compare interrupt enable 0: Disabled

		1: Enabled
3	CH2IE	Channel 2 capture/compare interrupt enable 0: Disabled 1: Enabled
2	CH1IE	Channel 1 capture/compare interrupt enable 0: Disabled 1: Enabled
1	CH0IE	Channel 0 capture/compare interrupt enable 0: Disabled 1: Enabled
0	UPIE	Update interrupt enable 0: Disabled 1: Enabled

Interrupt flag register (TIMERx_INTF)

Address offset: 0x10

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH3COMADDIF	CH2COMADDIF	CH1COMADDIF	CH0COMADDIF	Reserved										DECDISIF	DECJIF
rc_w0	rc_w0	rc_w0	rc_w0											rc_w0	rc_w0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			CH3OF	CH2OF	CH1OF	CH0OF	Reserved		TRGIF	Reserved	CH3IF	CH2IF	CH1IF	CH0IF	UPIF
			rc_w0	rc_w0	rc_w0	rc_w0			rc_w0		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

Bits	Fields	Descriptions
31	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).
27:18	Reserved	Must be kept at reset value.
17	DECDISIF	Quadrature decoder signal disconnection interrupt flag 0: No quadrature decoder signal disconnection interrupt occurred 1: Quadrature decoder signal disconnection interrupt occurred **Note:** This bit just used for quadrature decoder signal disconnection detection is enabled (when DECDISDEN =1).
16	DECJIF	Quadrature decoder signal jump (the two signals jump at the same time) interrupt flag 0: No quadrature decoder signal jump interrupt occurred 1: Quadrature decoder signal jump interrupt occurred **Note:** This bit just used for quadrature decoder signal jump detection is enabled (when DECJDEN =1).
15:13	Reserved	Must be kept at reset value.
12	CH3OF	Channel 3 over capture flag Refer to CH0OF description
11	CH2OF	Channel 2 over capture flag Refer to CH0OF description
10	CH1OF	Channel 1 over capture flag Refer to CH0OF description
9	CH0OF	Channel 0 over capture flag When channel 0 is configured in input mode, this flag is set by hardware when a capture event occurs while CH0IF flag has already been set. This flag is cleared by software. 0: No over capture interrupt occurred 1: Over capture interrupt occurred
8:7	Reserved	Must be kept at reset value.
6	TRGIF	Trigger interrupt flag This flag is set by hardware on trigger event and cleared by software. When the slave mode controller is enabled in all modes but pause mode, an active edge of trigger input generates a trigger event. When the slave mode controller is enabled in pause mode, either edge of the trigger input can generate a trigger event. 0: No trigger event occurred 1: Trigger interrupt occurred
5	Reserved	Must be kept at reset value.
4	CH3IF	Channel 3 capture/compare interrupt flag

		Refer to CH0IF description
3	CH2IF	Channel 2 capture/compare interrupt flag Refer to CH0IF description
2	CH1IF	Channel 1 capture/compare interrupt flag Refer to CH0IF description
1	CH0IF	Channel 0 capture/compare interrupt flag This flag is set by hardware and cleared by software. If channel 0 is in input mode, this flag is set when a capture event occurs. If channel 0 is in output mode, this flag is set when a compare event occurs. If channel 0 is set to input mode, this bit will be reset by reading TIMERx_CH0CV. 0: No channel 0 interrupt occurred 1: Channel 0 interrupt occurred
0	UPIF	Update interrupt flag This bit is set by hardware when an update event occurs and cleared by software. 0: No update interrupt occurred 1: Update interrupt occurred

Software event generation register (TIMERx_SWEVG)

Address offset: 0x14

Reset value: 0x0000 0000

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

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

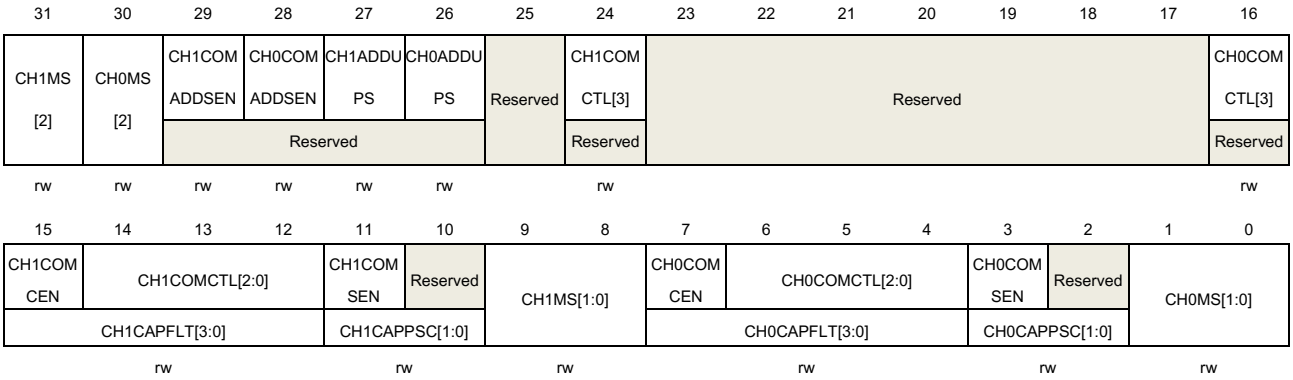
		automatically cleared by hardware. When this bit is set, the CH0COMADDIF flag will be set, and the corresponding interrupt will be sent if enabled. 0: No generate a channel 0 additional compare event 1: Generate a channel 0 additional compare event **Note:** This bit just used in composite PWM mode(when 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).



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	Reserved	Must be kept at reset value.
9:8	CH1MS[1:0]	Channel 1 mode selection This bit-field specifies the direction of the channel and the input signal selection. The CH1MS[2:0] bit-field is writable only when the channel is not active (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...4,22,23,30,31) 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. 1010~1111: 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	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 (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...4,22,23,30,31) register).
100~111: Reserved.

Input capture mode:

Bits	Fields	Descriptions
31	CH1MS[2]	Channel 1 I/O mode selection Same as output compare mode.
30	CH0MS[2]	Channel 0 I/O mode selection Same as output compare mode.
29:16	Reserved	Must be kept at reset value.
15:12	CH1CAPFLT[3:0]	Channel 1 input capture filter control Refer to CH0CAPFLT description.
11:10	CH1CAPPSC[1:0]	Channel 1 input capture prescaler Refer to CH0CAPPSC description.
9:8	CH1MS[1:0]	Channel 1 I/O mode selection Same as output compare mode.
7:4	CH0CAPFLT[3:0]	Channel 0 input capture filter control An event counter is used in the digital filter, in which a transition on the output occurs after N input events. This bit-field specifies the frequency used to sample CI0 input signal and the length of the digital filter applied to CI0. 0000: Filter disabled, $f_{SAMP}=f_{DTS}$, N=1. 0001: $f_{SAMP}=f_{CK_TIMER}$, N=2. 0010: $f_{SAMP}= f_{CK_TIMER}$, N=4. 0011: $f_{SAMP}= f_{CK_TIMER}$, N=8. 0100: $f_{SAMP}=f_{DTS}/2$, N=6. 0101: $f_{SAMP}=f_{DTS}/2$, N=8. 0110: $f_{SAMP}=f_{DTS}/4$, N=6. 0111: $f_{SAMP}=f_{DTS}/4$, N=8. 1000: $f_{SAMP}=f_{DTS}/8$, N=6. 1001: $f_{SAMP}=f_{DTS}/8$, N=8. 1010: $f_{SAMP}=f_{DTS}/16$, N=5. 1011: $f_{SAMP}=f_{DTS}/16$, N=6. 1100: $f_{SAMP}=f_{DTS}/16$, N=8. 1101: $f_{SAMP}=f_{DTS}/32$, N=5. 1110: $f_{SAMP}=f_{DTS}/32$, N=6. 1111: $f_{SAMP}=f_{DTS}/32$, N=8.
3:2	CH0CAPPSC[1:0]	Channel 0 input capture prescaler This bit-field specifies the factor of the prescaler on channel 0 input. The prescaler is reset when CH0EN bit in TIMERx_CHCTL2 register is cleared. 00: Prescaler disabled, capture is done on each channel input edge.

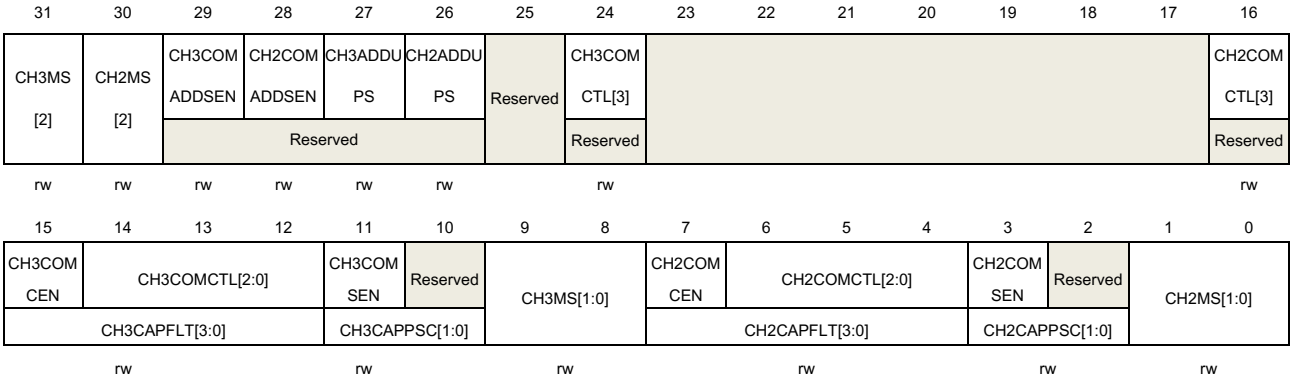
		01: Capture is done every 2 channel input edges. 10: Capture is done every 4 channel input edges. 11: Capture is done every 8 channel input edges.
1:0	CH0MS[1:0]	Channel 0 I/O mode selection Same as output compare mode.

Channel control register 1 (TIMERx_CHCTL1)

Address offset: 0x1C

Reset value: 0x0000 0000

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



Output compare mode:

Bits	Fields	Descriptions
31	CH3MS[2]	Channel 3 I/O mode selection Refer to CH3MS[1:0]description.
30	CH2MS[2]	Channel 2 I/O mode selection Refer to CH2MS[1:0] description.
29	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	Reserved	Must be kept at reset value.
9:8	CH3MS[1:0]	Channel 3 I/O mode selection This bit-field specifies the direction of the channel and the input signal selection. The CH3MS[2:0] bit-field is writable only when the channel is not active (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...4,22,23,30,31) 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.

1010~1111: 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	Reserved	Must be kept at reset value.
1:0	CH2MS[1:0]	Channel 2 I/O mode selection This bit-field specifies the work mode of the channel and the input signal selection. The CH2MS[2:0] bit-field is writable only when the channel is not active (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...4,22,23,30,31) register). 100~111: Reserved.

Input capture mode:

Bits	Fields	Descriptions
31	CH3MS[2]	Channel 3 I/O mode selection Same as output compare mode.
30	CH2MS[2]	Channel 2 I/O mode selection Same as output compare mode.
31:16	Reserved	Must be kept at reset value.
15:12	CH3CAPFLT[3:0]	Channel 3 input capture filter control Refer to CH2CAPFLT description.
11:10	CH3CAPPSC[1:0]	Channel 3 input capture prescaler Refer to CH2CAPPSC description.
9:8	CH3MS[1:0]	Channel 3 mode selection Same as output compare mode.
7:4	CH2CAPFLT[3:0]	Channel 2 input capture filter control An event counter is used in the digital filter, in which a transition on the output occurs after N input events. This bit-field specifies the frequency used to sample CI2 input signal and the length of the digital filter applied to CI2. 0000: Filter disabled, $f_{SAMP}=f_{DTS}$, N=1. 0001: $f_{SAMP}=f_{CK_TIMER}$, N=2. 0010: $f_{SAMP}= f_{CK_TIMER}$, N=4. 0011: $f_{SAMP}= f_{CK_TIMER}$, N=8. 0100: $f_{SAMP}=f_{DTS}/2$, N=6. 0101: $f_{SAMP}=f_{DTS}/2$, N=8. 0110: $f_{SAMP}=f_{DTS}/4$, N=6. 0111: $f_{SAMP}=f_{DTS}/4$, N=8. 1000: $f_{SAMP}=f_{DTS}/8$, N=6. 1001: $f_{SAMP}=f_{DTS}/8$, N=8.

1010: $f_{SAMP}=f_{DTS}/16$, N=5.
1011: $f_{SAMP}=f_{DTS}/16$, N=6.
1100: $f_{SAMP}=f_{DTS}/16$, N=8.
1101: $f_{SAMP}=f_{DTS}/32$, N=5.
1110: $f_{SAMP}=f_{DTS}/32$, N=6.
1111: $f_{SAMP}=f_{DTS}/32$, N=8.

3:2	CH2CAPPSC[1:0]	Channel 2 input capture prescaler This bit-field specifies the factor of the prescaler on channel 2 input. The prescaler is reset when CH2EN bit in TIMERx_CHCTL2 register is cleared. 00: Prescaler disabled, capture is done on each channel input edge. 01: Capture is done every 2 channel input edges. 10: Capture is done every 4 channel input edges. 11: Capture is done every 8 channel input edges.
1:0	CH2MS[1:0]	Channel 2 mode selection Same as output compare mode.

Channel control register 2 (TIMERx_CHCTL2)

Address offset: 0x20
Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH3PERFOREN	CH2PERFOREN	CH1PERFOREN	CH0PERFOREN	Reserved											
rw	rw	rw	rw												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3NP	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	CH3PERFOREN	Channel 3 in the composite PWM mode, the period point match moment O3CPRE level setting Refer to CH0PERFOREN description.
30	CH2PERFOREN	Channel 2 in the composite PWM mode, the period point match moment O2CPRE level setting Refer to CH0PERFOREN description.
29	CH1PERFOREN	Channel 1 in the composite PWM mode, the period point match moment O1CPRE level setting Refer to CH0PERFOREN description.
28	CH0PERFOREN	Channel 0 in the composite PWM mode, the period point match moment O0CPRE

		level setting 0: disable 1: In the composite PWM0 mode, the level will be forced to high at the time of period point matching; In the composite PWM1 mode, the level at the matching time of the period point is forced to low. **Note:** In the central alignment mode, the period point is the counter underflow time; In the edge-aligned mode, the period point is the counter flow moment.
27:16	Reserved	Must be kept at reset value.
15	CH3NP	Channel 3 complementary capture/compare polarity Refer to CH0NP description.
14	Reserved	Must be kept at reset value.
13	CH3P	Channel 3 capture/compare function polarity Refer to CH0P description
12	CH3EN	Channel 3 capture/compare function enable Refer to CH0EN description
11	CH2NP	Channel 2 complementary capture/compare polarity Refer to CH0NP description.
10	Reserved	Must be kept at reset value.
9	CH2P	Channel 2 capture/compare function polarity Refer to CH0P description
8	CH2EN	Channel 2 capture/compare function enable Refer to CH0EN description
7	CH1NP	Channel 1 complementary capture/compare polarity Refer to CH0NP description.
6	Reserved	Must be kept at reset value.
5	CH1P	Channel 1 capture/compare function polarity Refer to CH0P description
4	CH1EN	Channel 1 capture/compare function enable Refer to CH0EN description
3	CH0NP	Channel 0 complementary output polarity When channel 0 complementary is configured in output mode, this bit must be kept at reset value. When CH0 is configured in input mode, in conjunction with CH0P, this bit is used to define the polarity of CH0. This bit cannot be modified when PROT[1:0] bit-field in TIMERx_CCHP register is 11 or 10.

2	Reserved	Must be kept at reset value.
1	CH0P	Channel 0 capture/compare function polarity When channel 0 is configured in output mode, this bit specifies the output signal polarity. 0: Channel 0 active high 1: Channel 0 active low When channel 0 is configured in input mode, this bit specifies the 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-field in TIMERx_CCHP register is 11 or 10.
0	CH0EN	Channel 0 capture/compare function enable When channel 0 is configured in output mode, setting this bit enables CH0_O signal in active state. When channel 0 is configured in input mode, setting this bit enables the capture event in channel0. 0: Channel 0 disabled 1: Channel 0 enabled

Counter register (TIMERx_CNT) (TIMERx, x= 2,3,30,31)

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.

Counter register (TIMERx_CNT) (TIMERx, x= 1,4,22,23)

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] / UPIFBU	CNT[30:16]														
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.

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.

Prescaler register (TIMERx_PSC)

Address offset: 0x28
Reset value: 0x0000 0000

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

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

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

PSC[15:0]
rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	PSC[15:0]	Prescaler value of the counter clock The PSC clock is divided by (PSC+1) to generate the counter clock. The value of this bit-field will be loaded to the corresponding shadow register at every update event.

Counter auto reload register (TIMERx_CAR)

Address offset: 0x2C

Reset value: 0xFFFF FFFF

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

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

Bits	Fields	Descriptions
31:16	CARL[31:16]	Counter auto reload value (bit 16 to 31) This bit-field only for TIMER1/ 4/ 22/ 23.
15:0	CARL[15:0]	Counter auto reload value This bit-field specifies the auto reload value of the counter.

Channel 0 capture/compare value register (TIMERx_CH0CV)

Address offset: 0x34

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH0VAL[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH0VAL[15:0]															
rw															

Bits	Fields	Descriptions

31:16	CH0VAL[31:16]	Capture/compare value of channel 0 (bit 16 to 31) This bit-field only for TIMER1/ 4/ 22/ 23.
15:0	CH0VAL[15:0]	Capture/compare value of channel 0 When channel 0 is configured in input mode, this bit-field indicates the counter value at the last capture event. And this bit-field is read-only. When channel 0 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event.

Channel 1 capture/compare value register (TIMERx_CH1CV)

Address offset: 0x38
Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH1VAL[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1VAL[15:0]															
rw															

Bits	Fields	Descriptions
31:16	CH1VAL[31:16]	Capture/compare value of channel 1(bit 16 to 31) This bit-field only for TIMER1/ 4/ 22/ 23.
15:0	CH1VAL[15:0]	Capture/compare value of channel 1 When channel 1 is configured in input mode, this bit-field indicates the counter value at the last capture event. And this bit-field is read-only. When channel 1 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event.

Channel 2 capture/compare value register (TIMERx_CH2CV)

Address offset: 0x3C
Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH2VAL[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH2VAL[15:0]															

rw

Bits	Fields	Descriptions
31:16	CH2VAL[31:16]	Capture/compare value of channel 2 (bit 16 to 31) This bit-field only for TIMER1/ 4/ 22/ 23.
15:0	CH2VAL[15:0]	Capture/compare value of channel 2 When channel 2 is configured in input mode, this bit-field indicates the counter value at the last capture event. And this bit-field is read-only. When channel 2 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event.

Channel 3 capture/compare value register (TIMERx_CH3CV)

Address offset: 0x40

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH3VAL[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3VAL[15:0]															
rw															

Bits	Fields	Descriptions
31:16	CH3VAL[31:16]	Capture/compare value of channel 3 (bit 16 to 31) This bit-field only for TIMER1/ 4/ 22/ 23.
15:0	CH3VAL[15:0]	Capture/compare value of channel 3 When channel 3 is configured in input mode, this bit-field indicates the counter value at the last capture event. And this bit-field is read-only. When channel 3 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event.

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/ 22/ 23.
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
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/ 22/ 23.
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).

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/ 22/ 23.
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).

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/ 22/ 23.
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: 0x0000 0000

This register 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	Reserved								DECDISDEN	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.

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 continues to count to this value, the counter will timeout and the interrupt flag DECDISIF is set. If DECDISIE=1, the corresponding interrupt is generated. Note: This register is just used in quadrature decoder signal disconnection detection function(with DECDISDEN =1).

DMA configuration register (TIMERx_DMACFG)

Address offset: 0xE0

Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value.
13:8	DMATC[5:0]	DMA transfer count This field defines the times of accessing(R/W) the TIMERx_DMATB register by DMA. 6'b000000: transfer 1 time 6'b000001: transfer 2 times … 6'b100101: transfer 38 times

7:6	Reserved	Must be kept at reset value.
5:0	DMATA[5:0]	DMA transfer access start address This field defines the start address of accessing the TIMERx_DMATB register by DMA. When the first access to the TIMERx_DMATB register is done, this bit-field specifies the address just accessed. And then the address of the second access to the TIMERx_DMATB register will be (start address + 0x4). 6'b000000: TIMERx_CTL0 6'b000001: TIMERx_CTL1 … In a word: start address = TIMERx_CTL0 + DMATA*4

DMA transfer buffer register (TIMERx_DMATB)

Address offset: 0xE4

Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:0	DMATB[31:0]	DMA transfer buffer When a read or write operation is assigned to this register, the register located at the address ranges from (start address) to (start address + transfer count * 4) will be accessed. The transfer count is calculated by hardware, and ranges from 0 to DMATC.

Configuration register (TIMERx_CFG)

Address offset: 0xFC

Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:5	Reserved	Must be kept at reset value.
4	OCPRECIVM	Output prepare signal clear invalid mode select This bit determines when the output prepare signal clear is set to invalid by the hardware. 0：The next counter overflow or underflow event 1：The next counter update event
3:2	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.

15.3. General level3 timer (TIMERx, x=14,40,41,42,43,44)

15.3.1. Overview

The general level3 timer module (TIMER14/40/41/42/43/44) is a three-channel timer that supports both input capture and output compare. They can generate PWM signals to control motor or be used for power management applications. The general level3 timer has a 16-bit counter that can be used as an unsigned counter.

In addition, the general level3 timers can be programmed and be used for counting, their external events can be used to drive other timers.

Timer also includes a dead-time Insertion module which is suitable for motor control applications.

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

15.3.2. Characteristics

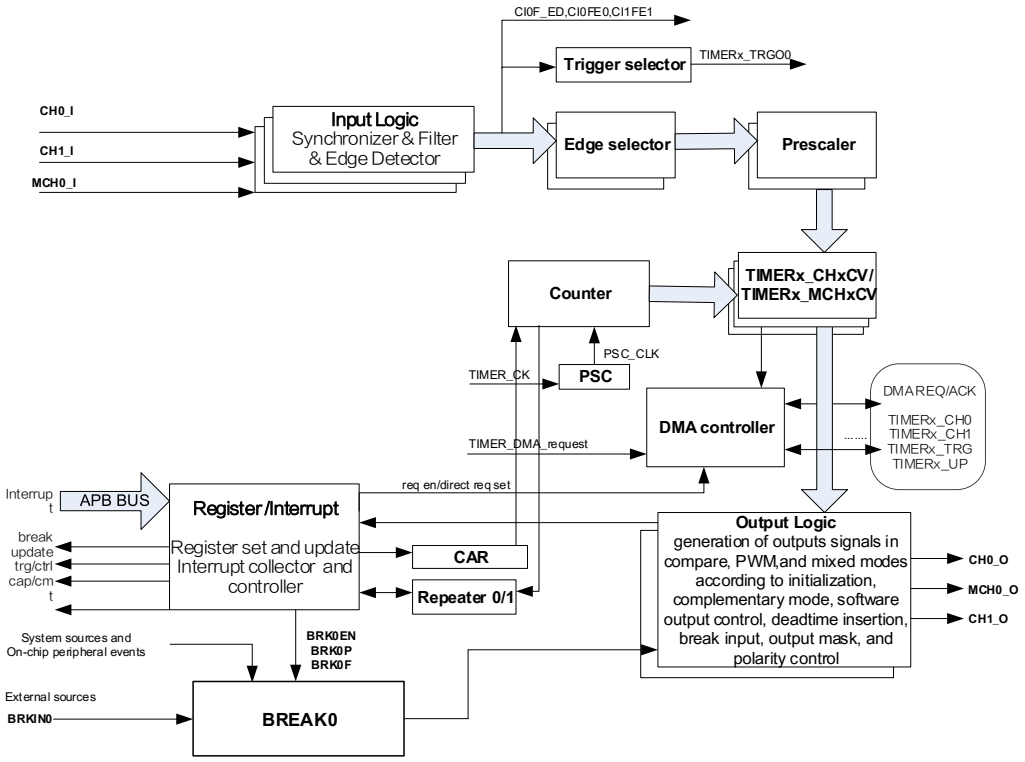
- Total channel num: 3.
- Counter width: 16-bit.
- Source of counter clock is selectable: internal clock, internal trigger, external input.
- Multiple counter modes: up counting, down counting and center-aligned counting.
- Programmable prescaler: 16-bit. The factor can be changed on the go.
- Each channel is user-configurable: input capture mode, output compare mode, programmable PWM mode, single pulse mode.
- Programmable dead time insertion.
- Auto reload function.
- Programmable counter repetition function.
- Break input function: 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.

15.3.3. Block diagram

Figure 15-80. General level3 timer block diagram provides details of the internal

configuration of the general level3 timer.

Figure 15-80. General level3 timer block diagram



15.3.4. 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=14,40,41,42,43,44) registers.

- TSCFGy[4:0] (y=3..7,15) = 5'b00000 in SYSCFG_TIMERxCFG(x=14,40,41,42,43,44) registers. Internal clock CK_TIMER is selected as timer clock source which is from module RCU.

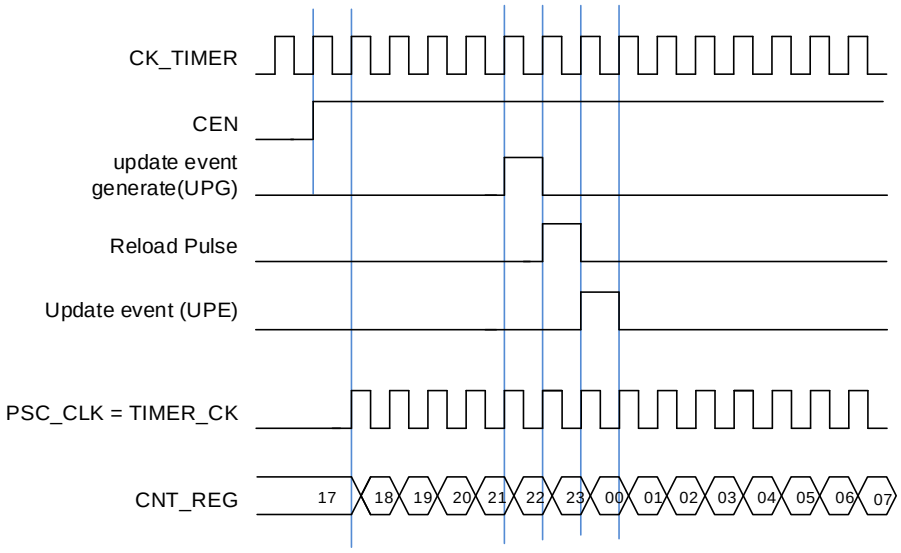
The default clock source is the CK_TIMER for driving the counter prescaler when TSCFGy[4:0] (y=3..7,15) = 5'b00000 in SYSCFG_TIMERxCFG(x=14,40,41,42,43,44) registers. When the CEN is set, the CK_TIMER will be divided by PSC value to generate PSC_CLK.

In this mode, the TIMER_CK, which drives counter's prescaler to count, is equal to CK_TIMER which is from RCU.

If TSCFG6[4:0] bit-field in SYSCFG_TIMERxCFG(x=14,40,41,42,43,44) registers are setting to a nonzero value, the prescaler is clocked by other clock sources selected in the TSCFG6[4:0] bit-field, 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 15-81. Normal mode, internal clock divided by 1



- TSCFG6[4:0] are setting to a nonzero value (external clock mode 0). External input pin is selected as timer clock source

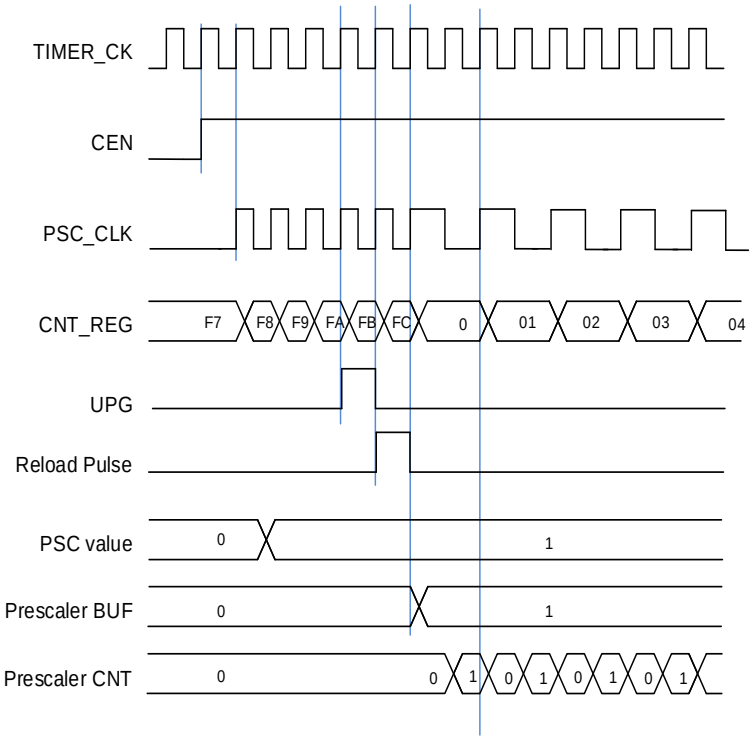
The TIMER_CK, which drives counter's prescaler to count, can be triggered by the event of rising or falling edge on the external pin TIMERx_CH0/TIMERx_CH1. This mode can be selected by setting TSCFG6[4:0] to 0x5~0x7.

And, the counter prescaler can also be driven by rising edge on the internal trigger input pin ITI0/1/2/3. This mode can be selected by setting TSCFG6[4:0] to 0x1~0x4.

Clock prescaler

The prescaler can divide the timer clock (TIMER_CK) to a counter clock (PSC_CLK) by any factor between 1 and 65536. It is controlled by prescaler register (TIMERx_PSC) which can be changed on the go but is taken into account at the next update event.

Figure 15-82. 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 15-83. Timing diagram of up counting mode, PSC=0/2 and *Figure 15-84. 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 15-83. Timing diagram of up counting mode, PSC=0/2

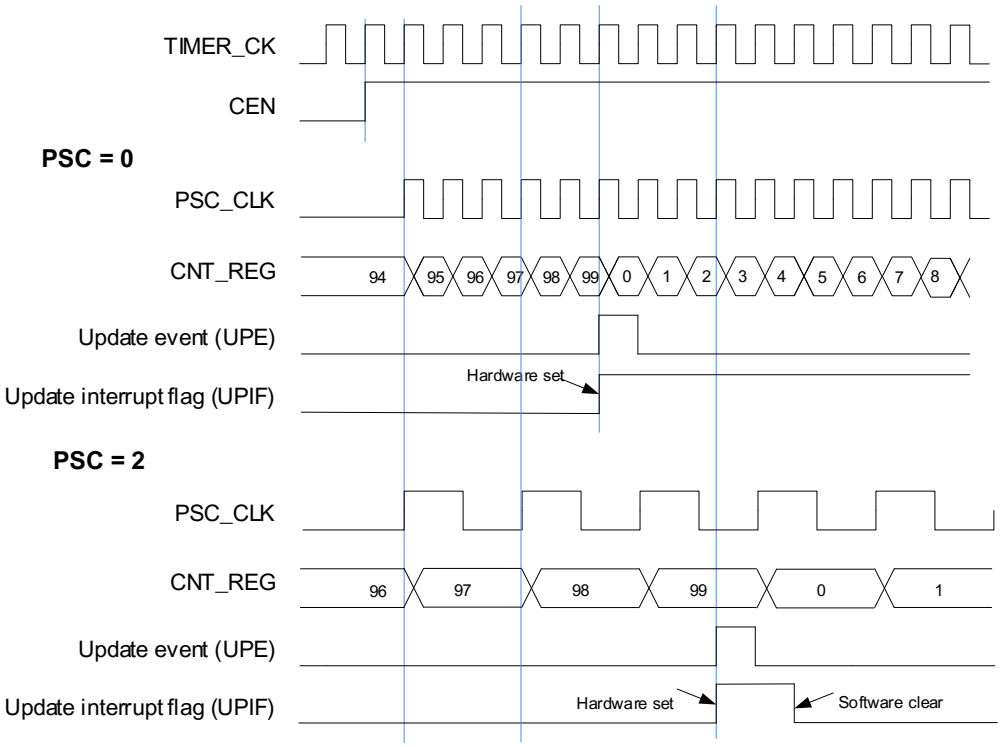
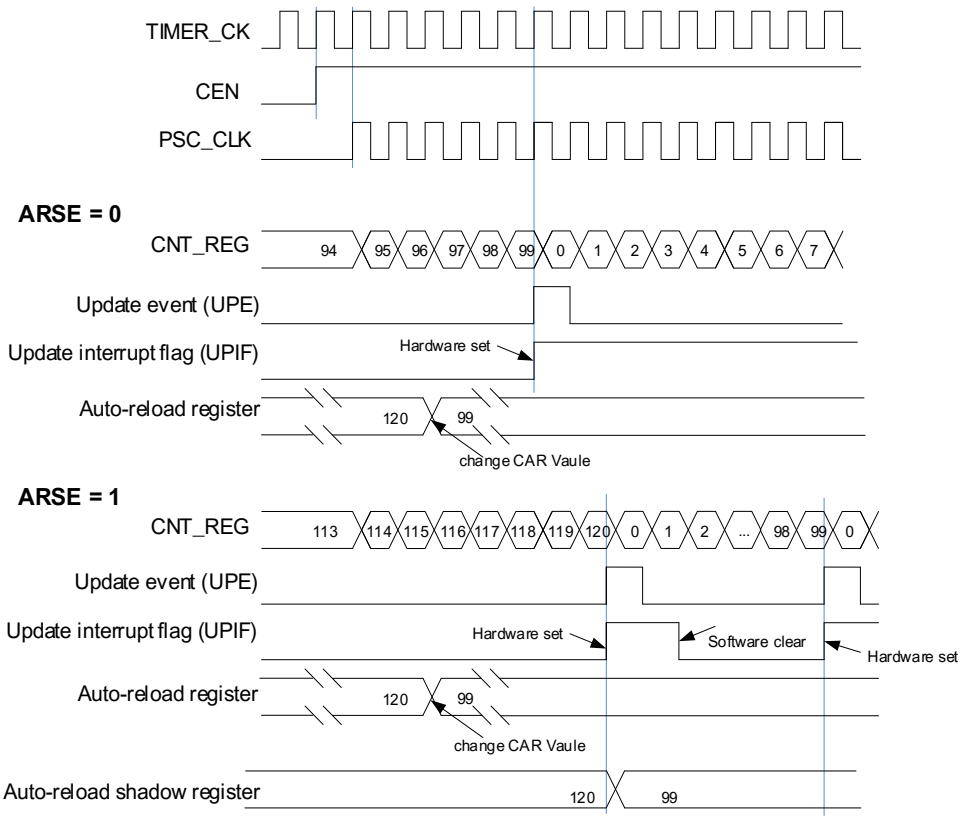


Figure 15-84. Timing diagram of up counting mode, change TIMERx_CAR on the go



Down counting mode

In this mode, the counter counts down continuously from the counter reload value, which is defined in the TIMERx_CAR register, in a count-down direction. Once the counter reaches 0, the counter restarts to count again from the counter reload value. If the repetition counter is set, the update event will be generated after (TIMERx_CREP0 / 1 + 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 15-85. Timing diagram of down counting mode, PSC=0/2 and *Figure 15-86. 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 15-85. Timing diagram of down counting mode, PSC=0/2

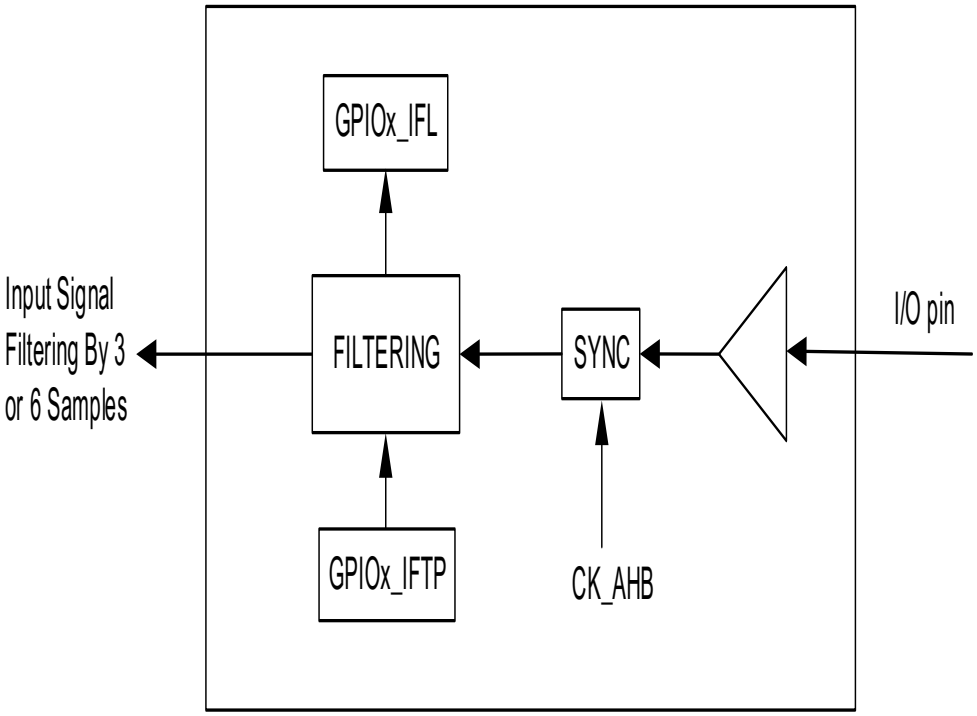
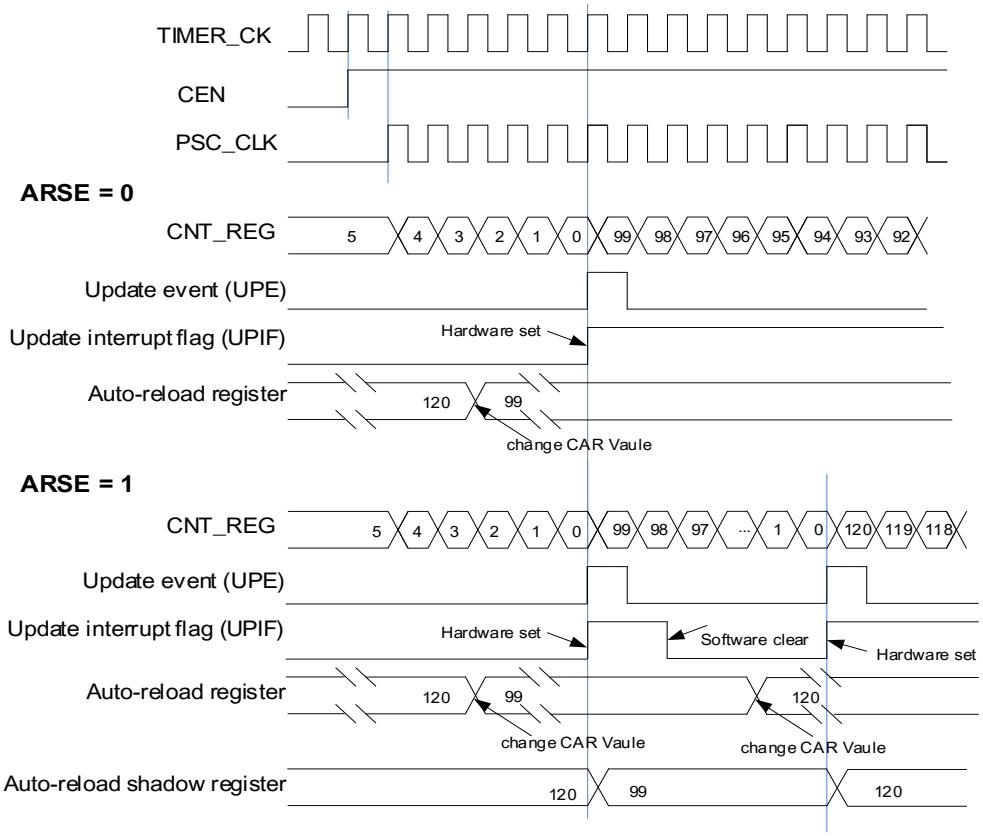


Figure 15-86. Timing diagram of down counting mode, change TIMERx_CAR ongoing



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 15-87. Timing diagram of center-aligned counting mode***.

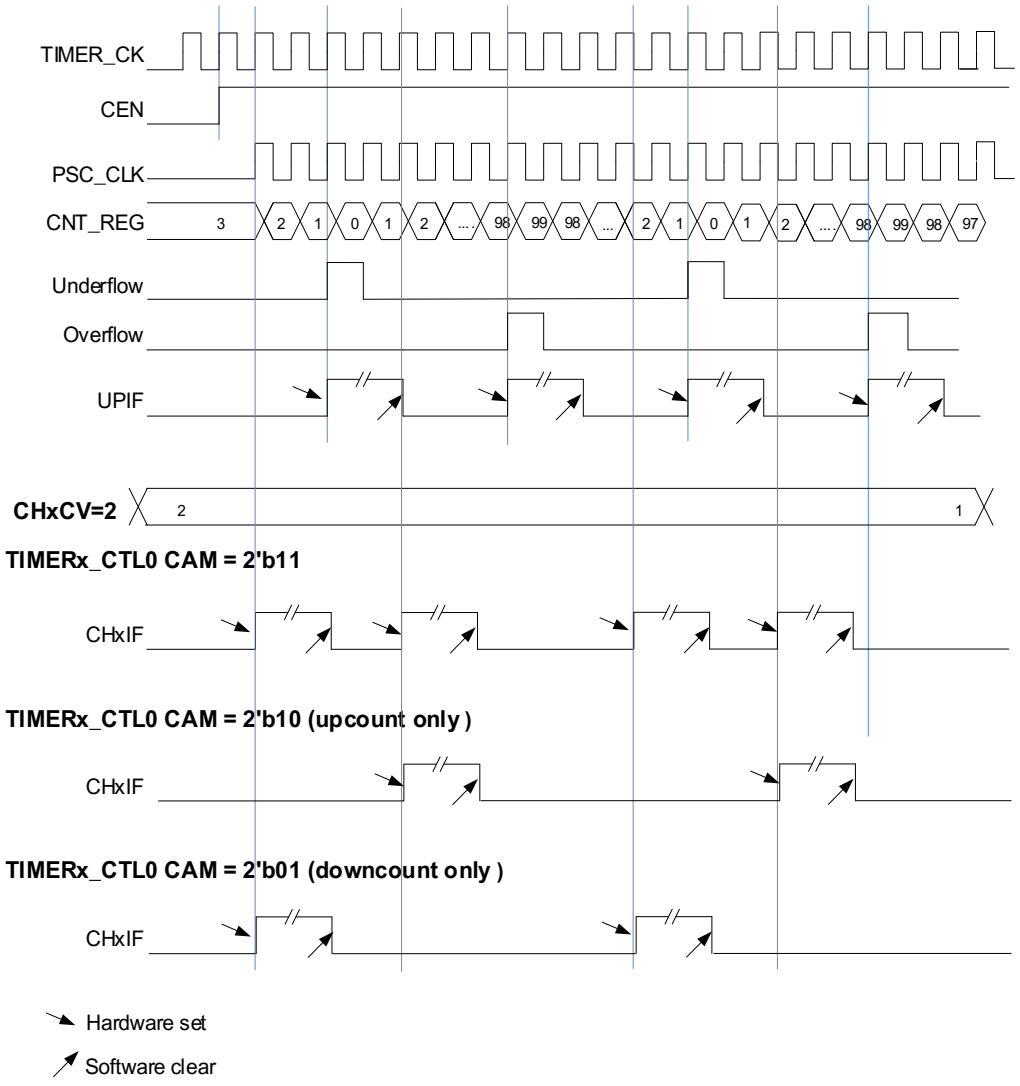
If the UPDIS bit in the TIMERx_CTL0 register is set, the update event is disabled.

When an update event occurs, all the registers (repetition counter register, auto-reload

register, prescaler register) are updated.

Figure 15-87. Timing diagram of center-aligned counting mode shows some examples of the counter behavior when TIMERx_CAR=0x99. TIMERx_PSC=0x0.

Figure 15-87. Timing diagram of center-aligned counting mode



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 32 bits 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 15-88. Repetition counter timing diagram of center-aligned counting mode

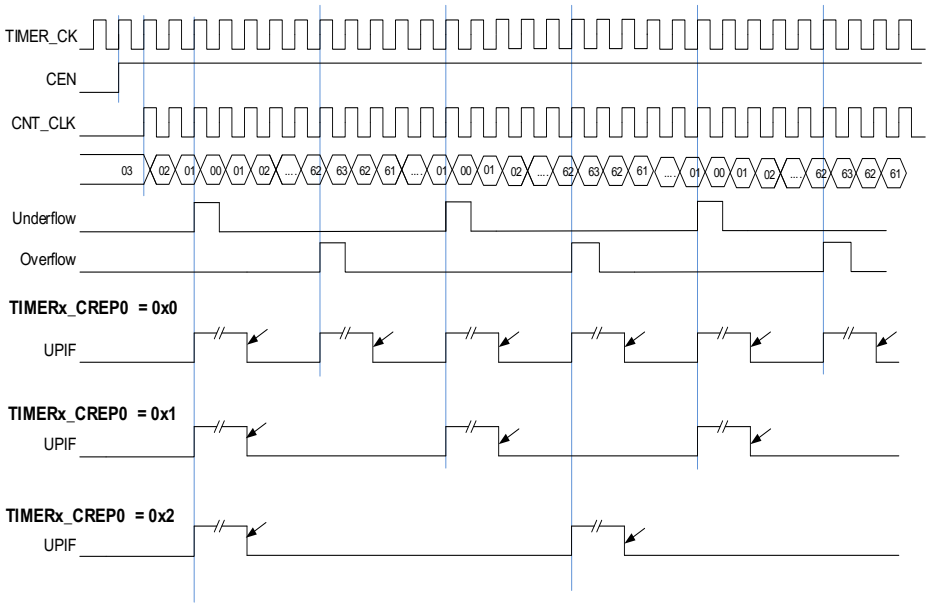


Figure 15-89. Repetition counter timing diagram of up counting mode

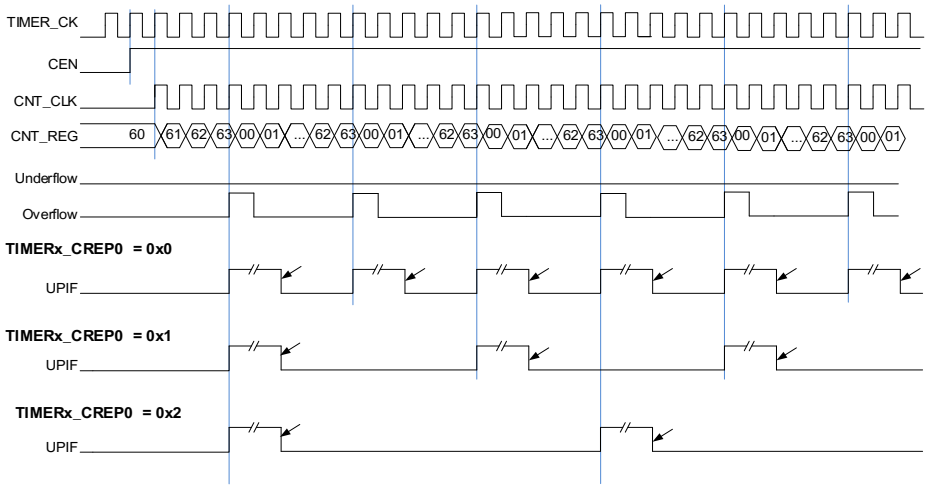
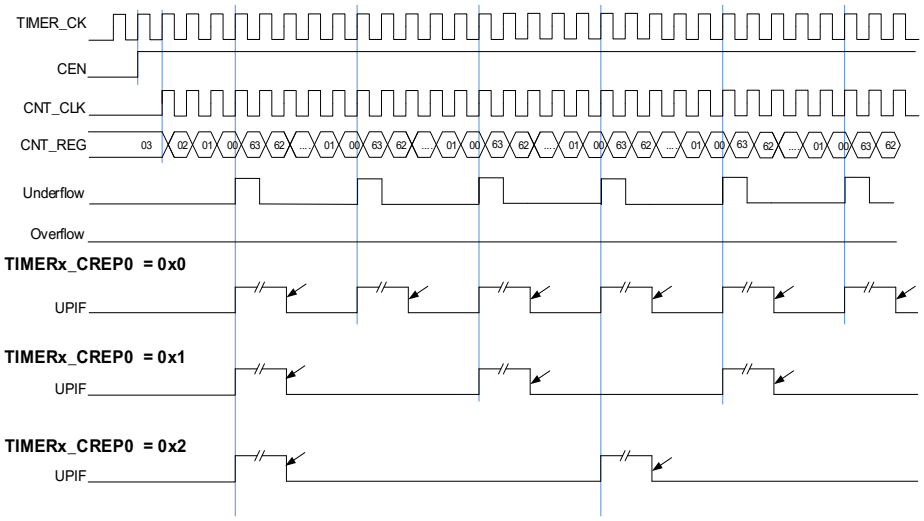


Figure 15-90. Repetition counter timing diagram of down counting mode



Capture/compare channels

The general level3 timer has three independent channels which can be used as capture inputs or compare match outputs. Each channel is built around a channel capture compare register including an input stage, channel controller and an output stage.

When the channels are used for input, channel 0 and multi mode channel 0 can perform input capture independently; when the channels are used for comparison output, the channel 0 and multi mode channel 0 can output independent and complementary outputs.

■ Input capture mode

When MCHxMSEL=2'b00(independent mode), channel x and multi mode channel x can perform input capture independently.

Input capture mode allows the channel to perform measurements such as pulse timing, frequency, period, duty cycle and so on. The input stage consists of a digital filter, a channel polarity selection, edge detection and a channel prescaler. When a selected edge occurs on the channel input, the current value of the counter is captured into the TIMERx_CHxCV/ TIMERx_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 15-91. Input capture logic for channel 0

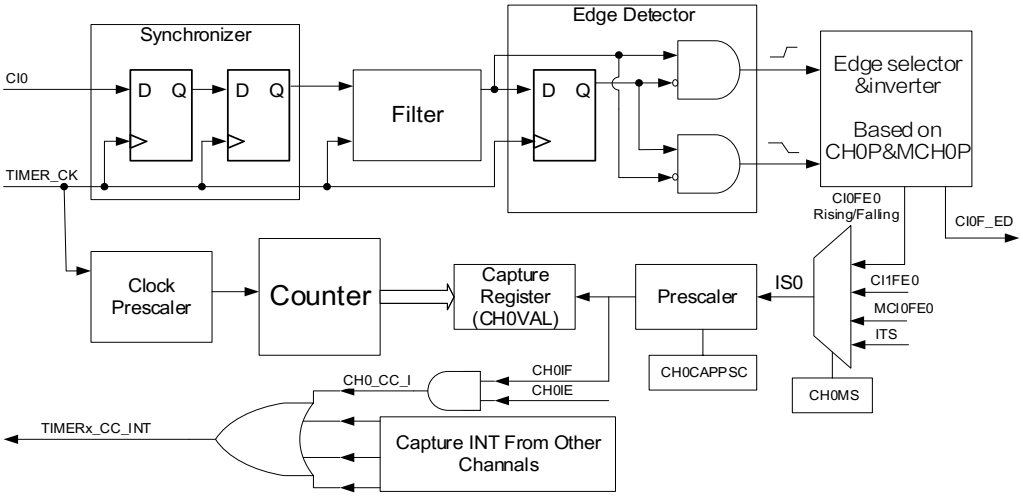
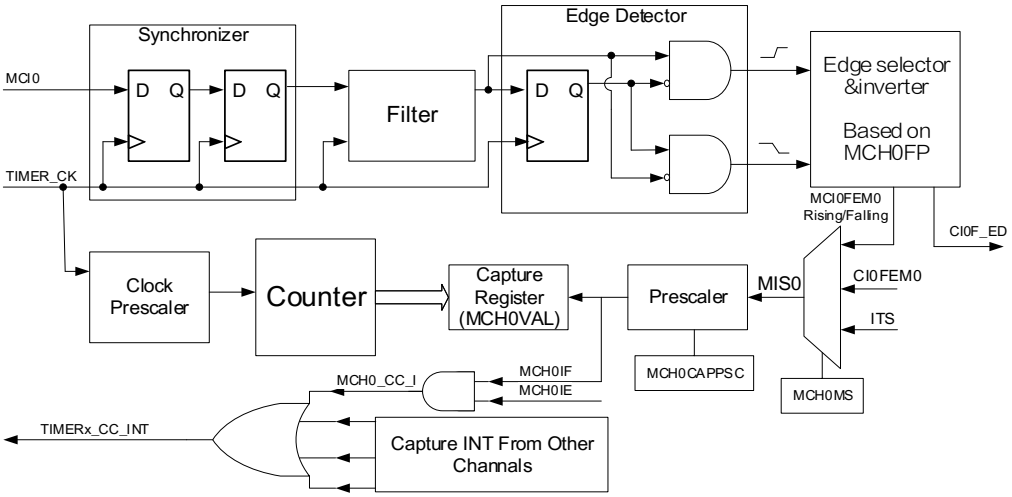


Figure 15-92. Input capture logic for multi mode channel 0



The input signals of channelx (Cix/ MCIx) are the TIMERx_CHx/ TIMERx_MCHxCV signal.

First, the input signal of channel (Cix/ MCIx) is synchronized to TIMER_CK signal, and then sampled by a digital filter to generate a filtered input signal. Then through the edge detector, the rising or falling edge is detected by configuring CHxP/ MCHxP or MCHxFP bits. The input capture signal can also be selected from the input signal of other channel or the internal trigger signal by configuring CHxMS/ MCHxMS bits. The IC prescaler makes several input events generate one effective capture event. On the capture event, TIMERx_CHxCV/ TIMERx_MCHxCV will store the value of counter.

So, the process can be divided into several steps as below:

Step1: Filter configuration (CHxCAPFLT bit in TIMERx_CHCTL0 register and MCHxCAPFLT bit in TIMERx_MCHCTL0 register).

Based on the input signal and quality of requested signal, configure compatible CHxCAPFLT or MCHxCAPFLT bit.

Step2: Edge selection (CHxP and MCHxP bits in TIMERx_CHCTL2 register, MCHxFP[1:0] bits in TIMERx_MCHCTL2 register).

Rising edge or falling edge, choose one by configuring CHxP and MCHxP bits or MCHxFP[1:0] bits.

Step3: Capture source selection (CHxMS bit in TIMERx_CHCTL0 register, MCHxMS bit in TIMERx_MCHCTL0 register).

As soon as selecting one input capture source by CHxMS, the channel must be set to input mode (CHxMS! =0x000 or MCHxMS != 0x000) and TIMERx_CHxCV/ TIMERx_MCHxCV cannot be written any more.

Step4: Interrupt enable (ChxIE and CHxDEN bits, MCHxIE and MCHxDEN bits in TIMERx_DMAINTEN).

Enable the related interrupt to get the interrupt and DMA request.

Step5: Capture enable (ChxEN and MCHxEN bits in TIMERx_CHCTL2).

Result: When the wanted input signal is captured, TIMERx_CHxCV/ TIMERx_MCHxCV will be set by counter's value and ChxIF/ MCHxIF bit is asserted. If the ChxIF/ MCHxIF bit is 1, the ChxOF/ MCHxOF bit will also be asserted. The interrupt and DMA request will be asserted or not based on the configuration of ChxIE and CHxDEN bits, MCHxIE and MCHxDEN bits in TIMERx_DMAINTEN.

Direct generation: A DMA request or interrupt is generated by setting CHxG directly.

The input capture mode can be also used for pulse width measurement from signals on the TIMERx_CHx and TIMERx_MCHx pins. For example, PWM signal connects to CI0 input. Select CI0 as channel 0 capture signals by setting CH0MS to 3'b001 in the channel control register (TIMERx_CHCTL0) and set capture on rising edge. Select CI0 as channel 1 capture signal by setting CH1MS to 3'b010 in the channel control register (TIMERx_CHCTL0) and set capture on falling edge. The counter is set to restart mode and is restarted on channel 0 rising edge. Then the TIMERX_CH0CV can measure the PWM period and the TIMERx_CH1CV can measure the PWM duty cycle.

Output compare mode

Figure 15-93. Output compare logic (when MCHxMSEL = 2'b00, x=0), *Figure 15-94. Output compare logic (when MCHxMSEL = 2'b11, x=0)* and *Figure 15-95. Output compare logic (x=1)* show the logic circuit of output compare mode.

Figure 15-93. Output compare logic (when MCHxMSEL = 2'b00, x=0)

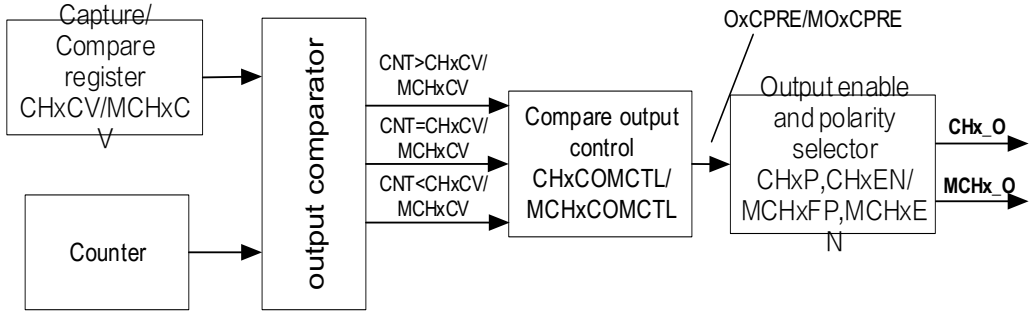


Figure 15-94. Output compare logic (when MCHxMSEL = 2'b11, x=0)

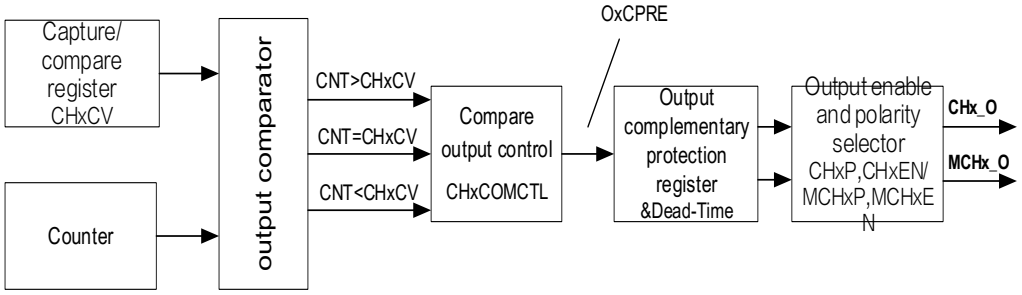
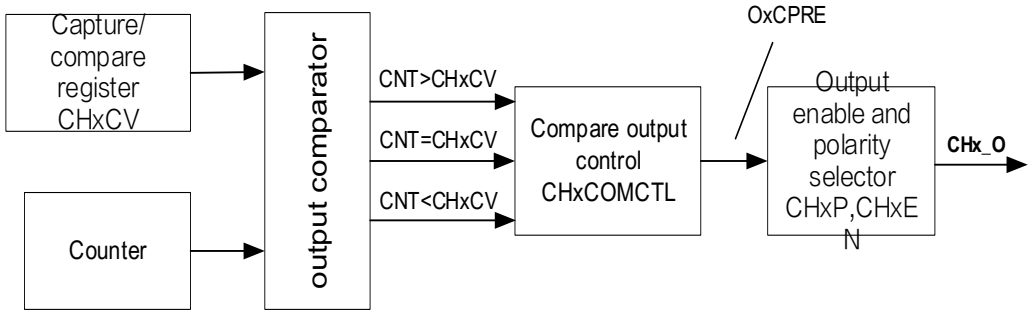


Figure 15-95. Output compare logic (x=1)



The relationship between the channel output signal CHx_O/MCHx_O and the OxCPRE/ MOxCPRE signal (more details refer to ***Channel output prepare signal***) is described as below(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 15-93. 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 15-94. 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_CCHP register. Please refer to ***Outputs complementary*** for more details.

In output compare mode, the TIMERx can generate timed pulses with programmable position, polarity, duration and frequency. When the counter matches the value in the TIMERx_CHxCV/ TIMERx_MCHxCV register of an output compare channel, the channel (n) output can be set, cleared, or toggled based on CHxCOMCTL/ MCHxCOMCTL. When the counter reaches the value in the TIMERx_CHxCV/ TIMERx_MCHxCV register, the ChxIF/ MCHxIF bit will be set and the channel (n) interrupt is generated if ChxIE/ MCHxIE = 1. And the DMA request will be asserted, if CHxDEN/ MCHxDEN =1.

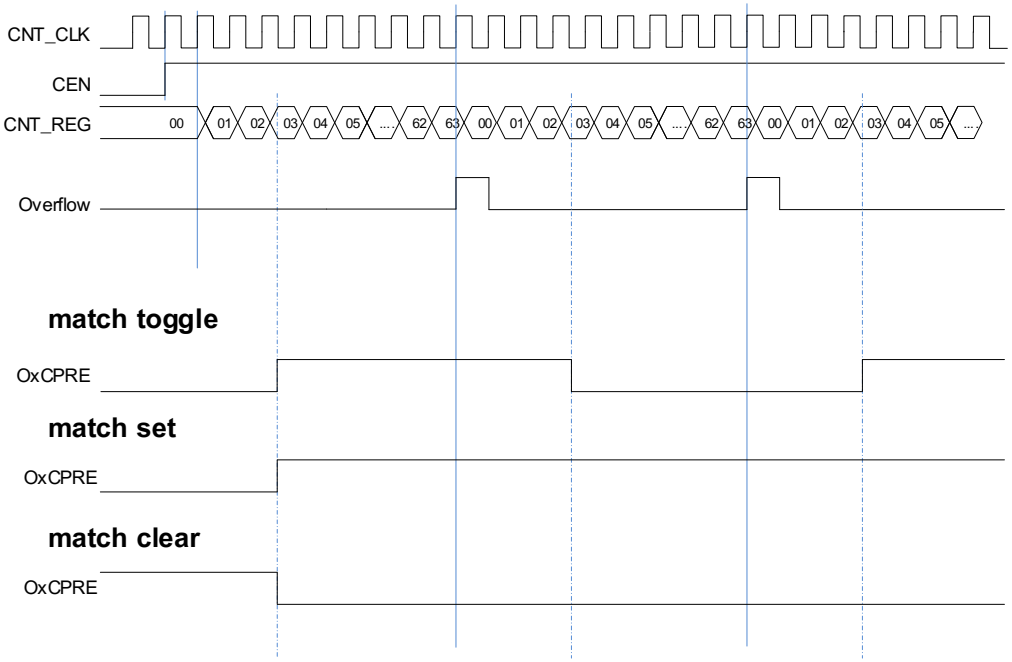
So, the process can be divided into several steps as below:

Step1: Clock Configuration. Such as clock source, clock prescaler and so on.
Step2: Compare mode configuration.
- Set the shadow enable mode by CHxCOMSEN/ MCHxCOMSEN.
- Set the output mode (set/clear/toggle) by CHxCOMCTL/ MCHxCOMCTL.
- Select the active polarity by CHxP/MCHxP/ MCHxFP.
- Enable the output by ChxEN/ MCHxEN.

Step3: Interrupt/DMA request enable configuration by ChxIE/ MCHxIE /CHxDEN/ MCHxDEN.
Step4: Compare output timing configuration by TIMERx_CAR and TIMERx_CHxCV/ TIMERx_MCHxCV.
The TIMERx_CHxCV/ TIMERx_MCHxCV can be changed onging to meet the expected waveform.
Step5: Start the counter by configuring CEN to 1.

Figure 15-96. Output-compare in three modes shows the three compare modes: toggle/set/clear. CARL=0x63, CHxVAL=0x3.

Figure 15-96. 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 15-97. 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 15-98. 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 15-97. Timing diagram of EAPWM

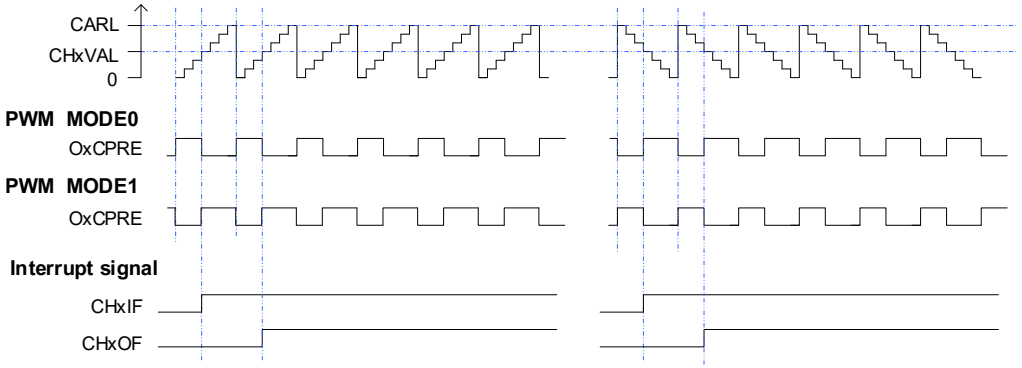
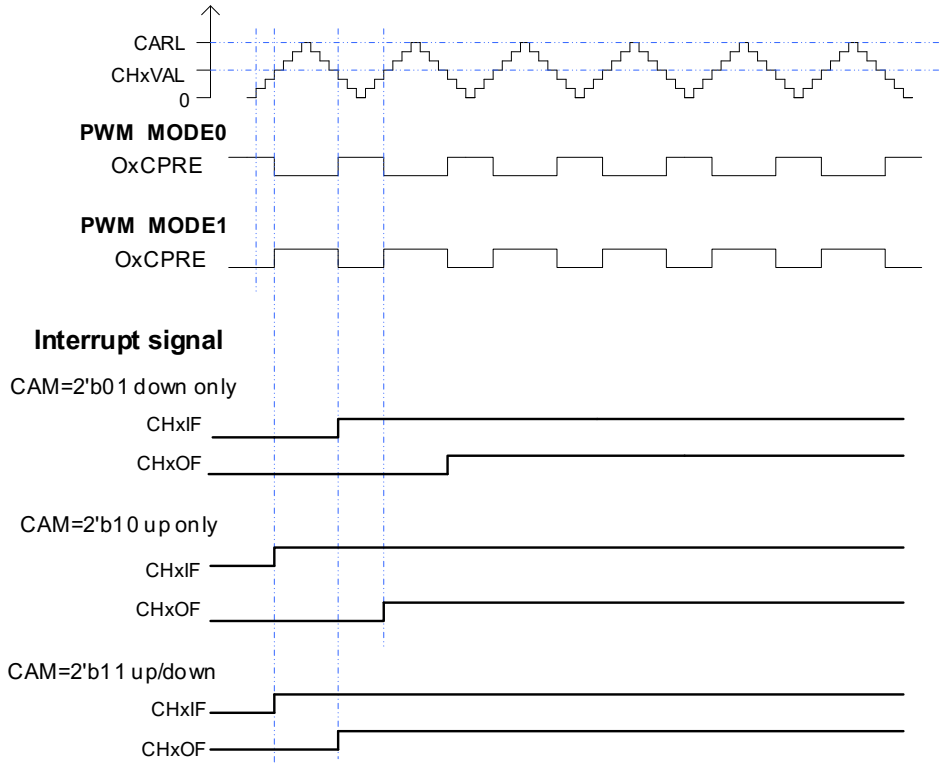


Figure 15-98. Timing diagram of CAPWM



Composite PWM mode

In the Composite PWM mode (CHxCPWMEN = 1'b1, CHxMS[2:0] = 3'b000 and CHxCOMCTL = 4'b0110 or 4'b0111), the PWM signal output in channel x (x=0,1) is composited by CHxVAL and CHxCOMVAL_ADD bits.

If CHxCOMCTL = 4'b0110 (PWM mode 0) and DIR = 1'b0 (up counting mode), or CHxCOMCTL = 4'b0111 (PWM mode 1) and DIR = 1'b1 (Down counting mode), the channel x output is forced low when the counter matches the value of CHxVAL. It is forced high when the counter matches the value of CHxCOMVAL_ADD.

If CHxCOMCTL = 4'b0111 (PWM mode 1) and DIR = 1'b0 (up counting mode), or

CHxCOMCTL = 4'b0110 (PWM mode 0) and DIR = 1'b1 (down counting mode) the channel x output is forced high when the counter matches the value of CHxVAL. It is forced low when the counter matches the value of CHxCOMVAL_ADD.

When CHxVAL or CHxCOMVAL_ADD = 0 / CARL, the channel x output is specially processed, and by setting the CHxPERFOREN bit in the TIMERx_CHCTL2 register, the OxCPRE output is forced to either high or low level (determined according to the selected composite PWM mode).

The PWM period is determined by (CARL + 0x0001) and the PWM pulse width is determined by the following table.

Table 15-14.The Composite PWM pulse width

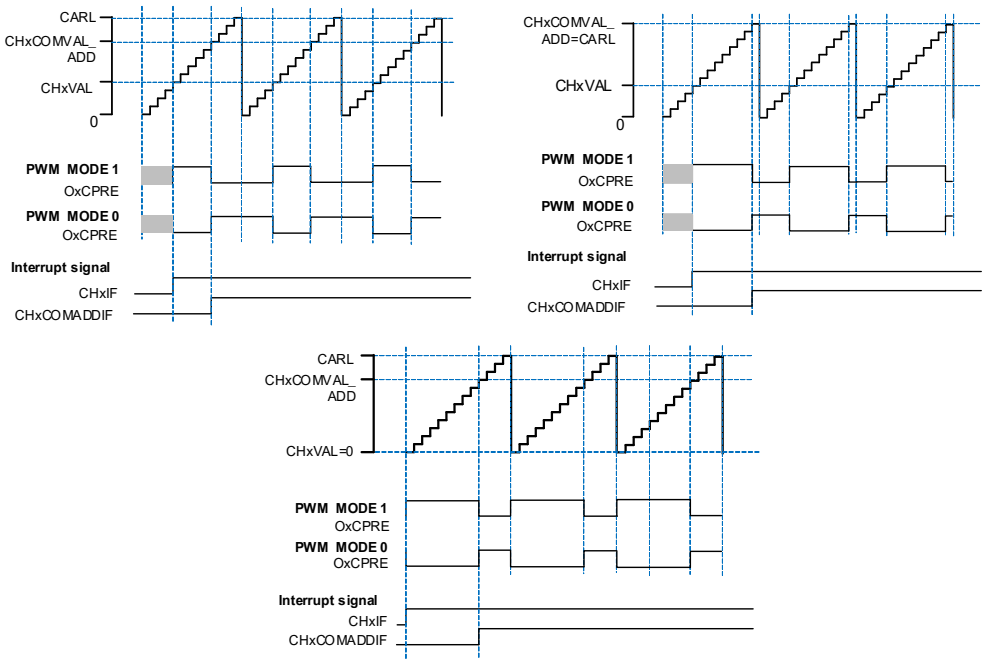
Condition	Mode	PWM pulse width
CHxVAL < CHxCOMVAL_ADD <= CARL	PWM mode 0	(CARL + 0x0001) + (CHxVAL - CHxCOMVAL_ADD)
	PWM mode 1	(CHxCOMVAL_ADD - CHxVAL)
CHxCOMVAL_ADD < CHxVAL <= CARL	PWM mode 0	(CHxVAL - CHxCOMVAL_ADD)
	PWM mode 1	(CARL + 0x0001) + (CHxCOMVAL_ADD - CHxVAL)
(CHxVAL = CHxCOMVAL_ADD <= CARL) or (CHxVAL > CARL > CHxCOMVAL_ADD)	PWM mode 0 (up counting) or PWM mode 1 (down counting)	100%
	PWM mode 0 (down counting) or PWM mode 1 (up counting)	0%
CHxCOMVAL_ADD > CARL > CHxVAL	PWM mode 0(up counting) or PWM mode 1(down counting)	0%
	PWM mode 0(down counting) or PWM mode 1(up counting)	100%
(CHxVAL>CARL) and (CHxCOMVAL_ADD > CARL)	-	The output of CHx_O is keeping

When the counter reaches the value of CHxVAL, the ChxIF bit is set and the channel x interrupt is generated if ChxIE = 1, and the DMA request will be asserted, if CHxDEN=1. When the counter reaches the value of CHxCOMVAL_ADD, the CHxCOMADDIF bit is set (this flag just used in composite PWM mode, when CHxCPWMEN=1) and the channel x additional compare interrupt is generated if CHxCOMADDIE = 1 (Only interrupt is generated, no DMA request is generated).

According to the relationship among CHxVAL, CHxCOMVAL_ADD and CARL, it can be divided into four situations:

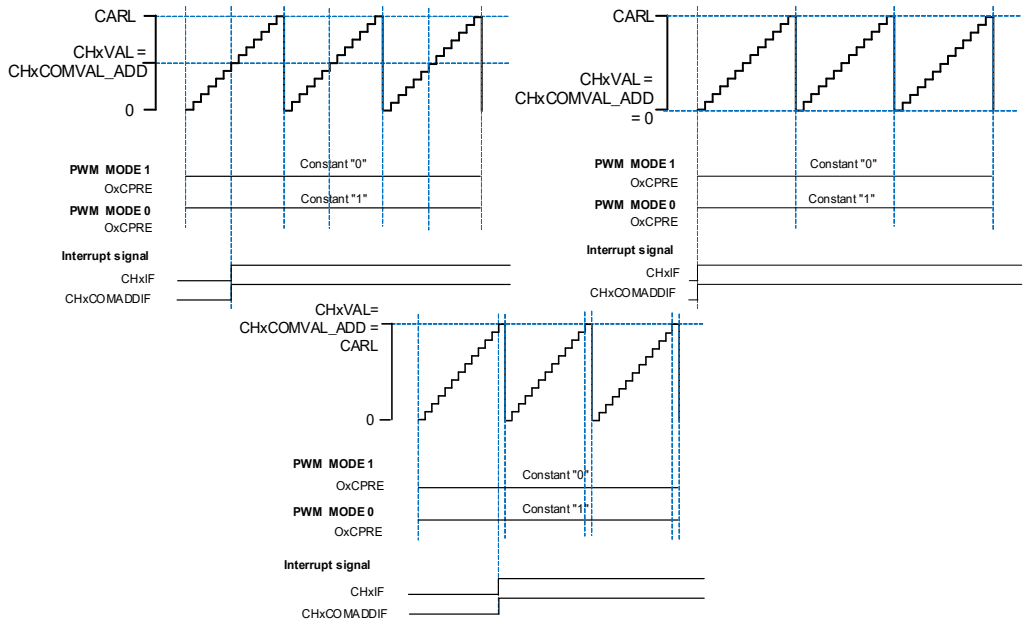
1) CHxVAL < CHxCOMVAL_ADD, and the values of CHxVAL and CHxCOMVAL_ADD between 0 and CARL.

Figure 15-99. Channel x output PWM with (CHxVAL < CHxCOMVAL_ADD)



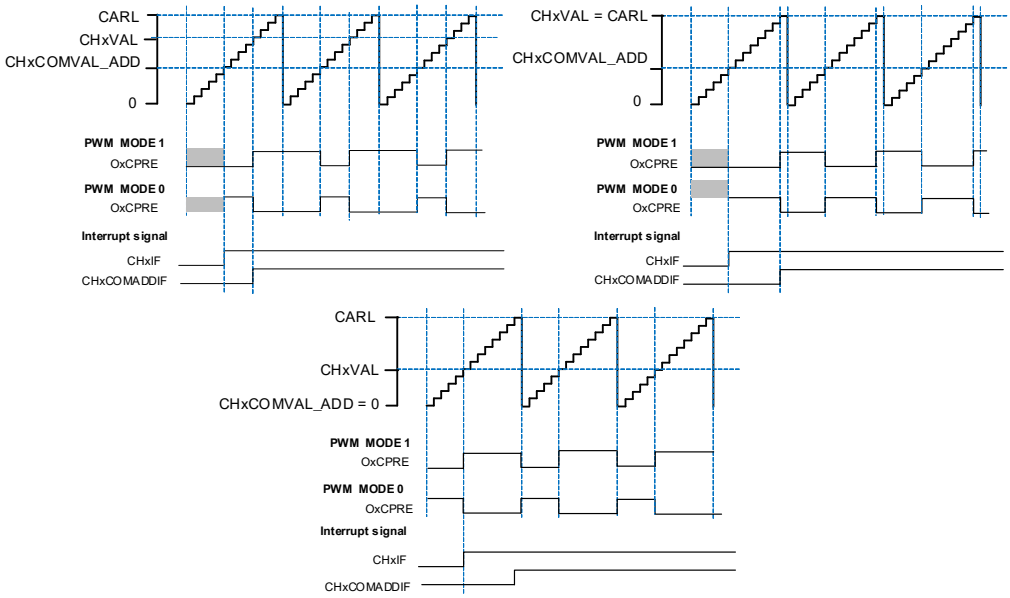
2) CHxVAL = CHxCOMVAL_ADD, and the value of CHxVAL and CHxCOMVAL_ADD between 0 and CARL.

Figure 15-100. Channel x output PWM with (CHxVAL = CHxCOMVAL_ADD)



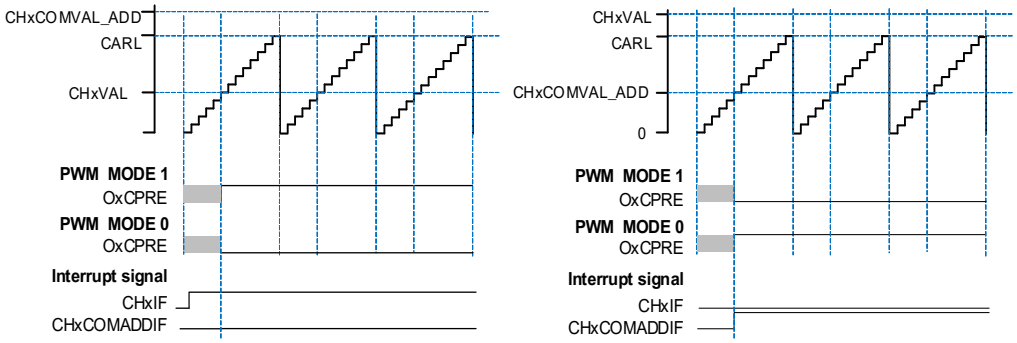
3) CHxVAL > CHxCOMVAL_ADD, and the value of CHxVAL and CHxCOMVAL_ADD between 0 and CARL.

Figure 15-101. Channel x output PWM with (CHxVAL > CHxCOMVAL_ADD)



4) One of the value of CHxVAL and CHxCOMVAL_ADD exceeds CARL.

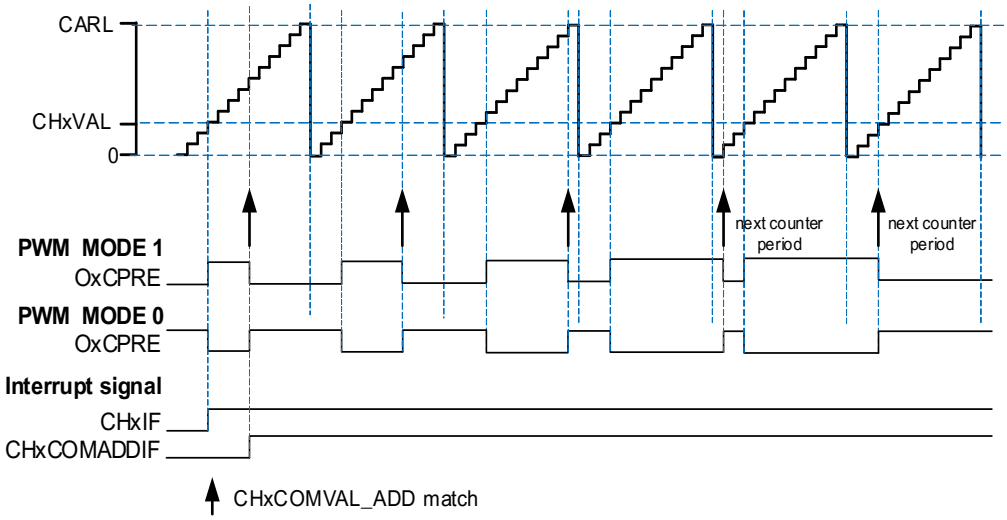
Figure 15-102. Channel x output PWM with CHxVAL or CHxCOMVAL_ADD exceeds CARL



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 15-103. 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 15-103. Channel x output PWM duty cycle changing with CHxCOMVAL_ADD



If more than one channels are configured in composite PWM mode, it is possible to fix an offset for the channel x match edge of each pair with respect to other channels. This behavior is useful in the generation of lighting PWM control signals where it is desirable that edges are not coincident with each other pair to help eliminate noise generation. The CHxVAL register value is the shift of the PWM pulse with respect to the beginning of counter period.

Output match pulse select

Basing on that CHx_O (x=0, 1) outputs are configured by CHxCOMCTL[3:0] (x=0, 1) bits when the match events occur, the output signal is configured by CHxOMPSEL[1:0] (x=0, 1) bit to be normal or a pulse.

When the match events occur, the CHxOMPSEL[1:0] (x=0, 1) bits are used to select the output of OxCPRE which drives CHx_O:

- CHxOMPSEL = 2'b00, the OxCPRE signal is output normally with the configuration of CHxCOMCTL[3:0] bits;
- CHxOMPSEL = 2'b01, only the counter is counting up, the OxCPRE signal is output a pulse when the match events occur, and the pulse width is one CK_TIMER clock cycle.
- CHxOMPSEL = 2'b10, only the counter is counting down, the OxCPRE signal is output a pulse when the match events occur, and the pulse width is one CK_TIMER clock cycle.
- CHxOMPSEL = 2'b11, both the counter is counting up and counting down, the OxCPRE signal is output a pulse when the match events occur, and the pulse width is one CK_TIMER clock cycle.

Figure 15-104. CHx_O output with a pulse in edge-aligned mode (CHxOMPSEL!=2'b00)

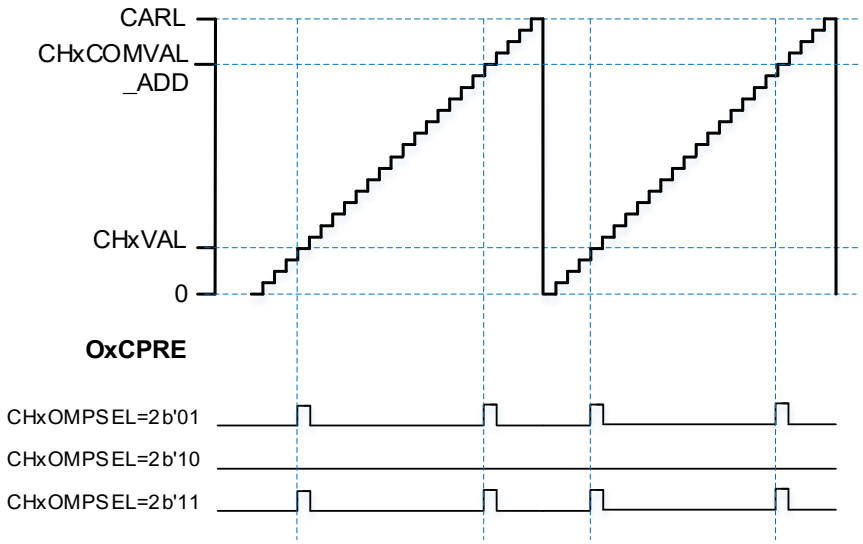
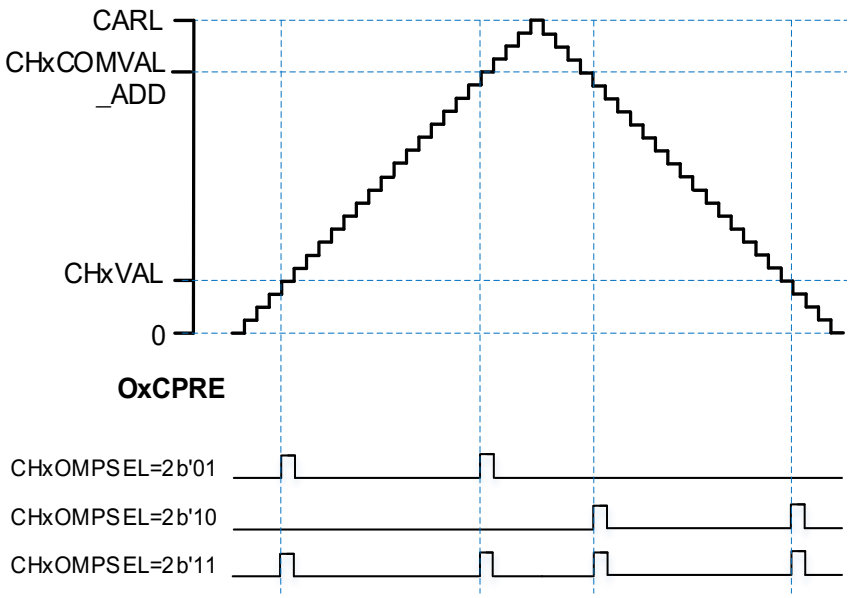


Figure 15-105. CHx_O output with a pulse in center-aligned mode (CHxOMPSEL!=2'b00)



Channel output prepare signal

As is shown in ***Figure 15-93. Output compare logic (when MCHxMSEL = 2'b00, x=0)***, ***Figure 15-94. Output compare logic (when MCHxMSEL = 2'b11, x=0)*** and ***Figure 15-95. Output compare logic (x=1)***, when TIMERx is configured in compare match output mode, a middle signal named OxCPRE or MOxCPRE (channel x output or multi mode channel x output prepare signal) will be generated before the channel outputs signal.

The OxCPRE and MOxCPRE signal have several types of output function. The OxCPRE signal type is defined by configuring the CHxCOMCTL bit and the MOxCPRE signal type is defined by configuring the MCHxCOMCTL bit.

Take OxCPRE as an example for description below, these include keeping the original level

by configuring the CHxCOMCTL field to 0x00, setting to high by configuring the CHxCOMCTL field to 0x01, setting to low by configuring the CHxCOMCTL field to 0x02 or toggling signal by configuring the CHxCOMCTL field to 0x03 when the counter value matches the content of the TIMERx_CHxCV register.

The PWM mode 0/ PWM mode 1 output is another output type of OxCPRE which is setup by configuring the CHxCOMCTL field to 0x06/0x07. In these modes, the OxCPRE signal level is changed according to the counting direction and the relationship between the counter value and the TIMERx_CHxCV content. Refer to the definition of relative bit for more details.

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

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

Outputs complementary

The outputs of CHx_O and MCHx_O have two situations:

- MCHxMSEL=2'b00: The MCHx_O output is independent from the CHx_O output;
- MCHxMSEL=2'b11: The outputs of MCHx_O and CHx_O are complementary and the MCHxOMCTL bits are not used in the generation of the MCHx_O output.

Function of complementary is for a pair of channels, CHx_O and MCHx_O, the two output signals cannot be active at the same time. TIMERx's one pair of channel has this function. The complementary signals CHx_O and MCHx_O are controlled by a group of parameters: the ChxEN and MCHxEN bits in the TIMERx_CHCTL2 register, the POEN, ROS and IOS bits in the TIMERx_CCHP register, ISOx and ISOxN bits in the TIMERx_CTL1 register. The output polarity is determined by CHxP and MCHxP bits in the TIMERx_CHCTL2 register.

When the 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 15-15. Complementary outputs controlled by parameters (MCHxMSEL =2'b11)***.

Table 15-15. Complementary outputs controlled by parameters (MCHxMSEL =2'b11)

Complementary Parameters					Output Status	
POEN	ROS	IOS	ChxEN	MCHxEN	CHx_O	MCHx_O
0	0/1	0	0	0	CHx_O / CHx_ON = LOW CHx_O / CHx_ON output disable(1).	
				1	CHx_O/ CHx_ON output "off-state" (2): the CHx_O/ CHx_ON output inactive level firstly: CHx_O = CHxP, CHx_ON = CHxNP; If the clock for deadtime generator is present, after a deadtime: CHx_O = ISOx, CHx_ON = ISOxN. (3)	
			1	0		
				1		
		1	x	x	CHx_O/ CHx_ON output "off-state": the CHx_O/ CHx_ON output inactive level firstly: CHx_O = CHxP, CHx_ON = CHxNP; If the clock for deadtime generator is present, after a deadtime: CHx_O = ISOx, CHx_ON = ISOxN.	
1	0	0/1	0	0	CHx_O/MCHx_O = LOW CHx_O/MCHx_O output disable.	
				1	CHx_O = LOW CHx_O output disable.	MCHx_O=OxCPRE⊕(4)MCHxP MCHx_O output enable.
			1	0	CHx_O=OxCPRE⊕CHxP CHx_O output enable.	MCHx_O = LOW MCHx_O output disable.
				1	CHx_O=OxCPRE⊕CHxP CHx_O output enable.	MCHx_O = (!OxCPRE)(5)⊕MCHxP. MCHx_O output enable.
	1		0	0	CHx_O = CHxP CHx_O output "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 / CHx_ON are disconnected to corresponding pins, the pin is floating with GPIO pull up/down setting which will be Hi-Z if no pull.

(2) "off-state": CHx_O / CHx_ON output with inactive state (e.g., CHx_O = 0⊕CHxP = CHxP).

(3) See Break mode section for more details.

(4) ⊕: Xor calculate.

(5) (! OxCPRE)： the complementary output of the OxCPRE signal.

Dead time insertion

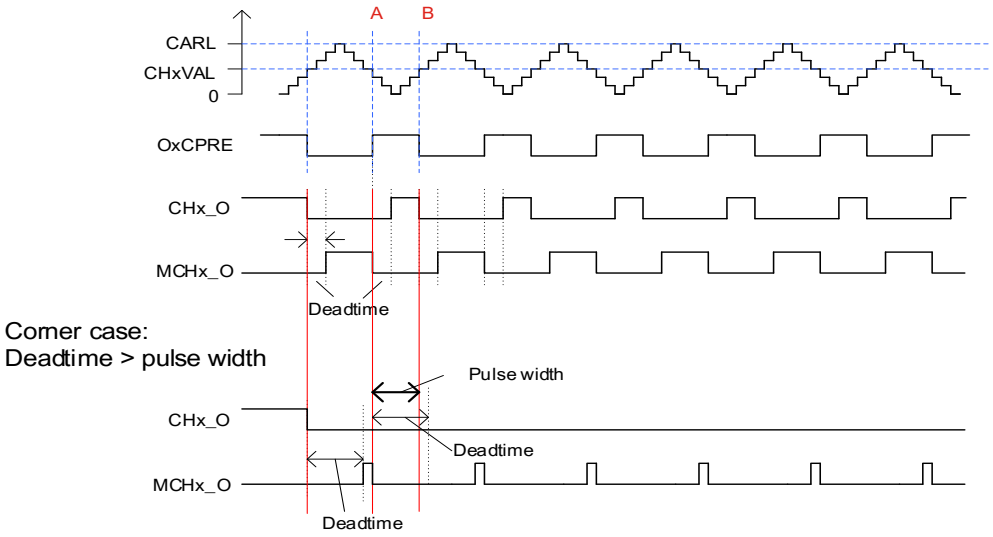
The dead time insertion is enabled when MCHxMSEL=2'b11 and both ChxEN and MCHxEN are configured to 1'b1, it is also necessary to configure POEN to 1. The field named DTCFG defines the dead time delay that can be used for all channels. Refer to the ***Complementary channel protection register (TIMERx_CCHP)*** for details about the delay time.

The dead time delay insertion ensures that two complementary signals are not active at the same time.

When the channel x match event (TIMERx_CNT = CHxVAL) occurs, OxCPRE will be toggled in PWM mode 0. At point A in ***Figure 15-106. 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 837pprox. 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 15-106. Complementary output with dead-time insertion***).

Figure 15-106. Complementary output with dead-time insertion



Break function

The MCHx_O output is the inverse of the CHx_O output when the MCHxMSEL=2'b11 (and the MCHxOMCTL bits are not used in the generation of the MCHx_O output). In this case, CHx_O and MCHx_O signals cannot be set to active level at the same time.

The general level3 timers have the BREAK0 function. The BREAK0 function can be enabled by setting the BRK0EN bit in the TIMERx_CCHP register. The break input polarity is configured by the BRK0P bit in TIMERx_CCHP register, the input is active on level.

In BREAK0 function, CHx_O and MCHx_O are controlled by the POEN, OAEN, IOS and ROS bits in the TIMERx_CCHP register, ISOx and ISOxN bits in the TIMERx_CTL1 register.

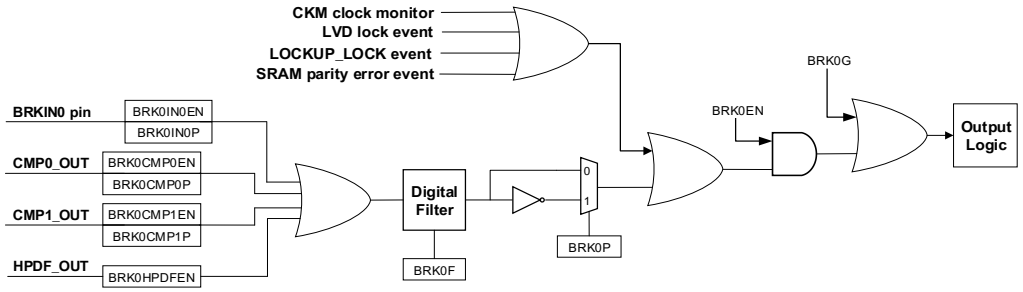
The break event is the result of logic Ored of all sources. The BREAK0 function can handle three types of event sources:

- External sources: coming from BRKIN0 input;
- System sources: HXTAL stuck event which is generated by Clock Monitor CKM in RCU, LVD lock event, Cortex®-M7 LOCKUP_LOCK event or SRAM parity error event;
- 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 15-107. 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)***.

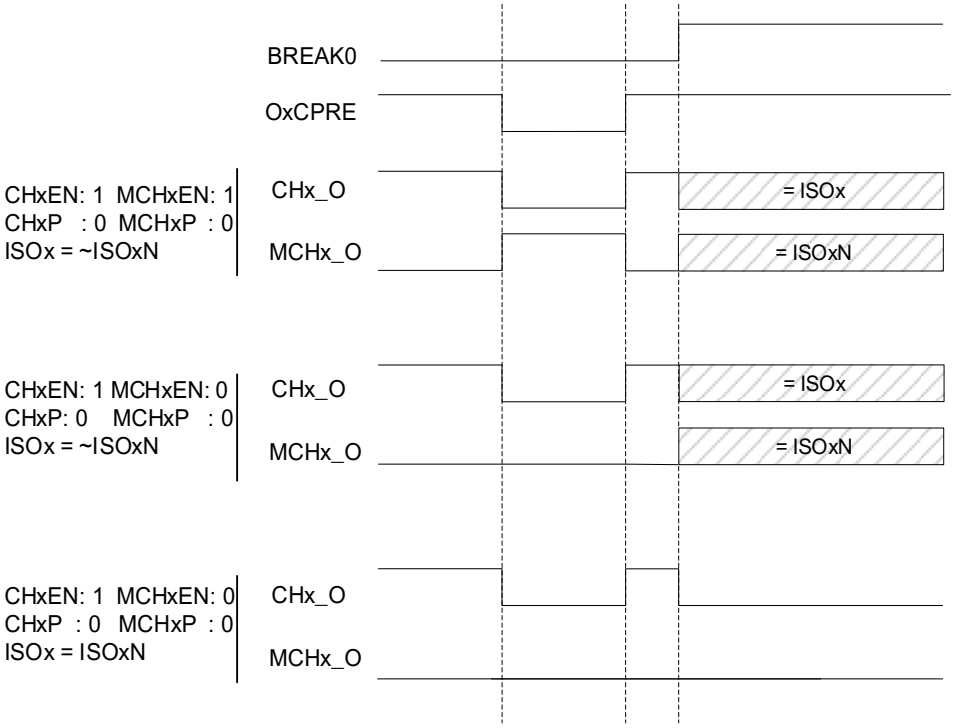
Figure 15-107. BREAK0 function logic diagram



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 15-108. 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 15-108. 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_CCHP 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_CCHP 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:

- 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.
- POEN=1: when the channel outputs are enabled, the BRKIN0 pin cannot be released even if the BRK0REL is set.

Table 15-16. Break function input pins locked/ released conditions

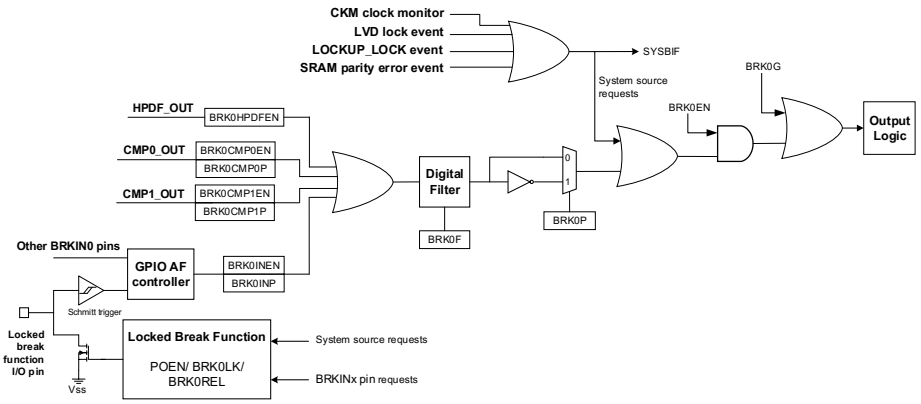
POEN	BRK0LK	BRK0REL	Break input pin state
0	1	0	Locked
	1	1	Released

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:

- Set the BRK0REL bit to 1 and released the BRKIN0 pin;
- The software waits for the system break sources inactive, and then clears the SYSBIF flag;
- 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 15-109. BRKIN0 pin logic with BREAK0 function

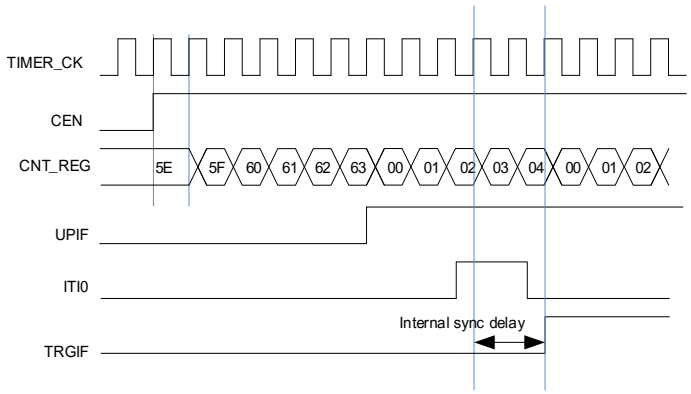
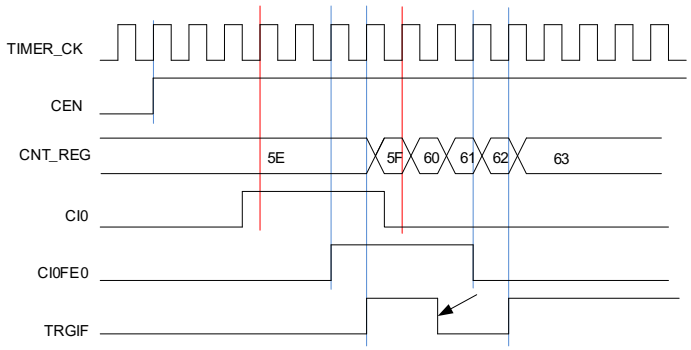
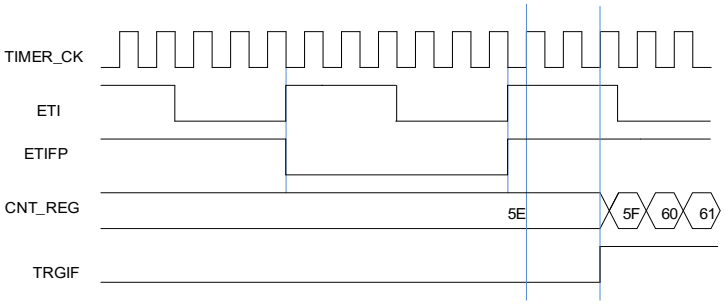


Master-slave management

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=14,40,41,42,43,44).

Table 15-17. 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 10010: MCI0FEM0 10011: ITI14	If you choose the CixFEx(x=0, 1) or MCIxFEMx(x=0), configure the CHxP, MCHxP and MCHxFP for the polarity selection and inversion.	For the ITIx no filter and prescaler can be used. For the Cix/ MCIx, configure Filter by CHxCAPFLT/ MCHxCAPFLT, no prescaler can be used.
Exam1	**Restart mode** The counter can be clear and restart when a rising trigger input.	TSCFG3[4:0] 5'b00001, ITI0 is the selection.	For ITI0, no polarity selector can be used.	For the ITI0, no filter and prescaler can be used.

	Mode Selection	Source Selection	Polarity Selection	Filter and Prescaler
	Figure 15-110. Restart mode  			
Exam2	**Pause mode** The counter can be paused when the trigger input is low.	TSCFG4[4:0] =5'b00110, CI0FE0 is the selection.	TI0S=0.(Non-xor) [MCHxP=0, CH0P=0] no inverted. Capture will be sensitive to the rising edge only.	Filter is bypass in this example.
	Figure 15-111. 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 15-112. Event mode  			

Single pulse mode

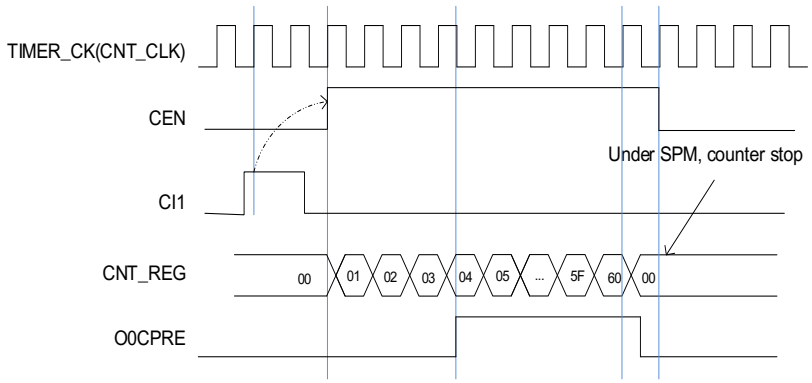
Single pulse mode is opposite to the repetitive mode, which can be enabled by setting SPM in TIMERx_CTL0. If SPM is set, the counter will be cleared and stopped automatically when the next update event occurs. In order to get a pulse waveform, the TIMERx is configured to PWM mode or compare mode by CHxCOMCTL or MCHxCOMCTL bits.

Once the timer is set to the single pulse mode, it is not necessary to configure the timer enable bit CEN in the TIMERx_CTL0 register to 1 to enable the counter. Setting the CEN bit to 1 or a trigger signal edge can generate a pulse and then keep the CEN bit at a high state until the update event occurs or the CEN bit is written to 0 by software. If the CEN bit is cleared to 0 by software, the counter will be stopped and its value will be held. If the CEN bit is automatically cleared to 0 by a hardware update event, the counter will be reinitialized.

In the single pulse mode, the active edge of trigger which sets the CEN bit to 1 will enable the counter. However, there exists several clock delays to perform the comparison result between the counter value and the TIMERx_CHxCV value. After a trigger rising occurs in the single pulse mode, the OxCPRE signal will immediately be forced to the state which the OxCPRE/ MOxCPRE signals will change to, as the compare match event occurs without taking the comparison result into account.

Single pulse mode is also applicable to composite PWM mode (CHxCPWMEN = 1'b1 and CHxMS[2:0] = 3'b000).

Figure 15-113. 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);

- 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).

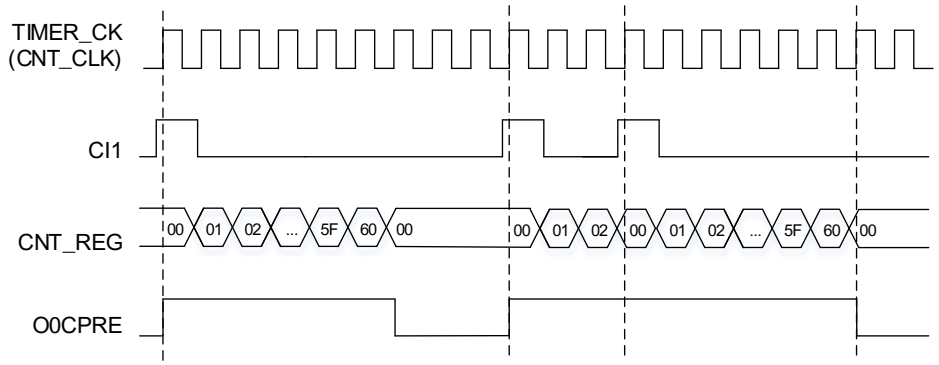
In delayable SPM mode 0. The behavior of OxCPRE/ MOxCPRE is performed as in PWM mode 0. When counting up, the OxCPRE/ MOxCPRE is active. When a trigger event occurs, the OxCPRE/ MOxCPRE is inactive. The OxCPRE/ MOxCPRE is active again at the next update event; When counting down, the OxCPRE/ MOxCPRE 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.

Note:

3) The center-aligned counting mode cannot be used in this mode and the CAM[1:0] = 2'b00(in TIMERx_CTL0 register);
4) 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 15-114. delayable single pulse mode TIMERx_CHxCV=0x00, TIMERx_CAR=0x60



Timers interconnection

Please refer to ***Advanced timer (TIMERx, x=0, 7)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®-M7 halted, and the TIMERx_HOLD configuration bit in DBG register set to 1, the TIMERx counter stops.

15.3.5. Register definition (TIMERx, x=14,40,41,42,43,44)

TIMER14 base address: 0x4001 4000

TIMER40 base address: 0x4001 D000

TIMER41 base address: 0x4001 D400

TIMER42 base address: 0x4001 D800

TIMER43 base address: 0x4001 DC00

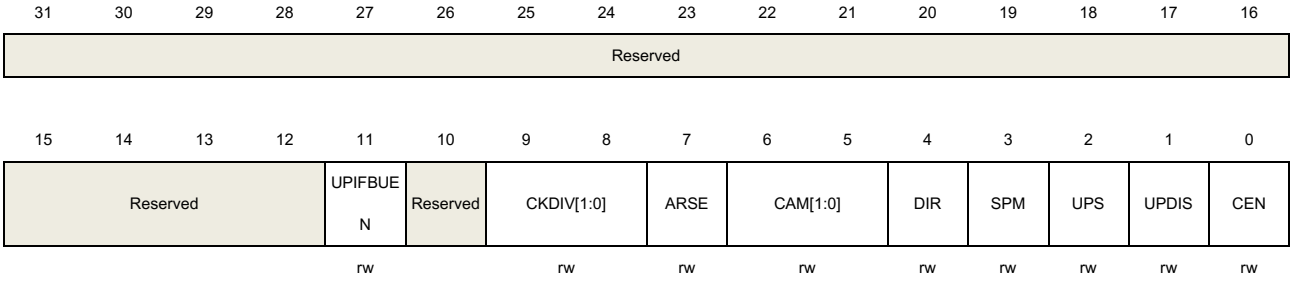
TIMER44 base address: 0x4001 F000

Control register 0 (TIMERx_CTL0)

Address offset: 0x00

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
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 generates an update interrupt or a DMA request.
1	UPDIS	Update disable This bit is used to enable or disable the update event generation. 0: Update event enable. The update event is generated and the buffered registers are loaded with their preloaded values when one of the following events occurs: – The UPG bit is set. – The counter generates an overflow or underflow event. – The slave mode controller generates an update event. 1: Update event disable. The buffered registers keep their value, while the counter

		and the prescaler are reinitialized if the UPG bit is set or the slave mode controller generates a hardware reset event.
0	CEN	Counter enable 0: Counter disable 1: Counter enable The CEN bit must be set by software when timer works in external clock mode, pause mode or decoder mode. While in event mode, the hardware can set the CEN bit automatically.

Control register 1 (TIMERx_CTL1)

Address offset: 0x04
Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					ISO1	ISO0N	ISO0	TI0S	MMC0[2:0]			DMAS	CCUC[0]	Reserved	CCSE
					rw	rw	rw	rw	rw			rw	rw		rw

Bits	Fields	Descriptions
31:30	CCUC[2:1]	Commutation control shadow register update control Refer to CCUC [0] description.
29:11	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_CCHP register is 00.
8	ISO0	Idle state of channel 0 output 0: When POEN bit is reset, CH0_O is set low. 1: When POEN bit is reset, CH0_O is set high. The CH0_O output changes after a dead time if MCH0_O is implemented. This bit can be modified only when PROT[1:0] bits in TIMERx_CCHP register is 00.
7	TI0S	Channel 0 trigger input selection 0: The TIMERx_CH0 pin input is selected as channel 0 trigger input. 1: The result of combinational XOR of TIMERx_CH0, TIMERx_CH1 and

		TIMERx_CH2 pins is selected as channel 0 trigger input.
6:4	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. 101: The shadow registers update when the counter generates an underflow event. 110: The shadow registers update when the counter generates an overflow or underflow event. When a channel does not have a complementary output, this bit has no effect. **Note**: When CCUC[2:0] bit-field are set to 100, the update of the shadow registers also considers the value the CCUSEL bit in the TIMERx_CFG register.

1	Reserved	Must be kept at reset value.
0	CCSE	Commutation control shadow enable 0: The shadow registers (for ChxEN, MCHxEN and CHxCOMCTL bits) are disabled. 1: The shadow registers (for ChxEN, MCHxEN and CHxCOMCTL bits) are enabled. After these bits have been written, they are updated when commutation event comes. When a channel does not have a complementary output, this bit has no effect.

Slave mode configuration register (TIMERx_SMCFG)

Address offset: 0x08

Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7	MSM	Master-slave mode This bit can be used to synchronize the selected timers to begin counting at the same time. The TRGI is used as the start event, and through TRGO, timers are connected. 0: Master-slave mode disabled 1: Master-slave mode enabled
6:0	Reserved	Must be kept at reset value.

DMA and interrupt enable register (TIMERx_DMAINTEN)

Address offset: 0x0C

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		CH1COMADDIE	CH0COMADDIE	Reserved			MCH0DEN	Reserved			MCH0IE	Reserved			
		rw	rw				rw				rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	TRGDEN	CMTDEN	Reserved		CH1DEN	CH0DEN	UPDEN	BRKIE	TRGIE	CMTIE	Reserved		CH1IE	CH0IE	UPIE

rw rw rw rw rw rw rw rw rw rw rw 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 MCH0SEL[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 MCH0SEL[1:0] = 2'b00).
19:15	Reserved	Must be kept at reset value.
14	TRGDEN	Trigger DMA request enable 0: Disabled 1: Enabled
13	CMTDEN	Commutation DMA request enable 0: Disabled 1: Enabled
12:11	Reserved	Must be kept at reset value.
10	CH1DEN	Channel 1 capture/compare DMA request enable 0: Disabled 1: Enabled
9	CH0DEN	Channel 0 capture/compare DMA request enable 0: Disabled

		1: Enabled
8	UPDEN	Update DMA request enable 0: Disabled 1: Enabled
7	BRKIE	Break interrupt enable 0: Disabled 1: Enabled
6	TRGIE	Trigger interrupt enable 0: Disabled 1: Enabled
5	CMTIE	Commutation interrupt enable 0: Disabled 1: Enabled
4:3	Reserved	Must be kept at reset value.
2	CH1IE	Channel 1 capture/compare interrupt enable 0: Disabled 1: Enabled
1	CH0IE	Channel 0 capture/compare interrupt enable 0: Disabled 1: Enabled
0	UPIE	Update interrupt enable 0: Disabled 1: Enabled

Interrupt flag register (TIMERx_INTF)

Address offset: 0x10

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		CH1COMADDIF	CH0COMADDIF	Reserved			MCH0OF	Reserved			MCH0IF	Reserved			
		rc_w0	rc_w0				rc_w0				rc_w0				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					CH1OF	CH0OF	Reserved	BRK0IF	TRGIF	CMTIF	Reserved		CH1IF	CH0IF	UPIF
					rc_w0	rc_w0		rc_w0	rc_w0	rc_w0			rc_w0	rc_w0	rc_w0

Bits	Fields	Descriptions

31:30	Reserved	Must be kept at reset value.
29	CH1COMADDIF	Channel 1 additional compare interrupt flag. Refer to CH0COMADDIF description.
28	CH0COMADDIF	Channel 0 additional compare interrupt flag. This flag is set by hardware and cleared by software. If channel 0 is in output mode, this flag is set when a compare event occurs. 0: No channel 0 output compare interrupt occurred 1: Channel 0 output compare interrupt occurred **Note:** This flag just used in composite PWM mode.
27:25	Reserved	Must be kept at reset value.
24	MCH0OF	Multi mode channel 0 over capture flag When multi mode channel 0 is configured in input mode, this flag is set by hardware when a capture event occurs while MCH0IF flag has already been set. This flag is cleared by software. 0: No over capture interrupt occurred 1: Over capture interrupt occurred
23:21	Reserved	Must be kept at reset value.
20	MCH0IF	Multi mode channel 0 capture/compare interrupt flag This flag is set by hardware and cleared by software. If multi mode channel 0 is in input mode, this flag is set when a capture event occurs. If multi mode channel 0 is in output mode, this flag is set when a compare event occurs. If multi mode channel 0 is set to input mode, this bit will be reset by reading TIMERx_MCH0CV. 0: No multi mode channel 0 capture/compare interrupt occurred 1: Multi mode channel 0 capture/compare interrupt occurred
19:11	Reserved	Must be kept at reset value.
10	CH1OF	Channel 1 over capture flag Refer to CH0OF description
9	CH0OF	Channel 0 over capture flag When channel 0 is configured in input mode, this flag is set by hardware when a capture event occurs while CH0IF flag has already been set. This flag is cleared by software. 0: No over capture interrupt occurred 1: Over capture interrupt occurred
8	Reserved	Must be kept at reset value.
7	BRK0IF	BREAK0 interrupt flag This flag is set by hardware when the break input is active, and cleared by software

		if the BREAK0 input is not at active level. 0: No active level on BREAK0 input has been detected. 1: An active level on BREAK0 input has been detected.
6	TRGIF	Trigger interrupt flag This flag is set by hardware on trigger event and cleared by software. When the slave mode controller is enabled in all modes but pause mode, an active edge of trigger input generates a trigger event. When the slave mode controller is enabled in pause mode, either edge of the trigger input can generate a trigger event. 0: No trigger event occurred 1: Trigger interrupt occurred
5	CMTIF	Channel commutation interrupt flag This flag is set by hardware when the commutation event of channel occurs, and cleared by software. 0: No channel commutation interrupt occurred 1: Channel commutation interrupt occurred
4:3	Reserved	Must be kept at reset value.
2	CH1IF	Channel 1 capture/compare interrupt flag Refer to CH0IF description
1	CH0IF	Channel 0 capture/compare interrupt flag This flag is set by hardware and cleared by software. If channel 0 is in input mode, this flag is set when a capture event occurs. If channel 0 is in output mode, this flag is set when a compare event occurs. If channel 0 is set to input mode, this bit will be reset by reading TIMERx_CH0CV. 0: No channel 0 interrupt occurred 1: Channel 0 interrupt occurred
0	UPIF	Update interrupt flag This bit is set by hardware when an update event occurs and cleared by software. 0: No update interrupt occurred 1: Update interrupt occurred

Software event generation register (TIMERx_SWEVG)

Address offset: 0x14
Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		CH1COMADDG	CH0COMADDG	Reserved							MCH0G	Reserved			
		w	w								w				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Reserved	BRK0G	TRGG	CMTG	Reserved	CH1G	CH0G	UPG
	w	w	w		w	w	w

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29	CH1COMADDG	Channel 1 additional compare event generation. Refer to CH0COMADDG description.
28	CH0COMADDG	Channel 0 additional compare event generation. This bit is set by software to generate a compare event in channel 0 additional, it is automatically cleared by hardware. When this bit is set, the CH0COMADDIF flag will be set, and the corresponding interrupt will be sent if enabled. 0: No generate a channel 0 additional compare event 1: Generate a channel 0 additional compare event **Note:** This bit just used in composite PWM mode.
27:21	Reserved	Must be kept at reset value.
20	MCH0G	Multi mode channel 0 capture or compare event generation. This bit is set by software to generate a capture or compare event in multi mode channel 0, it is automatically cleared by hardware. When this bit is set, the MCH0IF flag will be set, and the corresponding interrupt or DMA request will be sent if enabled. In addition, if multi mode channel 0 is configured in input mode, the current value of the counter is captured to TIMERx_MCH0CV register, and the MCH0OF flag is set if the MCH0IF flag has been set. 0: No generate a multi mode channel 0 capture or compare event 1: Generate a multi mode channel 0 capture or compare event
19:8	Reserved	Must be kept at reset value.
7	BRK0G	BREAK0 event generation This bit is set by software to generate an event and cleared by hardware automatically. When this bit is set, the POEN bit will be cleared and BRK0IF flag will be set, related interrupt can occur if enabled. 0: No generate a BREAK0 event 1: Generate a BREAK0 event
6	TRGG	Trigger event generation This bit is set by software and cleared by hardware automatically. When this bit is set, the TRGIF flag in TIMERx_INTF register will be set, related interrupt or DMA transfer can occur if enabled. 0: No generate a trigger event 1: Generate a trigger event
5	CMTG	Channel commutation event generation

		This bit is set by software and cleared by hardware automatically. When this bit is set, channel's capture/compare control registers (ChxEN, MCHxEN and CHxCOMCTL bits) are updated based on the value of CCSE (in the TIMERx_CTL1). 0: No affect 1: Generate channel commutation update event
4:3	Reserved	Must be kept at reset value.
2	CH1G	Channel 1 capture or compare event generation Refer to CH0G description
1	CH0G	Channel 0 capture or compare event generation This bit is set by software to generate a capture or compare event in channel 0, it is automatically cleared by hardware. When this bit is set, the CH0IF flag will be set, and the corresponding interrupt or DMA request will be sent if enabled. In addition, if channel 0 is configured in input mode, the current value of the counter is captured to TIMERx_CH0CV register, and the CH0OF flag is set if the CH0IF flag has been set. 0: No generate a channel 0 capture or compare event 1: Generate a channel 0 capture or compare event
0	UPG	Update event generation This bit can be set by software, and automatically cleared by hardware. When this bit is set, the counter is cleared if the up counting mode is selected. The prescaler counter is cleared at the same time. 0: No generate an update event 1: Generate an update event

Channel control register 0 (TIMERx_CHCTL0)

Address offset: 0x18

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH1MS [2]	CH0MS [2]	CH1COM ADDSEN	CH0COM ADDSEN	CH1ADDU PS	CH0ADDU PS	Reserved	CH1COM CTL[3]	Reserved							CH0COM CTL[3]
		Reserved					Reserved								Reserved
rw	rw	rw	rw	rw	rw		rw								rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	CH1COMCTL[2:0]			CH1COM SEN	Reserved	CH1MS[1:0]		Reserved	CH0COMCTL[2:0]			CH0COM SEN	Reserved	CH0MS[1:0]	
CH1CAPFLT[3:0]				CH1CAPPSC[1:0]				CH0CAPFLT[3:0]				CH0CAPPSC[1:0]			
rw				rw		rw		rw				rw		rw	

Output compare mode:

Bits	Fields	Descriptions
31	CH1MS[2]	Channel 1 I/O mode selection Refer to CH1MS[1:0]description
30	CH0MS[2]	Channel 0 I/O mode selection Refer to CH0MS[1:0] description
29	CH1COMADDSEN	Channel 1 additional compare output shadow enable Refer to CH0COMADDSEN description.
28	CH0COMADDSEN	Channel 0 additional compare output shadow enable When this bit is set, the shadow register of TIMERx_CH0COMV_ADD register which updates at each update event will be enabled. 0: Channel 0 additional compare output shadow disabled 1: Channel 0 additional compare output shadow enabled The PWM mode can be used without validating the shadow register only in single pulse mode (SPM bit in TIMERx_CTL0 register is set). This bit cannot be modified when PROT[1:0] bit-field in TIMERx_CCHP register is 11 and CH0MS bit-field is 000.
27	CH1ADDUPS	Channel 1 additional update source 0: TIMERx_CH1COMV_ADD register is updated when an update event occurs. 1: TIMERx_CH1COMV_ADD register is updated when the counter matches the value of CH1VAL.
26	CH0ADDUPS	Channel 0 additional update source 0: TIMERx_CH0COMV_ADD register is updated when an update event occurs. 1: TIMERx_CH0COMV_ADD register is updated when the counter matches the value of CH0VAL.
25	Reserved	Must be kept at reset value.
24	CH1COMCTL[3]	Channel 1 compare output control Refer to CH0COMCTL[2:0] description
23:17	Reserved	Must be kept at reset value.
16	CH0COMCTL[3]	Channel 0 compare output control Refer to CH0COMCTL[2:0] description
15	Reserved	Must be kept at reset value.
14:12	CH1COMCTL[2:0]	Channel 1 compare output control Refer to CH0COMCTL[2:0] description
11	CH1COMSEN	Channel 1 output compare shadow enable Refer to CH0COMSEN description
10	Reserved	Must be kept at reset value.
9:8	CH1MS[1:0]	Channel 1 mode selection

		This bit-field specifies the direction of the channel and the input signal selection. The CH1MS[2:0] bit-field is writable only when the channel is not active (When MCH1MSEL[1:0] = 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=14,40,41,42,43,44) register). 100~111: Reserved.
7	Reserved	Must be kept at reset value.
6:4	CH0COMCTL[2:0]	Channel 0 compare output control The CH0COMCTL[3] and CH0COMCTL[2:0] bit-field control the behavior of O0CPRE which drives CH0_O. 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_CCHP register is 11 and CH0MS bit-field is 000 (compare mode).
3	CH0COMSEN	Channel 0 compare output shadow enable When this bit is set, the shadow register of TIMERx_CH0CV register which updates at each update event will be enabled. 0: Channel 0 output compare shadow disabled 1: Channel 0 output compare shadow enabled The PWM mode can be used without validating the shadow register only in single pulse mode (SPM bit in TIMERx_CTL0 register is set). This bit cannot be modified when PROT[1:0] bit-field in TIMERx_CCHP register is 11 and CH0MS bit-field is 000.
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: 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=14,40,41,42,43,44) 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 word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		CH1PERFOREN	CH0PERFOREN	Reserved											
		rw	rw												

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								MCH1P	Reserved	CH1P	CH1EN	MCH0P	MCH0EN	CH0P	CH0EN
								rw		rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29	CH1PERFOREN	Channel 1 in the composite PWM mode, the period point match moment O1CPRE level setting Refer to CH0PERFOREN description.
28	CH0PERFOREN	Channel 0 in the composite PWM mode, the period point match moment O0CPRE level setting 0: disable 1: In the composite PWM0 mode, the level will be forced to high at the time of period point matching; In the composite PWM1 mode, the level at the matching time of the period point is forced to low. **Note:** In the central alignment mode, the period point is the counter underflow time; In the edge-aligned mode, the period point is the counter flow moment.
27:8	Reserved	Must be kept at reset value.
7	MCH1P	Multi mode channel 1 output polarity Refer to MCH0P description.
6	Reserved	Must be kept at reset value.
5	CH1P	Channel 1 capture/compare polarity Refer to CH0P description.
4	CH1EN	Channel 1 capture/compare enable

		Refer to CH0EN description.
3	MCH0P	Multi mode channel 0 output polarity When Multi mode channel 0 is configured in output mode, and the MCH0MSEL[1:0] = 2'b11, this bit specifies the MCH0_O output signal polarity. 0: Multi mode channel 0 output active high 1: Multi mode channel 0 output active low When CH0 is configured in input mode, in conjunction with CH0P, this bit is used to define the polarity of CH0. This bit cannot be modified when PROT[1:0] bit-field in TIMERx_CCHP register is 11 or 10.
2	MCH0EN	Multi mode channel 0 capture/compare enable When multi mode channel 0 is configured in output mode, setting this bit enables MCH0_O signal in active state. When multi mode channel 0 is configured in input mode, setting this bit enables the capture event in multi mode channel 0. 0: Multi mode channel 0 disabled 1: Multi mode channel 0 enabled
1	CH0P	Channel 0 capture/compare polarity When channel 0 is configured in output mode, this bit specifies the output signal polarity. 0: Channel 0 active high 1: Channel 0 active low When channel 0 is configured in input mode, these bits 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_CCHP register is 11 or 10.
0	CH0EN	Channel 0 capture/compare enable When channel 0 is configured in output mode, setting this bit enables CH0_O signal in active state. When channel 0 is configured in input mode, setting this bit enables the capture event in channel 0. 0: Channel 0 disabled 1: Channel 0 enabled

Counter register (TIMERx_CNT)

Address offset: 0x24

Reset value: 0x0000 0000

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

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

Prescaler register (TIMERx_PSC)

Address offset: 0x28

Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	PSC[15:0]	Prescaler value of the counter clock The PSC clock is divided by (PSC+1) to generate the counter clock. The value of this bit-field will be loaded to the corresponding shadow register at every update event.

Counter auto reload register (TIMERx_CAR)

Address offset: 0x2C

Reset value: 0x0000 FFFF

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

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

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

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CARL[15:0]	Counter auto reload value This bit-field specifies the auto reload value of the counter.

Counter repetition register 0 (TIMERx_CREP0)

Address offset: 0x30

Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								CREP0[7:0]							
								rw							

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	CREP0[7:0]	Counter repetition value 0 This bit-field specifies the update event generation rate. Each time the repetition counter counts down to zero, an update event will be generated. The update rate of the shadow registers is also affected by this bit-field when these shadow registers are enabled. **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
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH0VAL[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH0VAL[15:0]	Capture/compare value of channel 0 When channel 0 is configured in input mode, this bit-field indicates the counter value at the last capture event. And this bit-field is read-only. When channel 0 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event.

Channel 1 capture/compare value register (TIMERx_CH1CV)

Address offset: 0x38

Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1VAL[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH1VAL[15:0]	Capture/compare value of channel 1 When channel 1 is configured in input mode, this bit-field indicates the counter value at the last capture event. And this bit-field is read-only. When channel 1 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event.

Complementary channel protection register (TIMERx_CCHP)

Address offset: 0x44

Reset value: 0x0000 0000

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

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

Reserved			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_CCHP 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_CCHP 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_CCHP 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_CCHP 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_CCHP register is 00.

11 ROS
Run mode "off-state" enable
When POEN bit is set (Run mode), this bit can be set to enable the "off-state" for the channels which has been configured in output mode. Please refer to ***Table 15-15. Complementary outputs controlled by parameters (MCHxMSEL =2'b11)***.
0: "off-state" disabled. If the ChxEN or CHxNEN bit is reset, the corresponding

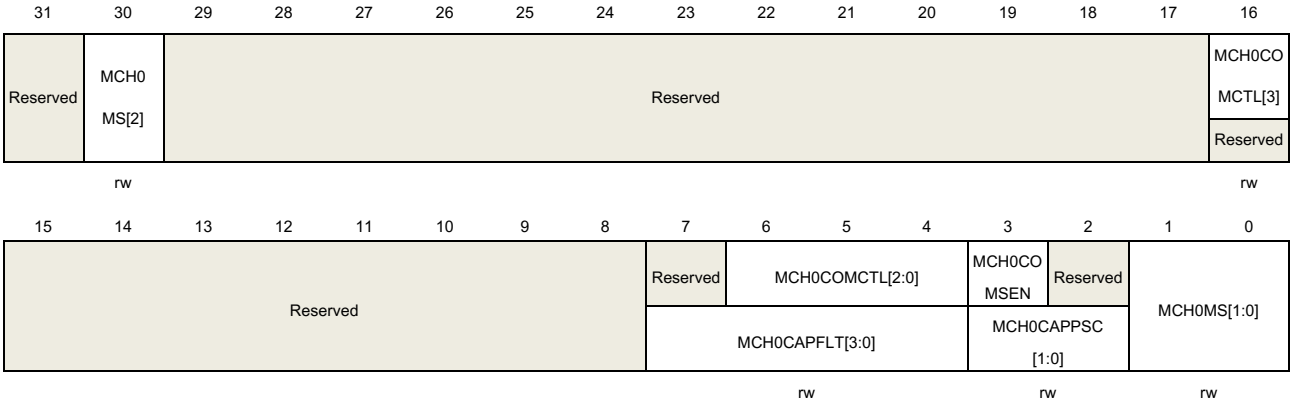
		channel is output disabled. 1: "off-state" enabled. If the ChxEN or CHxNEN bit is reset, the corresponding channel is "off-state". This bit cannot be modified when PROT [1:0] bit-field in TIMERx_CCHP register is 10 or 11.
10	IOS	Idle mode "off-state" enable When POEN bit is reset (Idle mode), this bit can be set to enable the "off-state" for the channels which has been configured in output mode. Please refer to ***Table 15-15. 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-field in TIMERx_CCHP register is 10 or 11.
9:8	PROT[1:0]	Complementary register protect control This bit-field specifies the write protection property of registers. 00: Protect disabled. No write protection. 01: PROT mode 0. The ISOx/ISOxN bits in TIMERx_CTL1 register, the BRK0EN/ BRK0P/OAEN/DTCFG bits in TIMERx_CCHP register are writing protected. 10: PROT mode 1. In addition to the registers in PROT mode 0, the CHxP/MCHxP bits in TIMERx_CHCTL2 register (if related channel is configured in output mode) the ROS/IOS bits in TIMERx_CCHP register 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_CCHP 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] x t_{DT}, t_{DT} = t_{DTS}. DTCFG[7:5] = 3'b10x: DT value = (64+DTCFG[5:0]) x t_{DT}, t_{DT} = t_{DTS}*2. DTCFG[7:5] = 3'b110: DT value = (32+DTCFG[4:0]) x t_{DT}, t_{DT} = t_{DTS}*8. DTCFG[7:5] = 3'b111: DT value = (32+DTCFG[4:0]) x t_{DT}, t_{DT} = t_{DTS}*16. This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP register is 00.

Multi mode channel control register 0 (TIMERx_MCHCTL0)

Address offset: 0x48

Reset value: 0x0000 0000

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



Output compare mode:

Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30	MCH0MS[2]	Multi mode channel 0 I/O mode selection Refer to MCH0MS[1:0] description.
29:17	Reserved	Must be kept at reset value.
16	MCH0COMCTL[3]	Multi mode channel 0 compare output control. Refer to MCH0COMCTL[2:0] description.
15:7	Reserved	Must be kept at reset value.
6:4	MCH0COMCTL[2:0]	Multi mode channel 0 output compare control When multi mode channel 0 is configured in output mode, and the MCH0MSEL[1:0] = 2'b00, the MCH0COMCTL[3] and MCH0COMCTL[2:0] bit-field control the behavior of MO0CPRE which drives MCH0_O. The active level of MO0CPRE is high, while the active level of MCH0_O depends on MCH0FP[1:0] bits. **Note**: When multi mode channel 0 is configured in output mode, and the MCH0MSEL[1:0] = 2'b11, the CH0COMCTL[2:0] bit-field controls the behavior of O0CPRE which drives CH0_O and MCH0_O, while the active level of CH0_O and MCH0_O depends on CH0P and MCH0P bits. 0000: Timing mode. The MO0CPRE signal keeps stable, independent of the comparison between the register TIMERx_MCH0CV and the counter TIMERx_CNT. 0001: Set the channel output on match. MO0CPRE signal is forced high when the counter matches the output compare register TIMERx_MCH0CV. 0010: Clear the channel output on match. MO0CPRE signal is forced low when the counter matches the output compare register TIMERx_MCH0CV.

		0011: Toggle on match. MO0CPRE toggles when the counter matches the output compare register TIMERx_MCH0CV. 0100: Force low. MO0CPRE is forced low level. 0101: Force high. MO0CPRE is forced high level. 0110: PWM mode 0. When counting up, MO0CPRE is active as long as the counter is smaller than TIMERx_MCH0CV, otherwise it is inactive. When counting down, MO0CPRE is inactive as long as the counter is larger than TIMERx_MCH0CV, otherwise it is active. 0111: PWM mode 1. When counting up, MO0CPRE is inactive as long as the counter is smaller than TIMERx_MCH0CV, otherwise it is active. When counting down, MO0CPRE is active as long as the counter is larger than TIMERx_MCH0CV, otherwise it is inactive. 1000: 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 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_CCHP register is 11 and MCH0MS bit-field is 00(compare mode).
3	MCH0COMSEN	Multi mode channel 0 output compare shadow enable When this bit is set, the shadow register of TIMERx_MCH0CV register which updates at each update event will be enabled. 0: Multi mode channel 0 output compare shadow disabled 1: Multi mode channel 0 output compare shadow enabled The PWM mode can be used without validating the shadow register only in single pulse mode (SPM bit in TIMERx_CTL0 register is set). This bit cannot be modified when PROT[1:0] bit-field in TIMERx_CCHP register is 11 and MCH0MS bit-field is 00.
2	Reserved	Must be kept at reset value.

1:0	MCH0MS[1:0]	Multi mode channel 0 I/O mode selection This bit-field specifies the work mode of the channel and the input signal selection. This bit-field is writable only when the channel is not active (MCH0EN bit in TIMERx_CHCTL2 register is reset). 000: Multi mode channel 0 is 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=14,40,41,42,43,44) 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.

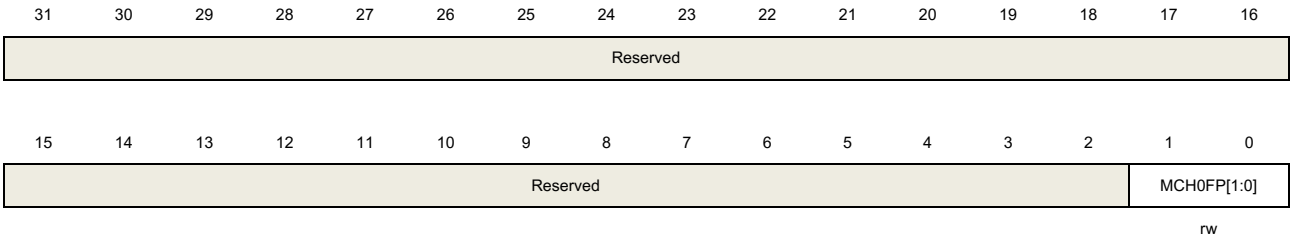
3:2	MCH0CAPPSC[1:0]	Multi mode channel 0 input capture prescaler This bit-field specifies the factor of the prescaler on channel 0 input. The prescaler is reset when MCH0EN bit in TIMERx_CHCTL2 register is cleared. 00: Prescaler disabled, capture is done on each channel input edge. 01: Capture is done 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 word(32-bit).



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_CCHP register is 11 or 10.

Multi mode channel 0 capture/compare value register (TIMERx_MCH0CV)

Address offset: 0x54

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	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		CH1CPWMEN	CH0CPWMEN	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					BRK0CMP1P	BRK0CMP0P	Reserved								BRK0IN0P
					rw	rw									rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					BRK0CMP1EN	BRK0CMP0EN	BRK0HPDFEN	Reserved							BRK0IN0EN
					rw	rw	rw								rw

Bits	Fields	Descriptions
31:27	Reserved	Must be kept at reset value.
26	BRK0CMP1P	BREAK0 CMP1 input polarity This bit is used to configure the CMP1 input polarity, and the specific polarity is determined by this bit and the BRK0P bit. 0: CMP1 input signal will not be inverted (BRK0P =0, the input signal is active low; BRK0P =1, the input signal is active high) 1: CMP1 input signal will be inverted (BRK0P=0, the input signal is active high; BRK0P =1, the input signal is active low) This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP 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_CCHP 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_CCHP register is 00.
15:11	Reserved	Must be kept at reset value.
10	BRK0CMP1EN	BREAK0 CMP1 enable 0: CMP1 input disabled 1: CMP1 input enabled This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP 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_CCHP register is 00.
8	BRK0HPDFEN	BREAK0 HPDF input(HPDF_BREAK[x], please refer to ***Table 33-2. HPDF internal signal*** enable 0: HPDF input disabled 1: HPDF input enabled This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP register is 00.
7:1	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_CCHP register is 00.

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															

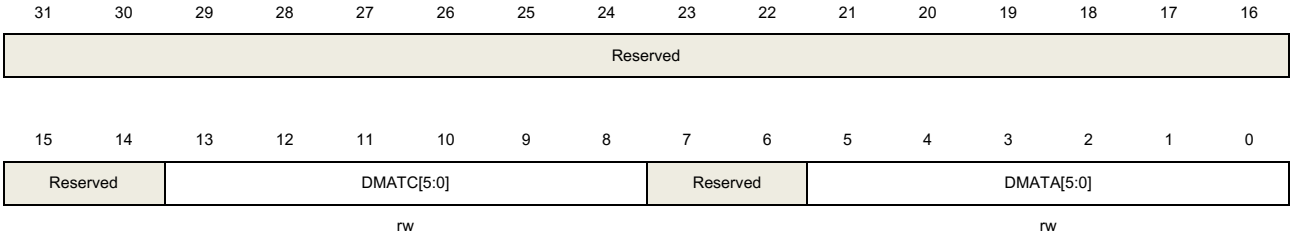
Bits	Fields	Descriptions
31:0	CREP1[31:0]	Counter repetition value 1 This bit-field is 32 bits and can be read on the fly. This bit-field specifies the update event generation rate. Each time the repetition counter counts down to zero, an update event will be generated. The update rate of the shadow 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).

DMA configuration register (TIMERx_DMACFG)

Address offset: 0xE0

Reset value: 0x0000 0000

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



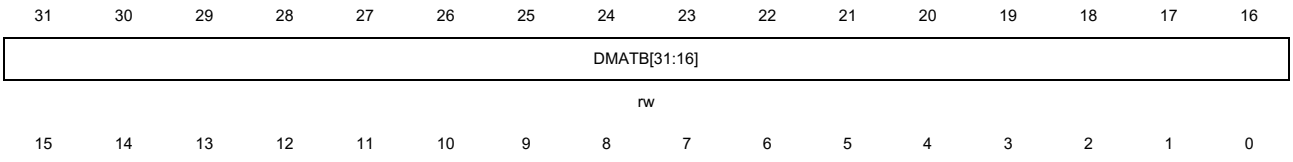
Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value.
13:8	DMATC[5:0]	DMA transfer count This field defines the times of accessing(R/W) the TIMERx_DMATB register by DMA. 6'b000000: transfer 1 time 6'b000001: transfer 2 times … 6'b100101: transfer 38 times
7:6	Reserved	Must be kept at reset 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).



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 100,101,110. 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.

15.4. General level4 timer (TIMERx, x=15,16)

15.4.1. 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.

15.4.2. Characteristics

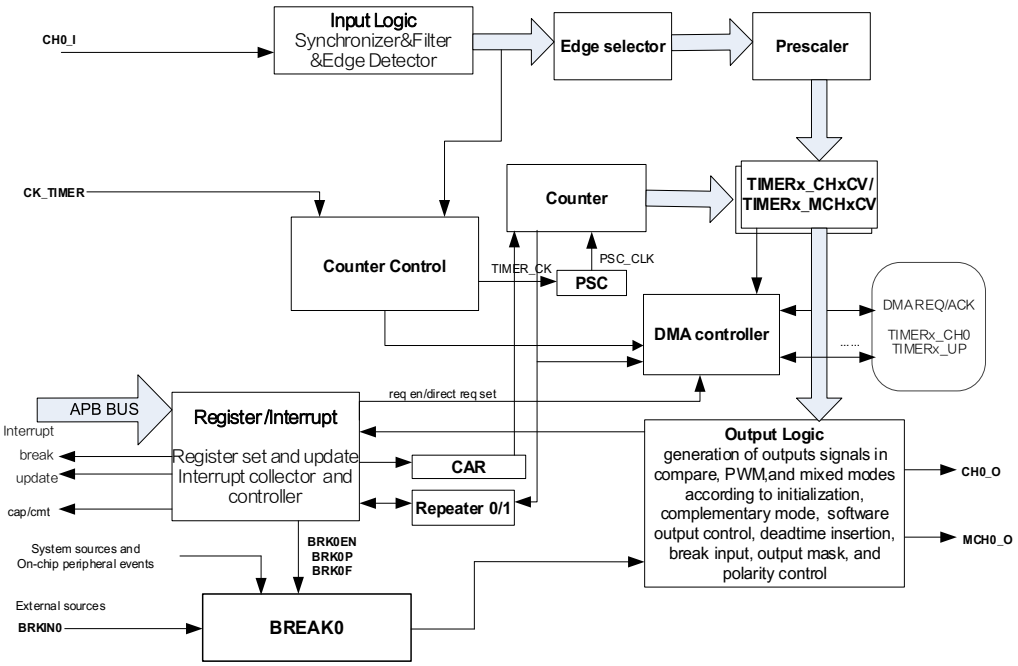
- Total channel num: 2.
- Counter width: 16-bit.
- Source of counter clock: internal clock.
- Counter modes: count up only.
- Programmable prescaler: 16-bit. The factor can be changed on the go.
- Each channel is user-configurable: input capture mode, output compare mode, programmable PWM mode, single 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.

15.4.3. Block diagram

Figure 15-115. General level4 timer block diagram provides details of the internal

configuration of the general level4 timer.

Figure 15-115. General level4 timer block diagram



15.4.4. Function overview

Clock selection

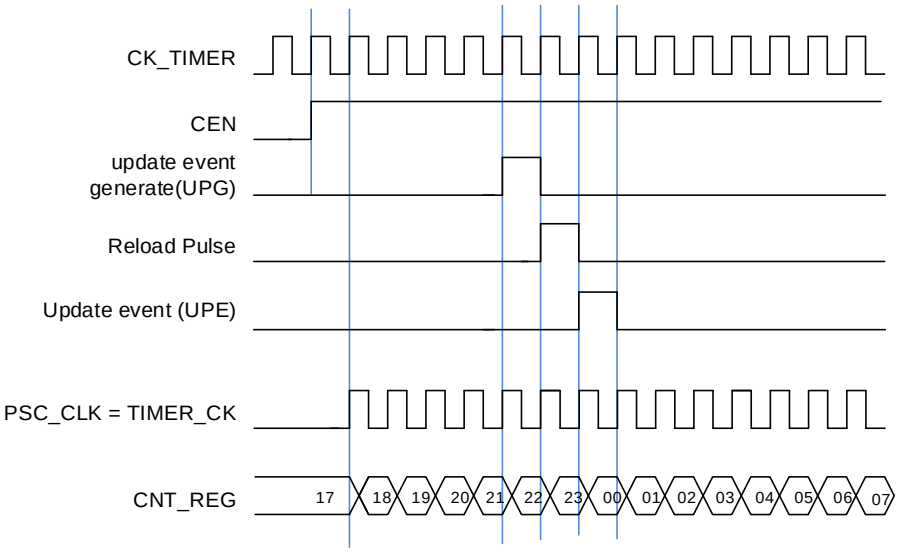
The general level4 TIMER can only being clocked by the CK_TIMER.

■ Internal timer clock CK_TIMER which is from module RCU

The general level4 TIMER has only one clock source which is the internal CK_TIMER, used to drive the counter prescaler. When the CEN is set, the CK_TIMER will be divided by PSC value to generate PSC_CLK.

The TIMER_CK, driven counter's prescaler to count, is equal to CK_TIMER which is from RCU

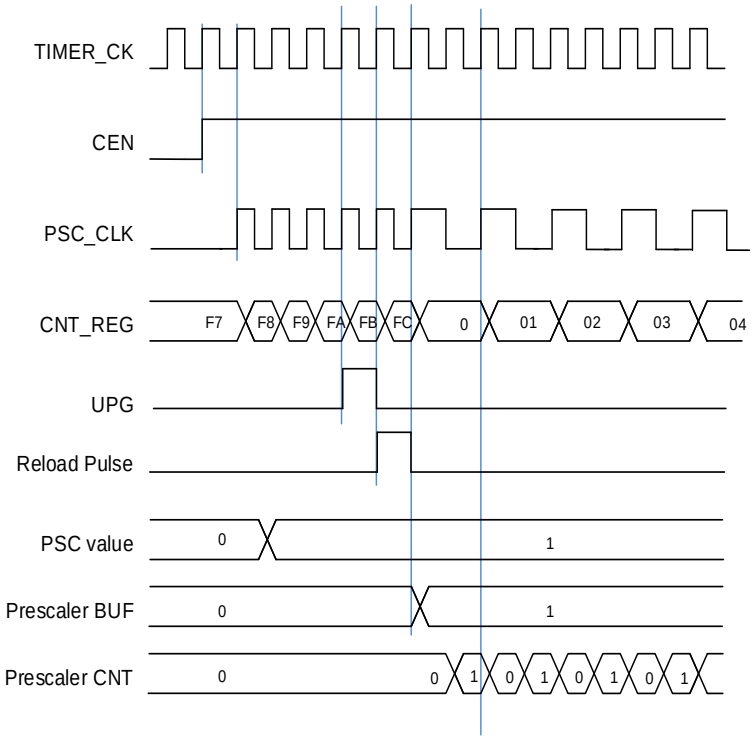
Figure 15-116. Normal mode, internal clock divided by 1



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 15-117. 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 15-118. Timing diagram of up counting mode, PSC=0/2 and *Figure 15-119. 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 15-118. Timing diagram of up counting mode, PSC=0/2

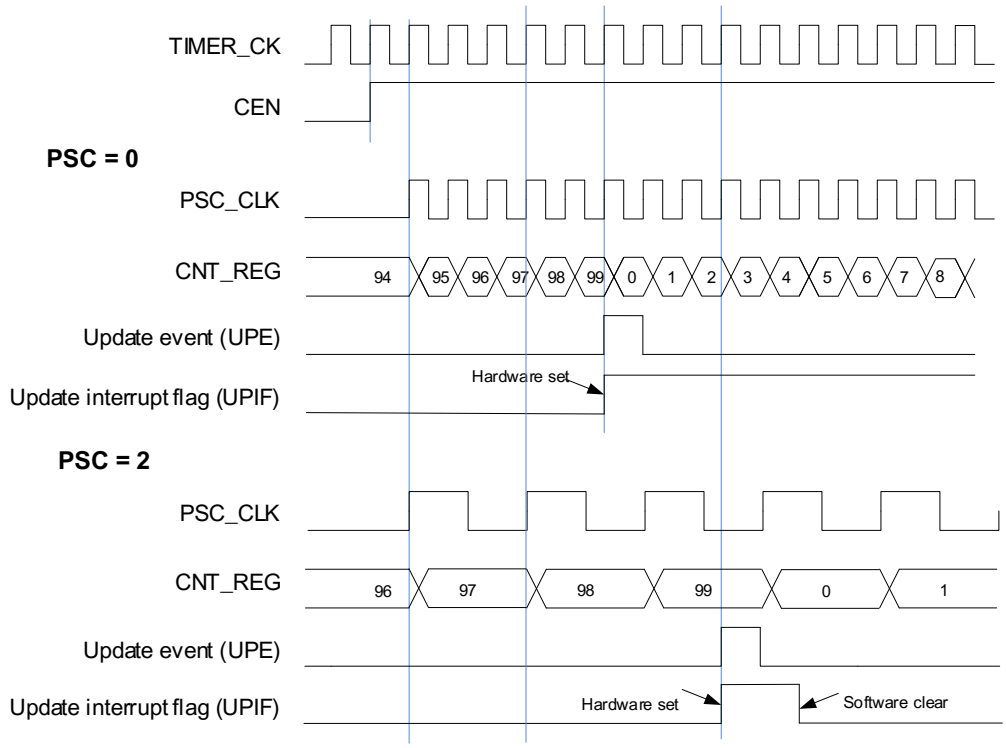
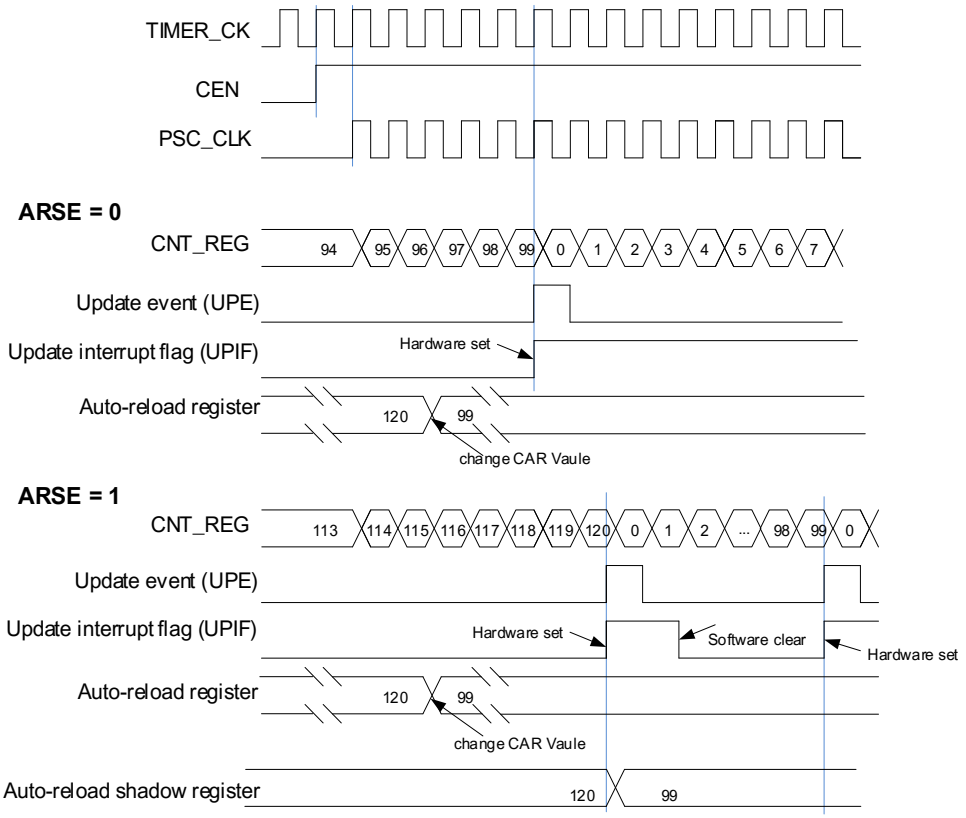


Figure 15-119. Timing diagram of up counting mode, change TIMERx_CAR on the go



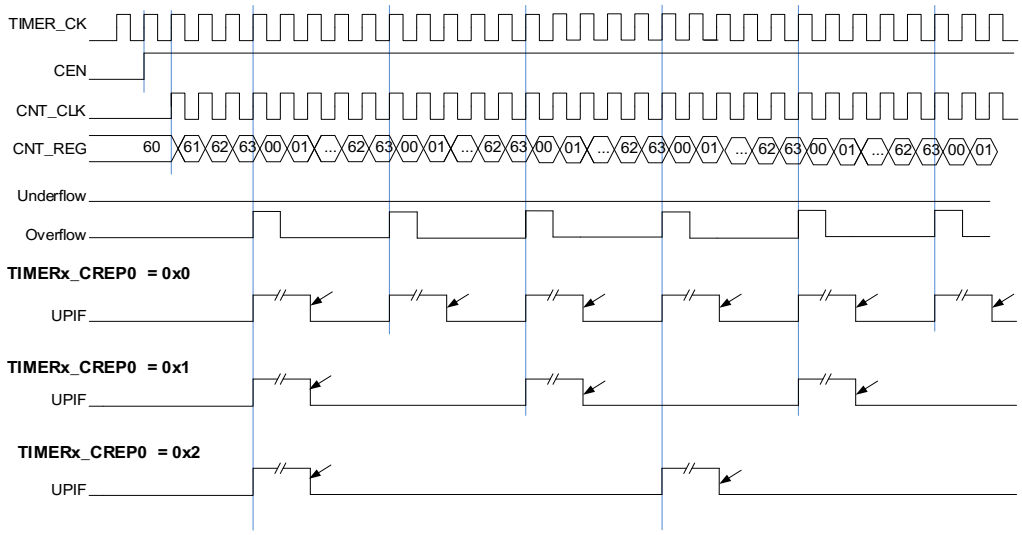
Counter repetition

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 15-120. 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 15-121. Input capture logic for channel 0

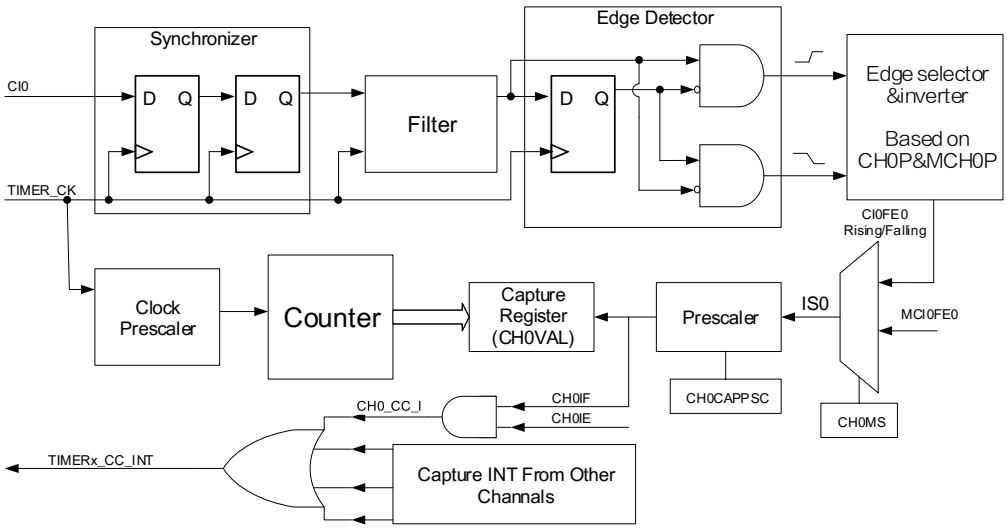
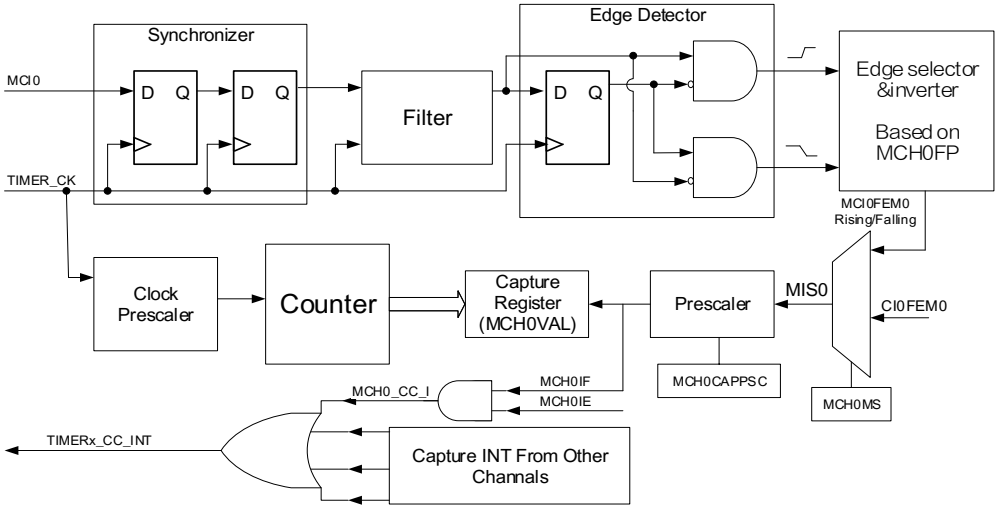


Figure 15-122. Input capture logic for multi mode channel 0



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 15-123. Output compare logic (when MCHxMSEL = 2'b00, x=0) and *Figure 15-124. Output compare logic (when MCHxMSEL = 2'b11, x=0)* show the logic circuit of output compare mode.

Figure 15-123. Output compare logic (when MCHxMSEL = 2'b00, x=0)

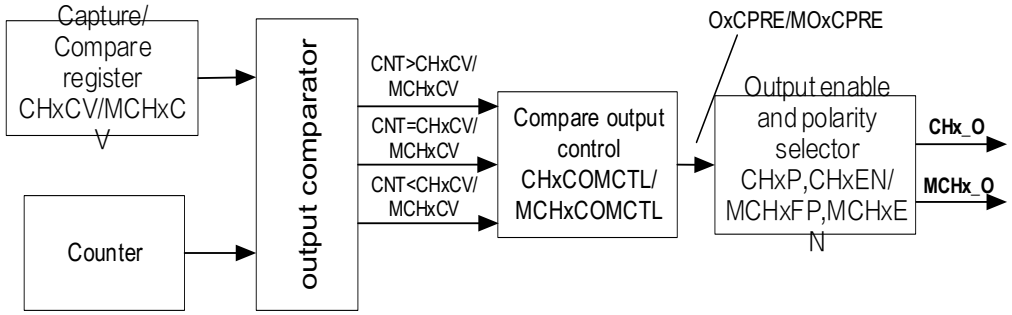
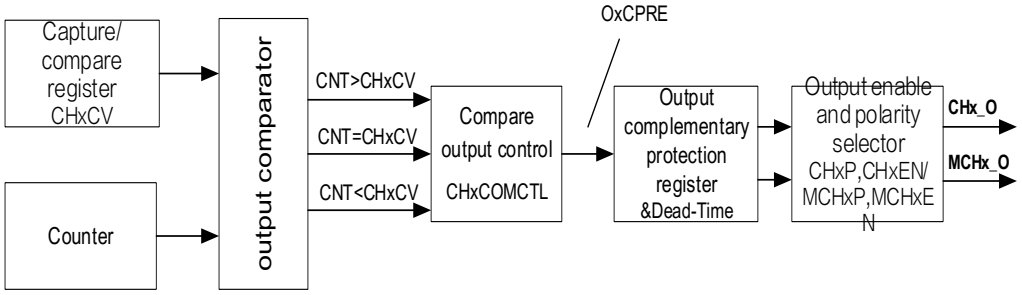


Figure 15-124. Output compare logic (when MCHxMSEL = 2'b11, x=0)



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 below(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 15-123. 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 15-124. 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_CCHP register. Please refer to ***Outputs complementary*** for more details.

In output compare mode, the TIMERx can generate timed pulses with programmable position, polarity, duration and frequency. When the counter matches the value in the TIMERx_CHxCV/ TIMERx_MCHxCV register of an output compare channel, the channel (n) output can be set, cleared, or toggled based on CHxCOMCTL/ MCHxCOMCTL. When the

counter reaches the value in the TIMERx_CHxCV/ TIMERx_MCHxCV register, the ChxIF/ MCHxIF bit will be set and the channel (n) interrupt is generated if ChxIE/ MCHxIE = 1. And the DMA request will be asserted, if CHxDEN/ MCHxDEN =1.

So, the process can be divided into several steps as below:

Step1: Clock Configuration. Such as clock source, clock prescaler and so on.

Step2: Compare mode configuration.

- Set the shadow enable mode by CHxCOMSEN/ MCHxCOMSEN.
- Set the output mode (set/clear/toggle) by CHxCOMCTL/ MCHxCOMCTL.
- Select the active polarity by CHxP/MCHxP/ MCHxFP.
- Enable the output by ChxEN/ MCHxEN.

Step3: Interrupt/DMA request enable configuration by ChxIE/ MCHxIE /CHxDEN/ MCHxDEN.

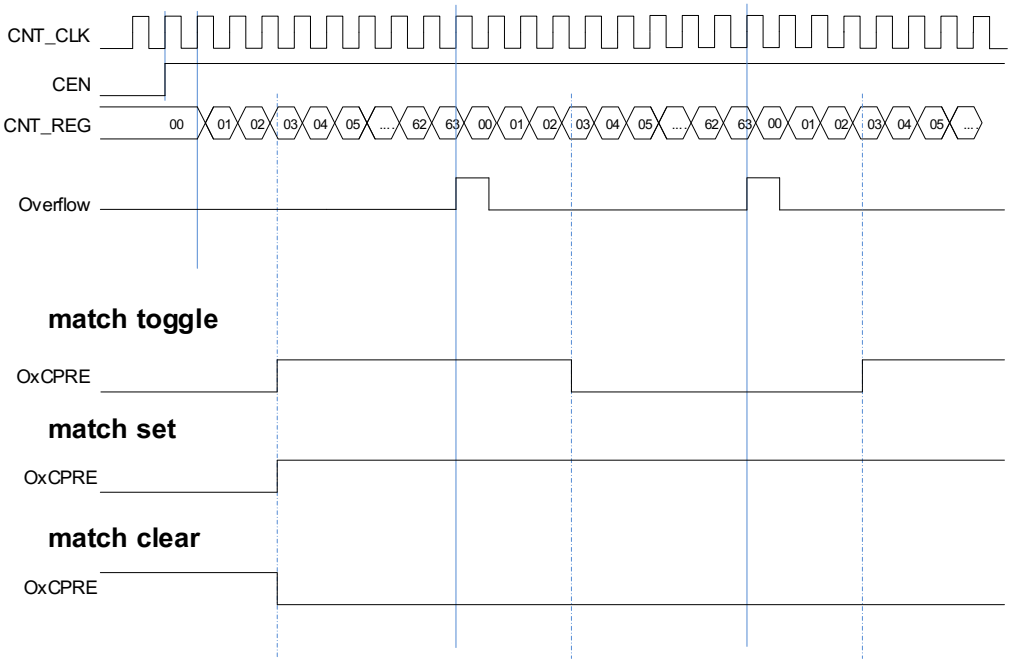
Step4: Compare output timing configuration by TIMERx_CAR and TIMERx_CHxCV/ TIMERx_MCHxCV.

The TIMERx_CHxCV/ TIMERx_MCHxCV can be changed onging to meet the expected waveform.

Step5: Start the counter by configuring CEN to 1.

Figure 15-125. Output-compare in three modes shows the three compare modes: toggle/set/clear. CARL=0x63, CHxVAL=0x3.

Figure 15-125. Output-compare in three modes



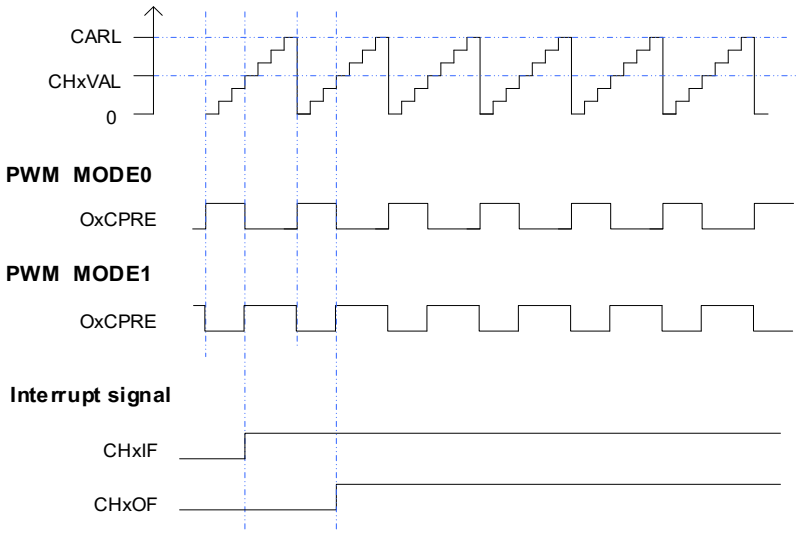
PWM mode

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 15-126. 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 15-126. PWM mode timechart



Channel output prepare signal

As is shown in ***Figure 15-123. Output compare logic (when MCHxMSEL = 2'b00, x=0)*** and ***Figure 15-124. 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.

Outputs complementary

The outputs of CHx_O and MCHx_O have two situations:

- MCHxMSEL=2'b00: The MCHx_O output is independent from the CHx_O output;
- MCHxMSEL=2'b11: The outputs of MCHx_O and CHx_O are complementary and the MCHxOMCTL bits are not used in the generation of the MCHx_O output.

Function of complementary is for a pair of channels, CHx_O and MCHx_O, the two output signals cannot be active at the same time. TIMERx's one pair of channel has this function. The complementary signals CHx_O and MCHx_O are controlled by a group of parameters: the ChxEN and MCHxEN bits in the TIMERx_CHCTL2 register, the POEN, ROS and IOS bits in the TIMERx_CCHP register, ISOx and ISOxN bits in the TIMERx_CTL1 register. The output polarity is determined by CHxP and MCHxP bits in the TIMERx_CHCTL2 register.

When the 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 15-18. Complementary outputs controlled by parameters (MCHxMSEL =2'b11)***.

Table 15-18. 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 / CHx_ON = LOW
CHx_O / CHx_ON output disable(1).</td></tr>
<tr><td>1</td><td colspan="2" rowspan="3">CHx_O/ CHx_ON output "off-state" (2):
the CHx_O/ CHx_ON output inactive level firstly: CHx_O = CHxP, CHx_ON = CHxNP; If the clock for deadtime generator is present, after a deadtime: CHx_O = ISOx, CHx_ON = ISOxN. (3)</td></tr>
<tr><td rowspan="2">1</td><td>0</td></tr>
<tr><td>1</td></tr>
<tr><td>1</td><td>x</td><td>x</td><td colspan="2">CHx_O/ CHx_ON output "off-state":
the CHx_O/ CHx_ON output inactive level firstly: CHx_O = CHxP, CHx_ON = CHxNP; If the clock for deadtime generator is present, after a deadtime: CHx_O = ISOx, CHx_ON = ISOxN.</td></tr>
<tr><td>1</td><td>0</td><td>0/1</td><td>0</td><td>0</td><td colspan="2">CHx_O/MCHx_O = LOW
CHx_O/MCHx_O output disable.</td></tr>
</table>

Complementary Parameters					Output Status	
POEN	ROS	IOS	ChxEN	MCHxEN	CHx_O	MCHx_O
				1	CHx_O = LOW CHx_O output disable.	MCHx_O=OxCPRE⊕(4)MCH xP 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⊕MCHx P 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 / CHx_ON are disconnected to corresponding pins, the pin is floating with GPIO pull up/down setting which will be Hi-Z if no pull.

(2) "off-state": CHx_O / CHx_ON output with inactive state (e.g., CHx_O = 0⊕CHxP = CHxP).

(3) See Break mode section for more details.

(4) ⊕: Xor calculate.

(5) (! OxCPRE): the complementary output of the OxCPRE signal.

Dead time insertion

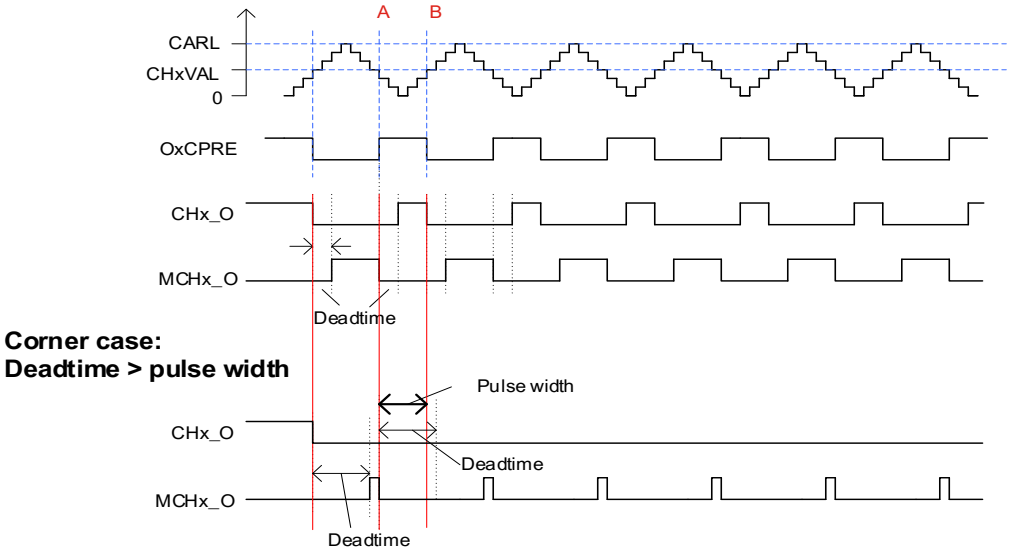
The dead time insertion is enabled when MCHxMSEL=2'b11 and both ChxEN and MCHxEN are configured to 1'b1, it is also necessary to configure POEN to 1. The field named DTCFG defines the dead time delay that can be used for all channels. Refer to the ***Complementary channel protection register (TIMERx_CCHP)*** for details about the delay time.

The dead time delay insertion ensures that two complementary signals are not active at the same time.

When the channel x match event (TIMERx_CNT = CHxVAL) occurs, OxCPRE will be toggled in PWM mode 0. At point A in ***Figure 15-127. 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 894pprox. 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 15-127. Complementary output with dead-time insertion***).

Figure 15-127. Complementary output with dead-time insertion



Break function

The MCHx_O output is the inverse of the CHx_O output when the MCHxMSEL=2'b11 (and the MCHxOMCTL bits are not used in the generation of the MCHx_O output). In this case, CHx_O and MCHx_O signals cannot be set to active level at the same time.

The general level3 timers have the BREAK0 function. The BREAK0 function can be enabled

by setting the BRK0EN bit in the TIMERx_CCHP register. The break input polarity is configured by the BRK0P bit in TIMERx_CCHP register, the input is active on level.

In BREAK0 function, CHx_O and MCHx_O are controlled by the POEN, OAEN, IOS and ROS bits in the TIMERx_CCHP register, ISOx and ISOxN bits in the TIMERx_CTL1 register.

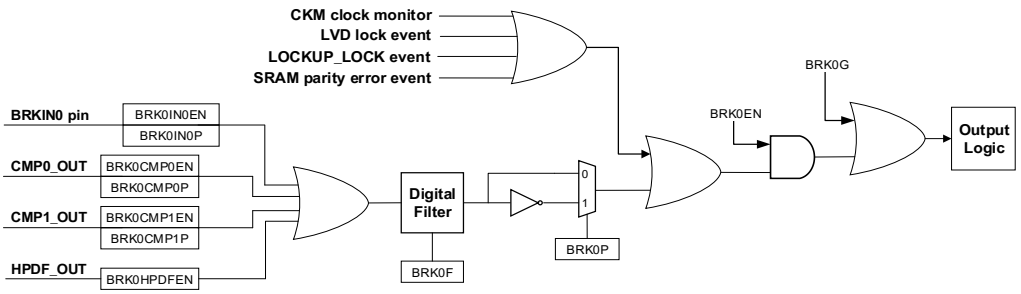
The break event is the result of logic Ored of all sources. The BREAK0 function can handle three types of event sources:

- External sources: coming from BRKIN0 input;
- System sources: HXTAL stuck event which is generated by Clock Monitor CKM in RCU, LVD lock event, Cortex®-M7 LOCKUP_LOCK event or SRAM parity error event;
- 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 15-128. BREAK0 function logic diagram***, BRKIN0 can select GPIO pins from the TRIGSEL module, which can select by TRICSEL_TIMERxBTKIN registers.

Figure 15-128. BREAK0 function logic diagram

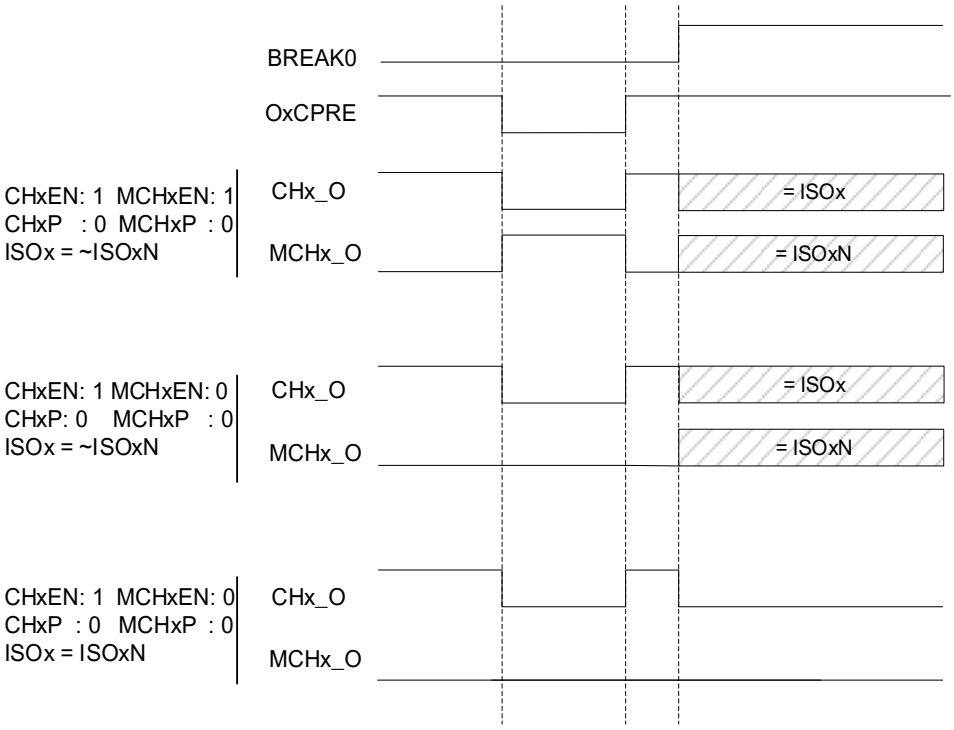


When a BREAK0 event occurs, the outputs are force at an inactive level, or at a predefined level (either active or inactive) after a deadtime duration.

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 15-129. 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 15-129. 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_CCHP 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_CCHP 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:

- 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.

- POEN=1: when the channel outputs are enabled, the BRKIN0 pin cannot be released even if the BRK0REL is set.

Table 15-19. Break function input pins locked/ released conditions

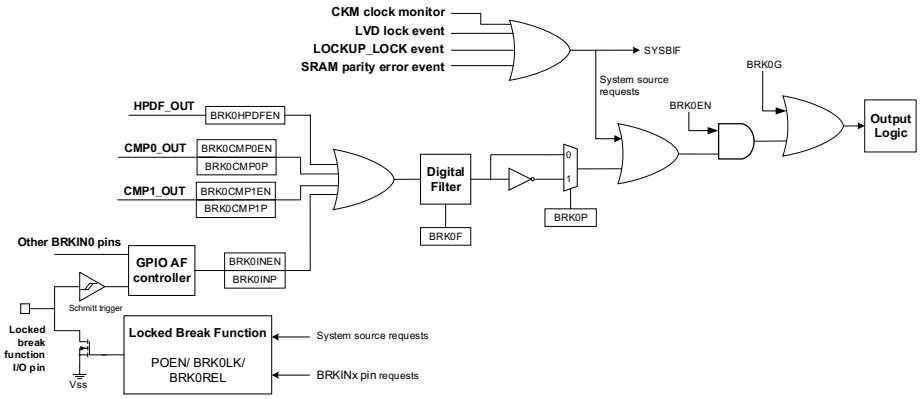
POEN	BRK0LK	BRK0REL	Break input pin state
0	1	0	Locked
	1	1	Released

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:

- Set the BRK0REL bit to 1 and released the BRKIN0 pin;
- The software waits for the system break sources inactive, and then clears the SYSBIF flag;
- 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 15-130. BRKIN0 pin logic with BREAK0 function



Single pulse mode

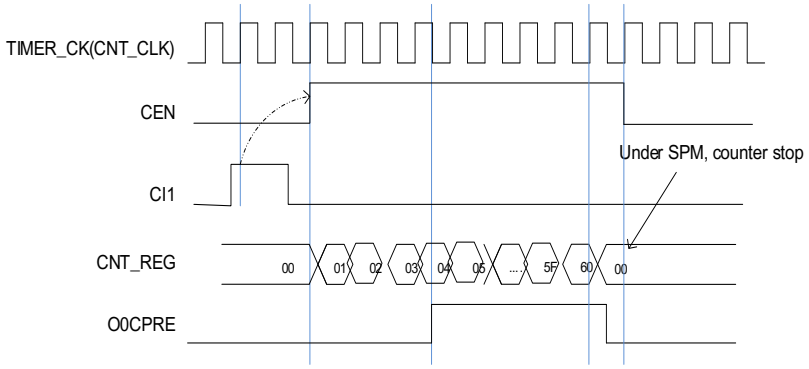
Single pulse mode is opposite to the repetitive mode, which can be enabled by setting SPM in TIMERx_CTL0. If SPM is set, the counter will be cleared and stopped automatically when the next update event occurs. In order to get a pulse waveform, the TIMERx is configured to PWM mode or compare mode by CHxCOMCTL 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 15-131. 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®-M7 halted, and the TIMERx_HOLD configuration bit in DBG register set to 1, the TIMERx counter stops.

15.4.5. 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).

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				UPIFBUEN	Reserved	CKDIV[1:0]		ARSE	Reserved			SPM	UPS	UPDIS	CEN
				rw		rw		rw				rw	rw	rw	rw

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

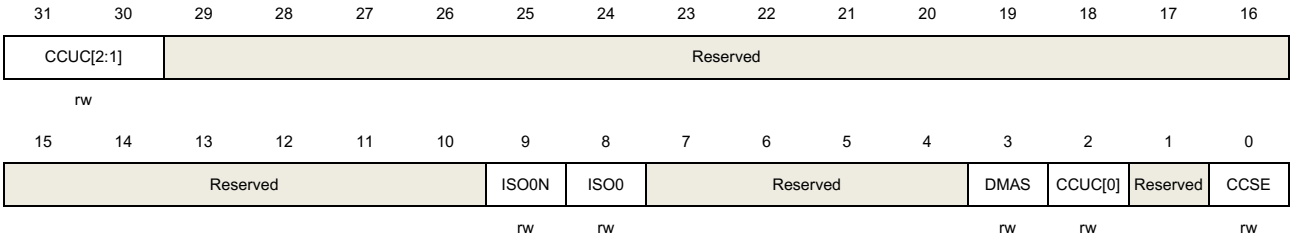
		counter stops at next update event.
2	UPS	Update source This bit is used to select the update event sources by software. 0: Any of the following events generates an update interrupt or a DMA request: – The UPG bit is set. – The counter generates an overflow event. – The slave mode controller generates 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 UPG bit is set or the slave mode controller generates a hardware reset event.
0	CEN	Counter enable 0: Counter disable 1: Counter enable The CEN bit must be set by software when timer works in external clock mode, pause mode or decoder mode. While in event mode, the hardware can set the CEN bit automatically.

Control register 1 (TIMERx_CTL1)

Address offset: 0x04

Reset value: 0x0000 0000

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



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_CCHP register is 00.
8	ISO0	Idle state of channel 0 output 0: When POEN bit is reset, CH0_O is set low. 1: When POEN bit is reset, CH0_O is set high. The CH0_O output changes after a dead time if MCH0_O is implemented. This bit can be modified only when PROT[1:0] bits in TIMERx_CCHP register is 00.
7:4	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 MCH0SEL[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 MCH0SEL[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
15:7	Reserved	Must be kept at reset value.
6:4	CH0COMCTL[2:0]	Channel 0 compare output control The CH0COMCTL[3] and CH0COMCTL[2:0] bit-field control the behavior of O0CPRE which drives CH0_O. 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_CCHP register is 11 and CH0MS bit-field is 000 (compare mode).
3	CH0COMSEN	Channel 0 compare output shadow enable When this bit is set, the shadow register of TIMERx_CH0CV register which updates at each update event will be enabled. 0: Channel 0 output compare shadow disabled 1: Channel 0 output compare shadow enabled The PWM mode can be used without validating the shadow register only in single pulse mode (SPM bit in TIMERx_CTL0 register is set). This bit cannot be modified when PROT[1:0] bit-field in TIMERx_CCHP register is 11 and CH0MS bit-field is 000.
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:

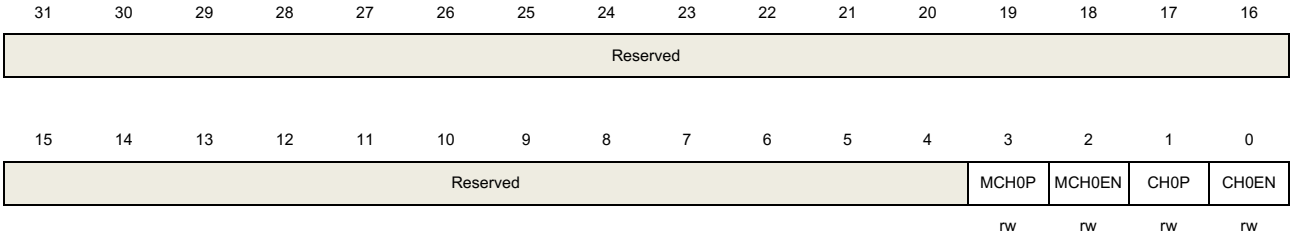
Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30	CH0MS[2]	Channel 0 I/O mode selection Refer to CH0MS[1:0] description.
29:8	Reserved	Must be kept at reset value.
7:4	CH0CAPFLT[3:0]	Channel 0 input capture filter control An event counter is used in the digital filter, in which a transition 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 word(32-bit).



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_CCHP register is 11 or 10.
2	MCH0EN	Multi mode channel 0 capture/compare enable When multi mode channel 0 is configured in output mode, setting this bit enables MCH0_O signal in active state. When multi mode channel 0 is configured in input mode, setting this bit enables the capture event in multi mode channel 0. 0: Multi mode channel 0 disabled 1: Multi mode channel 0 enabled
1	CH0P	Channel 0 capture/compare polarity When channel 0 is configured in output mode, this bit specifies the output signal polarity. 0: Channel 0 active high 1: Channel 0 active low When channel 0 is configured in input mode, these bits 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_CCHP register is 11 or 10.
0	CH0EN	Channel 0 capture/compare enable When channel 0 is configured in output mode, setting this bit enables CH0_O signal in active state. When channel 0 is configured in input mode, setting this bit enables the capture event in channel 0. 0: Channel 0 disabled 1: Channel 0 enabled

Counter register (TIMERx_CNT)

Address offset: 0x24
Reset value: 0x0000 0000

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

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

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]															

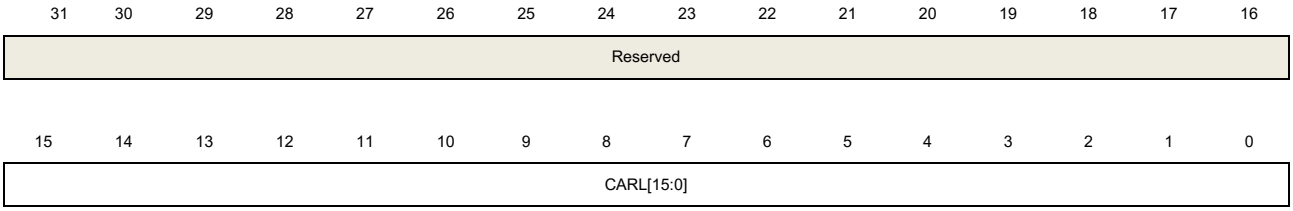
rw

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).



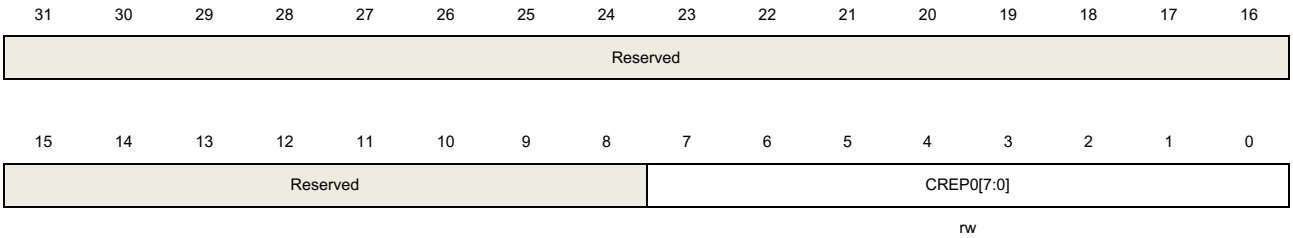
rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CARL[15:0]	Counter auto reload value This bit-field specifies the auto reload value of the counter.

Counter repetition register 0 (TIMERx_CREP0)

Address offset: 0x30
Reset value: 0x0000 0000

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



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
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH0VAL[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH0VAL[15:0]	Capture/compare value of channel 0 When channel 0 is configured in input mode, this bit-field indicates the counter value at the last capture event. And this bit-field is read-only. When channel 0 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event.

Complementary channel protection register (TIMERx_CCHP)

Address offset: 0x44
Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			BRK0LK	Reserved	BRK0REL	Reserved						BRK0F[3:0]			
	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

31:29	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_CCHP 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_CCHP 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_CCHP 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_CCHP 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_CCHP register is 00.
11	ROS	Run mode "off-state" enable When POEN bit is set (Run mode), this bit can be set to enable the "off-state" for the channels which has been configured in output mode. Please refer to ***Table 15-15. 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-field in TIMERx_CCHP register is 10 or 11.
10	IOS	Idle mode "off-state" enable When POEN bit is reset (Idle mode), this bit can be set to enable the "off-state" for

		the channels which has been configured in output mode. Please refer to ***Table 15-15. 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-field in TIMERx_CCHP register is 10 or 11.
9:8	PROT[1:0]	Complementary register protect control This bit-field specifies the write protection property of registers. 00: Protect disabled. No write protection. 01: PROT mode 0. The ISOx/ISOxN bits in TIMERx_CTL1 register, the BRK0EN/ BRK0P/OAEN/DTCFG bits in TIMERx_CCHP register are writing protected. 10: PROT mode 1. In addition to the registers in PROT mode 0, the CHxP/MCHxP bits in TIMERx_CHCTL2 register (if related channel is configured in output mode), the ROS/IOS bits in TIMERx_CCHP register 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_CCHP 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] x t_{DT}, t_{DT} = t_{DTS}. DTCFG[7:5] = 3'b10x: DT value = (64+DTCFG[5:0]) x t_{DT}, t_{DT} = t_{DTS}*2. DTCFG[7:5] = 3'b110: DT value = (32+DTCFG[4:0]) x t_{DT}, t_{DT} = t_{DTS}*8. DTCFG[7:5] = 3'b111: DT value = (32+DTCFG[4:0]) x t_{DT}, t_{DT} = t_{DTS}*16. This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP register is 00.

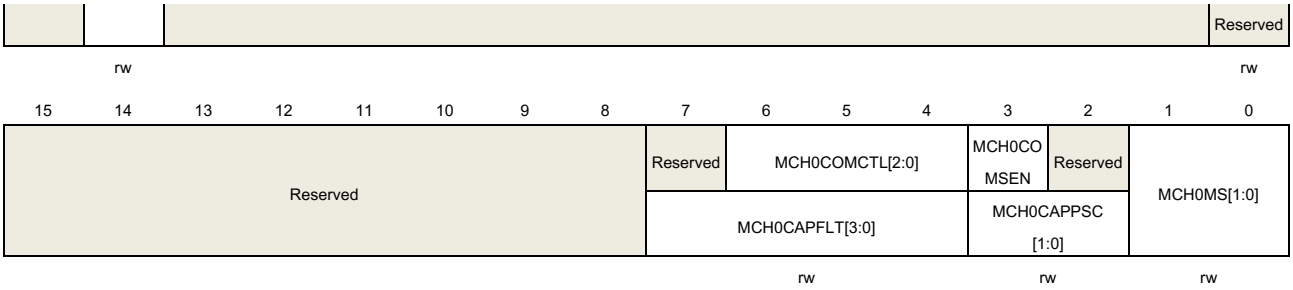
Multi mode channel control register 0 (TIMERx_MCHCTL0)

Address offset: 0x48

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	MCH0MS[2]	Reserved													MCH0COMCTL[3]



Output compare mode:

Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30	MCH0MS[2]	Multi mode channel 0 I/O mode selection Refer to MCH0MS[1:0] description.
29:17	Reserved	Must be kept at reset value.
16	MCH0COMCTL[3]	Multi mode channel 0 compare output control. Refer to MCH0COMCTL[2:0] description.
15:7	Reserved	Must be kept at reset value.
6:4	MCH0COMCTL[2:0]	Multi mode channel 0 output compare control When multi mode channel 0 is configured in output mode, and the MCH0MSEL[1:0] = 2'b00, the MCH0COMCTL[3] and MCH0COMCTL[2:0] bit-field control the behavior of MO0CPRE which drives MCH0_O. The active level of MO0CPRE is high, while the active level of MCH0_O depends on MCH0FP[1:0] bits. **Note**: When multi mode channel 0 is configured in output mode, and the MCH0MSEL[1:0] = 2'b11, the CH0COMCTL[2:0] bit-field controls the behavior of O0CPRE which drives CH0_O and MCH0_O, while the active level of CH0_O and MCH0_O depends on CH0P and MCH0P bits. 0000: Timing mode. The MO0CPRE signal keeps stable, independent of the comparison between the register TIMERx_MCH0CV and the counter TIMERx_CNT. 0001: Set the channel output on match. MO0CPRE signal is forced high when the counter matches the output compare register TIMERx_MCH0CV. 0010: Clear the channel output on match. MO0CPRE signal is forced low when the counter matches the output compare register TIMERx_MCH0CV. 0011: Toggle on match. MO0CPRE toggles when the counter matches the output compare register TIMERx_MCH0CV. 0100: Force low. MO0CPRE is forced low level. 0101: Force high. MO0CPRE is forced high level. 0110: PWM mode 0. When counting up, MO0CPRE is active as long as the counter is smaller than TIMERx_MCH0CV, otherwise it is inactive. When counting down, MO0CPRE is inactive as long as the counter is larger than TIMERx_MCH0CV, otherwise it is active.

		0111: PWM mode 1. When counting up, MO0CPRE is inactive as long as the counter is smaller than TIMERx_MCH0CV, otherwise it is active. When counting down, MO0CPRE is active as long as the counter is larger than TIMERx_MCH0CV, otherwise it is inactive. 1000~1111: Reserved. If configured in PWM mode, the MO0CPRE level changes only when the output compare mode switches from "Timing" mode to "PWM" mode or the result of the comparison changes. When the outputs of CH0 and MCH0 are complementary, this bit-field is preloaded. If CCSE =1, this bit-field will only be updated when a channel commutation event is generated. This bit cannot be modified when PROT[1:0] bit-field in TIMERx_CCHP register is 11 and MCH0MS bit-field is 00(compare mode).
3	MCH0COMSEN	Multi mode channel 0 output compare shadow enable When this bit is set, the shadow register of TIMERx_MCH0CV register which updates at each update event will be enabled. 0: Multi mode channel 0 output compare shadow disabled 1: Multi mode channel 0 output compare shadow enabled The PWM mode can be used without validating the shadow register only in single pulse mode (SPM bit in TIMERx_CTL0 register is set). This bit cannot be modified when PROT[1:0] bit-field in TIMERx_CCHP register is 11 and MCH0MS bit-field is 00.
2	Reserved	Must be kept at reset value.
1:0	MCH0MS[1:0]	Multi mode channel 0 I/O mode selection This bit-field specifies the work mode of the channel and the input signal selection. This bit-field is writable only when the channel is not active (MCH0EN bit in TIMERx_CHCTL2 register is reset). 000: Multi mode channel 0 is 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 word(32-bit).

31	30	29	28	27	26	25	24	23	22	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	

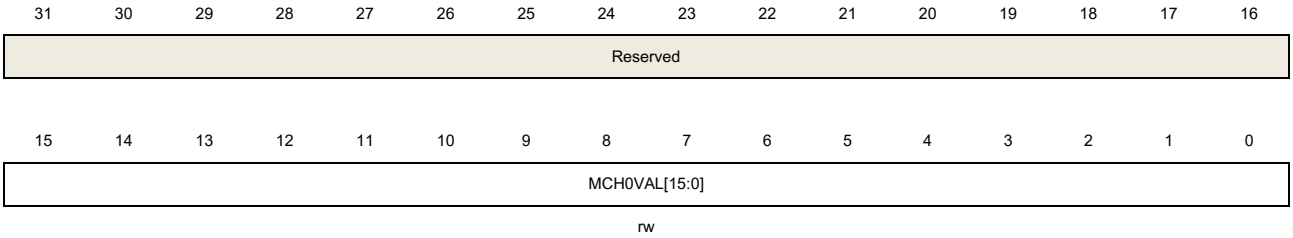
Bits	Fields	Descriptions

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_CCHP register is 11 or 10.

Multi mode channel 0 capture/compare value register (TIMERx_MCH0CV)

Address offset: 0x54
Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	MCH0VAL[15:0]	Capture/compare value of multi mode channel 0. When multi mode channel 0 is configured in input mode, this bit-field indicates the counter value at the last capture event. And this bit-field is read-only. When multi mode channel 0 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is

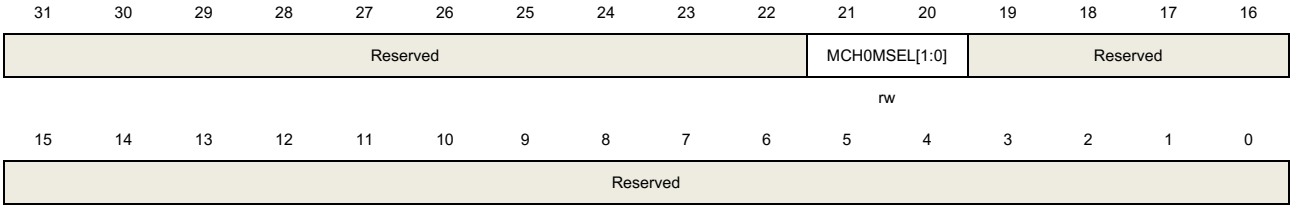
enabled, the shadow register updates by every update event.

Control register 2 (TIMERx_CTL2)

Address offset: 0x74

Reset value: 0x0030 0000

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



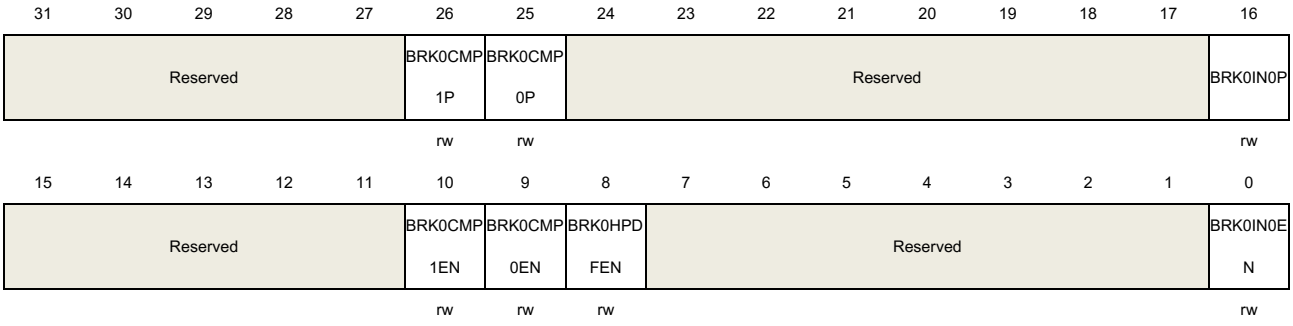
Bits	Fields	Descriptions
31:22	Reserved	Must be kept at reset value.
21:20	MCH0MSEL[1:0]	Multi mode channel 0 mode select 00: Independent mode, MCH0 is independent of CH0 01: Reserved 10: Reserved 11: Complementary mode, only the CH0 is valid for input, and the outputs of MCH0 and CH0 are complementary
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).



Bits	Fields	Descriptions
31:27	Reserved	Must be kept at reset value.

26	BRK0CMP1P	BREAK0 CMP1 input polarity This bit is used to configure the CMP1 input polarity, and the specific polarity is determined by this bit and the BRK0P bit. 0: CMP1 input signal will not be inverted (BRK0P =0, the input signal is active low; BRK0P =1, the input signal is active high) 1: CMP1 input signal will be inverted (BRK0P=0, the input signal is active high; BRK0P =1, the input signal is active low) This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP 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_CCHP 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_CCHP register is 00.
15:11	Reserved	Must be kept at reset value.
10	BRK0CMP1EN	BREAK0 CMP1 enable 0: CMP1 input disabled 1: CMP1 input enabled This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP 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_CCHP register is 00.
8	BRK0HPDFEN	BREAK0 HPDF input(HPDF_BREAK[x], please refer to *Table 33-2. HPDF internal*

		signal enable 0: HPDF input disabled 1: HPDF input enabled This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP register is 00.
7:1	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_CCHP register is 00.

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).

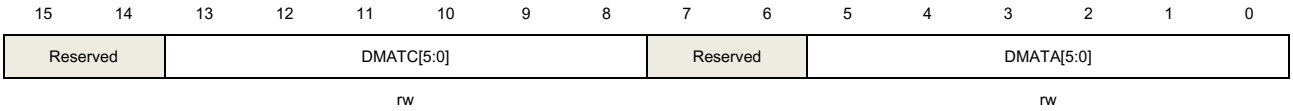
DMA configuration register (TIMERx_DMACFG)

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															



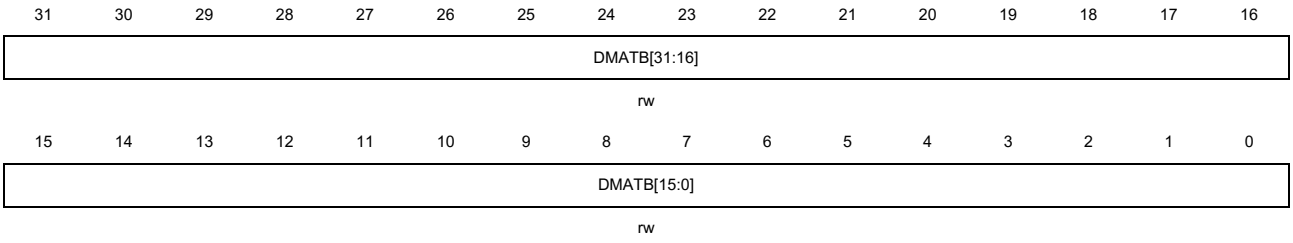
Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value.
13:8	DMATC[5:0]	DMA transfer count This field defines the times of accessing(R/W) the TIMERx_DMATB register by DMA. 6'b000000: transfer 1 time 6'b000001: transfer 2 times … 6'b100101: transfer 38 times
7:6	Reserved	Must be kept at reset 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).



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 100,101,110. 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.

15.5. Basic timer (TIMERx, x=5,6,50,51)

15.5.1. Overview

The basic timer module(TIMER5/6/50/51) has a 32-bit or 64-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.

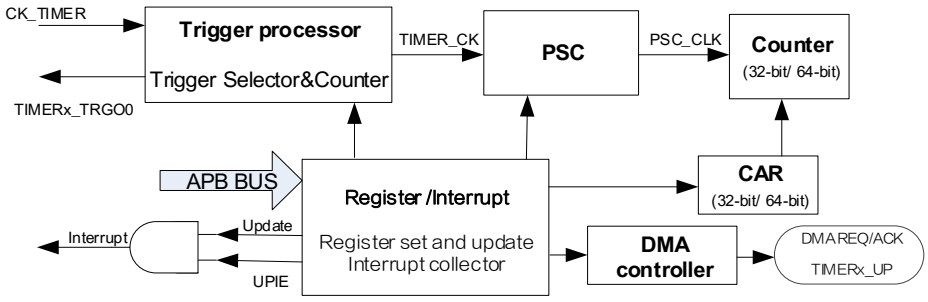
15.5.2. Characteristics

- Counter width: 32 bits (TIMER5/6), 64 bits (TIMER50/51).
- 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.

15.5.3. Block diagram

Figure 15-132. Basic timer block diagram provides details on the internal configuration of the basic timer.

Figure 15-132. Basic timer block diagram



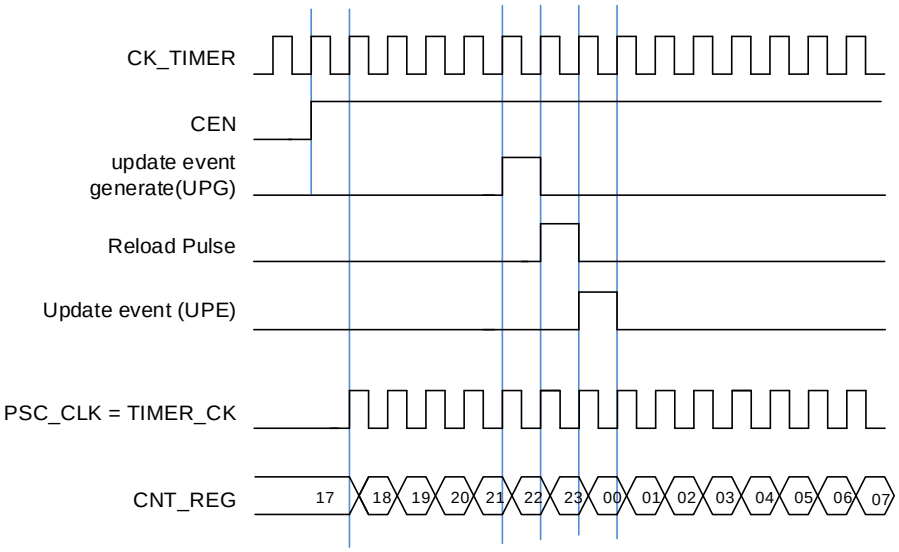
15.5.4. Function overview

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.

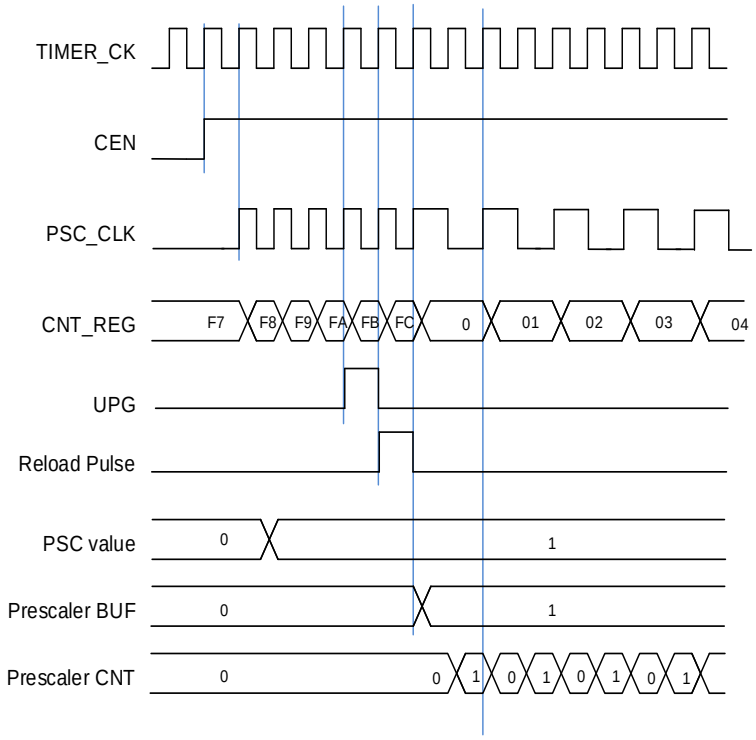
Figure 15-133. Normal mode, internal clock divided by 1



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 15-134. 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/ TIMERx_CARL/ TIMERx_CARH 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 (TIMERx, x=5,6).

Figure 15-135. Timing chart of up counting mode, PSC=0/2 (TIMERx, x=5,6)

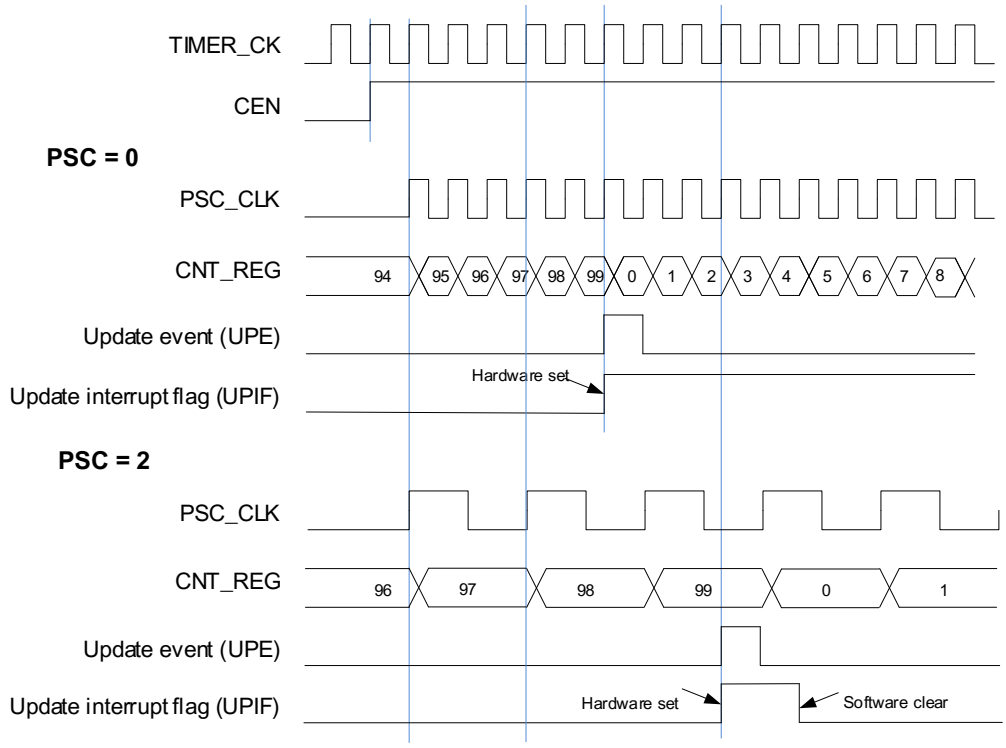
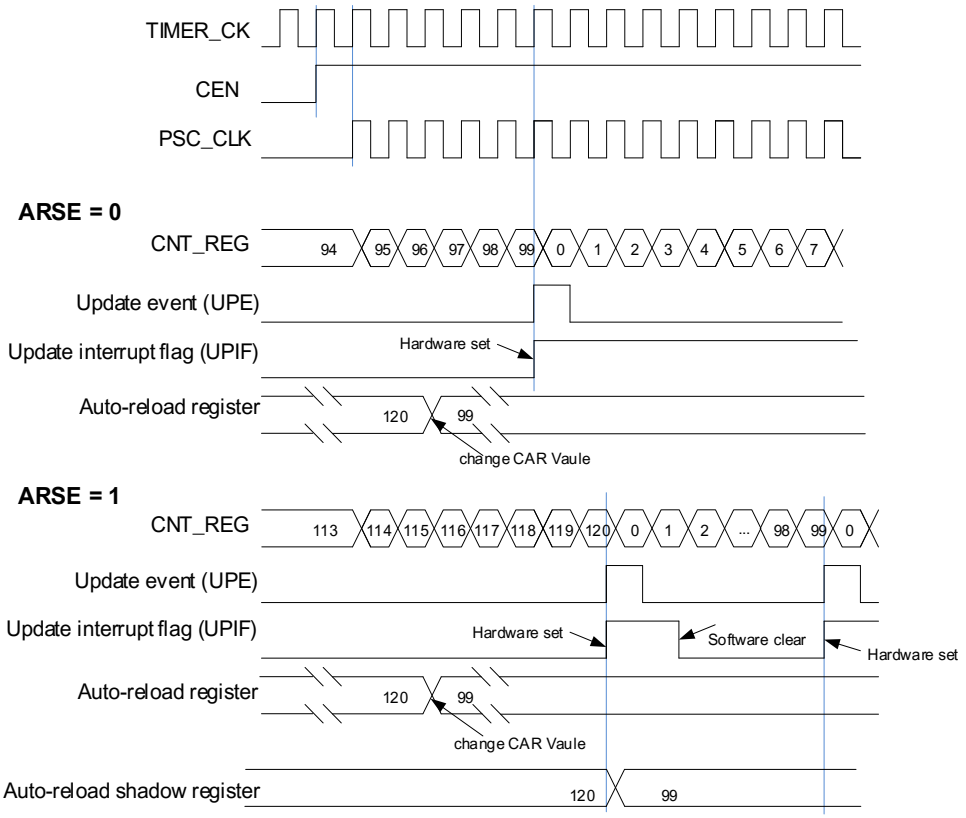


Figure 15-136. Timing chart of up counting mode, change TIMERx_CAR ongoing

(TIMERx, x=5,6)



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®-M7 is halted, and the TIMERx_HOLD configuration bit in DBG register set to 1, the TIMERx counter stops.

15.5.5. Registers definition (TIMERx, x=5,6,50,51)

TIMER5 base address: 0x4000 1000

TIMER6 base address: 0x4000 1400

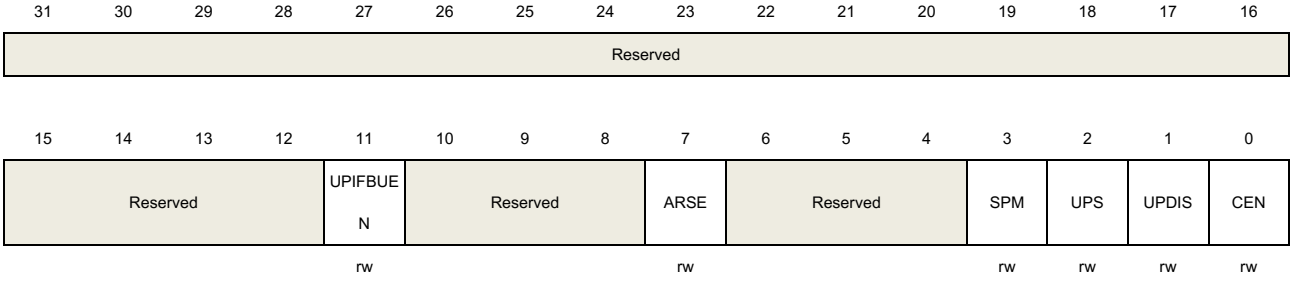
TIMER50 base address: 0x4000 F000

TIMER51 base address: 0x4000 F400

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
31:12	Reserved	Must be kept at reset value.
11	UPIFBUEN	UPIF bit backup enable 0: Backup disable. UPIF bit is not backuped to UPIFBU bit in TIMERx_CNT/ TIMERx_CNTH register. 1: Backup enabled. UPIF bit is backuped to UPIFBU bit in TIMERx_CNT/ TIMERx_CNTH register.
10:8	Reserved	Must be kept at reset value.
7	ARSE	Auto-reload shadow enable 0: The shadow register for TIMERx_CAR/ TIMERx_CARL/ TIMERx_CARH register is disabled 1: The shadow register for TIMERx_CAR/ TIMERx_CARL/ TIMERx_CARH 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

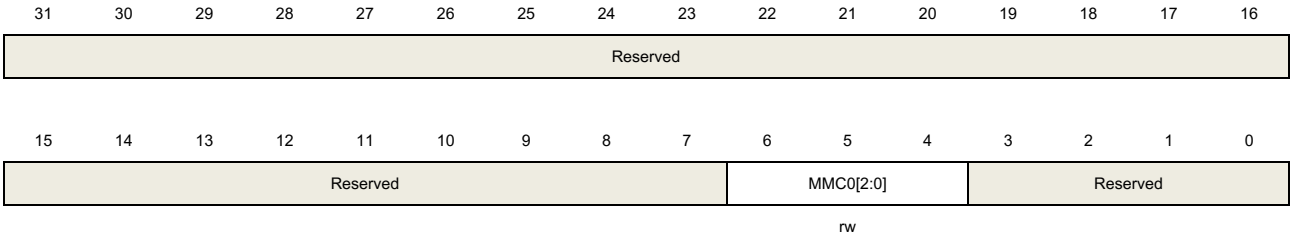
		This bit is used to select the update event sources by software. 0: When enabled, any of the following events generates an update interrupt or a DMA request: – The UPG bit is set – The counter generates an overflow event – The slave mode controller generates an update event. 1: When enabled, only counter 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 UPG bit is set or the slave mode controller generates a hardware reset event.
0	CEN	Counter enable 0: Counter disable 1: Counter enable The CEN bit must be set by software when timer works in external clock mode, pause mode or decoder mode. While in event mode, the hardware can set the CEN bit automatically.

Control register 1 (TIMERx_CTL1)

Address offset: 0x04

Reset value: 0x0000 0000

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



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) (TIMERx, x=5,6)

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.

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.

Counter low register (TIMERx_CNTL) (TIMERx, x=50,51)

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

Bits	Fields	Descriptions
31:0	CNTL[31:0]	This bit-field indicates the current counter low value. Writing to this bit-field can change the value of the counter.

Prescaler register (TIMERx_PSC)

Address offset: 0x28
Reset value: 0x0000 0000

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

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

Reserved

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

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	PSC[15:0]	Prescaler value of the counter clock The PSC clock is divided by (PSC+1) to generate the counter clock. The value of this bit-field will be loaded to the corresponding shadow register at every update event.

Counter auto reload register (TIMERx_CAR) (TIMERx, x=5,6)

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 This bit-field specifies the auto reload value of the counter.

Counter auto reload low register (TIMERx_CARL) (TIMERx, x=50,51)

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
CARLL[31:16]															
rw															

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

Bits	Fields	Descriptions
31:0	CARLL[31:0]	Counter auto reload low value This bit-field specifies the auto reload low value of the counter.

Counter high register (TIMERx_CNTH) (TIMERx, x=50,51)

Address offset: 0xD0

Reset value: 0x0000 0000

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

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

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

UPIFBUEN = 0:

Bits	Fields	Descriptions
31:0	CNTH[63:32]	This bit-field indicates the current counter high value. Writing to this bit-field can change the value of the counter.

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.
30:0	CNTH[62:32]	This bit-field indicates the current counter high value. Writing to this bit-field can change the value of the counter.

Counter auto reload high register 1(TIMERx_CARH)(TIMERx, x=50,51)

Address offset: 0xD4

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
CARLH[63:48]															
rw															

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

Bits	Fields	Descriptions
31:0	CARLH[63:32]	Counter auto reload high value This bit-field specifies the auto reload high value of the counter.

16. Real time clock (RTC)

16.1. 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.

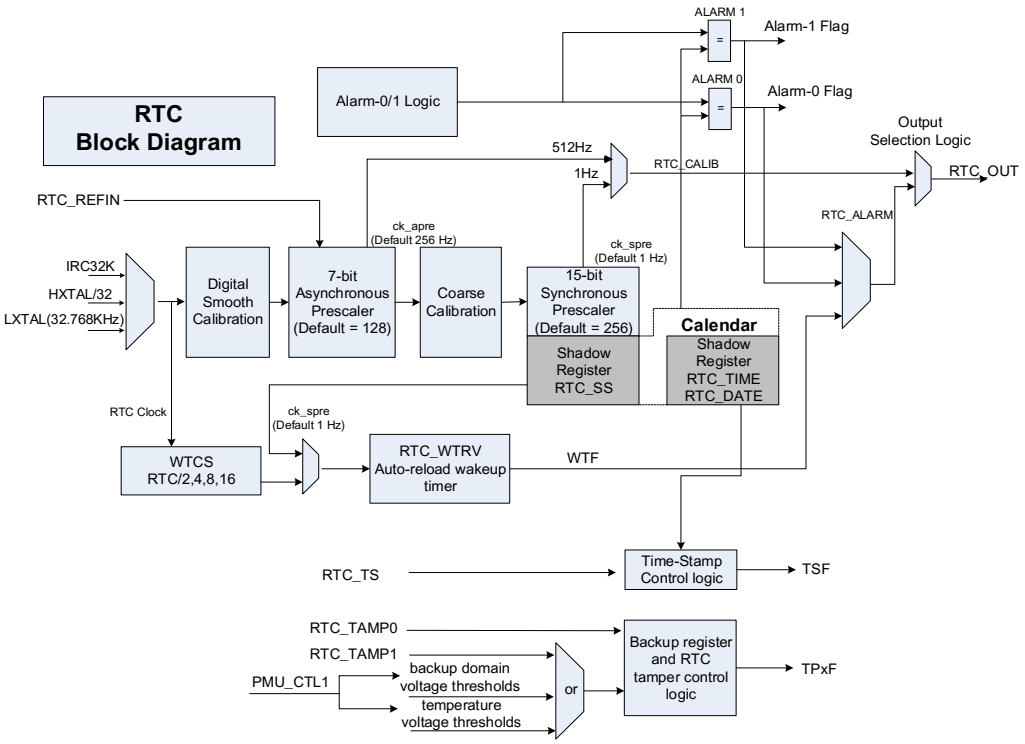
16.2. Characteristics

- 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
- Two tamper sources can be chosen and tamper type is configurable (RTC_TAMP0 and RTC_TAMP1)
- Programmable calendar and two field maskable alarms
- Maskable interrupt source:
  - Alarm 0 and Alarm 1
  - Time-stamp detection
  - Tamper detection
  - Auto wakeup event
- 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

16.3. Function overview

16.3.1. Block diagram

Figure 16-1. 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 V_{BAT}, 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)
  - tamper 1 event detection: RTC_TAMP1(PC1)
  - reference clock input: RTC_REFIN(PB15 or PB13)
  - tamper 1 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 1 event detection.

16.3.2. Clock source and prescalers

RTC unit has three independent clock sources: LXTAL, IRC32K and HXTAL with divided by RTCDIV[5:0] (configured in RCU_CFG0 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:

$$f_{ck_apre} = \frac{f_{rtcclk}}{FACTOR_A + 1} \quad (16\text{-}1)$$

$$f_{ck_spre} = \frac{f_{ck_apre}}{FACTOR_S + 1} = \frac{f_{rtcclk}}{(FACTOR_A + 1)*(FACTOR_S + 1)} \quad (16\text{-}2)$$

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.

Note: When the RTC uses the divided clock of HXTAL as the working clock source, a system reset will cause the HXTAL clock enable to reset, resulting in the RTC being unable to function properly. The clock needs to be reconfigured for the RTC to work normally.

16.3.3. 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.

16.3.4. 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.

16.3.5. 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.

16.3.6. 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)

16.3.7. 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.

16.3.8. 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.

16.3.9. 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.

16.3.10. 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.

16.3.11. 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:

$$f_{cal}=f_{rtcclk} \times (1+\frac{FREQI\times 512-CMSK}{2^{N}+CMSK-FREQI\times 512}) \quad (16\text{-}3)$$

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:

$$f_{cal}=f_{rtcclk} \times (1+\frac{256-CMSK}{2^{N}+CMSK-256}) \quad (16\text{-}4)$$

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

Multiple calibaration window

Software can select single window or cwindows mode of calibration. If MWE in RTC_HRFC is not set, only one calibration window will be active when software write RTC_HRFC. Otherwise, calibration will keep running according to the 947pprox.947er947io in RTC_HRFC. When the current calibration window finishes its work, another calibration window will start immediately.

Software write 0 to MWE will close the multiple windows mode. Current calibration window on the run will not be affect by the closing action. It is not allowed to change the configuration of the smoothing calibration window and calibration value at the same time when writing 0 to MWE.

16.3.12. 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.

16.3.13. 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, BKP sram and RTDEC register
- 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.

16.3.14. 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:

$$f_{rtc_calib} = \frac{f_{rtcclk}}{(FACTOR_A+1)\times(FACTOR_S+1)} \quad (16\text{-}5)$$

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

16.3.15. 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.

16.3.16. 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 16-1 RTC pin configuration and function***.

Table 16-1 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.

16.3.17. RTC power saving mode management

Table 16-2 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

16.3.18. 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 16-3 RTC interrupts control

Interrupt	Event flag	Control Bit	Exit Sleep	Exit Deep-sleep And Standby
Alarm 0	ALRM0F	ALRM0IE	Y	Y(1)
Alarm 1	ALRM1F	ALRM1IE	Y	Y(1)
Wakeup	WTF	WTIE	Y	Y(1)
Timestamp	TSF	TSIE	Y	Y(1)
Tamper 0	TP0F	TPIE	Y	Y(1)
Tamper 1	TP1F	TPIE	Y	Y(1)

0 – Only active when RTC clock source is LXTAL or IRC32K.

16.4. Register definition

RTC base address: 0x5800 4000

16.4.1. 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)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									PM	HRT[1:0]		HRU[3:0]			
									rw	rw		rw			

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

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

16.4.2. 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

16.4.3. 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							ITSEN	COEN	OS[1:0]		OPOL	COS	DSM	S1H	A1H
							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
31:25	Reserved	Must be kept at reset value.
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.

16.4.4. 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
TP1F	Reserved	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	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	TP1F	RTC_TAMP1 detected flag Set to 1 by hardware when tamper detection is found on tamper1 input pin. Software can clear this bit by writing 0 into this bit.
14	Reserved	Must be kept at reset value.
13	TP0F	RTC_TAMP0 detected flag Set to 1 by hardware when tamper detection is found on tamper0 input pin. Software can clear this bit by writing 0 into this bit.
12	TSOVRF	Time-stamp overflow flag This bit is set by hardware when a time-stamp 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.

16.4.5. 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)

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

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

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)

16.4.6. 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.

16.4.7. 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

16.4.8. 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

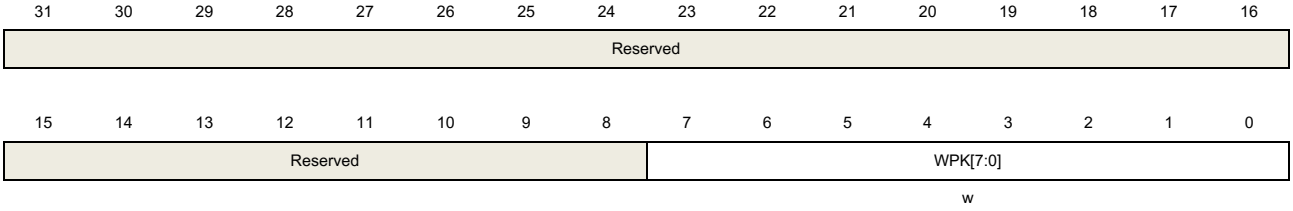
		0: Not mask hour field 1: Mask hour field
22	PM	AM/PM flag 0: AM or 24-hour format 1: PM
21:20	HRT[1:0]	Hour tens in BCD code
19:16	HRU[3:0]	Hour units in BCD code
15	MSKM	Alarm minutes mask bit 0: Not mask minutes field 1: Mask minutes field
14:12	MNT[2:0]	Minutes tens in BCD code
11:8	MNU[3:0]	Minutes 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

16.4.9. 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

16.4.10. 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)

16.4.11. 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.

16.4.12. Time of time stamp register (RTC_TTS)

Address offset: 0x30

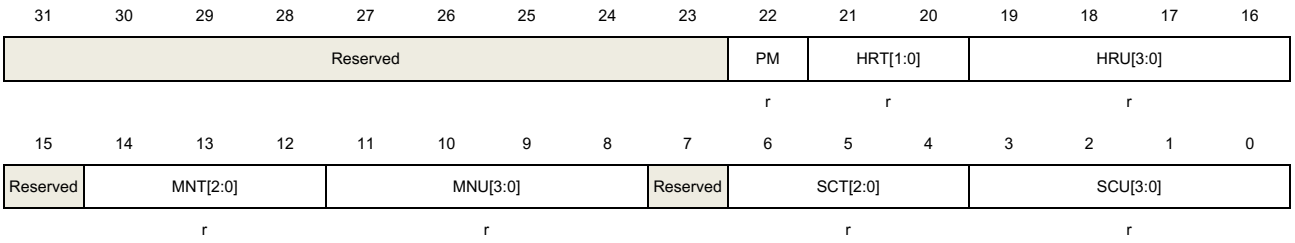
Backup domain reset value: 0x0000 0000

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

16.4.13. 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

16.4.14. 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.

16.4.15. High resolution frequency compensation register (RTC_HRFC)

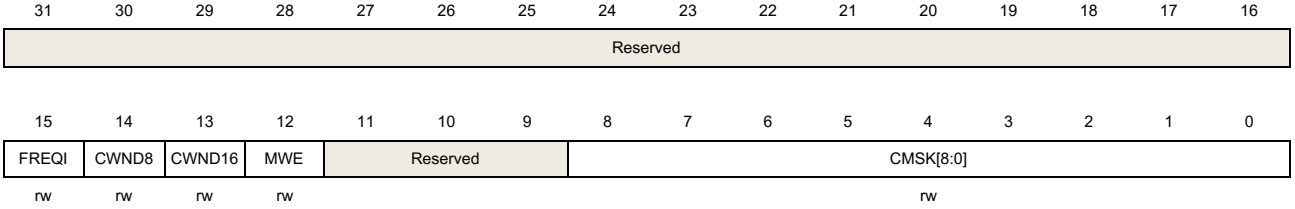
Address offset: 0x3C

Backup domain reset: 0x0000 0000

System Reset: no effect

This register is write protected.

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



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	MWE	Multiple windows calibration enable. 0: Single window mode 1: Multiple windows mode In multiple windows mode, the calibration will start again at the end of current calibration window. During continuous smoothing calibration, closing the smoothing calibration requires writing 0 to MWE. While writing 0 to MWE, other bit fields in the RTC_HRFC register must not be changed.
11:9	Reserved	Must be kept at reset value.
8:0	CMSK[8:0]	Calibration mask number The number of mask pulse out of 2^{20} RTCCLK pulse. This feature will decrease the frequency of calendar with a resolution of 0.9537 PPM.

16.4.16. 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															

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

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	DISPU	RTC_TAMPx pull up disable bit 0: Enable inner pull-up before sampling for pre-charge RTC_TAMPx pin 1: Disable pre-charge duration
14:13	PRCH[1:0]	Pre-charge duration time of RTC_TAMPx This setting determines the pre-charge time before each sampling. 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	TP1EG	Tamper 1 event trigger edge If tamper detection is in edge mode(FLT =0): 0: Rising edge triggers a tamper detection event 1: Falling edge triggers a tamper detection event If tamper detection is in level mode(FLT !=0): 0: Low level triggers a tamper detection event 1: High level triggers a tamper detection event
5	TP1EN	Tamper 1 detection enable 0:Disable tamper 1 detection function 1:Enable tamper 1 detection function
4:3	Reserved	Must be kept at reset value.
2	TPIE	Tamper detection interrupt enable 0: Disable tamper interrupt 1: Enable tamper interrupt
1	TP0EG	Tamper 0 event trigger edge If tamper detection is in edge mode(FLT =0): 0: Rising edge triggers a tamper detection event 1: Falling edge triggers a tamper detection event If tamper detection is in level mode(FLT !=0): 0: Low level triggers a tamper detection event 1: High level triggers a tamper detection event
0	TP0EN	Tamper 0 detection enable 0:Disable tamper 0 detection function 1:Enable tamper 0 detection function

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

16.4.17. Alarm 0 sub second register (RTC_ALRM0SS)

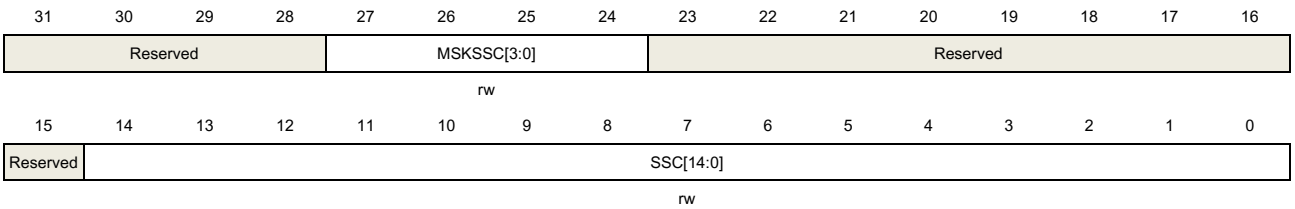
Address offset: 0x44

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.

16.4.18. Alarm 1 sub second register (RTC_ALRM1SS)

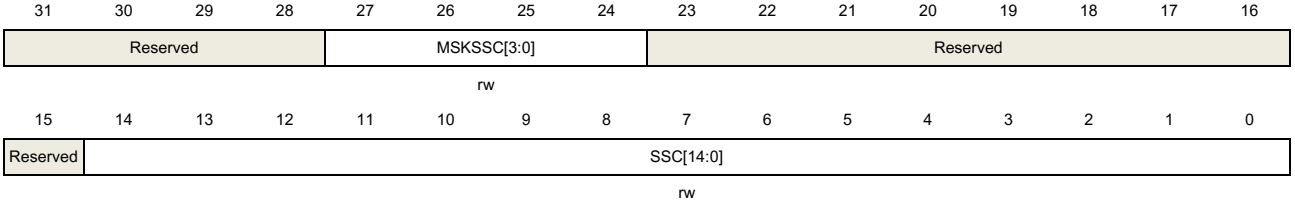
Address offset: 0x48

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.

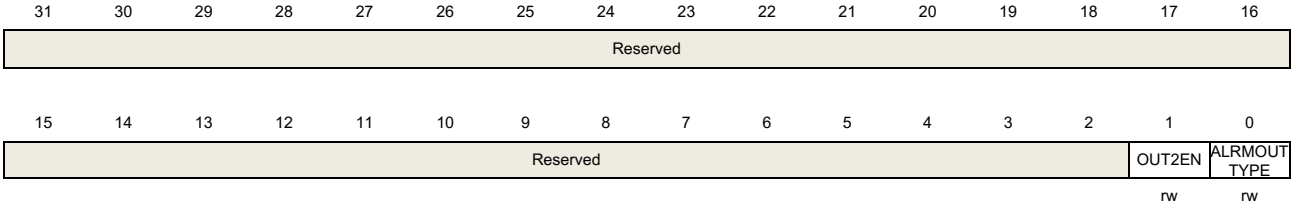
16.4.19. Configuration register (RTC_CFG)

Address offset: 0x4C

Backup domain reset: 0x0000 0000

System reset: no effect

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



Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	OUT2EN	RTC_OUT pin select

		0:RTC_OUT is output on PC13 1: RTC_OUT is output on PB2
0	ALRMOUTTYPE	RTC_ALARM Output Type 0: Open-drain output type 1: Push-pull output type

16.4.20. 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)

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

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

Bits	Fields	Descriptions
31:0	DATA[31:0]	Data These registers can be wrote or read by software. The content remains valid even in power saving mode because they can powered-on by VBAT. Tamper detection flag TPxF assertion will reset these registers.

17. 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.

17.1. Free watchdog timer (FWDGT)

17.1.1. 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.

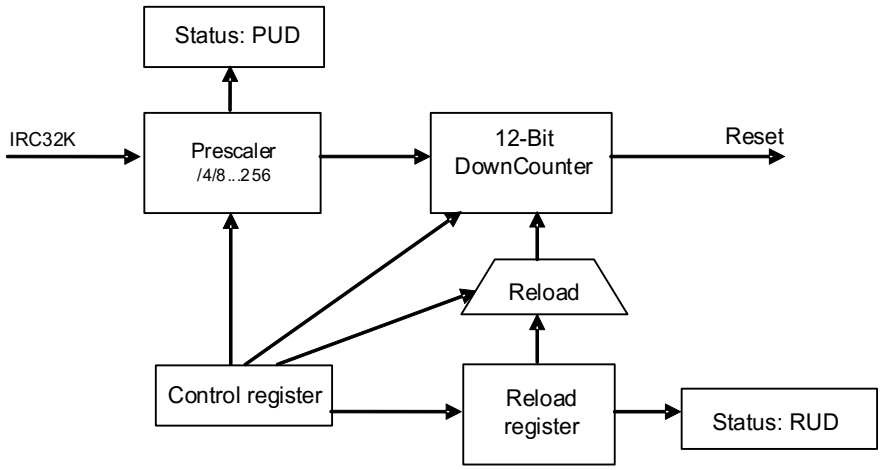
17.1.2. Characteristics

- 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.

17.1.3. Function overview

The free watchdog consists of an 8-stage prescaler and a 12-bit down counter. ***Figure 17-1. Free watchdog block diagram*** shows the functional block of the free watchdog module.

Figure 17-1. 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™-M7 core halted (Debug mode). The FWDGT stops in Debug mode if the FWDGT_HOLD bit is set.

Table 17-1. 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.

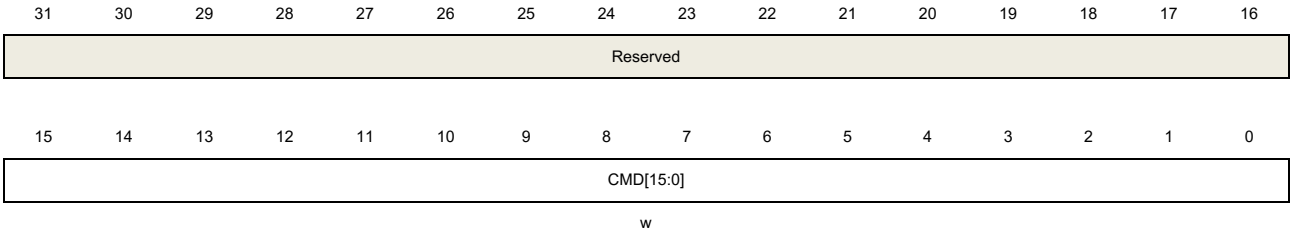
17.1.4. Register definition

FWDGT base address: 0x5800 4800

Control register (FWDGT_CTL)

Address offset: 0x00
Reset value: 0x0000 0000

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

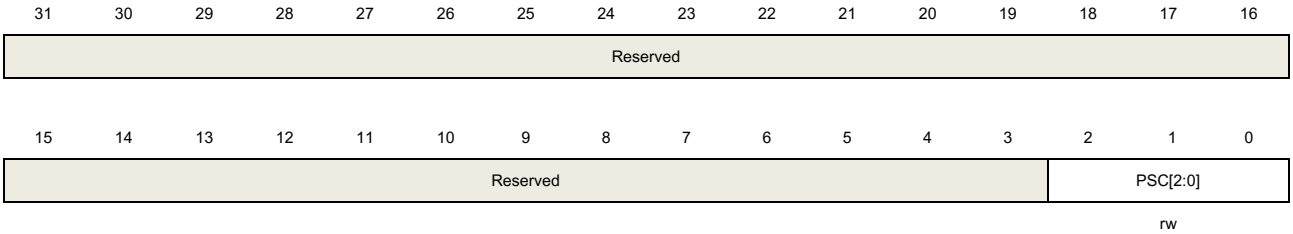


Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CMD[15:0]	Write only. Several different functions are realized by writing these bits with different values. 0x5555: Disable the FWDGT_PSC, FWDGT_RLD and FWDGT_WND write protection. 0xCCCC: Start the free watchdog timer counter. When the counter reduces to 0, the free watchdog generates a reset 0xAAAA: Reload the counter

Prescaler register (FWDGT_PSC)

Address offset: 0x04
Reset value: 0x0000 0000

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



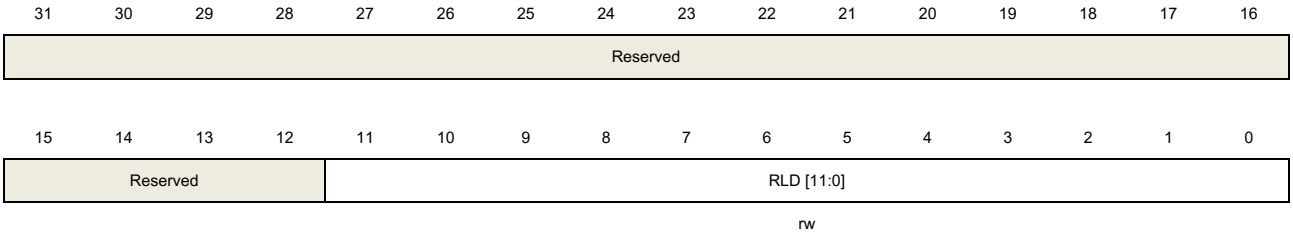
Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value.
2:0	PSC[2:0]	Free watchdog timer prescaler selection. Write 0x5555 in the FWDGT_CTL register before writing these bits. During a write operation to this register, the PUD bit in the

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 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 counter with the RLD value. These bits are write-protected. Write 0X5555 to the FWDGT_CTL register before writing these bits. During a write operation to this register, the RUD bit in the FWDGT_STAT register is set and the value read from this register is invalid. If several reload values are used by the application, it is mandatory to wait until RUD bit has been reset before changing the reload value. If the reload value has been updated, it is not necessary to wait until RUD has been reset before continuing code execution (Before entering low-power mode, it is necessary to wait until RUD is reset).

Status register (FWDGT_STAT)

Address offset: 0x0C

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	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 word(32-bit).

31	30	29	28	27	26	25	24	23	22	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 window 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).

17.2. Window watchdog timer (WWDGT)

17.2.1. 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 APB3 clock. The window watchdog timer is suitable for the situation that requires an accurate timing.

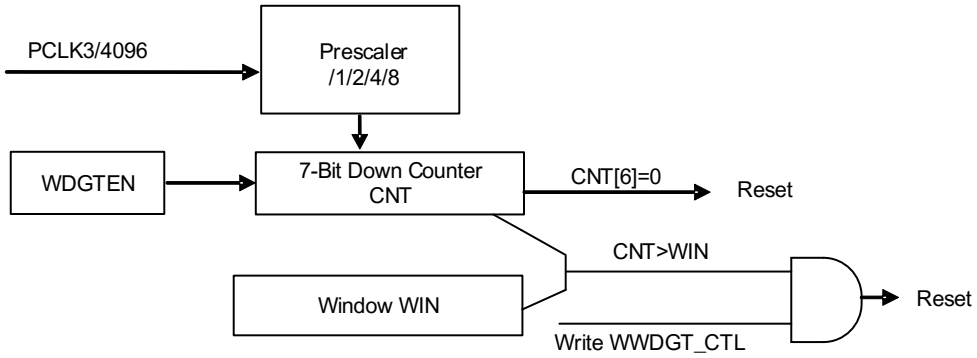
17.2.2. Characteristics

- 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.

17.2.3. 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 17-2. Window watchdog timer block diagram



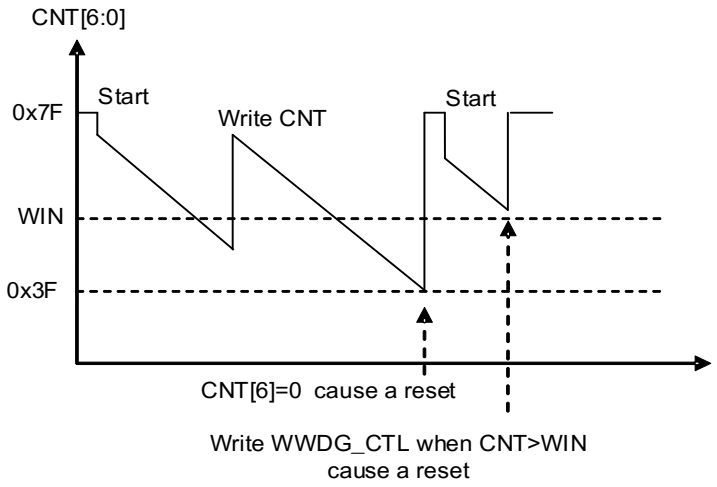
The watchdog is always disabled after power on reset. The software starts the watchdog by setting the WDGTEN bit in the WWDGT_CTL register. When window watchdog timer is enabled, the counter counts down all the time, the configured value of the counter should be greater than 0x3F(it implies that the 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 APB3 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 17-3. Window watchdog timing diagram



Calculate the WWDGT timeout by using the formula below.

$$T_{WWDGT}=t_{PCLK3} \times 4096 \times 2^{PSC} \times (CNT[5:0]+1) \quad (ms) \qquad (17\text{-}1)$$

where:

t_{WWDGT}: WWDGT timeout

t_{PCLK3}: APB3 clock period measured in ms

The *Table 17-2. Min-max timeout value at 175 MHz (fPCLK3)* and *Table 17-3. Min-max timeout value at 187.5 MHz (fPCLK3)* shows the minimum and maximum values of the t_{WWDGT}.

Table 17-2. Min-max timeout value at 175 MHz (f_{PCLK3})

Prescaler divider	PSC[1:0]	Min timeout value CNT[6:0] =0x40	Max timeout value CNT[6:0]=0x7F
1/1	00	23.41 µs	1.50 ms
1/2	01	46.81 µs	3.00 ms
1/4	10	93.62 µs	5.99 ms
1/8	11	187.25 µs	11.98 ms

Table 17-3. Min-max timeout value at 187.5 MHz (fPCLK3)(1)

Prescaler divider	PSC[1:0]	Min timeout value CNT[6:0] =0x40	Max timeout value CNT[6:0]=0x7F
1/1	00	21.85 µs	1.40 ms
1/2	01	43.69 µs	2.80 ms
1/4	10	87.38 µs	5.59 ms
1/8	11	174.76 µs	11.18 ms

Note: (1) Applicable to the high-performance mode of GD32H78x.

If the WWDGT_HOLD bit in DBG module is cleared, the WWDGT continues to work even the Cortex®-M7 core halted (Debug mode). While the WWDGT_HOLD bit is set, the WWDGT stops in Debug mode.

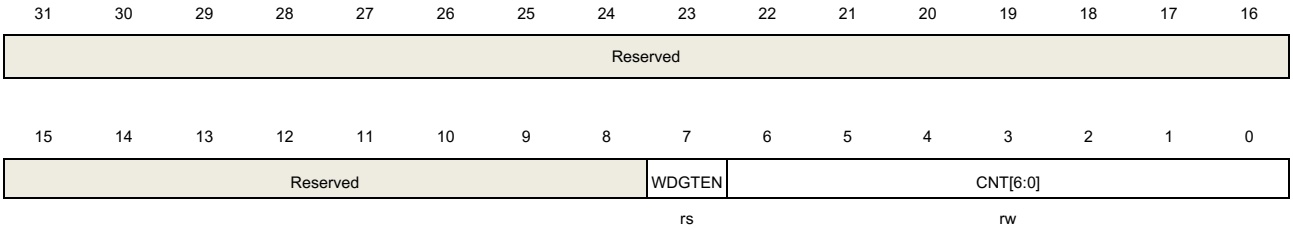
17.2.4. Register definition

WWDGT base address: 0x5000 3000

Control register (WWDGT_CTL)

Address offset: 0x00
Reset value: 0x0000 007F

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

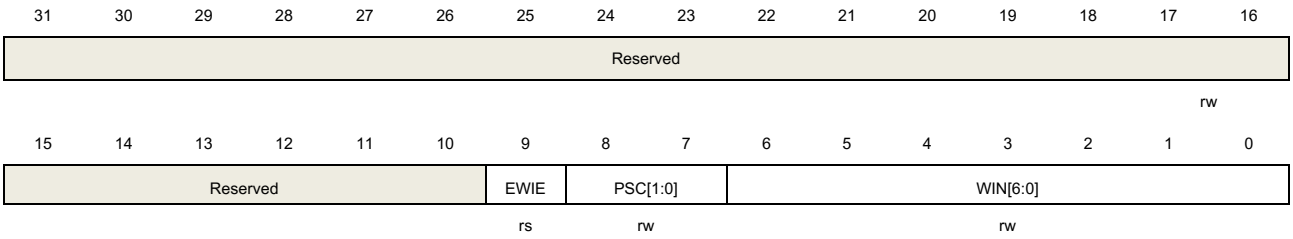


Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7	WDGTEN	Start the Window watchdog timer. Cleared by a hardware reset. Writing 0 has no effect. 0: Window watchdog timer disabled 1: Window watchdog timer enabled
6:0	CNT[6:0]	The value of the watchdog timer counter. A reset occur when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

Configuration register (WWDGT_CFG)

Address offset: 0x04
Reset value: 0x0000 007F

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



Bits	Fields	Descriptions
31:10	Reserved	Must be kept at reset value.
9	EWIE	Early wakeup interrupt enable. If the bit is set, an interrupt occurs when the counter reaches 0x40. It can be cleared by a hardware reset or software clock reset. A write

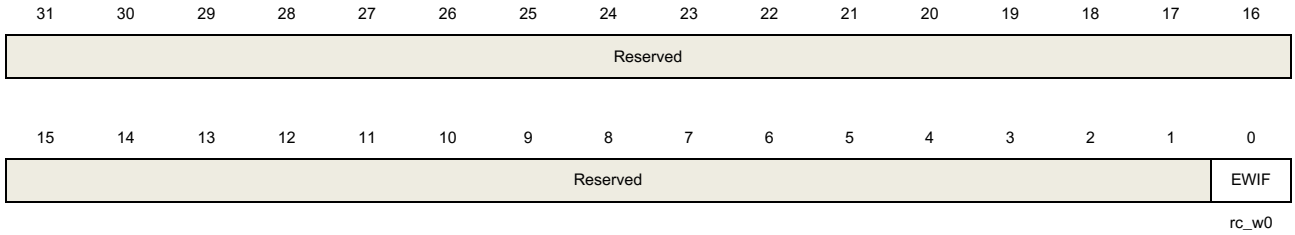
		operation of 0 has no effect.
8:7	PSC[1:0]	Prescaler. The time base of the watchdog counter 00: (PCLK3 / 4096) / 1 01: (PCLK3 / 4096) / 2 10: (PCLK3 / 4096) / 4 11: (PCLK3 / 4096) / 8
6:0	WIN[6:0]	The Window value. A reset occur if the watchdog counter (CNT bits in WWDGT_CTL) is written when the value 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 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.

18. Debug (DBG)

18.1. Overview

The GD32H77x_78x 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®-M7. 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®-M7 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 or CAN.

18.2. 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).

18.2.1. Switch JTAG or SW interface

By default, the SWD interface is active. The JTAG and SWD debug interface switching is realized through the JTAGNSW bit of the NVMC_OTP_USER_CTL0 register.

18.2.2. 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 18-1. Pin assignment

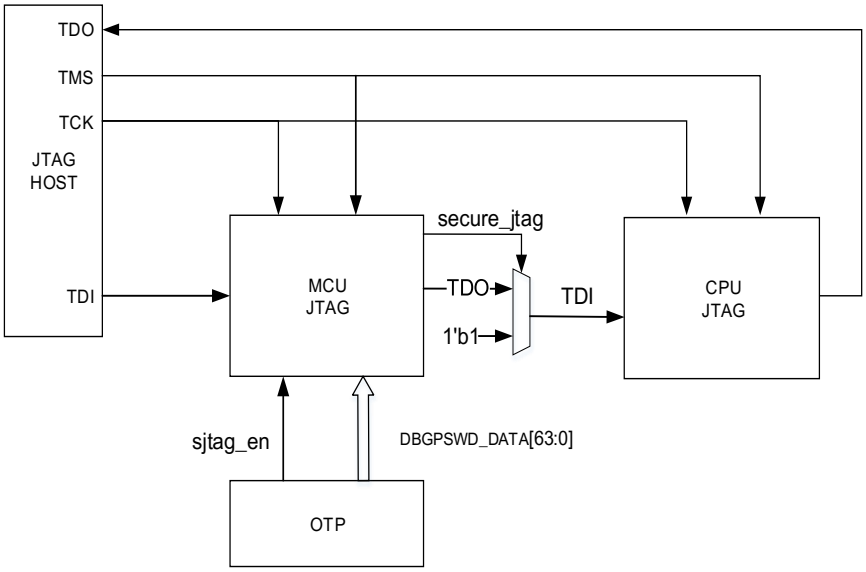
Pin	Debug interface
PA15	JTDI
PA14	JTCK/SWCLK
PA13	JTMS/SWDIO

Pin	Debug interface
PB4	NJTRST
PB3	JTDO

By default, 5-pin standard JTAG debug mode is chosen after reset. Users can also use JTAG function without NJTRST pin, then the PB4 can be used to other GPIO functions (NJTRST tied to 1 by hardware). If switch to SW debug mode, the PA15/PB3/PB4 are released to other GPIO functions. If JTAG and SW not used, all 5-pin can be released to other GPIO functions.

18.2.3. JTAG

Figure 18-1. Block diagram of JTAG unit



JTAG daisy chained structure

The Cortex®-M7 JTAG TAP (CPU JTAG) is connected to a boundary-scan (BSD) JTAG TAP (MCU JTAG). The BSD JTAG IR is 5-bit width, while the Cortex®-M7 JTAG IR is 4-bit width. So when JTAG in IR shift step, it first shift 5-bit BYPASS instruction (5'b 11111) for BSD JTAG, and then shift normal 4-bit instruction for Cortext-M7 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 0x000717A3.

Secure JTAG

1. Secure JTAG only supports JTAG but not SW

2. OTP configuration:

OTP related bits: JTAGNSW, NDBG[1:0], DBGPSWD_DATA[63:0]

Mode	Register configuration
No debug	NDBG[1:0] = 2b'10 or 2b'11

Mode	Register configuration
	JTAGNSW: Don't care DBGPSWD_DATA[63:0]: Don't care
SW	NDBG[1:0] = 2b'00 or 2b'01 JTAGNSW = 1b'0 DBGPSWD_DATA[63:0]: Don't care
Normal JTAG	NDBG[1:0] = 2b'00 JTAGNSW = 1b'1 DBGPSWD_DATA[63:0]: Don't care
Secure JTAG	NDBG[1:0] = 2b'01 JTAGNSW = 1b'1 DBGPSWD_DATA[63:0]: OTP debug password value

1－ Using Secure JTAG

a) Configure OTP for secure JTAG mode: Configure DBGPSWD_DATA[63:0] as the JTAG secure password. JTAGNSW=1b'1, NDBG[1:0]= 2b'01.

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 DBGPSWD_DATA [31:0] to DR

Write 5'b10110 to IR, Write DBGPSWD_DATA[63:32] to DR

Note: 1. If the password is entered incorrectly, a power reset is required.

2. Any wrong input sequence 986pprox, a power reset is required to re-decrypt.

3. Entering the correct password to open debug.

d) Get written value and JTAG status:

Get the written value and JTAG status via JTAG:

Write 5'b11000 to IR, Read value from DR: It can be judged whether the read value is the DBGPSWD_DATA [31:0]

Write 5'b11001 to IR, Read value from DR: It can be judged whether the read value is DBGPSWD_DATA[63:32].

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.

18.2.4. 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®-M7, 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.

18.2.5. JEDEC-106 ID code

The Cortex®-M7 integrates JEDEC-106 ID code, which is located in ROM table and mapped on the address of 0xE00FD000_0xE00FDFFF.

18.3. Debug hold function overview

18.3.1. 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 remain the same, 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 remain the same, and the debugger can debug in deep-sleep mode. When exit the deep-sleep mode, PLL is off, system clock switches to IRC64M or LPIRC4M.

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.

18.3.2. Debug support for TIMER, I2C, WWDGT, FWDGT, RTC and CAN

When the core halted and the corresponding bit in DBG control register x (DBG_CTLx, x=1,2,3,4) is set, the following behaved.

For TIMER, the timer counters stopped and hold for debug.

For I2C, SMBUS timeout hold for debug.

For WWDGT or FWDGT, the counter clock stopped for debug.

For RTC, the counter clock stopped for debug.

For CAN, the receive register stopped counting for debug.

18.4. Register definition

DBG base address: 0xE00E1000

18.4.1. ID code register (DBG_ID)

Address offset: 0x00

Read only

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

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

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

Bits	Fields	Descriptions
31:0	ID_CODE[31:0]	DBG ID code register These bits read by software. These bits are unchanged constant.

18.4.2. Control register0 (DBG_CTL0)

Address offset: 0x04

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											TRACECLKEN	TRACE_MODE[1:0]		Reserved	
											rw	rw			

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

Bits	Fields	Descriptions
31:21	Reserved	Must be kept at reset value.
20	TRACECLKEN	Trace port clock enable 0: trace port clock disable 1: trace port clock enable.
19:18	TRACE_MODE[1:0]	Trace pin allocation mode This bit is set and reset by software 00: Trace pin used in asynchronous mode

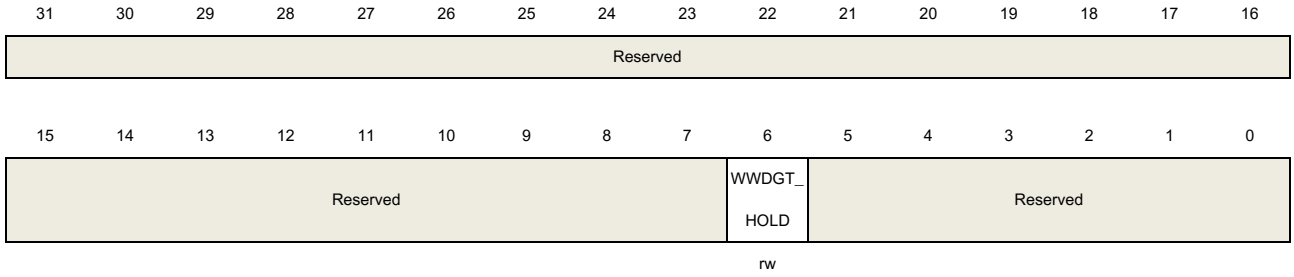
		01: Trace pin used in synchronous mode and the data length is 1 10: Trace pin used in synchronous mode and the data length is 2 11: Trace pin used in synchronous mode and the data length is 4.
17:3	Reserved	Must be kept at reset value.
2	STB_HOLD	Standby mode hold bit This bit is set and reset by software. 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.

18.4.3. Control register1 (DBG_CTL1)

Address offset: 0x34

Reset value: 0x0000 0000

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



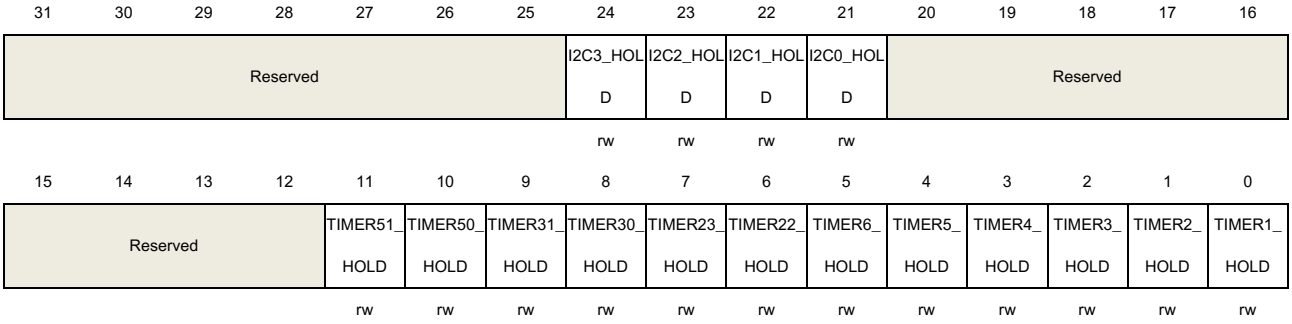
Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value.
6	WWDGT_HOLD	WWDGT hold bit This bit is set and reset by software. 0: no effect 1: Hold the WWDGT counter clock for debug when core halted.
5:0	Reserved	Must be kept at reset value.

18.4.4. Control register2 (DBG_CTL2)

Address offset: 0x3C

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:25	Reserved	Must be kept at reset value.
24	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 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 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 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 halted.
20:12	Reserved	Must be kept at reset value.
11	TIMER51_HOLD	TIMER51 hold bit This bit is set and reset by software. 0: no effect 1: Hold the TIMER51 counter for debug when core halted.
10	TIMER50_HOLD	TIMER50 hold bit This bit is set and reset by software. 0: no effect

		1: Hold the TIMER50 counter for debug when core halted.
9	TIMER31_HOLD	TIMER31 hold bit This bit is set and reset by software. 0: no effect 1: Hold the TIMER31 counter for debug when core halted.
8	TIMER30_HOLD	TIMER30 hold bit This bit is set and reset by software. 0: no effect 1: Hold the TIMER30 counter for debug when core halted.
7	TIMER23_HOLD	TIMER23 hold bit This bit is set and reset by software. 0: no effect 1: Hold the TIMER23 counter for debug when core halted.
6	TIMER22_HOLD	TIMER22 hold bit This bit is set and reset by software. 0: no effect 1: Hold the TIMER22 counter for debug when core halted.
5	TIMER6_HOLD	TIMER6 hold bit This bit is set and reset by software. 0: no effect 1: Hold the TIMER6 counter for debug when core halted.
4	TIMER5_HOLD	TIMER5 hold bit This bit is set and reset by software. 0: no effect 1: Hold the TIMER5 counter for debug when core halted.
3	TIMER4_HOLD	TIMER4 hold bit This bit is set and reset by software. 0: no effect 1: Hold the TIMER4 counter for debug when core halted.
2	TIMER3_HOLD	TIMER3 hold bit This bit is set and reset by software. 0: no effect 1: Hold the TIMER3 counter for debug when core halted.
1	TIMER2_HOLD	TIMER2 hold bit This bit is set and reset by software. 0: no effect 1: Hold the TIMER2 counter for debug when core halted.
0	TIMER1_HOLD	TIMER1 hold bit This bit is set and reset by software.

0: no effect

1: Hold the TIMER1 counter for debug when core halted.

18.4.5. Control register3 (DBG_CTL3)

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								TIMER44_HOLD	TIMER43_HOLD	TIMER42_HOLD	TIMER41_HOLD	TIMER40_HOLD	TIMER16_HOLD	TIMER15_HOLD	TIMER14_HOLD
								rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											CAN2_HOLD	CAN1_HOLD	CAN0_HOLD	TIMER7_HOLD	TIMER0_HOLD
											rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23	TIMER44_HOLD	TIMER44 hold bit This bit is set and reset by software. 0: no effect 1: Hold the TIMER44 counter for debug when core halted.
22	TIMER43_HOLD	TIMER43 hold bit This bit is set and reset by software. 0: no effect 1: Hold the TIMER43 counter for debug when core halted.
21	TIMER42_HOLD	TIMER42 hold bit This bit is set and reset by software. 0: no effect 1: Hold the TIMER42 counter for debug when core halted.
20	TIMER41_HOLD	TIMER41 hold bit This bit is set and reset by software. 0: no effect 1: Hold the TIMER41 counter for debug when core halted.
19	TIMER40_HOLD	TIMER40 hold bit This bit is set and reset by software. 0: no effect 1: Hold the TIMER40 counter for debug when core halted.
18	TIMER16_HOLD	TIMER16 hold bit

		This bit is set and reset by software. 0: no effect 1: Hold the TIMER16 counter for debug when core halted.
17	TIMER15_HOLD	TIMER15 hold bit This bit is set and reset by software. 0: no effect 1: Hold the TIMER15 counter for debug when core halted.
16	TIMER14_HOLD	TIMER14 hold bit This bit is set and reset by software. 0: no effect 1: Hold the TIMER14 counter for debug when core halted.
15:5	Reserved	Must be kept at reset value.
4	CAN2_HOLD	CAN2 hold bit This bit is set and reset by software. 0: no effect 1: Hold the CAN2 for debug when core halted.
3	CAN1_HOLD	CAN1 hold bit This bit is set and reset by software. 0: no effect 1: Hold the CAN1 for debug when core halted.
2	CAN0_HOLD	CAN0 hold bit This bit is set and reset by software. 0: no effect 1: Hold the CAN0 for debug when core halted.
1	TIMER7_HOLD	TIMER7 hold bit This bit is set and reset by software. 0: no effect 1: Hold the TIMER7 counter for debug when core halted.
0	TIMER0_HOLD	TIMER0 hold bit This bit is set and reset by software. 0: no effect 1: Hold the TIMER0 counter for debug when core halted.

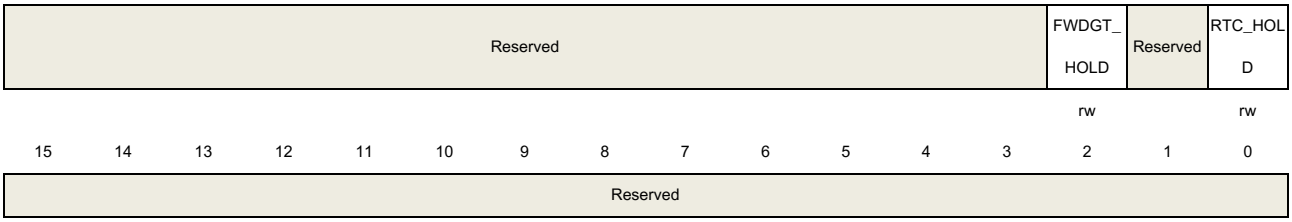
18.4.6. Control register4 (DBG_CTL4)

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



Bits	Fields	Descriptions
31:19	Reserved	Must be kept at reset value.
18	FWDGT_HOLD	FWDGT hold bit This bit is set and reset by software. 0: no effect 1: Hold the FWDGT counter clock for debug when core halted.
17	Reserved	Must be kept at reset value.
16	RTC_HOLD	RTC hold bit This bit is set and reset by software. 0: no effect 1: Hold the RTC counter clock for debug when core halted.
15:0	Reserved	Must be kept at reset value.

19. Cyclic redundancy checks management unit (CRC)

19.1. Overview

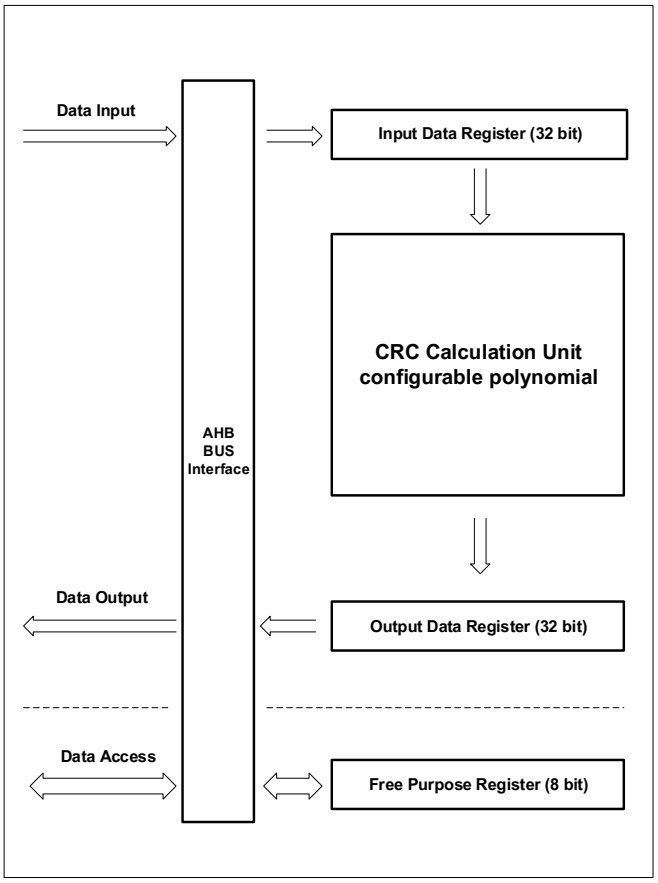
A cyclic redundancy check (CRC) is an error-detecting code commonly used in digital networks and storage devices to detect accidental changes to raw data.

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

19.2. Characteristics

- Supports 8 / 16 / 32 bits data input.
- For 7(8) / 16 / 32 bits 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 19-1. Block diagram of CRC calculation unit



19.3. 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) bits data size. During this period, AHB will not be hanged because of the existence of the 32bits 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 bits, 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.

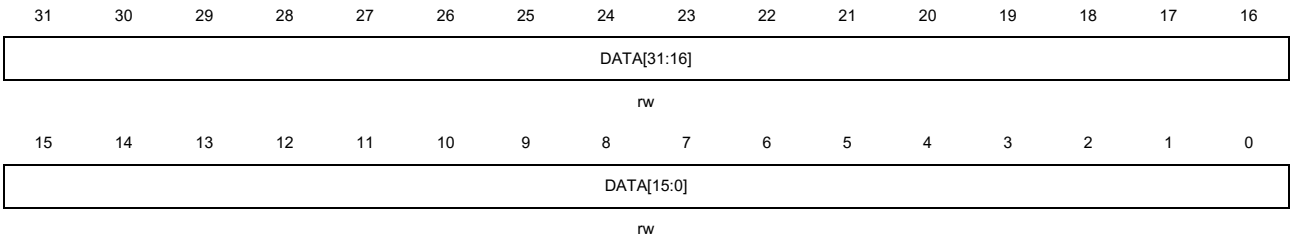
19.4. Register definition

CRC base address: 0x5802 4C00

19.4.1. Data register (CRC_DATA)

Address offset: 0x00
Reset value: 0xFFFF FFFF

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

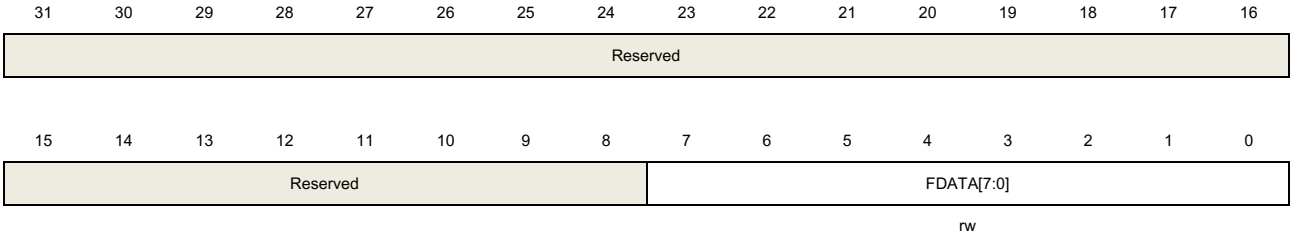


Bits	Fields	Descriptions
31:0	DATA[31:0]	CRC calculation result bits Software writes and reads. This register is used to calculate new data, and the register can be written the new data directly. Write value cannot be read because the read value is the previous CRC calculation result.

19.4.2. Free data register (CRC_FDATA)

Address offset: 0x04
Reset value: 0x0000 0000

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



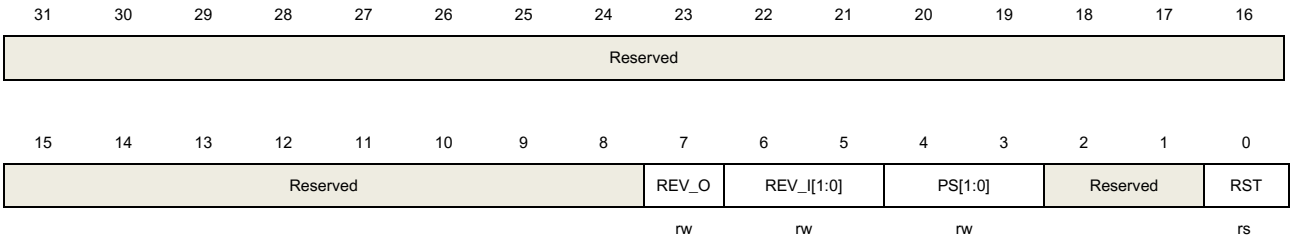
Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	FDATA[7:0]	Free data register bits Software writes and reads. These bits are unrelated with CRC calculation. This byte can be used for any goal by any other peripheral. The CRC_CTL register will generate no effect to the byte.

19.4.3. Control register (CRC_CTL)

Address offset: 0x08

Reset value: 0x0000 0000

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



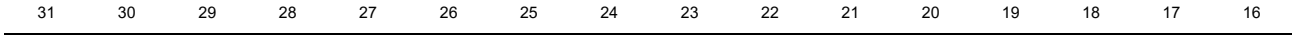
Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7	REV_O	Reverse output data value in bit order 0:Not bit reversed for output data 1:Bit reversed for output data
6:5	REV_I[1:0]	Reverse type for input data 0: Dot not use reverse for input data 1: Reverse input data with every 8-bit 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.

19.4.4. Initialization data register (CRC_IDATA)

Address offset: 0x10

Reset value: 0xFFFF FFFF

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



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

Bits	Fields	Descriptions
31:0	IDATA[31:0]	Configurable initial CRC data value When RST bit in CRC_CTL asserted, CRC_DATA will be programmed to this value.

19.4.5. 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.

20. True random number generator (TRNG)

20.1. Overview

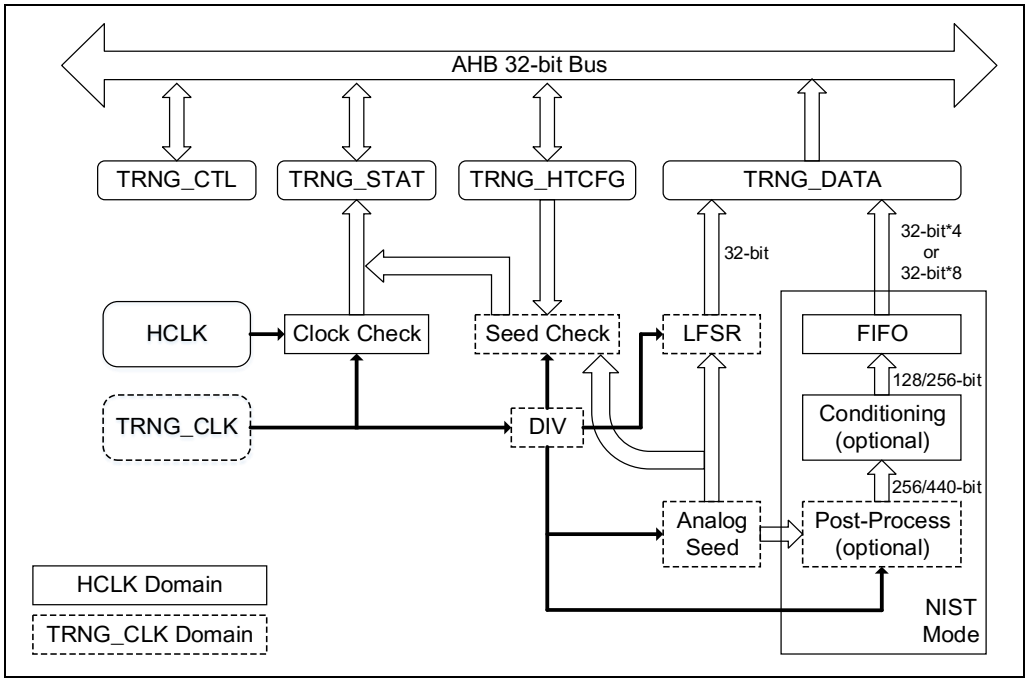
The true random number generator (TRNG) module can generate a 32-bit random value by using continuous analog noise and it has been pre-certified NIST SP800-90B.

20.2. Characteristics

- LFSR (Linear Feedback Shift Register) mode and NIST (National Institute of Standards and Technology) mode to generate random number.
- About 40 periods of TRNG_CLK are needed between two consecutive random numbers in LFSR mode.
- 32-bit random numbers are generated each operation in LFSR mode.
- TRNG NIST mode follows the NIST SP800-90B.
- Support health tests recommended by the NIST SP800-90B.
- 4 * 32-bit or 8 * 32-bit random numbers are generated each operation in NIST mode.
- TRNG has the functions of startup and health tests, associated with specific error flags.

20.3. Function overview

Figure 20-1. TRNG block diagram



There are two modes in TRNG module, NIST mode and LFSR mode.

In the NIST mode, the analog seeds of random number come 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-bit or 256-bit data output. A HASH function complied with NIST SP800-90B is implemented in the conditioning stage to increase the quality of random number. And the output data can be read out by reading the data register TRNG_DATA 4 or 8 times continuously after the data ready flag (DRDY) in TRNG_STAT register is set.

In the LFSR mode, 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 RCU chapter), so that the quality of the generated random number depends on TRNG_CLK exclusively, no matter what HCLK frequency was.

20.3.1. 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 quality of the random number. In this mode, TRNG generates a random number of 32 bits each time, which is not affected by the configuration of OUTMOD bit in the TRNG_CTL register.

20.3.2. 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.

20.3.3. Conditioning

The conditioning component in the TRNG is a deterministic HASH function that increases the quality 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).

20.3.4. 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 20-1. ALGO configurations***.

Table 20-1. ALGO configurations

ALGO	00	01	10	11
algorithm	SHA1	MD5	SHA224	SHA256
valid length	160	128	224	256

20.3.5. Health tests

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 APT_TH 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 RCT_TH 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 correct 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.

20.3.6. NIST mode state

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, and GSSTA bit in the TRNG_STAT register indicates the state of gen-samp. 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 RT_EN bit in the TRNG_CTL register is set, the first set of 128/256bit output number is a replace test result, and it should be discarded.

20.3.7. 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, source of seed for NIST 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 TRNG_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.

Note: If DRDYIMSK bit in the TRNG_CTL register is set, there is no interrupt even if DRDY bit in the TRNG_STAT register is 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.

20.3.8. Error flags

10.1.1.1 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.

10.1.1.2 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.

20.3.9. 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.

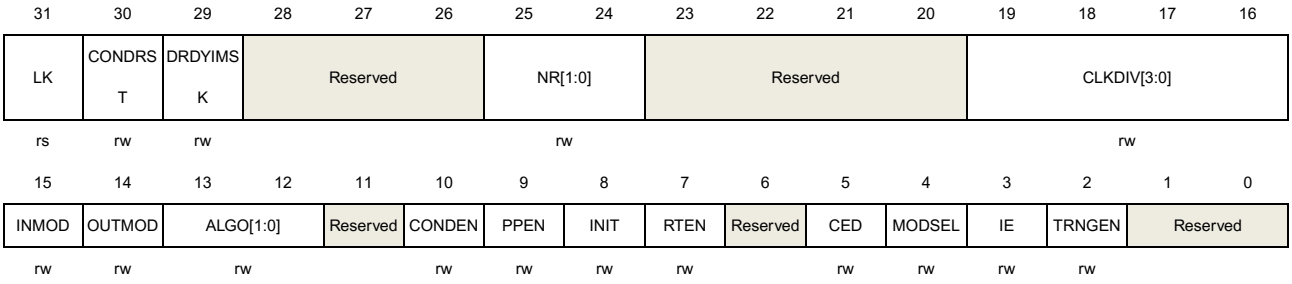
20.4. Register definition

TRNG base address: 0x4802 1800

20.4.1. Control register (TRNG_CTL)

Address offset: 0x00
Reset value: 0x0300 0410

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



Bits	Fields	Descriptions
31	LK	TRNG_CTL register lock bit This bit can only be reset to 0 if TRNG is reset. 0: Write bits[28:4] are allowed 1: Lock bits[28:4] and write bits[28:4] 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[28:4] in the TRNG_CTL register can only be written when CONDRST is 1. **Note**: During the period from 1 to 0 of this bit, at least three TRNG_CLK clock signals after frequency division need to be maintained to ensure the synchronization of configuration data
29	DRDYIMSK	DRDY interrupt mask bit. This bit can mask the interrupt generated when DRDY is set. 0: Do not mask TRNG interrupt generated when DRDY is set. 1: Mask TRNG interrupt generated when DRDY is set.
28:26	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 post_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	Reserved	Must be kept at reset value.
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 DRDY, SEIF CEIF or ERRSTA 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.

20.4.2. Status register (TRNG_STAT)

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									SEIF	CEIF	GSSTA	ERRSTA	SECS	CECS	DRDY
									rc_w0	rc_w0	r	r	r	r	r

Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value.
6	SEIF	Seed error interrupt flag This bit will be set if more than 64 consecutive same bit or more than 32 consecutive 01(or 10) changing or adaptive proportion test window exceeds APT_TH repetition during operation are detected. 0: No fault detected 1: Seed error has been detected. The bit is cleared by writing 0 or conditioning reset.
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 or conditioning reset.
4	GSSTA	This bit indicates whether the module is in generating random number sample state, when the startup health test has been completed. This bit can be cleared when FSM transitions to another error state or conditioning reset occurs. 0: The module is not in generating random number sample state 1: The module is in generating random number sample state

3	ERRSTA	NIST mode error status, 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 or adaptive proportion test window exceeds APT_TH repetition during operation 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.

20.4.3. 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).

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

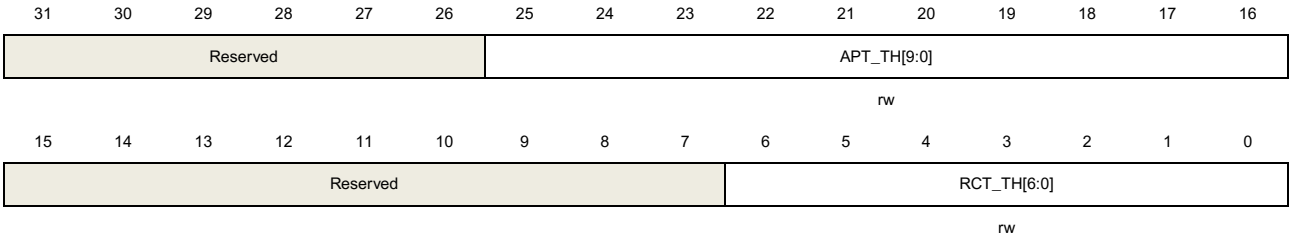
Bits	Fields	Descriptions
31:0	TRNDATA[31:0]	32-bit random data

20.4.4. Health tests configure register (TRNG_HTCFG)

Address offset: 0x10
Reset value: 0x02B3 0028

Set CONDRST bit and clear CTL_LK 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	APT_TH[9:0]	Adaptive proportion test threshold. Default 691.
15:7	Reserved	Must be kept at reset value.
6:0	RCT_TH[6:0]	Repetition count test threshold. Default 40.

21. Cryptographic Acceleration Unit (CAU)

21.1. Overview

The cryptographic acceleration unit (CAU) is used to encipher and decipher data with AES (128, 192, or 256) algorithms. This module follows the following standards:

- The Advanced Encryption Standard (AES) is announced by Federal Information Processing Standards Publication 197, November 26, 2001.

AES algorithms with different key sizes are supported to perform data encryption and decryption in the CAU in multiple modes.

The CAU is a 32-bit peripheral, DMA transfer is supported and data can be accessed in the input and output FIFO.

21.2. Characteristics

- AES encryption / decryption algorithms are supported.
- Multiple modes are supported respectively in 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 message authentication code mode (CMAC), Cipher Feedback mode (CFB) and Output Feedback mode(OFB).
- DMA transfer for incoming and outgoing data is supported.

AES

- Supports the ECB, CBC, CTR, GCM, GMAC, CCM, CMAC, 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, CMAC, 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.
- Support to use initial key from OTP by setting KEY_SEL register.

21.3. CAU data type and initialization vectors

21.3.1. Data type

The cryptographic acceleration unit receives data of 32 bits at a time, while they are

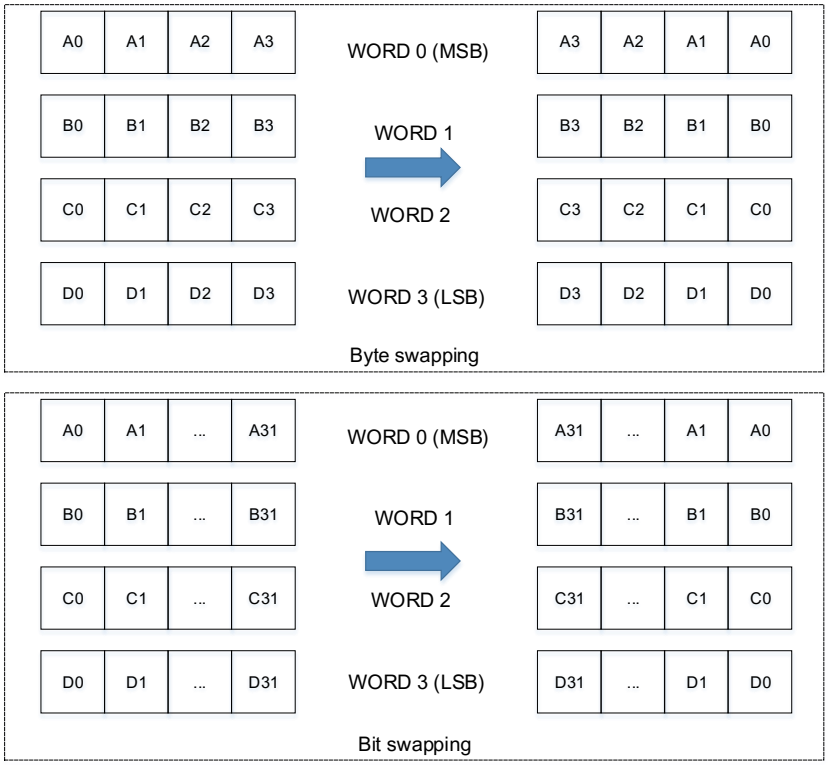
processed in 128 bits for AES algorithms. For each data block, according to the data type, the data could be bit / byte / half-word / no swapped before they are transferred into the cryptographic acceleration processor. The same swapping operation should be also performed on the processor output data before they are collected. Note the least-significant data always occupies the lowest address location no matter which data type is configured, because the system memory is little-endian.

Figure 21-1. DATAM No swapping and Half-word swapping and *Figure 21-2. DATAM Byte swapping and Bit swapping* illustrate the 128-bit AES block data swapping according to different data types.

Figure 21-1. DATAM No swapping and Half-word swapping



Figure 21-2. DATAM Byte swapping and Bit swapping



21.3.2. Initialization vectors

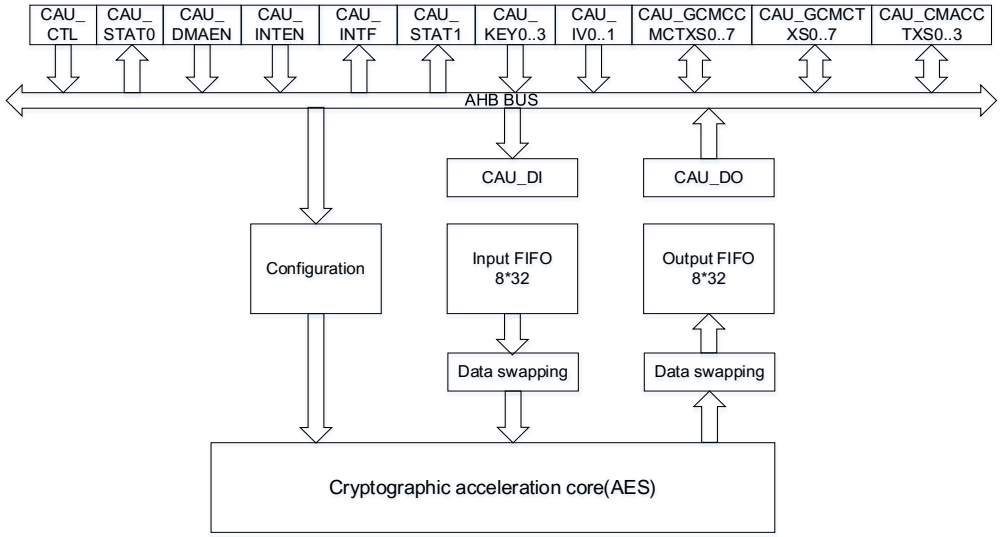
The initialization vectors are used in CBC, CTR, GCM, GMAC, CCM, CMAC, CFB and OFB modes to XOR with data blocks. They are independent of plaintext and ciphertext, and the DATAM value will not affect them. Note the initialization vector registers CAU_IV0..1(H/L) can only be written when BUSY is 0, otherwise the write operations are invalid.

21.4. Cryptographic acceleration processor

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

Figure 21-3. CAU diagram shows the block diagram of the cryptographic acceleration unit.

Figure 21-3. CAU diagram



21.4.1. AES cryptographic acceleration processor

The AES cryptographic acceleration processor consists of three components, including the AES algorithm (AEA), multiple keys and the initialization vectors or Nonce.

Three lengths of AES keys are supported: 128, 192 and 256 bits, and different initialization vectors or nonce are used depends on the operation mode.

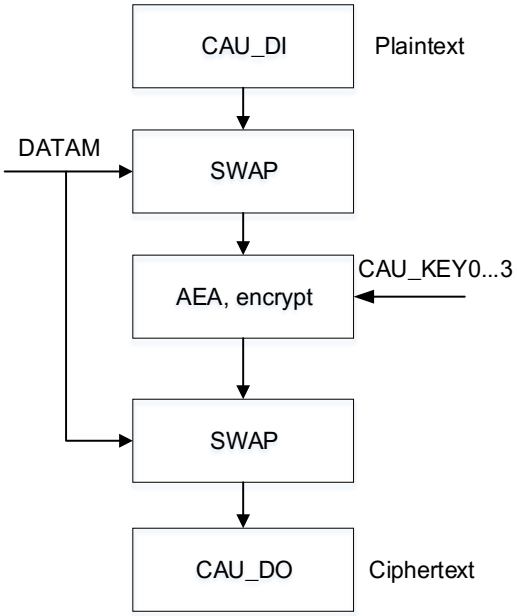
The AES key is used as [KEY3 KEY2] when the key size is configured as 128, [KEY3 KEY2 KEY1] when the key size is configured as 192 and [KEY3 KEY2 KEY1 KEY0] when the key size is configured as 256.

The thorough explanation of the key used in the AES is provided in FIPS PUB 197 (November 26, 2001), and the explanation process is omitted in this manual.

AES-ECB mode encryption

The 128-bit input plaintext is first obtained after data swapping according to the data type. The input data block is read in the AEA and encrypted using the 128, 192 or 256 -bit key. The output after above process is then swapped back according to the data type again, and a 128-bit ciphertext is produced and stored in the out FIFO. The procedure of AES ECB mode encryption is illustrated in ***Figure 21-4. AES ECB encryption***.

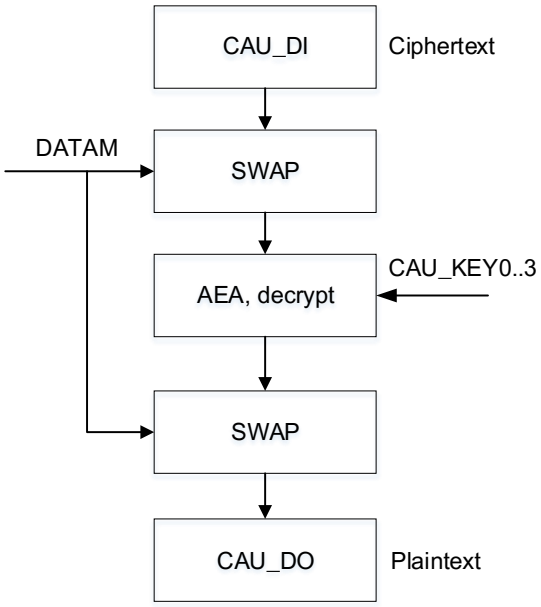
Figure 21-4. AES ECB encryption



AES-ECB mode decryption

First of all, the key derivation must be completed to prepare the decryption keys, the input key of the key schedule is the same to that used in encryption. The last round key obtained from the above operation is then used as the first round key in the decryption. After the key derivation, the 128-bit input ciphertext is first obtained after data swapping according to the data type. The input data block is read in the AEA and decrypted using keys prepared above. The output is then swapped back according to the data type again, and a 128-bit plaintext is produced. The procedure of AES ECB mode decryption is illustrated in ***Figure 21-5. AES ECB decryption***.

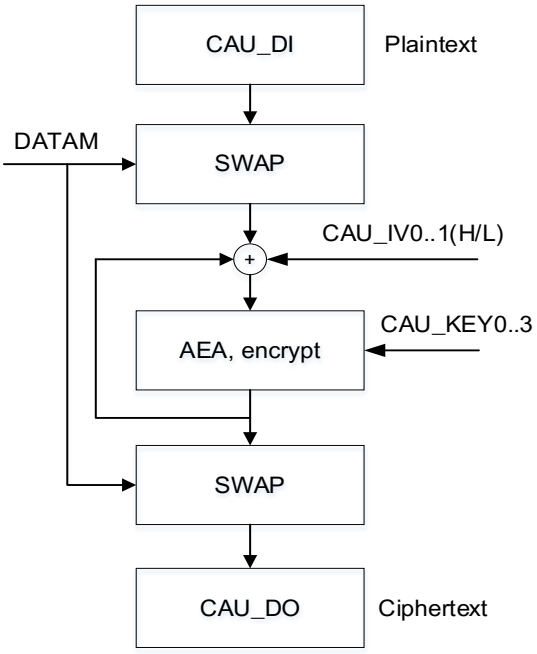
Figure 21-5. AES ECB decryption



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 that the final partial data block should be encrypted in a specified manner if the plaintext message does not consist of an integral number of data blocks. 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 21-6. AES CBC encryption***.

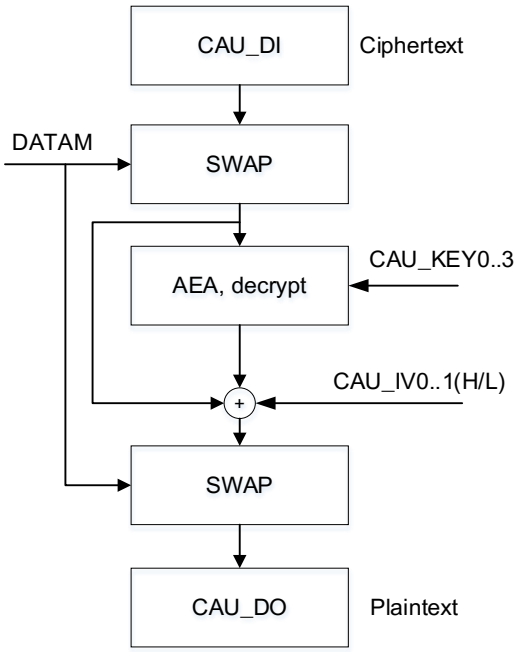
Figure 21-6. AES CBC encryption



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 21-7. AES CBC decryption*.

Figure 21-7. 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 21-8. Counter block structure*** illustrates the counter block structure and ***Figure 21-9. AES CTR encryption/decryption*** shows the AES CTR encryption / decryption.

Figure 21-8. Counter block structure

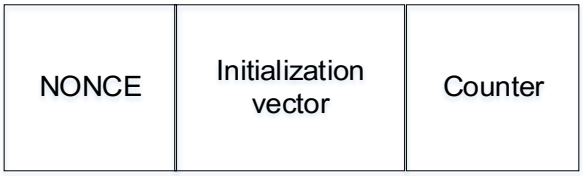
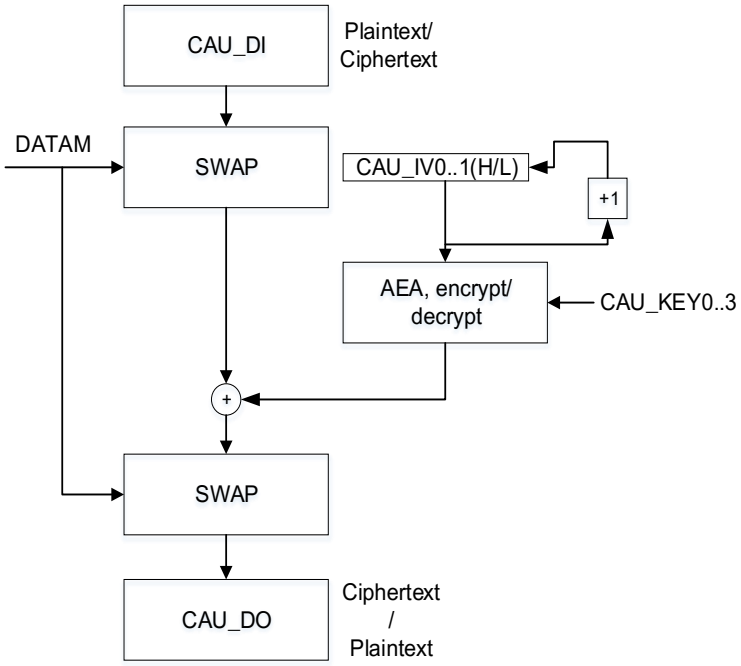


Figure 21-9. AES CTR encryption/decryption



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.

   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 additional authenticated 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-CMAC mode

CMAC allows to authenticate the plaintext, generating the corresponding TAG. The CMAC mode follows NIST special publication 800-38B. The message is composed by the prepare phase and the mac generate phase.

1. CMAC prepare phase

The configuration process is as follows:

(a) Clear the CAUEN bit to make sure CAU is disabled.
(b) Configure the ALGM[3:0] bits to '1100'.
(c) Configure GCM_CCMPH [1:0] bits to '00'.
(d) Configure key registers CAU_KEY
(e) initialization vectors with zero values.
(f) Enable CAU by writing 1 to CAUEN bit.
(g) Wait until CAUEN is cleared by hardware, and then move to the following phases.

2. generate mac phase

The configuration process is as follows:

(a) Configure GCM_CCMPH [1:0] bits to '01'.
(b) Enable CAU by writing 1 to CAUEN bit.
(c) Insert B0 for first transfer, and then B for further transfers. Write data into CAU_DI register, INF and IEM flags can be used to determine if the input FIFO can receive data. DMA can also be used.
(d) the CMAC_FB and the NBPILB bitfield should be configured after the encryption of the penultimate data block is completed.
(e) Wait until the ONE flag is set to 1, and then read CAU_DO 4 times. The output corresponds to the message authentication code (MAC).
(f) Disable the CAU.

NOTE: if only insert a 128bit block, please configure the CMAC_FB and the NBPILB bitfield after the generate mac phase(b).

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 the decryption process is similar to the encryption described before.

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 the decryption process is similar to the encryption described before.

21.5. Operating modes

Encryption

1. If users want to use AES key from OTP, go to step 2, otherwise, go to step 3.
2. Write AES keys (in little endian) in sequence into the CAU_AES_KEY field located in the OTP, then check whether the CRC values are right.
3. Disable the CAU by resetting the CAUEN bit in the CAU_CTL register.
4. If the AES key comes from registers of the CAU, skip this step. If the AES key comes from OTP, make sure that the key has been written to OTP and set KEY_SEL bit.
5. Select and configure the key length with the KEYM bits in the CAU_CTL register if AES algorithm is chosen. If the AES key comes from OTP, skip this step.
6. Reset KEY_SEL bit in the CAU_CTL register then configure the CAU_KEY0..3(H / L) registers according to the algorithm. If the AES key comes from OTP, skip this step.
7. Configure the DATAM bit in the CAU_CTL register to select the data swapping type.
8. Configure the algorithm (AES) and the chaining mode (ECB / CBC / CTR / GCM / GMAC / CCM / CMAC / CFB / OFB) by writing the ALGM[3:0] bit in the CAU_CTL register.
9. Configure the encryption direction by writing 0 to the CAUDIR bit in the CAU_CTL register.
10. Configure the initialization vectors by writing the CAU_IV0..1 registers.
11. Flush the input FIFO and output FIFO by configure the FFLUSH bit in the CAU_CTL register when CAUEN is 0.
12. Enable the CAU by set the CAUEN bit as 1 in the CAU_CTL register.
13. 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.
14. Wait for ONE bit in the CAU_STAT0 register is 1 then read the CAU_DO register. The output data can also be transferred by DMA / CPU during interrupts / no DMA or interrupts.
15. Repeat steps 13, 14 until all data blocks has been encrypted.

Note: When users use AES key from OTP (KEY_SEL is set), AES algorithm mode is forced AES-CTR mode (ALGM[3:0] is forced to 0b0110) and AES key length is forced to 128-bits.

Decryption

1. Disable the CAU by resetting the CAUEN bit in the CAU_CTL register.

2. Select and configure the key length with the KEYM bits in the CAU_CTL register if AES algorithm is chosen.
3. Reset KEY_SEL bit in the CAU_CTL register then configure the CAU_KEY0..3(H / L) registers according to the algorithm. If the key comes from OTP, make sure that the key has been written to OTP and then set KEY_SEL bit.
4. Configure the DATAM bit in the CAU_CTL register to select the data swapping type.
5. Configure the ALGM[3:0] bits to "0111" in the CAU_CTL register to complete the key derivation.
6. Enable the CAU by set the CAUEN bit as 1.
7. Wait until the BUSY and CAUEN bit return to 0 to make sure that the decryption keys are prepared.
8. Configure the algorithm (AES) and the chaining mode (ECB / CBC / CTR / GCM / GMAC / CCM / CMAC / CFB / OFB) by writing the ALGM[3:0] bit in the CAU_CTL register.
9. Configure the decryption direction by writing 1 to the CAUDIR bit in the CAU_CTL register.
10. Configure the initialization vectors by writing the CAU_IV0..1 registers.
11. Flush the input FIFO and output FIFO by configure the FFLUSH bit in the CAU_CTL register when CAUEN is 0.
12. Enable the CAU by set the CAUEN bit as 1 in the CAU_CTL register.
13. 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.
14. Wait for ONE bit in the CAU_STAT0 register is 1, then read the CAU_DO register. The output data can also be transferred by DMA / CPU during interrupts / no DMA or interrupts.
15. 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 number of filled pads and encrypt / decrypt it. It should be noted that the NBPILB[3:0] bitfield should be configured after the encryption / decryption of the penultimate data block is completed.

For CMAC mode, CAU supports non 128 bit integer multiple data block processing. When the last data block is less than 128 bits, the remaining bits are padded with 10...0, and the number of bytes used for padding is configured in the NBPILB field of the CAU_CTL register. Note that the NBPILB field can only be configured after the encryption of the second-to-last data block is completed.

21.6. CAU DMA interface

The DMA can be used to transfer data blocks with the interface of the cryptographic acceleration unit. The operations can be controlled by the CAU_DMAEN register. DMAIEN is used to enable the DMA request during the input phase, then a word is written into CAU_DI from DMA. DMAOEN is used to enable the DMA request during the output phase, then a word is read from the CAU.

Single and Burst transfers are both supported to ensure the data transfer if the number of words is not an integral multiple of burst size. Note the DMA controller should be configured to perform burst of 4 words or less to make sure no data will be lost. DMA channel for output data has a higher priority than that channel for input data so that the output FIFO can be empty earlier than that the input FIFO is full.

21.7. 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.

21.8. 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. Wait until 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.
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, CMAC, 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) and CAU_CMACCTXSx (x=0..3) 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) and CAU_CMACCTXSx (x=0..3) 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, CMAC, 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) and CAU_CMACCTXSx (x=0..3) 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) and CAU_CMACCTXSx (x=0..3) should also be restored. Then enable CAU by setting the CAUEN bit in the CAU_CTL register.

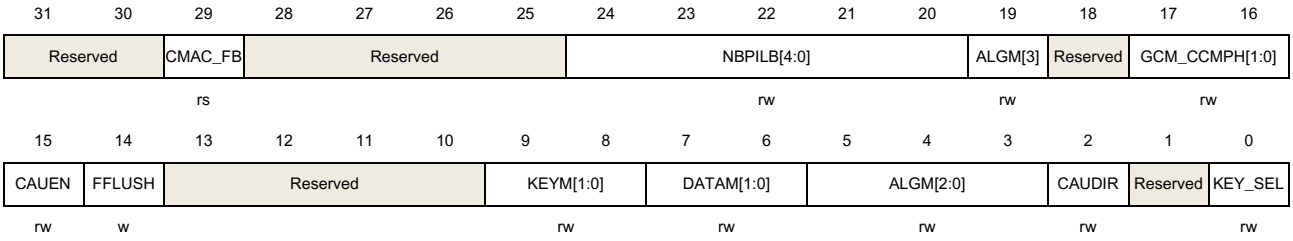
21.9. Register definition

CAU base address: 0x4802 1000

21.9.1. Control register (CAU_CTL)

Address offset: 0x00
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	CMAC_FB	CMAC generate mac final block control This bit field be cleared by hardware
28:25	Reserved	Must be kept at reset value.
24:20	NBPILB[4:0]	Number of bytes padding in last block of payload 00000: all bytes are valid (no padding) 00001: one padding byte of last block ... 01111: 15 padding bytes of last block 10000: padding for 16 least-significant bytes of last block (only for cmac mode，gcm / ccm no padding) 10001-11111: reserved
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 and CMAC phase 00: prepare phase 01: AAD phase or CMAC mac generate 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 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 1100: CMAC mode
2	CAUDIR	CAU direction, must be configured when BUSY=0 0: encryption 1: decryption
1	Reserved	Must be kept at reset value.

0	KEY_SEL	Key select 0: Use key from CAU_KEY0..3(H/L) 1:Use key from OTP

21.9.2. Status register 0 (CAU_STAT0)

Address offset: 0x04
Reset value: 0x0000 0003

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											BUSY	OFU	ONE	INF	IEM
											r	r	r	r	r

Bits	Fields	Descriptions
31:5	Reserved	Must be kept at reset value.
4	BUSY	Busy bit 0: No processing. This is because: - CAU is disabled by CAUEN=0 or the processing has been completed. - No enough data or no enough space in the input / output FIFO to perform a data block 1: CAU is processing data or key derivation.
3	OFU	Output FIFO is full 0: Output FIFO is not full 1: Output FIFO is full
2	ONE	Output FIFO is not empty 0: Output FIFO is empty 1: Output FIFO is not empty
1	INF	Input FIFO is not full 0: Input FIFO is full 1: Input FIFO is not full
0	IEM	Input FIFO is empty 0: Input FIFO is not empty 1: Input FIFO is empty

21.9.3. 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 input FIFO, read these bits will return input FIFO value if CAUEN is 0, or it will return an undefined value

21.9.4. 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 output FIFO value.

21.9.5. DMA enable register (CAU_DMAEN)

Address offset: 0x10

Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	DMAOEN	DMA output enable 0: DMA for OUT FIFO data is disabled 1: DMA for OUT FIFO data is enabled
0	DMAIEN	DMA input enable 0: DMA for IN FIFO data is disabled 1: DMA for IN FIFO data is enabled

21.9.6. Interrupt enable register (CAU_INTEN)

Address offset: 0x14

Reset value: 0x0000 0000

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

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

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

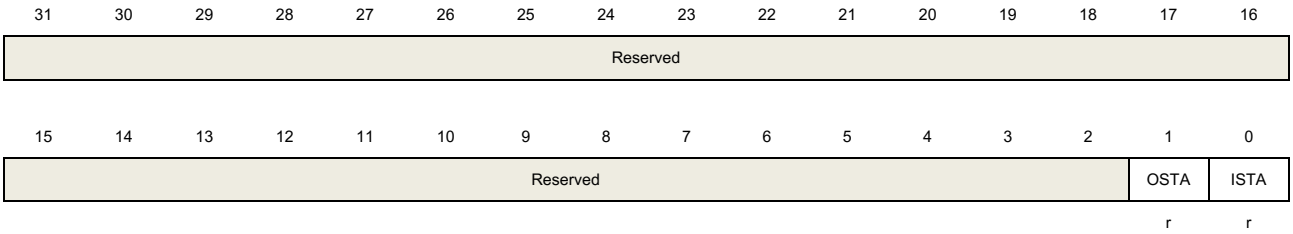
Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	OINTEN	OUT FIFO interrupt enable 0: OUT FIFO interrupt is disable 1: OUT FIFO interrupt is enable
0	IINTEN	IN FIFO interrupt enable 0: IN FIFO interrupt is disable 1: IN FIFO interrupt is enable

21.9.7. Status register 1 (CAU_STAT1)

Address offset: 0x18

Reset value: 0x0000 0001

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



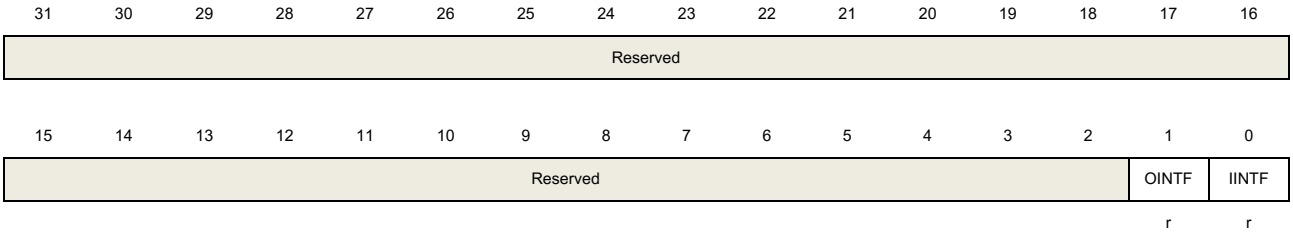
Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	OSTA	OUT FIFO interrupt status 0: OUT FIFO interrupt status not pending 1: OUT FIFO interrupt status pending
0	ISTA	IN FIFO interrupt status 0: IN FIFO interrupt not pending 1: IN FIFO interrupt flag pending

21.9.8. 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

21.9.9. Key registers (CAU_KEY0..3(H / L))

Address offset: 0x20 to 0x3C
Reset value: 0x0000 0000

These registers have to be accessed by word (32-bit), and all of them must be written when BUSY is 0.

In AES-128 mode, KEY2H[31:0] || KEY2L[31:0] is used as AES_KEY[0:63], and KEY3H[31:0] || KEY3L[31:0] is used as AES_KEY[64:127].

In AES-192 mode, KEY1H[31:0] || KEY1L[31:0] is used as AES_KEY[0:63], KEY2H[31:0] || KEY2L[31:0] is used as AES_KEY[64:127], and KEY3H[31:0] || KEY3L[31:0] is used as AES_KEY[128:191].

In AES-256 mode, KEY0H[31:0] || KEY0L[31:0] is used as AES_KEY[0:63], KEY1H[31:0] || KEY1L[31:0] is used as AES_KEY[64:127], KEY2H[31:0] || KEY2L[31:0] is used as AES_KEY[128:191], and KEY3H[31:0] || KEY3L[31:0] is used as AES_KEY[192:255].

Note: "||" is a concatenation operator. For example, X || Y denotes the concatenation of two bit strings X and Y.

CAU_KEY0H

Address offset: 0x20
Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY0H[31:16]															
w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY0H[15:0]															
w															

CAU_KEY0L

Address offset: 0x24
Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY0L[31:16]															
w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY0L[15:0]															

w

CAU_KEY1H

Address offset: 0x28

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY1H[31:16]															
w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY1H[15:0]															
w															

CAU_KEY1L

Address offset: 0x2C

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY1L[31:16]															
w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY1L[15:0]															
w															

CAU_KEY2H

Address offset: 0x30

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY2H[31:16]															
w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY2H[15:0]															
w															

CAU_KEY2L

Address offset: 0x34

Reset value: 0x0000 0000

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

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

KEY2L[31:16]															
w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY2L[15:0]															
w															

CAU_KEY3H

Address offset: 0x38
Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY3H[31:16]															
w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY3H[15:0]															
w															

CAU_KEY3L

Address offset: 0x3C
Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY3L[31:16]															
w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY3L[15:0]															
w															

Bits	Fields	Descriptions
31:0	KEY0...3(H / L)	The key for AES

21.9.10. 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 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

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

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

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

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

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

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

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

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 AES

21.9.11. 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.

21.9.12. 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.

21.9.13. CMAC mode context switch register x (CAU_CMACCTXSx) (x=0..3)

Address offset: 0x90 to 0x9C

Reset value: 0x0000 0000

This register 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 CMAC mode is selected.

22. Hash Acceleration Unit (HAU)

22.1. Overview

The hash acceleration unit is used for information security. The secure hash algorithm (SHA-1, SHA-224, SHA-256), the message-digest algorithm (MD5) and the keyed-hash message authentication code (HMAC) algorithm are supported for various applications. The digest will be computed and the length is 160 / 224 / 256 / 128 bits for a message up to (2^{64} - 1) bits computed by SHA-1, SHA-224, SHA-256 and MD5 algorithms respectively. In HMAC algorithm, SHA-1, SHA-224, SHA-256 or MD5 will be called twice as hash functions and authenticating messages can be produced.

The HAU is fully compliant implementation of the following standards:

- Federal Information Processing Standards Publication 180-4(FIPS PUB 180-4).
- Secure Hash Standard specifications (SHA-1, SHA-224, SHA-256).
- Internet Engineering Task Force Request for Comments number 1321 (IETF RFC 1321) specifications (MD5).

22.2. Characteristics

- 32-bit AHB slave peripheral.
- High performance of computation of hash algorithms.
- Little-endian data representation.
- Multiple data types are supported, including no swapping, half-word swapping, byte swapping, and bit swapping with 32-bit data words.
- Automatic data padding to fill the 512-bit message block for digest computation.
- DMA transfer is supported.
- Hash / HMAC process suspended mode.

22.3. HAU data type

The hash acceleration unit receives data words of 32 bits at a time, while they are processed in 512-bits blocks. For each input word, according to the data type, the data could be bit / byte / half-word / no swapped before they are transferred into the hash acceleration core. The same swapping operation should be also performed on the core output data before they are collected. Note the least-significant data always occupies the lowest address location no matter which data type is configured, because the system memory is little-endian. However, the computation of SHA-1, SHA-224 and SHA-256 are big-endian.

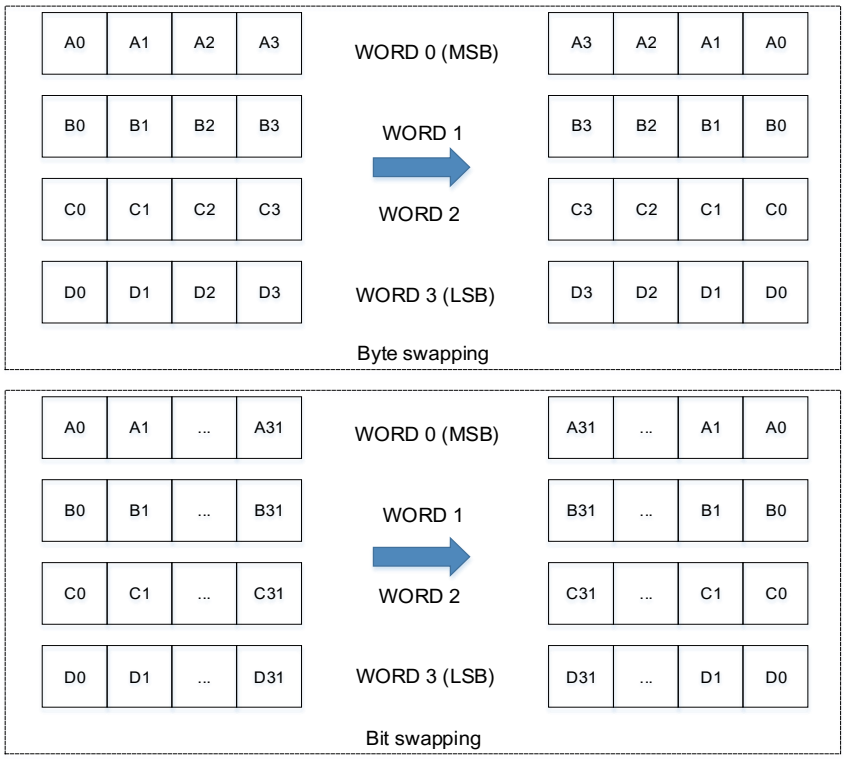
Figure 22-1. DATAM No swapping and Half-word swapping and *Figure 22-2. DATAM Byte swapping and Bit swapping* illustrate the data swapping according to different data

types.

Figure 22-1. DATAM No swapping and Half-word swapping



Figure 22-2. DATAM Byte swapping and Bit swapping

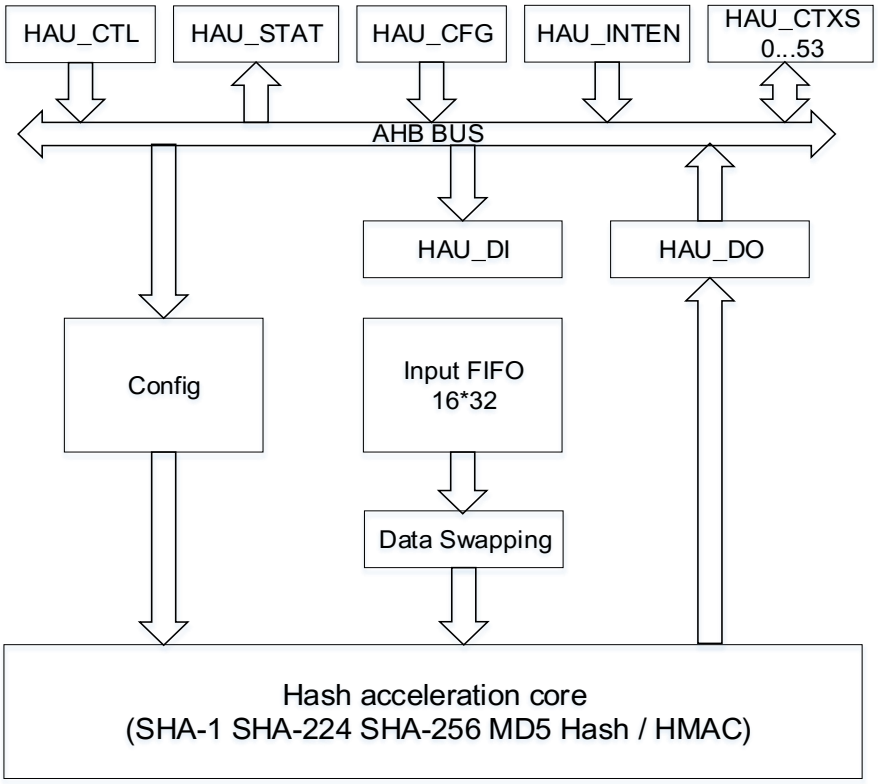


22.4. HAU core

The hash acceleration unit is used to compute condensed information of input messages with secure hash algorithms. The digest result has a length of 160 / 224 / 256 / 128 bits for a message up to (2^{64}-1) bits computed by SHA-1, SHA-224, SHA256 and MD5 algorithms respectively. It can be used to generate or verify the signature of a message with a higher efficiency because of the much simpler of the information.

A message which need to be processed in the HAU should be considered as bit information. And the length is the number of bits of the message. The information security is ensured because that, to find the original message using the digest is computationally impossible and, the result will be completely different with any change to the input message.

Figure 22-3. HAU block diagram



22.4.1. Automatic data padding

The input message should be padded first so that the number of bits in the input of the HAU core can be an integral multiple of 512. First of all, a "1" is added to follow the last bit of the input message, and then several "0" should be padded to ensure the result modulo 512 is 448, at last, a 64-bit length information of input is added.

After the message padding is correctly performed, the VBL bits in the HAU_CFG register is

configured as the 64-bit length value above, and CALEN bit in the HAU_CFG register can be set 1 to start the calculation of the digest of the last block.

Data Padding Example: The input message is "HAU", which ASCII hexadecimal code is:

484155

Then the VBL bits in the HAU_CFG register is set as decimal 24 because of the valid bit length. A "1" is added at bit location 24 then, and several "0" are padded so that the result modulo 512 is 448, the hexadecimal result is as follows:

48415580 00000000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000

After that, a 64-bit length information of the input message is padded, which hexadecimal value is 18, and the final result will be:

48415580 00000000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000018

22.4.2. Digest computing

After data padding, for each block calculation of HAU, 512 bits are written into the HAU core by DMA or CPU. To start the processing of the HAU core, the peripheral must obtain the information as to whether the HAU_DI register contains the last bits of the message or not. This can be confirmed with the status of the input FIFO and the HAU_DI register.

When DMA is used to transfer data:

The status of the block transfer is automatically interpreted with the information from the DMA controller. And padding and digest computation are performed automatically as if CALEN bit in the HAU_CFG register is set as 1.

Note: If hash message is large files and multiple DMA transfers are needed, then MDS bit should be set as 1. And the VBL bits need to be set before the transfer. The CALEN bit is not set automatically after an intermediate DMA transfer completed. Only when the last DMA transfer is processing, the MDS bit is cleared so that the CALEN bit is automatically set after data transferring.

Otherwise, the MDS bit is set as 0. And the CALEN bit is set automatically after a DMA transfer. Also, VBL bits need to set before the DMA transfer.

When CPU is used to transfer data without DMA:

- The intermediate block computing can be started when HAU_DI is filled with another new word of the next block.
- The last block computing can be started when CALEN bit in the HAU_CFG register is 1.

22.4.3. Hash mode

The hash mode is selected when the HMS bit in the HAU_CTL register is set as 0. A new digest calculation is started when the START bit in the HAU_CTL register is 1, and SHA-1, SHA-224, SHA-256 and MD5 mode computation is chosen by the ALGM bits in the HAU_CTL register.

After a message block of 512 bit has been received through the HAU_DI register and the input FIFO, the processor starts the calculation with the information from DMA or the status of the CALEN bit.

The results can be finally read from the HAU_DO0...7 registers.

22.4.4. HMAC mode

HMAC mode is used for message authentication with a unique key chosen by the user. More information about the HMAC specifications please refer to "HMAC: keyed-hashing for message authentication, H. Krawczyk, M. Bellare, R. Canetti, February 1997".

The HMAC algorithm can be represented as:

HMAC(input) = HASH[((key | opad) XOR 0x5c) | HASH(((key | ipad) XOR 0x36) | input)]

where ipad and opad are used to extend the key to 512 bits with several '0' and '1' is the concatenation operator.

There are four different phases in the HMAC mode:

1. Configure the HMS bit in the HAU_CTL register as 1 and set the ALGM bits for the desired algorithm. If the key size is longer than 64 bytes, then the KLM bit in the HAU_CTL register should also be set. After that, start the HAU core by setting the START bit.
2. The key is used as the input message to complete the calculation in HASH mode.
3. The new key used for the inner hush function is elaborated when the last word is accessed and computation has started.
4. After the first hash round, HAU core starts to receive the key for the outer hash function, usually, the outer hash function uses the same new key as the inner hash function. And when the last word of the key is entered and computation starts, the results are available in the HAU_DO registers.

22.5. HAU suspended mode

It is possible to suspend HASH or HMAC operation to perform a high-prior task first, then after the high-prior task is finished, resume the suspended operation.

When suspending the current task, it is necessary to save the context of the current task from registers to memory, and then the task can be resumed by restoring the context from memory to the HAU registers.

The following steps can be performed to complete the HAU process of the suspended data blocks.

22.5.1. Transfer data by CPU

1. Stop the current data transmission and calculation. Wait for BUSY = 0, and wait for DIF = 1 when NWIF[3:0] is larger than 0 (do not wait for DIF = 1 when NWIF[3:0] is 0).Only when no data block is processing, users can save context.
2. Save the configuration. Save the content of HAU_INTEN, HAU_CFG, HAU_CTL, HAU_CTXS0 to HAU_CTXS37 (HAU_CTXS0 to HAU_CTXS53 when HMAC operation is processing) registers to memory.
3. Configure and process the new message.
4. Restore the process before. Restore the content from memory to HAU_INTEN, HAU_CFG and HAU_CTL registers.
5. Resume the message calculation. Set START bit of HAU_CTL register to 1, to restart a new message digest calculation.
6. Resume the previous core state. Restore the content from memory to HAU_CTXS0 ~ HAU_CTXS37 (HAU_CTXS0 ~ HAU_CTXS53 when HMAC operation is to be resumed) registers.
7. Continue the operation from where it suspended before.

22.5.2. Transfer data by DMA

1. Wait for BUSY = 0, then if the CCF bit of HAU_STAT register is set, the proceeding context switch is no longer need, otherwise wait for BUSY = 1 again.
2. Stop the current data transfer. Disable DMA data transmission, then clear DMAE bit of HAU_CTL register to disable DMA request.
3. Save the current configuration. Wait for BUSY = 0, then if the CCF bit of HAU_STAT register is set, the proceeding context switch is no longer need, otherwise save the content of HAU_INTEN, HAU_CFG, HAU_CTL, HAU_CTXS0 to HAU_CTXS37 (HAU_CTXS0 to HAU_CTXS53 when HMAC operation is processing) registers to memory.
4. Configure and process the new message.
5. Restore the process before. Restore the content from memory to HAU_INTEN, HAU_CFG and HAU_CTL registers.

6. Resume DMA channel transmission. Reconfigure the DMA channel to transfer data.
7. Resume the message calculation. Set START bit of HAU_CTL register to 1, to restart a new message digest calculation.
8. Resume the previous core state. Restore the content from memory to HAU_CTXS0 ~ HAU_CTXS37 (HAU_CTXS0 ~ HAU_CTXS53 when HMAC operation is to be resumed) registers.
9. Set DMAE bit of HAU_CTL register to 1, continue the operation from where it suspended before.

Note: If the value of NWIF[3:0] bits of HAU_CTL register is 0, it means the context switch occurs between two data blocks, at the time when the previous block is completely processed, and the next block has not been pushed into input FIFO, so there is no need to save and restore HAU_CTXS22 ~ HAU_CTXS37 registers.

22.6. HAU interrupt

There are two types of interrupt registers in HAU, which are both in HAU_STAT register. In HAU, the interrupt is used to indicate the situation of the input FIFO and the status of whether the digest calculation is completed.

Any of interrupts can be enabled or disabled by configuring the HAU interrupt enable register HAU_INTEN. Value 1 of the register bits enable the interrupts.

22.6.1. Input FIFO interrupt

When the processing of data pushed in the input FIFO is completed, then DIF is asserted. If input FIFO interrupt is enabled, when DIF is asserted, input FIFO interrupt will be asserted.

22.6.2. Calculation completion interrupt

When the digest calculation is finished, then CCF is asserted. If calculation completion interrupt is enabled, when CCF is asserted, calculation completion interrupt will be asserted.

22.7. Register definition

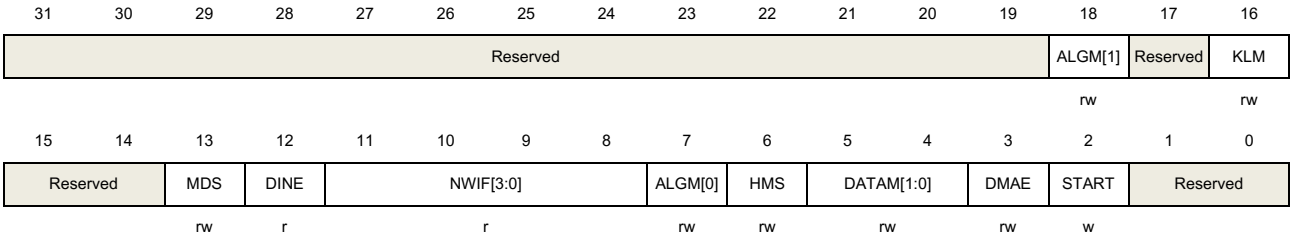
HAU base address: 0x4802 1400

22.7.1. Control register (HAU_CTL)

Address offset: 0x00

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:19	Reserved	Must be kept at reset value.
18	ALGM[1]	Algorithm selection bit 1
17	Reserved	Must be kept at reset value.
16	KLM	Key length mode 0: Key length <= 64 bytes 1: Key length > 64 bytes **Note:** This bit must be changed when no computation is processing.
15:14	Reserved	Must be kept at reset value.
13	MDS	Multiple DMA Selection Set this bit if hash message is large files and multiple DMA transfers are needed. 0: Single DMA transfers needed and CALEN bit is automatically set at the end of a DMA transfer 1: Multiple DMA transfers needed and CALEN bit is not automatically set at the end of a DMA transfer
12	DINE	DI register not empty 0: The DI register is empty 1: The DI register is not empty **Note:** This bit is cleared when START bit or CALEN bit is set as 1.
11:8	NWIF[3:0]	Number of words in the input FIFO **Note:** These bits are cleared when START bit set or a digest calculation starts (CALEN bit is set as 1, or DMA end of transfer)

7	ALGM[0]	Algorithm selection bit 0 This bit and bit 18 of CTL are written by software to select the SHA-1, SHA-224, SHA256 or the MD5 algorithm: 00: Select SHA-1 algorithm 01: Select MD5 algorithm 10: Select SHA224 algorithm 11: Select SHA256 algorithm
6	HMS	HAU mode selection, must be changed when no computation is processing 0: HASH mode selected 1: HMAC mode selected. If the key length is longer than 64 bytes, then KLM bit must also be set
5:4	DATAM[1:0]	Data type mode Defines the format of the data entered into the HAU_DI register: 00: No swapping. The data written to HAU_DI is direct write to FIFO without swapping. 01: Half-word swapping. The data written into HAU_DI need half-word swapping before write to FIFO. 10: Byte swapping. The data written into HAU_DI need byte swapping before write to FIFO. 11: Bit swapping. The data written into HAU_DI need bit swapping before write to FIFO.
3	DMAE	DMA enable 0: DMA disabled 1: DMA enabled **Note:** 1. This bit is cleared when transferring the last data of the message, but not cleared because of START. 2. When DMA is transferring, writing 0 to this bit will not stop the current transfer until the transfer is completed or START is set as 1.
2	START	Start the digest calculation 1: Start the digest of a new message 0: No effect **Note:** Reading this bit always returns 0.
1:0	Reserved	Must be kept at reset value.

22.7.2. Data input register (HAU_DI)

Address offset: 0x04
Reset value: 0x0000 0000

The data input register is used to transfer message with 512-bit blocks into the input FIFO for processing. Any new write operation to this register will be extended while the digest calculation is in process until it has been finished.

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

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

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

Bits	Fields	Descriptions
31:0	DI[31:0]	Message data input When write to these registers, the current content pushed to IN FIFO and new value updates. When read, returns the current content.

22.7.3. Configuration register (HAU_CFG)

Address offset: 0x08

Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:9	Reserved	Must be kept at reset value.
8	CALEN	Digest calculation enable 0: No calculation 1: Start data padding with VBL prepared previously. Start the calculation of the last digest **Note:** Reading this bit always returns 0.
7:5	Reserved	Must be kept at reset value.
4:0	VBL[4:0]	Valid bits length in the last word 0x00: All 32 bits of the last data written to HAU_DI after data swapping are valid 0x01: Only bit [31] of the last data written to HAU_DI after data swapping are valid 0x02: Only bits [31:30] of the last data written to HAU_DI after data swapping are valid 0x03: Only bits [31:29] of the last data written to HAU_DI after data swapping are valid ...

0x1F: Only bits [31:1] of the last data written to HAU_DI after data swapping are valid

Note: These bits must be configured before setting the CALEN bit.

22.7.4. Data output register (HAU_DO0..7)

Address offset: 0x0C
Reset value: 0x0000 0000

The data output registers are read only registers. They are used to receive results from the output FIFO. And they are reset by the START bit. Any read access when calculating will be extended until the calculation is completed.

In SHA-1 mode, HAU_DO0…4 are used.

In MD5 mode, HAU_DO0…3 are used.

In SHA-224 mode, HAU_DO0…6 are used.

In SHA-256 mode, HAU_DO0…7 are used.

HAU_DO0

Address offset: 0x0C and 0x310
Reset value: 0x0000 0000

These registers have to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DO0[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DO0[15:0]															
r															

HAU_DO1

Address offset: 0x10 and 0x314
Reset value: 0x0000 0000

These registers have to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DO1[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DO1[15:0]															
r															

HAU_DO2

Address offset: 0x14 and 0x318

Reset value: 0x0000 0000

These registers have to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DO2[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DO2[15:0]															
r															

HAU_DO3

Address offset: 0x18 and 0x31C

Reset value: 0x0000 0000

These registers have to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DO3[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DO3[15:0]															
r															

HAU_DO4

Address offset: 0x1C and 0x320

Reset value: 0x0000 0000

These registers have to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DO4[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DO4[15:0]															
r															

HAU_DO5

Address offset: 0x324

Reset value: 0x0000 0000

These registers have to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DO5[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DO5[15:0]															
r															

HAU_DO6

Address offset: 0x328

Reset value: 0x0000 0000

These registers have to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DO6[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DO6[15:0]															
r															

HAU_DO7

Address offset: 0x32C

Reset value: 0x0000 0000

These registers have to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DO7[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DO7[15:0]															
r															

Bits	Fields	Descriptions
31:0	DO0..7[31:0]	Message digest result of hash algorithm

22.7.5. Interrupt enable register (HAU_INTEN)

Address offset: 0x20

Reset value: 0x0000 0000

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

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

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

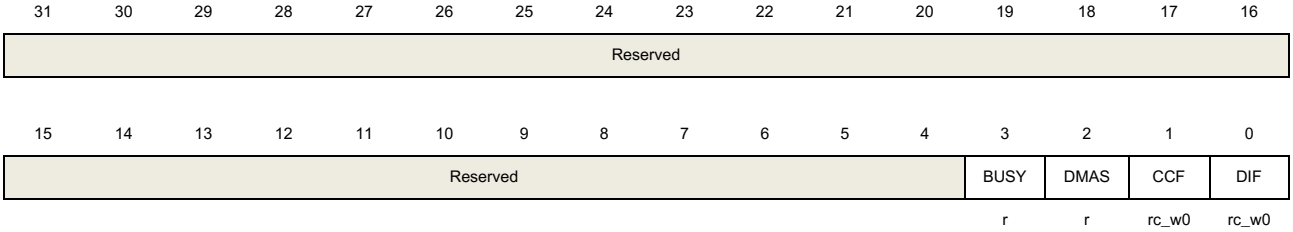
Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	CCIE	Calculation completion interrupt enable 0: Calculation completion interrupt is disabled 1: Calculation completion interrupt is enabled
0	DIIE	Data input interrupt enable 0: Data input interrupt is disabled 1: Data input interrupt is enabled

22.7.6. Status and flag register (HAU_STAT)

Address offset: 0x24

Reset value: 0x0000 0001

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



Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value.
3	BUSY	Busy bit 0: No processing 1: Data block is in process
2	DMAS	DMA status 0: DMA is disabled (DMAE = 0) and no transfer is processing 1: DMA is enabled (DMAE = 1) or a transfer is processing
1	CCF	Digest calculation completion flag 0: Digest calculation is not completed 1: Digest calculation is completed
0	DIF	Data input flag 0: A data is written to data input register 1: A data processing is completed (only the data in input FIFO will be processed)

22.7.7. Context switch register x (HAU_CTXSx) (x = 0...53)

Address offset: 0xF8 + 0x04 × x, (x = 0...53)

Reset value: 0x0000 0002 when x = 0, 0x0000 0000 when x = 1 to 53.

This register 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 complete internal status of the HAU 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.

23. Public Key Cryptographic Acceleration Unit (PKCAU)

23.1. Overview

Public key encryption is also called asymmetric encryption, asymmetric encryption algorithms use different keys for encryption and decryption. The Public Key Cryptographic Acceleration Unit (PKCAU) can accelerate RSA (Rivest, Shamir and Adleman), Diffie-Hellman (DH key exchange) and ECC (elliptic curve cryptography) in GF(p) (Galois domain). These operations are performed in the Montgomery domain to improve computational efficiency.

23.2. Characteristics

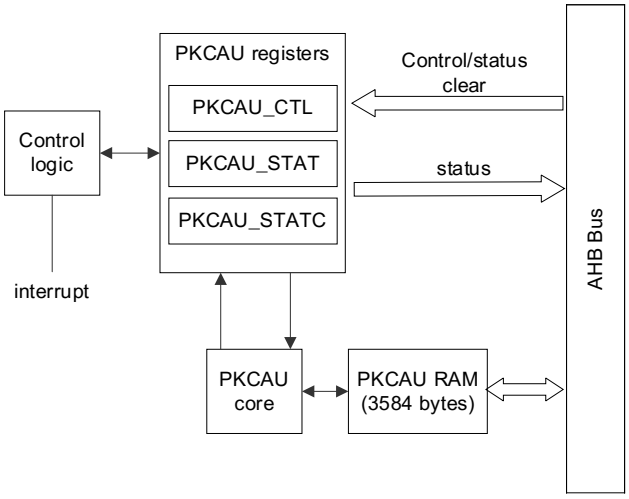
- Support RSA / DH algorithms with up to 3136 bits of operands.
- Support ECC algorithm with up to 640 bits of operands.
- RSA modular exponentiation, RSA CRT exponentiation.
- ECC scalar multiplication, check point on elliptic curve.
- ECDSA (Elliptic Curve Digital Signature Algorithm) signature and verification.
- Support Montgomery multiplication, accelerate RSA, DH and ECC operations.
- Embedded RAM of 3584 bytes.
- Conversion between the Montgomery domain and the natural domain.
- PKCAU is a 32 bit peripheral, only 32-bit access is supported.

23.3. Function overview

The Public Key Cryptographic Acceleration Unit (PKCAU) is used for accelerating RSA, DH and ECC operations in GF(p) domain. The PKCAU module contains PKCAU RAM, PKCAU core, and PKCAU registers. The PKCAU RAM is used to store the parameters required by the operation and save the calculation result after the calculation is completed.

Figure 23-1. PKCAU module block diagram below provides details on the internal configuration of the PKCAU interface.

Figure 23-1. PKCAU module block diagram



23.3.1. Operands

If the RSA operand size is ROS, the modulus length is ML, then the data size is ROS = (ML/32+1) words. If the ECC operand size is EOS, the prime modulus length is ML, then the data size is EOS = (ML / 32 + 1) words.

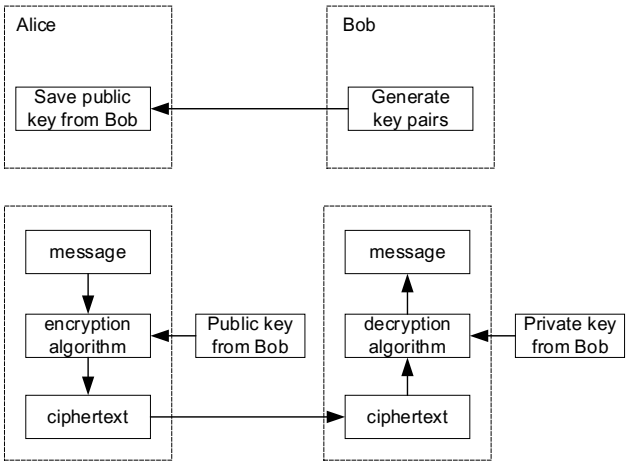
The PKCAU supports RSA / DH algorithms with up to 3136 bits (98 words) of operands and ECC algorithm with up to 640 bits (20 words) of operands. The maximum ROS is 99 words, and the maximum EOS is 21 words.

When writing the input parameters to the PKCAU RAM, a word 0x00000000 must be added. The PKCAU RAM is little-endian. For example, when writing the input parameter x_p of ECC P256 for ECC scalar multiplication to PKCAU RAM, the modulus length is 8 words, the lowest byte is stored at address offset 0x55C, and the highest byte at 0x57B (the highest word occupies 0x578-0x57B); address offset 0x57C stores 0x00000000.

23.3.2. RSA algorithm

RSA algorithm is a common public key cryptography algorithm and the most widely used asymmetric cryptography algorithm. The RSA algorithm flow is shown in ***Figure 23-2. Flow chart of RSA algorithm***.

Figure 23-2. Flow chart of RSA algorithm



A complete public key crypto system includes key pairs (public and private keys), encryption algorithms and decryption algorithms.

Generate key pairs of RSA

1. Select two large primes p and q (p != q);
2. Calculate n=p×q, n is the modulus of public and private keys;
3. Calculate L = Ø (n) = (p - 1)(q - 1), Ø (n) is euler function;
4. Select e, 1<e<L, e and L must be relatively prime;
5. Calculate d, 1<d<L and e * d mod L = 1.

The parameters show in *Table 23-1. Parameters of RSA algorithm* can be obtained by the above calculation.

Table 23-1. Parameters of RSA algorithm

Parameters	Description
n	modulus
e	public exponent
d	private exponent
(n,e)	public key
(n,d)	private key

RSA encryption

Bob generates a key pair that conforms to RSA algorithm standard, including public key and private key. Bob sends the public key to Alice, and holds the private key. Alice can encrypt the message m through the public key from Bob and obtain the ciphertext c. Then Alice sends the ciphertext to Bob. The ciphertext is $c = m^e \bmod n$.

RSA decryption

After receiving the ciphertext, Bob decrypts the ciphertext to get the plaintext by the private

key. The decryption process is $m = c^d \bmod n$.

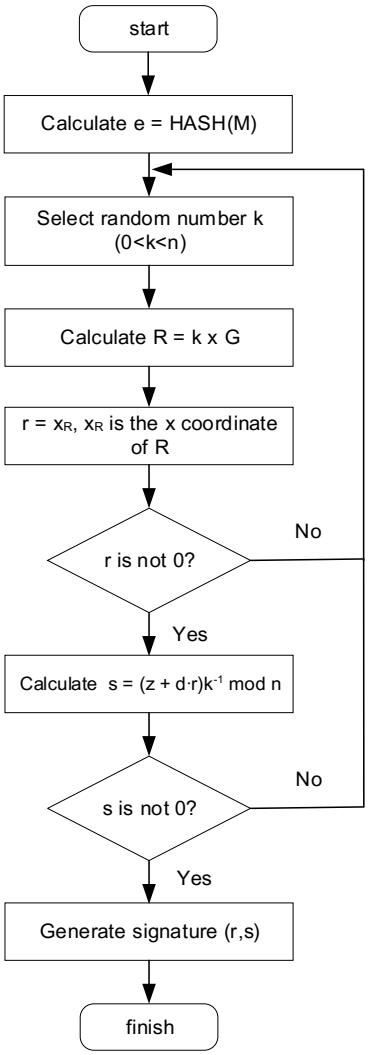
23.3.3. ECC algorithm

Suppose the message is M, d is the private key, G is the base point of the chosen elliptic curve, Q is a point of the chosen elliptic curve, with a prime order n. The hash function is HASH(), z is the Ln leftmost bits of HASH (M), where Ln is the bit length of the order n. The ECDSA sign and verification are detailed as follows:

ECDSA sign

The signature result of ECDSA consists of r and s. The process to generate ECDSA signature is shown in *Figure 23-3. Flow chart of ECDSA sign*.

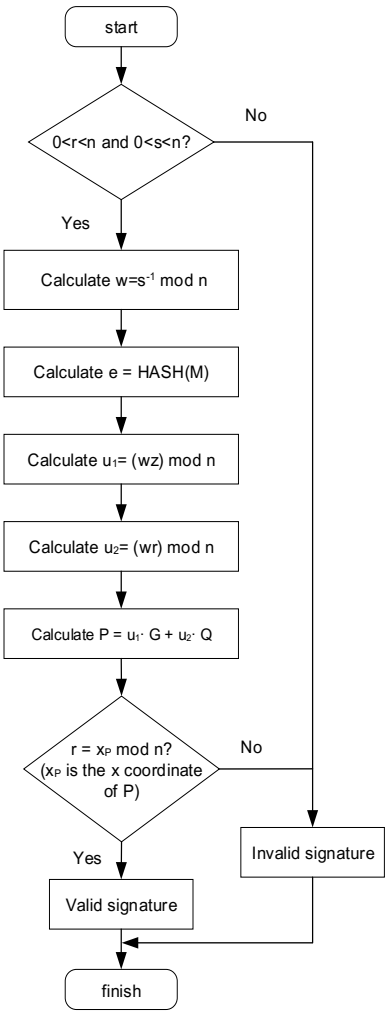
Figure 23-3. Flow chart of ECDSA sign



ECDSA verification

Before verifying the signature, be sure to get the signer's public key, message, and signature. The process to verify the ECDSA signature is shown in *Figure 23-4. Flow chart of ECDSA verification*.

Figure 23-4. Flow chart of ECDSA verification



Note: The HASH in the above diagram is the agreed cryptographic hash function.

23.3.4. Integer arithmetic operations

The integer arithmetic operation can be selected by configuring the MODSEL[5:0] in PKCAU_CTL register. The operation modes to be selected is shown in *Table 23-2. Integer arithmetic operations*.

Table 23-2. Integer arithmetic operations

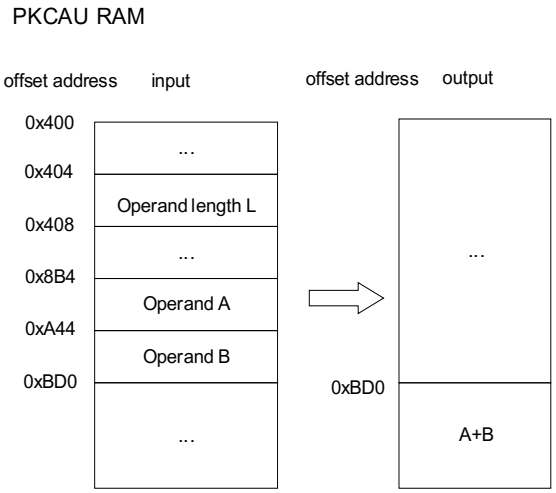
MODSEL[5:0]	Operation modes
000000	calculate Montgomery parameter and then modular exponentiation

MODSEL[5:0]	Operation modes
000001	calculate Montgomery parameter only
000010	modular exponentiation (the Montgomery parameter must be preloaded)
000111	RSA CRT exponentiation
001000	Modular inversion
001001	Arithmetic addition
001010	Arithmetic subtraction
001011	Arithmetic multiplication
001100	Arithmetic comparison
001101	Modular reduction
001110	Modular addition
001111	Modular subtraction
010000	Montgomery multiplication

Arithmetic addition

The arithmetic addition operation is selected by configuring MODSEL[5:0] in PKCAU_CTL register as "001001". The operation declaration is shown in ***Figure 23-5. Arithmetic addition***. The operation result is "result = A + B".

Figure 23-5. Arithmetic addition



$0<=A<2^{L}$, $0<=B<2^{L}$, $0<=result<2^{L+1}$, $0<L<=3136$.

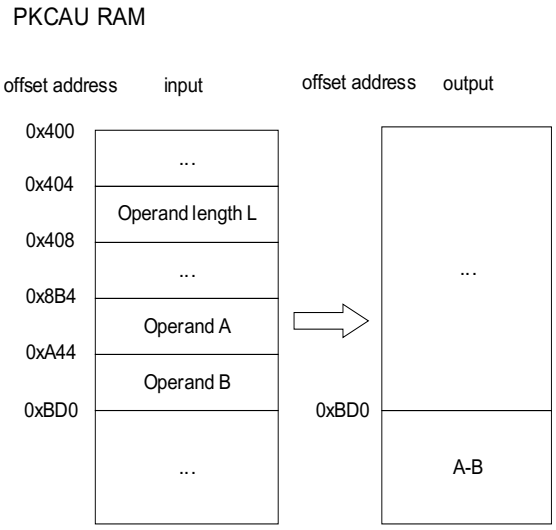
Arithmetic subtraction

The arithmetic subtraction operation is selected by configuring MODSEL[5:0] in PKCAU_CTL register as "001010". The operation declaration is shown in ***Figure 23-6. Arithmetic subtraction***.

If A>=B, the operation result is "result = A-B".

If A<B, the operation result is "result = $A-B+2^{L+ceil(L\%32)}$".

Figure 23-6. Arithmetic subtraction

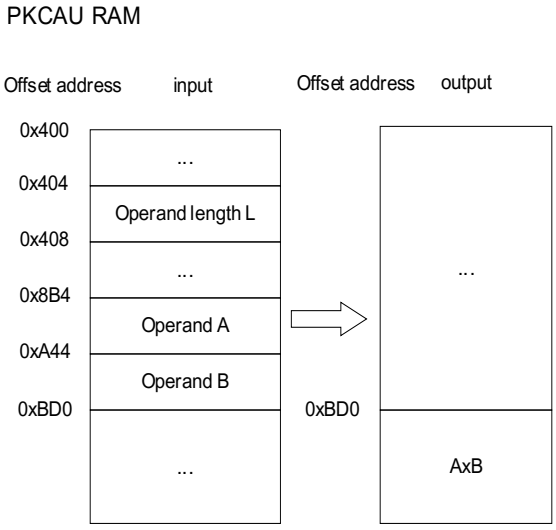


0<=A<2^L, 0<=B<2^L, 0<=result<2^L, 0<L<=3136.

Arithmetic multiplication

The arithmetic multiplication operation is selected by configuring MODSEL[5:0] in PKCAU_CTL register as "001011". The operation declaration is shown in ***Figure 23-7. Arithmetic multiplication***. The operation result is "result = A × B".

Figure 23-7. Arithmetic multiplication



0<=A<2^L, 0<=B<2^L, 0<=result<2^{2L}, 0<L<=3136.

Arithmetic comparison

The arithmetic comparison operation is selected by configuring MODSEL[5:0] in PKCAU_CTL register as "001100". The operation declaration is shown in ***Figure 23-8. Arithmetic***

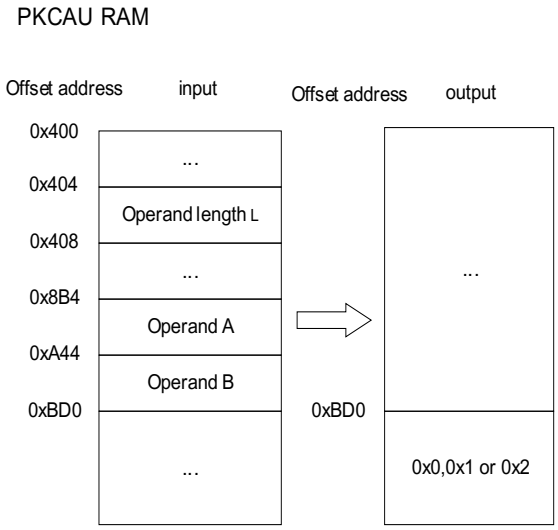
comparison.

If A=B, the operation result is "result = 0x0";

If A>B, the operation result is "result = 0x1";

If A<B, the operation result is "result = 0x2".

Figure 23-8. Arithmetic comparison

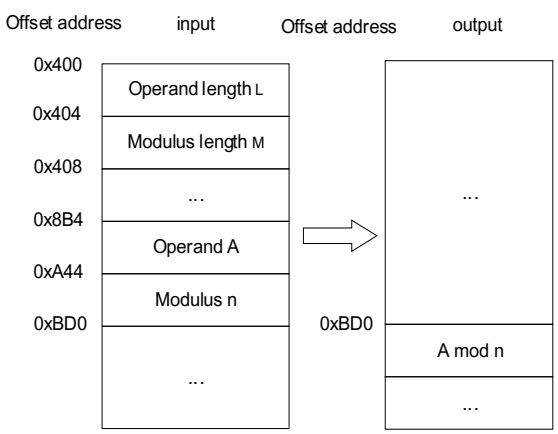


$0<=A<2^L$, $0<=B<2^L$, result = 0x0, 0x01, 0x2, $0<L<=3136$.

Modular reduction

The modular reduction operation is selected by configuring MODSEL[5:0] in PKCAU_CTL register as "001101". The operation declaration is shown in *Figure 23-9. Modular reduction*. The operation result is "result = A mod n".

Figure 23-9. Modular reduction

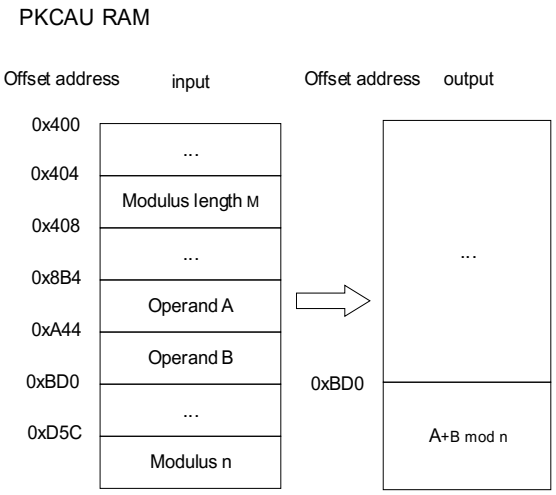


$0<L<=3136$, $0<M<=3136$, $0<=A<2^L$, $0<n<2^M$, $0<=result<n$.

Modular addition

The modular addition operation is selected by configuring MODSEL[5:0] in PKCAU_CTL register as "001110". The operation declaration is shown in *Figure 23-10. Modular addition*. The operation result is "result = A+B mod n".

Figure 23-10. Modular addition



$0<=A<n$, $0<=B<n$, $0<=result<n$, $0<n<2^M$, $0<M<=3136$.

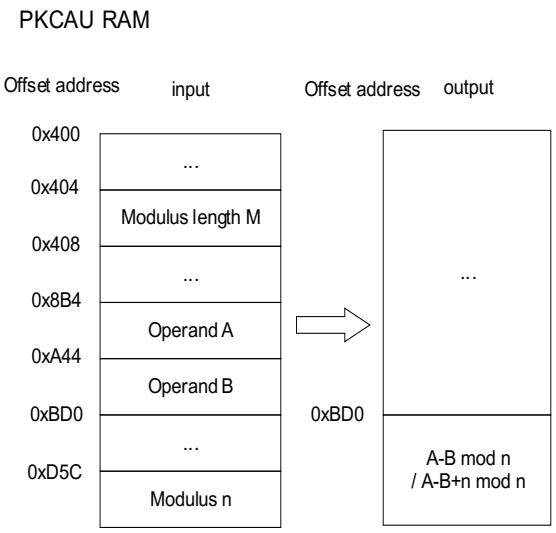
Modular subtraction

The modular subtraction operation is selected by configuring MODSEL[5:0] in PKCAU_CTL register as "001111". The operation declaration is shown in *Figure 23-11. Modular subtraction*.

If A>=B, the operation result is "result = A-B mod n";

If A<B, the operation result is "result = A-B+n mod n".

Figure 23-11. Modular subtraction



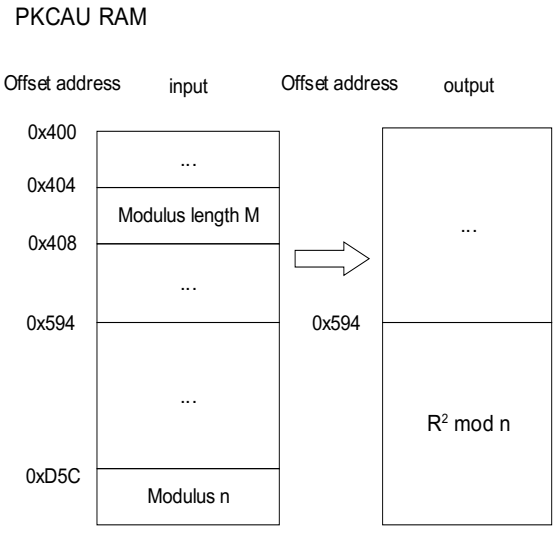
0<=A<n, 0<=B<n, 0<=result<n, 0<n<2^M, 0<M<=3136.

Montgomery parameter calculation

PKCAU needs to use the Montgomery parameter (R^2 mod n) to convert the operands to Montgomery residue system representation.

The Montgomery parameter calculation operation is selected by configuring MODSEL[5:0] in PKCAU_CTL register as "000001". The operation declaration is shown in ***Figure 23-12. Montgomery parameter calculation***.

Figure 23-12. Montgomery parameter calculation



0<M<=3136, 1<n<2^M(n must be odd integer).

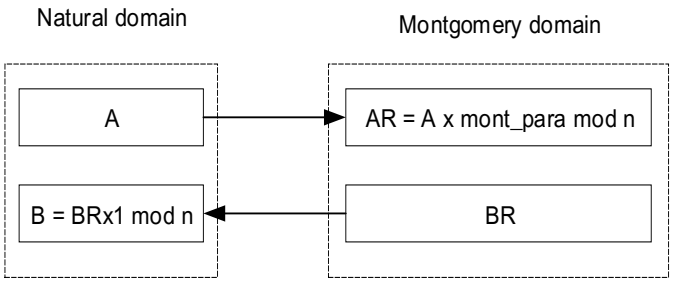
Montgomery multiplication

Suppose A, B and C are in natural domain. "x" function is Montgomery multiplication operation. The two main uses of this operation are as follows:

■ Mutual mapping between Montgomery domain and natural domain.

As is shown in ***Figure 23-13. Mutual mapping between Montgomery domain and natural domain***, if A is an integer in natural domain, the Montgomery parameter mont_para is R^2 mod n, the result AR = A x mont_para mod n is A in Montgomery domain. In turn, if BR is an integer in Montgomery domain, the calculation result B = BR x 1 mod n is in natural domain.

Figure 23-13. Mutual mapping between Montgomery domain and natural domain



■ Perform a modular multiplication operation A x B mod n.

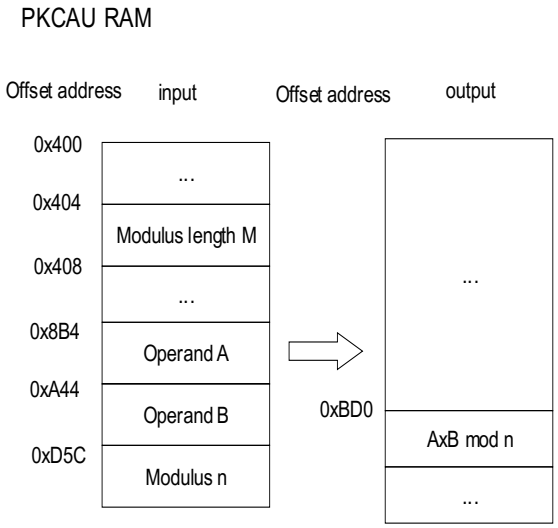
- Calculate Montgomery parameter mont_para = R^2 mod n.
- Calculate AR = A x mont_para mod n, the output is in Montgomery domain.
- Calculate AB = AR x B mod n, the output is in natural domain.

Multiple modular multiplication A x B x C mod n.

(1) Calculate Montgomery parameter mont_para = R^2 mod n.
(2) Calculate AR = A x mont_para mod n, the output is in Montgomery domain.
(3) Calculate BR = B x mont_para mod n, the output is in Montgomery domain.
(4) Calculate ABR = AR x BR mod n, the output is in Montgomery domain.
(5) Calculate CR = C x mont_para mod n, the output is in Montgomery domain.
(6) Calculate ABCR = ABR x CR mod n, the output is in Montgomery domain.
(7) Calculate ABC= ABCR x 1 mod n, the output is in natural domain.

The Montgomery multiplication operation is selected by configuring MODSEL[5:0] in PKCAU_CTL register as "010000". The operation declaration is shown in ***Figure 23-14. Montgomery multiplication***.

Figure 23-14. Montgomery multiplication



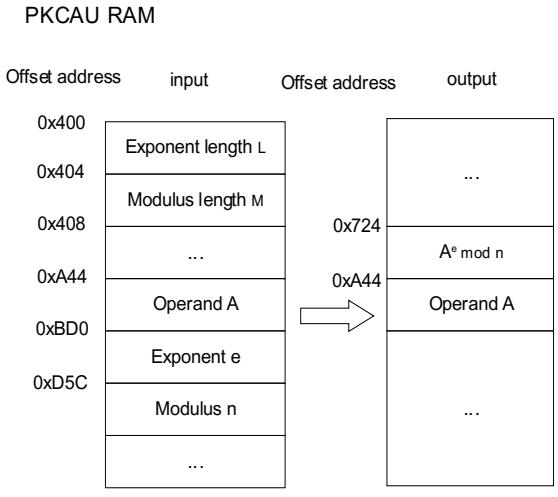
0<=A<n, 0<=B<n, 0<n<2^M, 0<M<=3136 (n must be odd integer).

Modular exponentiation

Normal mode

The Modular exponentiation of normal mode operation is selected by configuring MODSEL[5:0] in PKCAU_CTL register as "000000". The operation declaration is shown in *Figure 23-15. Modular exponentiation of normal mode*. The operation result is "result = A^e mod n".

Figure 23-15. Modular exponentiation of normal mode

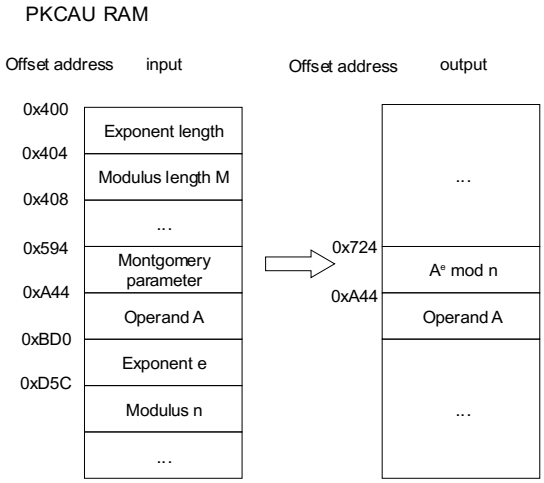


0<L<=3136, 0<M<=3136, 0<=A<n, 0<=e<2^L, 0<=result<n, 1<n<2^M (n must be odd integer).

Fast mode

The Modular exponentiation of fast mode operation is selected by configuring MODSEL[5:0] in PKCAU_CTL register as "000010". The operation declaration is shown in ***Figure 23-16. Modular exponentiation of fast mode***. The operation result is "result = A^e mod n".

Figure 23-16. Modular exponentiation of fast mode

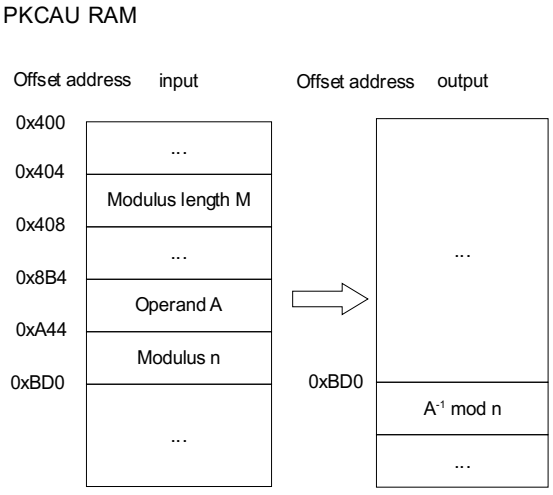


0<=A<n, 0<=e<n, 0<=result<n, 0<n<2^M, 0<M<=3136, 0<Montgomery parameter (R^2 mod n)<n.

Modular inversion

The modular inversion operation is selected by configuring MODSEL[5:0] in PKCAU_CTL register as "001000". The operation declaration is shown in ***Figure 23-17. Modular inversion***. The operation result is "result = A^{-1} mod n".

Figure 23-17. Modular inversion



0<A<n, 0<result<n, 0<n<2^M, 0<M<=3136.

Note:

1. If the modulus n is prime, for all values of A that satisfy the condition "1<=A<n", the modular inversion output is valid.
2. If the modulus n is not prime, only when the greatest common divisor of A and n is 1, the modular inversion output is valid.

RSA CRT exponentiation

The RSA CRT exponentiation operation is selected by configuring MODSEL[5:0] in PKCAU_CTL register as "000111".

P and q are part of the private key, and are primes

$d_P = d \bmod (p-1)$

$d_Q = d \bmod (q-1)$

$q_{inv} = q^{-1} \bmod p$

These parameters above allow the recipient to compute the exponentiation $m = A^d \pmod{pq}$ more efficiently as follows:

$m = A^d \pmod{pq}$

$m_1 = A^{dP} \bmod p$

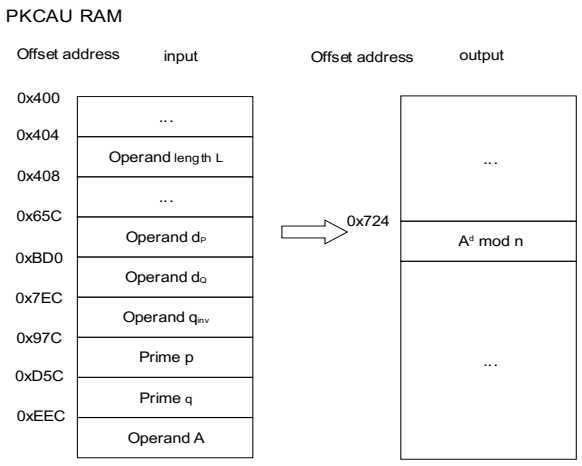
$m_2 = A^{dQ} \bmod p$

$h = q_{inv}(m_1 - m_2) \bmod p,\ m_1 > m_2$

$m = m_2 + hq$

The operation declaration is shown in *Figure 23-18. RSA CRT exponentiation*. The operation result is "result = A^d mod pq".

Figure 23-18. RSA CRT exponentiation



The range of parameters used by RSA CRT exponentiation operation is shown in *Table 23-3. Range of parameters used by RSA CRT exponentiation operation*.

Table 23-3. Range of parameters used by RSA CRT exponentiation operation

Parameters		Range
Input	Operand d_P	$0<=d_P<2^{L/2}$
	Operand d_Q	$0<=d_Q<2^{L/2}$
	Operand q_{inv}	$0<q_{inv}<2^{L/2}$
	Prime p	$0<p<2^{L/2}$
	Prime q	$0<q<2^{L/2}$
	Operand A	$0<=A<2^{L}$
Output	result = A^d mod pq	0<=result<pq

23.3.5. Elliptic curve operations in Fp domain

The Elliptic curve operation mode in Fp domain can be selected by configuring the MODSEL[5:0] in PKCAU_CTL register. The operation modes to be selected is shown in ***Table 23-4. Elliptic curve operations in Fp domain***.

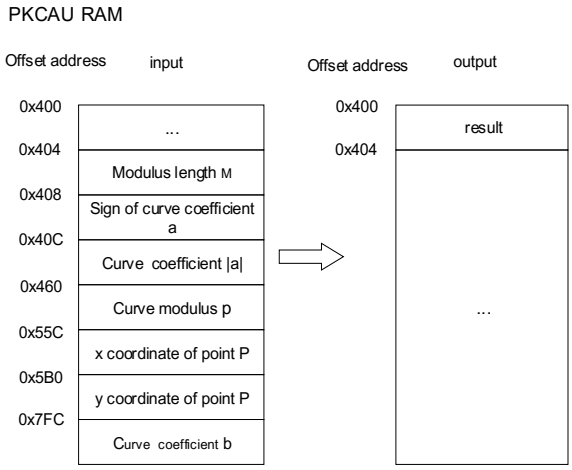
Table 23-4. Elliptic curve operations in Fp domain

MODSEL[5:0]	Operation modes
100000	Montgomery parameter computation then ECC scalar multiplication
100010	ECC scalar multiplication only (Montgomery parameter must be loaded first)
100100	ECDSA sign
100110	ECDSA verification
101000	Point on elliptic curve Fp check

Point on elliptic curve Fp check

The operation is used to check whether P(x,y) is on the $y^2= x^3+ax+b \bmod p$ in prime domain, where A and B are curve coefficients. The point on elliptic curve check operation in Fp domain is selected by configuring MODSEL[5:0] in PKCAU_CTL register as "101000". The operation declaration is shown in ***Figure 23-19. Point on elliptic curve Fp check***. If the operation result is 0, it indicates that point P is on the elliptic curve. Or else, it indicates that point P is not on the elliptic curve.

Figure 23-19. Point on elliptic curve Fp check



The range of parameters used by point on elliptic curve Fp check operation is shown in ***Table 23-5. Range of parameters used by point on elliptic curve Fp check***.

Table 23-5. Range of parameters used by point on elliptic curve Fp check

Parameters		Range
Input	Modulus length M	0<M<=640
	Sign of curve coefficient a	0x0: positive 0x1: negative
	Curve coefficient |a|	Absolute value |a|<p
	Curve coefficient b	Absolute value |b|<p
	Curve modulus p	Odd prime 0<p<=2^M
	x coordinate of point P	x<p
	y coordinate of point P	y<p

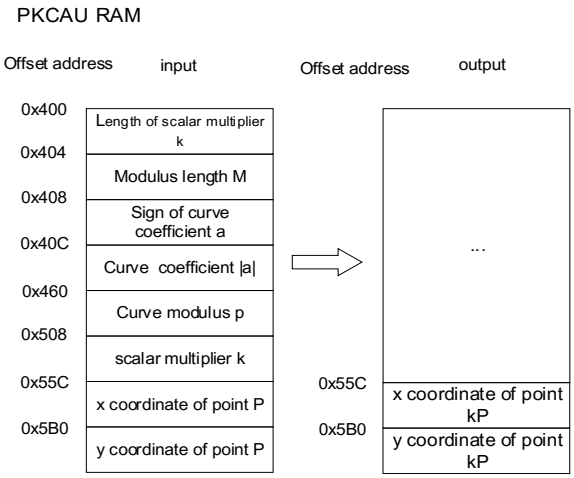
ECC scalar multiplication

ECC scalar multiplication operation is ak×P(x_P,y_P), where P is a point on the elliptic curve in the prime domain Fp. The operation result is also on the elliptic curve, or a point at infinity.

Normal mode

The ECC scalar multiplication operation in normal mode is selected by configuring MODSEL[5:0] in PKCAU_CTL register as "100000". The operation declaration is shown in ***Figure 23-20. ECC scalar multiplication of normal mode***.

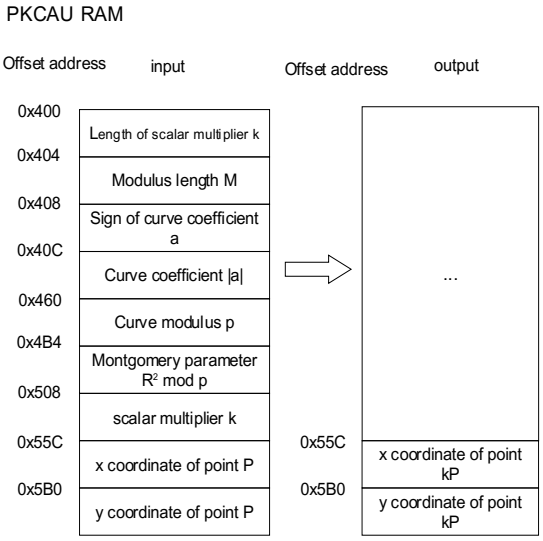
Figure 23-20. ECC scalar multiplication of normal mode



Fast mode

The ECC scalar multiplication operation in fast mode is selected by configuring MODSEL[5:0] in PKCAU_CTL register as "100010". The operation declaration is shown in ***Figure 23-21. ECC scalar multiplication of fast mode***.

Figure 23-21. ECC scalar multiplication of fast mode



The range of parameters used by ECC scalar multiplication operation is shown in ***Table 23-6. Range of parameters used by ECC scalar multiplication***.

Table 23-6. Range of parameters used by ECC scalar multiplication

Parameters		Range
input	Length of scalar multiplier k (LEN)	0<LEN<=640
	Modulus length M	0<M<=640
	Sign of curve coefficient a	0x0: positive 0x1: negative
	Curve coefficient |a|	Absolute value |a|<p
	Curve modulus p	Odd prime 0<p<=2^M
	scalar multiplier k	0<=k<2^{LEN} (k < n, and n is the curve prime order)
	x coordinate of point P	x_P<p
	y coordinate of point P	y_P<p
output	x coordinate of point kP	x<p
	y coordinate of point kP	y<p

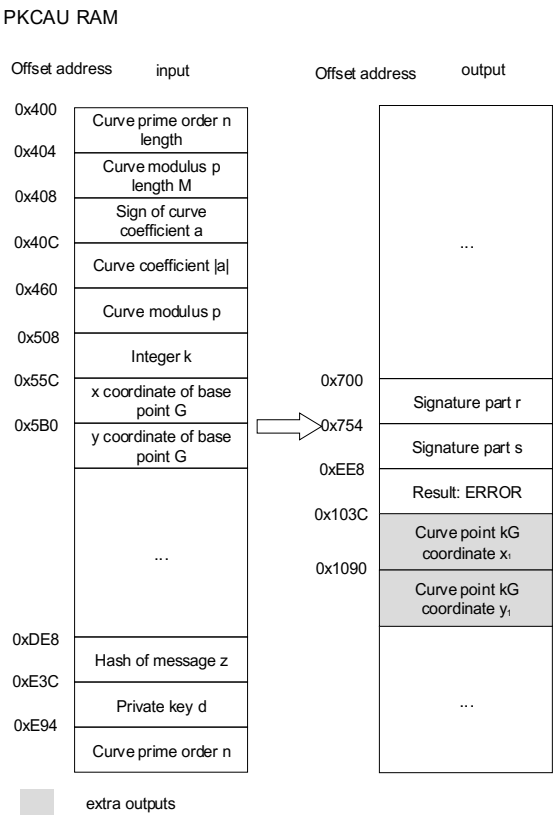
If k=0, the output is a point at infinity. When k is a multiple of curve prime order n, the output will also be a point at infinity. In this module, output is (0, 0) if the result is a point at infinity.

If k<0, the absolute value of k replaces k as the scalar multiplier for ECC scalar multiplication. After the computation is completed, -P = (x, -y) can be used to compute the final result of y.

ECDSA sign

The ECDSA sign operation is selected by configuring MODSEL[5:0] in PKCAU_CTL register as "100100". The operation declaration is shown in ***Figure 23-22. ECDSA sign***.

Figure 23-22. ECDSA sign



The range of parameters used ECDSA sign operation is shown in ***Table 23-7. Range of parameters used by ECDSA sign***.

Table 23-7. Range of parameters used by ECDSA sign

Parameters		Range
input	Curve prime order n length (LEN)	0<LEN<=640
	Curve modulus p length (M)	0<M<=640
	Sign of curve coefficient a	0x0: positive 0x1: negative
	Curve coefficient |a|	Absolute value |a|<p
	Curve modulus p	Odd prime 0<p<2^M
	Integer k	0<=k<2^{LEN}
	x coordinate of base point G	x<p
	y coordinate of base point G	y<p
	Hash of message z	z<2^{LEN}
	Private key d	positive integer d < n
	Curve prime order n	prime n<2^{LEN}

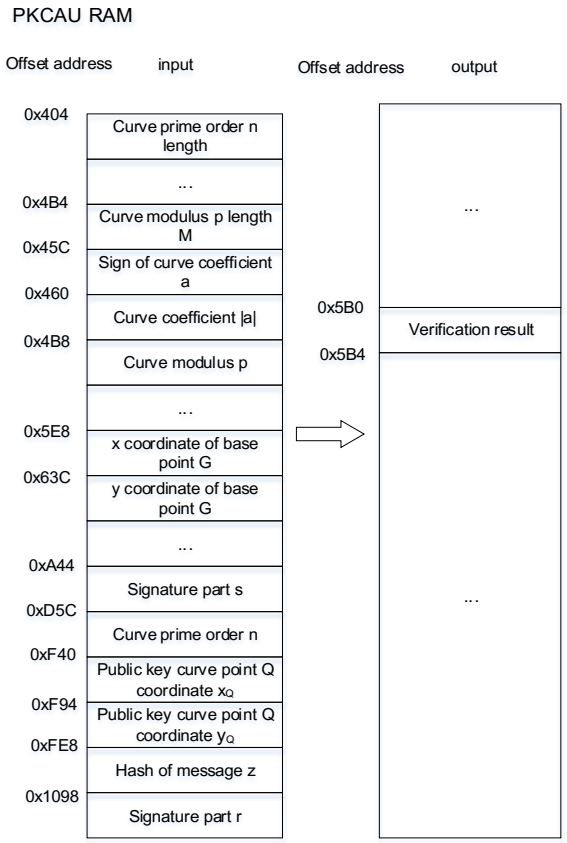
Parameters		Range
output	Signature part r	0<r<n
	Signature part s	0<s<n
	Signature result: ERROR	0x0: no error 0x1: Signature part r is 0 0x2: Signature part s is 0
	Curve point kG coordinate x_1	0<=x_1<n
	Curve point kG coordinate y_1	0<=y_1<n

If the error output is not 0, the content in PKCAU RAM will be cleared by hardware to avoid leaking private key related information.

ECDSA verification

The ECDSA verification operation is selected by configuring MODSEL[5:0] in PKCAU_CTL register as "100110". The operation declaration is shown in *Figure 23-23. ECDSA verification*.

Figure 23-23. ECDSA verification



The range of parameters used ECDSA verification operation is shown in *Table 23-8. Range*

of parameters used by ECDSA verification.

Table 23-8. Range of parameters used by ECDSA verification

Parameters		Range
input	Curve prime order n length (LEN)	0<LEN<=640
	Curve modulus p length (M)	0<M<=640
	Sign of curve coefficient a	0x0: positive 0x1: negative
	Curve coefficient |a|	Absolute value |a|<p
	Curve modulus p	Odd prime 0<p<2^M
	x coordinate of base point G	x<p
	y coordinate of base point G	y<p
	Public key curve point Q coordinate x_Q	x_Q<p
	Public key curve point Q coordinate y_Q	y_Q<p
	Signature part r	0<r<n
	Signature part s	0<s<n
	Hash of message z	z<2^{LEN}
	Curve prime order n	prime n<2^{LEN}
output	verification result	0x0: verification valid Not 0x0: verification invalid

23.3.6. PKCAU operation process

The PKCAU can be enabled by setting the PKCAUEN bit in PKCAU_CTL register. When the PKCAU is performing a calculation, the PKCAUEN should not be cleared, or else the ongoing operation is terminated and the content in PKCAU RAM will not be guaranteed.

When PKCAUEN is 0, the application can still access PKCAU RAM through the AHB interface.

Operation process in normal mode

The flows below applies to all operations listed in MODSEL[5:0] bits in the PKCAU_CTL register.

1. After system reset, the PKCAU RAM is to be erased from the beginning to the end. During this period, the BUSY bit in PKCAU_STAT register is set. All operation to PKCAU RAM should not carry out until the BUSY bit is 0.

2. Load the initial data into PKCAU RAM at offset address 0x400.
3. Specify the operation to be performed in MODSEL[5:0] bits in PKCAU_CTL register, then set START bit in PKCAU_CTL register.
4. Wait for the ENDF bit set in PKCAU_STAT register.
5. Read calculation result from the PKCAU RAM, then clear the ENDF bit by setting the ENDFC bit in PKCAU_STATC register.

Operation process in fast mode

The fast mode computes the Montgomery parameter only once when calculating many operations with the same modulus. When the operation is performed, the pre-calculated Montgomery parameter is loaded to perform the calculation.

The flow of operation in fast mode is as follows:

1. Load the initial data into PKCAU RAM at offset address 0x400.
2. Select the Montgomery parameter calculation mode by configuring the MODSEL[5:0] as "000001", then set START bit in PKCAU_CTL register.
3. Wait for the ENDF bit set in PKCAU_STAT register.
4. Read the Montgomery parameter from the PKCAU RAM, then clear the ENDF bit by setting the ENDFC bit in PKCAU_STATC register.
5. Load the initial data and Montgomery parameter into PKCAU RAM.
6. Specify the operation to be performed in MODSEL[5:0] bits in PKCAU_CTL register, then set START bit in PKCAU_CTL register.
7. Wait for the ENDF bit set in PKCAU_STAT register.
8. Read calculation result from the PKCAU RAM, then clear the ENDF bit by setting the ENDFC bit in PKCAU_STATC register.

23.3.7. Processing times

The following tables summarize the PKCAU computation times, expressed in clock cycles.

Table 23-9. Modular exponentiation computation times

Exponent length (in bits)	Mode	Operand length (in bits)		
		1024	2048	3072
1024	Normal	6780000	-	-
	Fast	6701000	-	-
	CRT	1853000		
2048	Normal	-	52196000	-
	Fast	-	51910000	-

Exponent length (in bits)	Mode	Operand length (in bits)		
		1024	2048	3072
	CRT	-	13651000	
3072	Normal	-	-	182783000
	Fast	-	-	181953000
	CRT	-	-	44905000

Table 23-10. ECC scalar multiplication computation times

Mode	Modulus length (in bits)					
	160	192	256	320	384	512
Normal	626000	951000	1997000	3617000	5762000	13134000
Fast	623000	946000	1990000	3607000	5749000	13111000

Table 23-11. ECDSA signature average computation times

Modulus length (in bits)					
160	192	256	320	384	512
634000	966000	2029000	3648000	5833000	13177000

Table 23-12. ECDSA verification average computation times

Modulus length (in bits)					
160	192	256	320	384	512
1261000	1901000	3997000	7225000	11477000	26287000

Table 23-13. Montgomery parameters average computation times

Modulus length (in bits)								
160	192	256	320	384	512	1024	2048	3072
3873	4658	7109	10330	14526	22301	79116	284359	626909

23.3.8. Status, errors and interrupts

There are several status and error flags in PKCAU, and interrupt may be asserted from these flags by setting some register bits.

■ Access address error (ADDRERR)

When the accessed address exceeds the expected range of PKCAU RAM, the address error flag ADDRERR bit in PKCAU_STAT register will be set. If the ADDRERRIE bit is set in PKCAU_CTL register, an interrupt will be generated. The ADDRERR bit can be cleared by setting the ADDRERRC bit in PKCAU_STATC register.

■ RAM error (RAMERR)

The PKCAU core is using the PKCAU RAM while an AHB access to the PKCAU RAM occurs, the RAM error flag RAMERR bit in PKCAU_STAT register will be set. If it is an AHB read, it returns 0, an AHB write will be ignored. If the RAMERRIE bit is set in PKCAU_CTL register, an interrupt will be generated. The RAMERR bit can be cleared by setting the RAMERRC bit in PKCAU_STATC register.

■ End of operation flag (ENDF)

When the operation specified in MODSEL[5:0] bits in the PKCAU_CTL register is completed, the ENDF bit will be set. If the ENDIE bit in PKCAU_CTL register is set, an interrupt will be generated. The ENDF bit can be cleared by setting the ENDFC bit in PKCAU_STATC register. The ENDF bit can also be cleared automatically if another operation is carried out by set the START bit.

The PKCAU interrupt events and flags are listed in *Table 23-14. PKCAU interrupt requests*.

Table 23-14. PKCAU interrupt requests

Interrupt event	Event flag	Flag clear	Enable control bit
Access address error	ADDRERR	ADDRERRC	ADDRERRIE
RAM error	RAMERR	RAMERRC	RAMERRIE
Operation end flag	ENDF	ENDFC	ENDIE

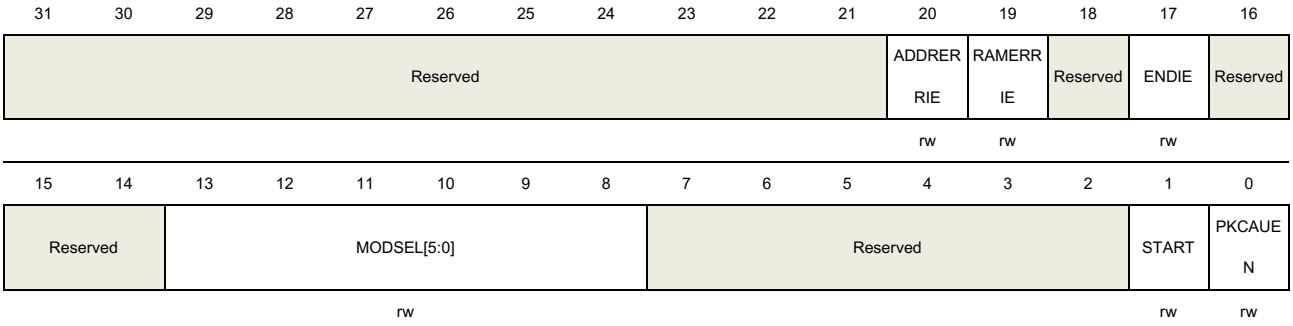
23.4. Register definition

PKCAU base address: 0x4800 0000

23.4.1. Control register (PKCAU_CTL)

Address offset: 0x00
Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:21	Reserved	Must be kept at reset value.
20	ADDRERRIE	Address error interrupt enable 0: Address error interrupt disable 1: Address error interrupt enable
19	RAMERRIE	RAM error interrupt enable 0: RAM error interrupt disable 1: RAM error interrupt enable
18	Reserved	Must be kept at reset value.
17	ENDIE	End of operation interrupt enable 0: End of operation interrupt disable 1: End of operation interrupt enable
16:14	Reserved	Must be kept at reset value.
13:8	MODSEL[5:0]	PKCAU operation mode selection 000000: Montgomery parameter computation then modular exponentiation 000001: Montgomery parameter computation only 000010: Modular exponentiation only (Montgomery parameter must be loaded first) 000111: RSA CRT exponentiation 001000: Modular inversion 001001: Arithmetic addition

001010: Arithmetic subtraction
001011: Arithmetic multiplication
001100: Arithmetic comparison
001101: Modular reduction
001110: Modular addition
001111: Modular subtraction
010000: Montgomery multiplication
100000: Montgomery parameter computation then ECC scalar multiplication
100010: ECC scalar multiplication only (Montgomery parameter must be loaded first)
100100: ECDSA sign
100110: ECDSA verification
101000: Point on elliptic curve Fp check
Other values are reserved.

7:2	Reserved	Must be kept at reset value.
1	START	PKCAU starts operation This bit is set by software to start the PKCAU operation which is specified in MODSEL[5:0] in PKCAU_CTL register. When the BUSY bit in PKCAU_STAT register is 1, writing 1 to this bit will be ignored.
0	PKCAUEN	PKCAU enable 0: PKCAU disable 1: PKCAU enable

23.4.2. Status register (PKCAU_STAT)

Address offset: 0x04
Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											ADDRERR	RAMERR	Reserved	ENDF	BUSY
											r	r		r	r

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

Bits	Fields	Descriptions
31:21	Reserved	Must be kept at reset value.
20	ADDRERR	Address error. 0: No address error.

		1: The accessed address exceeds the expected range of PKCAU RAM, an address error occurs.
19	RAMERR	PKCAU RAM error 0: No PKCAU RAM error. 1: When the PKCAU core is using the RAM, AHB accesses the PKCAU RAM, a PKCAU RAM error occurs.
18	Reserved	Must be kept at reset value.
17	ENDF	End of PKCAU operation When the operation executed completely, this bit is set by hardware.
16	BUSY	Busy flag When the START bit in PKCAU_CTL register is set, this bit is set by hardware. When the PKCAU operation is completed, this bit is cleared by hardware.
15:0	Reserved	Must be kept at reset value.

23.4.3. Status clear register (PKCAU_STATC)

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											ADDRERRC	RAMERRC	Reserved	ENDFC	Reserved
											w	w		w	

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	ADDRERRC	Address error flag clear. Software can clear the ADDRERR bit in PKCAU_STAT by writing 1 to this bit.
19	RAMERRC	PKCAU RAM error flag clear. Software can clear the RAMERR bit in PKCAU_STAT by writing 1 to this bit.
18	Reserved	Must be kept at reset value.
17	ENDFC	End of PKCAU operation flag clear. Software can clear the ENDF bit in PKCAU_STAT by writing 1 to this bit.

16:0	Reserved	Must be kept at reset value.

24. Real-time decryption (RTDEC)

24.1. Overview

The real-time decryption (RTDEC) allows to decrypt in real-time according to information of the read request address. RTDEC can configure four independent and different encrypted areas. And each area has the option of execute-only or execute-never enforcement to choose.

For a good real-time performance, RTDEC uses the counter (CTR) mode of AES-128. Since RTDEC using AES-128 in counter mode, the whole area has to be re-encrypted with an updated cryptographic context (key or initialization vector) when the data or code of one encrypted area is changed. This feature makes RTDEC only suitable for decrypting read-only content, like that stored in external flash.

Note: It is necessary to access the external read-only memory using the memory map mode when RTDEC is used with Octal-SPI (OSPI). It is important to know that CPU memories and RTDEC follow little endian notation, and AES hardware accelerator follows big endian notation.

24.2. Characteristics

- Software configurable encrypted areas up to 4.
- Granularity is 4096 bytes in RTDEC programmed areas.
- Area configuration can be locked.
- Optional decryption modes: execute-only or execute-never.
- Every area can be configured the independent 128-bits key, 16-bits area firmware version, and 64-bits application-defined nonce. If the user performed an encryption, one or more of those should be modified.
- Confidentiality and completeness protection for encryption keys.
  - 128-bits key registers are write-only, with software locking mechanism
  - 8-bits CRC is calculated automatically by hardware, and it's used as the public key information
- The real-time decryption when OSPI memory-mapped read operations.
  - Use of AES-128 in CTR mode
  - Support key stream FIFO with depth 4
  - Support various read size
  - Decryption / encryption with physical address of the reads
- Support for GD32 OSPI pre-fetching mechanism.

24.3. Function overview

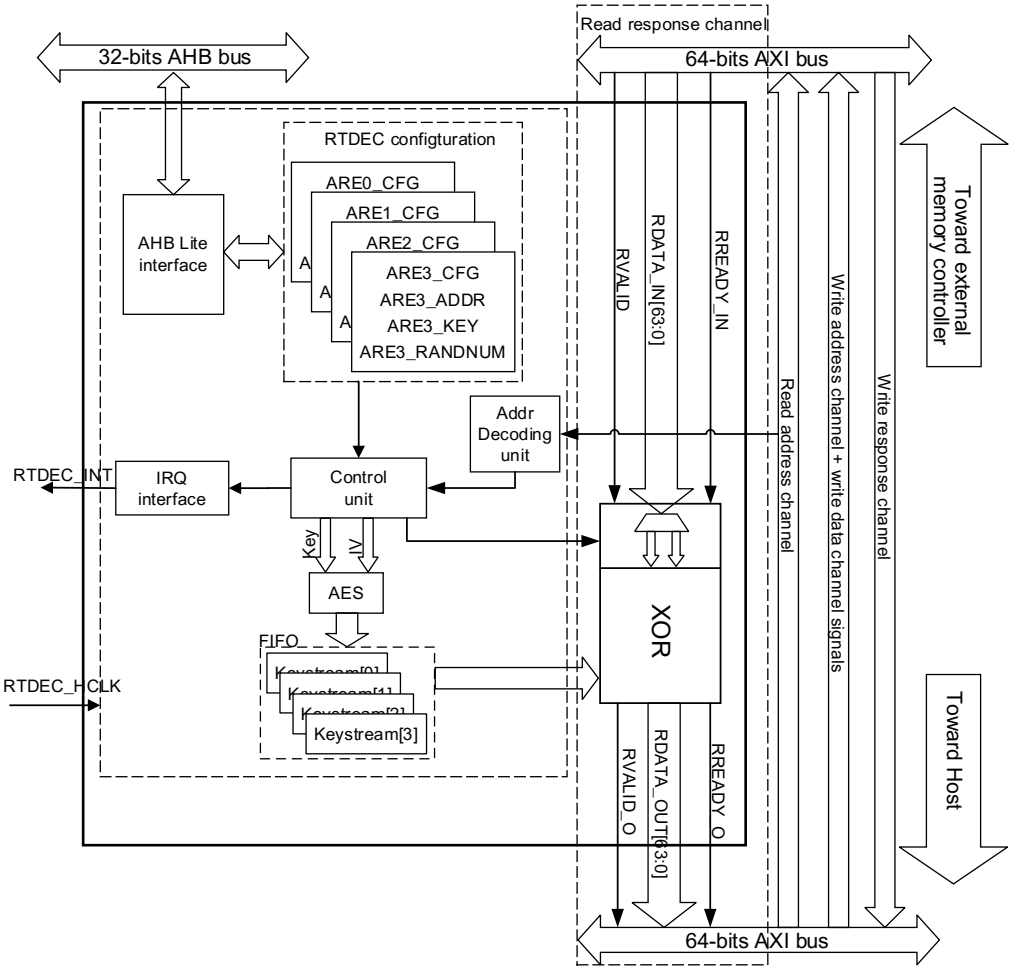
The RTDEC (real-time decryption) module allows real-time decryption of the Arm® AXI or AHB

traffic based on the read request address information. Original purpose of RTDEC is to protect the confidentiality of read-only firmware libraries stored in external SPI NOR Flash devices. The RTDEC performs real-time decryption during OSPI memory-mapped read operation.

24.3.1. RTDEC block diagram

Figure 24-1. RTDEC block diagram shows the block diagram of the RTDEC.

Figure 24-1. RTDEC block diagram



Note: RTDEC_HCLK is RTDEC clock input from AHB clock, RTDEC_INT is RTDEC global interrupt.

24.3.2. RTDEC real-time decryption introduction

RTDEC typical usage

RTDEC fundamental purpose is to protect the secrecy of the execute-only code. The code can be the execute-only firmware library is executed from external flash memory. RTDEC can also protect "code + data library" (which is read only), and data library (which is read only and

execute-never) stored in external flash memory.

In order to protect the completeness of the decryption keys and also to protect other RTDEC configurations against software denial of services attacks, RTDEC provides the configuration register locking mechanism.

It is necessary to access the external read-only memory using the memory map mode when RTDEC is used with OSPI.

RTDEC principle

The principle of RTDEC is to analyze all read address channel transactions between the MCU and a peripheral, the read address channel transactions is on the AXI interconnect bus.

The control logic will trigger a key stream computation if the read request is within one of the four areas programmed in RTDEC. The key stream is computed by RTDEC using AES-128 in counter mode. Any accesses outside the RTDEC programmed areas belong to a non-encrypted area. Granularity for the area determination is 4KB. That is, size of the RTDEC programmed areas is an integer multiple of 4KB.

Each RTDEC areas can be programmed by registers AREx_CFG, AREx_SADDR, AREx_EADDR, AREx_NONCE and AREx_KEY, where x=0 to 3. It is could be configured whether the content of the RTDEC area is code (execute-only), data (execute-never) or both through MODE[1:0] (in the register AREx_CFG).

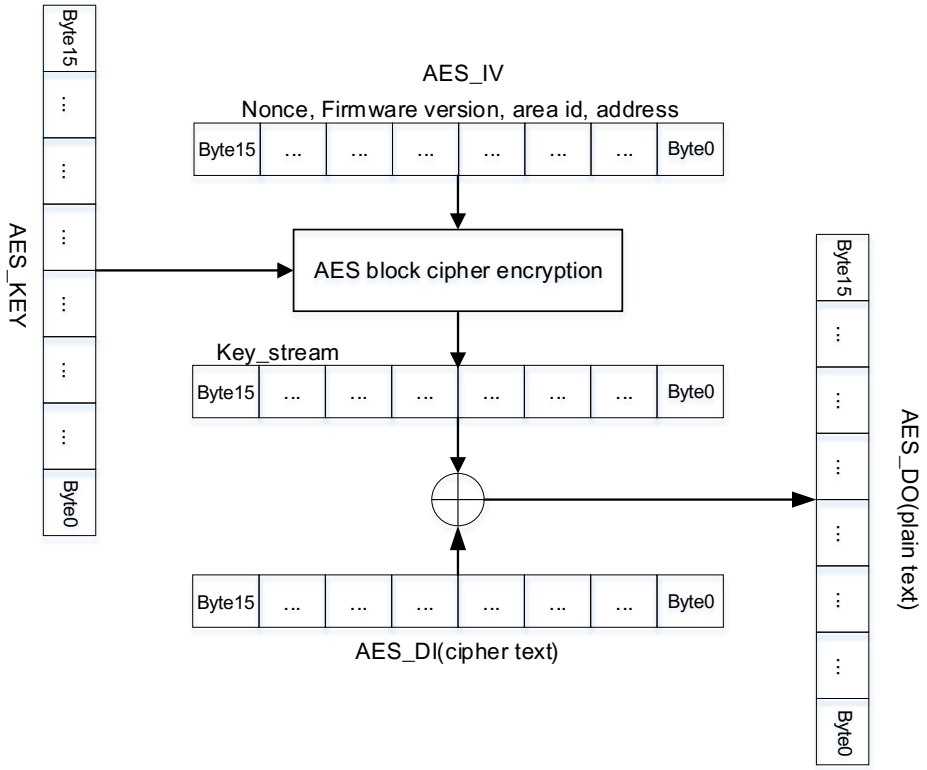
Note: RTDEC does not prevent overlapping of areas but it is an invalid programming, and the user should try to avoid areas overlapping in application software.

When RTDEC decrypts incremental or wrap bursts they should not cross the 4096-byte aligned address boundaries. RTDEC cannot decrypt part data of an AXI transaction.

24.3.3. Decryption with AES-128 in counter mode

Figure 24-2. Decryption flow with AES-128 in CTR mode shows RTDEC how to use Advanced Encryption Standard (AES) algorithm in counter mode. AES is announced by Federal Information Processing Standards Publication 197, November 26, 2001. And it is specified by NIST in *Special Publication 800-38A, Recommendation for Block Cipher Modes of Operation*.

Figure 24-2. Decryption flow with AES-128 in CTR mode



During decryption, a special key steam information will be computed by using AES block cipher for every 128-bit data block. The AES_IV and AES_KEY are defined as below:

1. The initialization vector AES_IV[127:0] = RTDEC_AREx_NONCE[63:0] || 16'b 0000 0000 0000 0000 || RTDEC_AREx_CFG[31:16] || 2'b 00 || x || ReadAddress[31:4].
2. The key material AES_KEY[127:0] = RTDEC_AREx_KEY3[31:0] || RTDEC_AREx_KEY2[31:0] || RTDEC_AREx_KEY1[31:0] || RTDEC_AREx_KEY0[31:0].

Note: Above x (x = 2'b 00 to 2'b 11) is the Area ID of the selected encrypted area.

After the 128-bit key stream is computed, the 128-bit clear text is produced by 128-bit key stream XORed with 128-bit cipher text data.

The 128-bit AES_DI and AES_DO data blocks are defined following the rule below: AES_Dx[127:0] = AXI_word(@+0x8)[63:0] || AXI_word(@)[63:0]. Where @ is the hexadecimal address, and it is used to compute the key stream.

When the read request is within a non-encrypted area, or in an encrypted area but the decryption is not enabled, the 128-bit AXI data is not changed.

24.3.4. RTDEC configuration

RTDEC initialization process

The trusted initialization of RTDEC registers is an important aspect when use RTDEC.

Because it involves secret keys initialization and critical options like MODE[1:0] bits initialization.

There are two solutions of RTDEC trusted initialization are introduced here below.

Note: Those sequences are suggested for production code. Because it is not always recommended to set bit ARE_K_LK or ARE_CFG_LK (lock the key or the area configuration) during code development.

Writes to area configuration registers are effective when the bit ARE_CFG_LK is 0, even if the bit ARE_EN is 1.

One key for all areas initialization solution

In this solution, the secret key used to decrypt the RTDEC four protected areas can belong to one entity.

The follow configuration sequence of RTDEC is recommended:

- Write the correct MODE[1:0] value in RTDEC_AREx_CFG register (where x=0 to 3).
- Program RTDEC_AREx_KEY0…3 registers (where x=0 to 3) using the sequence described in ARE_K_CRC (make sure to have a valid CRC). Note that key registers are write only, read from them is illegal.
- Check the key CRC. If it's ok, set bit ARE_K_LK in RTDEC_AREx_CFG register (where x=0 to 3). ARE_K_LK bit cannot be cleared, that is the key registers in this area x are no more changed.
- While you have an area x to decrypt do it. It is not necessarily have to be performed by the entity (that owns the decryption keys) when this work does.
  - Check if the key CRC matches to the encrypted binary stored in the area.
  - Configure the detailed information matching to this binary. The information are nonce, firmware version, start and end address.
  - Set bit ARE_EN in RTDEC_AREx_CFG (where x=0 to 3) to enable decryption of this area.
  - Set bit ARE_CFG_LK in RTDEC_AREx_CFG (where x=0 to 3). This bit cannot be cleared, that is the area configuration is no more changed.

Note: For a given area, the key registers and associated CRC are cleared by hardware when MODE[1:0] bits are changed. So MODE[1:0] bits must be written firstly before other operating, and after programming RTDEC_AREx_KEY0…3 (where x=0 to 3) registers the MODE[1:0] bits must not be modified.

One key for one area initialization solution

In this solution, the secret key used to decrypt the RTDEC one (or more) protected areas can belong to one entity.

The follow configuration sequence of RTDEC is recommended.

While you have an area x to decrypt do it, such as x is 0. It is have to be performed by the

entity (that owns the corresponding decryption key) when this work does.

- Write the correct MODE[1:0] value in RTDEC_ARE0_CFG register.
- Program RTDEC_ARE0_KEY0…3 registers using the sequence described in ARE_K_CRC (read from area configuration register to have a valid CRC). Note that key registers are write only, read from them is illegal.
- Check the key CRC. If it's ok, set bit ARE_K_LK in RTDEC_ARE0_CFG register. This bit cannot be cleared, that is the key registers are no more changed.
- Configure the detailed information matching to the protected firmware. The information are nonce, firmware version, start and end address.
- Set bit ARE_EN in RTDEC_ARE0_CFG to enable decryption of this area.
- Set bit ARE_CFG_LK in RTDEC_ARE0_CFG. This bit cannot be cleared, that is the area configuration is no more changed.

Note: For a given area, the key registers and associated CRC are cleared by hardware when MODE[1:0] bits are changed. So MODE[1:0] bits must be written firstly before other operating, and after programming RTDEC_ARE0_KEY0…3 registers the MODE[1:0] bits must not be modified.

Reset of RTDEC

The correct key loading sequence described in ***RTDEC initialization process*** must be performed when each time RTDEC is reset (in this case, bit ARE_K_CRC of RTDEC_AREx_CFG is 0). When the RTDEC is reset by hardware we suggested that user's application software to guarantee the correct key loading sequence.

24.3.5. RTDEC error management

RTDEC could manages below errors automatically,

- Illegal read to RTDEC_AREx_KEY0…3 registers. Where x is 0 to 3.
- While bit ARE_CFG_LK or ARE_K_LK in RTDEC_AREx_CFG is set, illegal write to RTDEC_AREx_KEY0…3 registers. Where x is 0 to 3.
- While bit ARE_CFG_LK in RTDEC_AREx_CFG is set, illegal write to RTDEC_AREx_CFG, RTDEC_AREx_SADDR, RTDEC_AREx_EADDR or RTDEC_AREx_NONCE0/1 registers. Where x is 0 to 3.
- While MODE[1:0] is 2'b00, illegal read to an execute-only area. While MODE[1:0] is 2'b01, illegal execution request to an execute-never area. These illegal requests return 0, without bus error.
- Key error: When an abort event happened, the key register will be cleared. At this time if there are some read requests, these requests will return 0, without bus error. The abort event maybe tamper detection, unauthorized debug connection, untrusted boot or SPC level low to no protection demotion.

If the bit SECEIE, ECONEIE or KEIE is set, one or more of above errors occurs will generate an interrupt. For details refer to ***RTDEC interrupts***.

Note: After a key error occurs, a system reset must be performed first, followed by reconfiguring the RTDEC, in order for the RTDEC to function normally again.

24.4. RTDEC interrupts

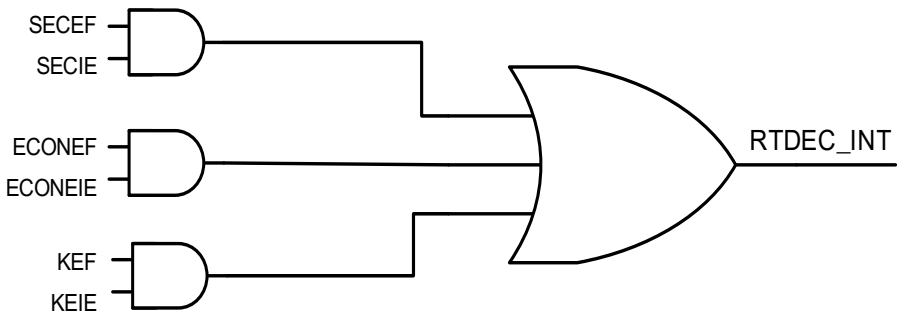
RTDEC can generate three independent maskable interrupt events, the interrupt flag is set after following security events:

- Illegal read or write access to keys (SECEF flag), refer to ***Interrupt flag register (RTDEC_INTF)***.
- Illegal write to an area's configuration register while ARE_CFG_LK=1 (SECEF flag), refer to ***Interrupt flag register (RTDEC_INTF)***.
- While MODE[1:0] is 2'b00, read access to an execute-only area. While MODE[1:0] is 2'b01, execute access to a execute-never data area. Both are triggering the ECONEF flag.
- Key error (encrypted area read as zero), triggering the KEF flag. Refer to ***Interrupt flag register (RTDEC_INTF)***.

All of the interrupt events are Ored together before being sent to the interrupt controller, so the RTDEC can only generate a single interrupt request to the controller at any given time.

Below block diagram illustrates the RTDEC module interrupt connection

Figure 24-3. RTDEC interrupt mapping diagram



By setting corresponding SECEIE, ECONEIE or KEIE bits in RTDEC_INTEN register, you can enable or disable RTDEC interrupt sources. It is shown as ***Table 24-1. RTDEC interrupt requests***. Status of the interrupt event is found in RTDEC_INTF register, and the interrupt flag can be cleared using RTDEC_INTC register.

Table 24-1. RTDEC interrupt requests

Interrupt event	Event flag	Enable control bit
Security error	SECEF	SECEIE
Execute-only, Execute-never error	ECONEF	ECONEIE
Key error	KEF	KEIE

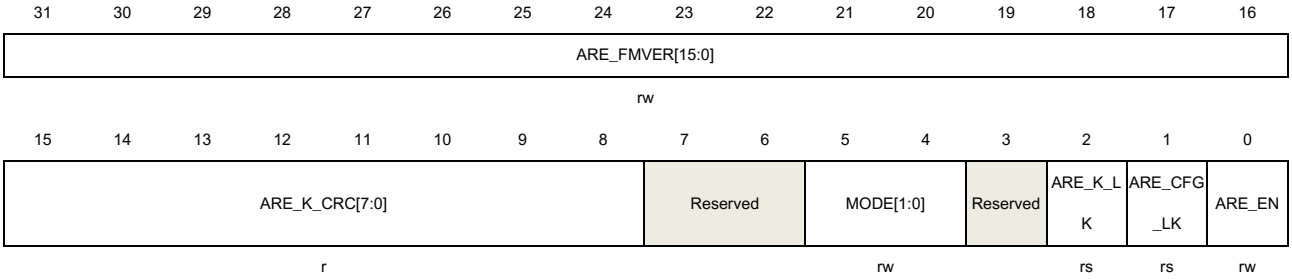
24.5. Registers definition

RTDEC0 base address: 0x5200 B800
RTDEC1 base address: 0x5200 BC00

24.5.1. Area x configuration register (RTDEC_AREx_CFG)

Address offset: 0x20 + 0x30 * x (x = 0 to 3)
Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:16	ARE_FMVER[15:0]	Area firmware version bits Before enabling the ARE_EN bit in the RTDEC_AREx_CFG register, the area firmware version must be configured first.
15:8	ARE_K_CRC[7:0]	8-bit CRC of area key bits As ARE_K_LK = 0, if user loads the key of the area in this exact sequence: KEY0 -> KEY1 -> KEY2 -> KEY3, the ARE_K_CRC[7:0] will automatically computed by hardware. When a new valid sequence is initiated, a new computation starts immediately, and ARE_K_CRC[7:0] is read as 0 until a valid sequence is completed. When ARE_K_LK=1, ARE_K_CRC remains unchanged until the next reset. The standard CRC-8-CCITT algorithm X8 + X2 + X + 1 is used for CRC calculation, the calculation result is an 8-bit checksum. These bits are read only. **Note:** When the last bit of the key has been written, CRC information is updated.
7:6	Reserved	Must be kept at reset value.
5:4	MODE[1:0]	RTDEC mode bits These bits configure the RTDEC operating mode for this area: 00: Only decrypt code accesses. 01: Only decrypt data accesses. 10: decrypt all read accesses (code or data). 11: Reserved. When the MODE[1:0] bits are changed, the area's key and associated CRC are

		automatically cleared.
3	Reserved	Must be kept at reset value.
2	ARE_K_LK	Area key lock bit 0: RTDEC_AREx_KEY registers write is enabled. 1: RTDEC_AREx_KEY registers write is disabled. ARE_K_CRC[7:0] is also locked. This bit is set by software only once. It cannot be cleared until a RTDEC reset. When this bit is set, it disables write access to this area key registers.
1	ARE_CFG_LK	Area configure lock bit 0: RTDEC_AREx_CFG, RTDEC_AREx_SADDR, RTDEC_AREx_EADDR, RTDEC_AREx_NONCEy (y = 0, 1), RTDEC_AREx_KEYy (y = 0 to 3) registers write is enabled. 1: RTDEC_AREx_CFG, RTDEC_AREx_SADDR, RTDEC_AREx_EADDR, RTDEC_AREx_NONCE, RTDEC_AREx_KEY registers write is disabled. This bit is set by software only once. It cannot be cleared until a RTDEC reset. When this bit is set ARE_K_LK bit is forced to 1, and it disables write access to this area configuration, start address, end address and nonce registers.
0	ARE_EN	Area real-time decryption enable bit 0: Area real-time decryption is disabled. 1: Area real-time decryption is enabled. The area configure information must be valid before this bit is set.

24.5.2. Area x start address register (RTDEC_AREx_SADDR)

Address offset: 0x24 + 0x30 * x (x = 0 to 3)

Reset value: 0x0000 0000

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

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

Bits	Fields	Descriptions
31:0	ARE_SADDR[31:0]	Area x AXI start address bits These bits should be written before the corresponding ARE_EN is set in the RTDEC_AREx_CFG register. Writing these bits is invalid if the ARE_CFG_LK bit is set in the RTDEC_AREx_CFG register. **Note:** The first 12 bits (LSB) and the last 4 bits (MSB) are ignored when determining the area.

The 4 MSB bits and the 12 LSB bits return 0, when read this register.

24.5.3. Area x end address register (RTDEC_AREx_EADDR)

Address offset: 0x28 + 0x30 * x (x = 0 to 3)
Reset value: 0x0000 0FFF

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

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

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

Bits	Fields	Descriptions
31:0	ARE_EADDR[31:0]	Area x AXI end address bits These bits should be written before the corresponding ARE_EN is set in the RTDEC_AREx_CFG register. And ARE_EADDR cannot be less than ARE_SADDR. Writing these bits is invalid if the ARE_CFG_LK bit is set in the RTDEC_AREx_CFG register. **Note:** The first 12 bits (LSB) and the last 4 bits (MSB) are ignored when determining the area. The 4 MSB bits return 0 and the 12 LSB bits return 1, when read this register.

24.5.4. Area x nonce register 0 (RTDEC_AREx_NONCE0)

Address offset: 0x2C + 0x30 * x (x = 0 to 3)
Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ARE_NONCE0[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARE_NONCE0[15:0]															
rw															

Bits	Fields	Descriptions
31:0	ARE_NONCE0[31:0]	Area x nonce bits, ARE_NONCE[31:0]. These bits should be written before the corresponding ARE_EN bit is set in the RTDEC_AREx_CFG register. Writing these bits is invalid if the ARE_CFG_LK bit is set in the

RTDEC_AREx_CFG register.

24.5.5. Area x nonce register 1 (RTDEC_AREx_NONCE1)

Address offset: 0x30 + 0x30 * x (x = 0 to 3)
Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ARE_NONCE1[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARE_NONCE1[15:0]															
rw															

Bits	Fields	Descriptions
31:0	ARE_NONCE1[31:0]	Area x nonce bits, ARE_NONCE[63:32]. Refer to ARE_NONCE0[31:0] description

24.5.6. Area x key register 0 (RTDEC_AREx_KEY0)

Address offset: 0x34 + 0x30 * x (x = 0 to 3)
Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ARE_KEY0[31:16]															
w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARE_KEY0[15:0]															
w															

Bits	Fields	Descriptions
31:0	ARE_KEY0[31:0]	Area x key bits, ARE_KEY[31:0]. These bits should be written before the corresponding ARE_EN bit is set in the RTDEC_AREx_CFG register. Writing these bits is invalid if the ARE_CFG_LK bit or ARE_K_LK bit is set in the RTDEC_AREx_CFG register. When read these bits it returns 0. **Note:** RTDEC_AREx_KEY0 register and associated ARE_K_CRC are erased when application successfully changes bits MODE[1:0] in RTDEC_AREx_CFG register.

24.5.7. Area x key register 1 (RTDEC_AREx_KEY1)

Address offset: 0x38 + 0x30 * x (x = 0 to 3)

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ARE_KEY1[31:16]															
w															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARE_KEY1[15:0]															
w															

Bits	Fields	Descriptions
31:0	ARE_KEY1[31:0]	Area x key bits, ARE_KEY[63:32]. Refer to RTDEC_ARE_KEY0[31:0] description

24.5.8. Area x key register 2 (RTDEC_AREx_KEY2)

Address offset: 0x3C + 0x30 * x (x = 0 to 3)

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ARE_KEY2[31:16]															
w															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARE_KEY2[15:0]															
w															

Bits	Fields	Descriptions
31:0	ARE_KEY2[31:0]	Area x key bits, ARE_KEY[95:64]. Refer to RTDEC_ARE_KEY0[31:0] description

24.5.9. Area x key register 3 (RTDEC_AREx_KEY3)

Address offset: 0x40 + 0x30 * x (x = 0 to 3)

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ARE_KEY3[31:16]															
w															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARE_KEY3[15:0]															
w															

Bits	Fields	Descriptions
31:0	ARE_KEY3[31:0]	Area x key bits, ARE_KEY[127:96]. Refer to RTDEC_ARE_KEY0[31:0] description

24.5.10. Interrupt flag register (RTDEC_INTF)

Address offset: 0x300

Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value.
2	KEF	Key error interrupt flag 0: No key error is detected. 1: Key error is detected. Read access detected on an enabled encrypted area after an abort event. An interrupt will occur if the KEIE bit is set in RTDEC_INTEN. This bit is set by hardware, when a read access occurs on encrypted or non-encrypted area following the reset of the key registers by an abort event (tamper detection, unauthorized debugger connection, untrusted boot, SPC level low to no protection demotion). Cleared by writing 1 to the KEC bit in the RTDEC_INTC register. Any subsequent read to encrypted or non-encrypted area returns 0 after KEF bit is set. It remains until RTDEC keys are initialized again.
1	ECONEF	Execute-only or execute-never error interrupt flag 0: No execute-only error or execute-never error is detected. 1: Execute-only error or execute-never error is detected. Read access detected on an area with MODE[1:0] set to 00, or execute access detected on an area with MODE [1:0] set to 01. An interrupt will occur if bit ECONEIE is set in RTDEC_INTEN register. Set by hardware, when a read access but not an instruction fetch is detected on any encrypted area with MODE[1:0] set to 00. It is also set when an instruction fetch but not a read access is detected on any encrypted area with MODE[1:0] set

		to 01. Cleared by writing 1 to the ECONEC bit in the RTDEC_INTC register. If the access is illegal, RTDEC returns 0.
0	SECEF	Security error interrupt flag 0: No security error is detected. 1: Security error is detected. An interrupt will occur if SECEIE bit is set in RTDEC_INTEN register. Set by hardware, when at least one security error has been detected (illegal access to keys, illegal write on locked configuration). Cleared by writing 1 to SECEC bit in the RTDEC_INTC register.

24.5.11. Interrupt flag clear register (RTDEC_INTC)

Address offset: 0x304
Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value.
2	KEC	Key error flag clear 0: No effect 1: Clear key error flag **Note:** To be able to read or execute again any encrypted region, RTDEC key registers should properly reinitialized but not just clear KEF.
1	ECONEC	Execute-only or execute-never error flag clear 0: No effect 1: Clear execute-only or execute-never error flag
0	SECEC	Security error interrupt flag clear 0: No effect 1: Clear security error flag

24.5.12. Interrupt enable register (RTDEC_INTEN)

Address offset: 0x308
Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value.
2	KEIE	Enable bit for key error interrupt Software set and cleared 0: Disable key error interrupt 1: Enable key error interrupt
1	ECONEIE	Enable bit for execute-only or execute-never error interrupt Software set and cleared 0: Disable execute-only or execute-never error interrupt 1: Enable execute-only or execute-never error interrupt
0	SECEIE	Enable bit for security error interrupt Software set and cleared 0: Disable security error interrupt 1: Enable security error interrupt

25. Analog-to-digital converter (ADC)

25.1. Overview

A 12 / 14-bit successive approximation analog-to-digital converter module(ADC) is integrated on the MCU chip. ADC0 has 20 external channels, 2 internal channel(DAC0_OUT0 channel, V_{REFINT} inputs channel), ADC1 has 18 external channels, 3 internal channels(the battery voltage, V_{REFINT} inputs channel and DAC0_OUT1 channel), ADC2 has 17 external channels, 3 internal channels(the battery voltage, temperature sensor and high-precision temperature sensor). 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(ADC0 / 1 are 32-bit data register, ADC2 is 16-bit data register). An on-chip hardware oversample scheme improves performance and reduces the computational burden of MCU.

25.2. Characteristics

- High performance.
  - ADC sampling resolution: ADC0 and ADC1, 14-bit,12-bit, 10-bit or 8-bit configurable resolution. ADC2, 12-bit, 10-bit, 8-bit or 6-bit configurable resolution.
  - Programmable sampling time.
  - Data storage mode: the most significant bit and the least significant bit
  - DMA support.
- Analog input channels.
  - 20 external analog inputs in ADC0, 18 external analog inputs in ADC1, 17 external analog inputs in ADC2.
  - Internal temperature sensor (V_{SENSE}).
  - Internal reference voltage (V_{REFINT}).
  - External battery power supply pin (V_{BAT}).
  - 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.

- Conversion result threshold monitor function: analog watchdog.
- Interrupt generation.
  - At the end of routine / inserted conversions.
  - Analog watchdog event.
  - Overflow event.
- Oversampler.
  - 32-bit data register in ADC0 and ADC1, 16-bit data register in ADC2.
  - In ADC0 and ADC1, Oversampling ratio arbitrarily adjustable from 2x to 1024x, In ADC2, Oversampling ratio arbitrarily adjustable from 2x to 256x.
  - In ADC0 and ADC1, Programmable data shift up to 11-bit, In ADC2, Programmable data shift up to 8-bit.
- Channel input range: V_{REFN} <=V_{IN} <=V_{REFP}.
- Data can be routed to HPDF for post processing.

25.3. Pins and internal signals

Figure 25-1. ADC module block diagram shows the ADC block diagram, *Table 25-1. ADC internal input signals* gives the ADC internal signals and *Table 25-2. ADC input pins definition* gives the ADC pin description.

Table 25-1. ADC internal input signals

Internal signal name	Description
V_{SENSE}	Internal temperature sensor output voltage
V_{SENSE2}	Internal high-precision temperature sensor2 output voltage
V_{REFINT}	Internal voltage reference output voltage
V_{BAT}	External battery voltage
DAC0_OUT0	DAC0_OUT0 channel
DAC0_OUT1	DAC0_OUT1 channel

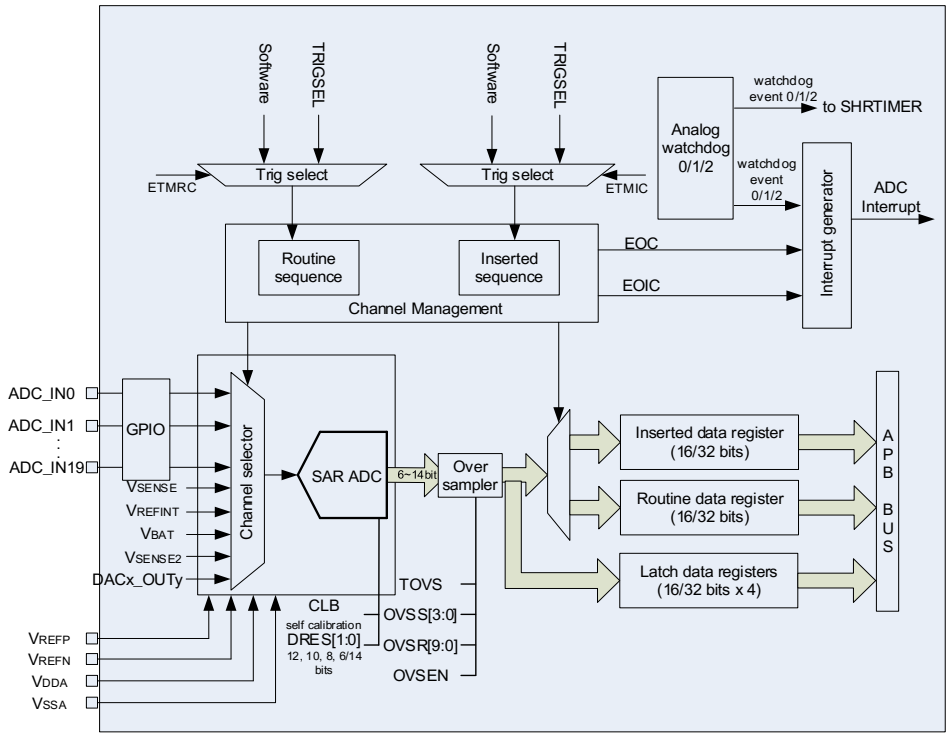
Table 25-2. ADC input pins definition

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,V_{REFN} = V_{SSA}
ADCx_IN[19:0]	Up to 20 external channels

Note: V_{DDA} and V_{SSA} have to be connected to V_{DD} and V_{SS}, respectively.

25.4. Function overview

Figure 25-1. ADC module block diagram



25.4.1. Foreground calibration function

During the foreground calibration procedure, the ADC calculates a calibration factor which is internally applied to the ADC until the next ADC power-off. The application must not use the ADC during calibration and must wait until it is completed. Calibration should be performed before starting A/D conversion. The calibration is initiated by setting bit CLB=1. CLB bit stays at 1 during all the calibration sequence. It is then cleared by hardware as soon as the calibration is completed.

When the ADC operating conditions change (such as supply power voltage V_{DDA}, positive reference voltage V_{REFP}, temperature and so on), it is recommended to re-run a calibration cycle.

The internal analog calibration can be reset by setting the RSTCLB bit in ADC_CTL1 register.

Calibration software procedure:

1. Ensure that ADCON=1
2. delay 14 CK_ADC to wait for ADC stability
3. Set RSTCLB (optional)
4. Set CLB=1
5. Wait until CLB=0

25.4.2. 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. The ADC0/1 maximum frequency is 80 MHz, ADC2 maximum frequency is 85 MHz.

The CK_ADC can also be fed by CK_PLL1P, CK_PLL2R, CK_PER or CK_PLL1Q, 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 ADCCK[3:0] of ADC_SYNCCTL.

The RCU controller has a dedicated programmable prescaler for the ADC clock.

Note: The ADC1 clock shares the ADC0 clock. When using the ADC1, ADC0 clock needs to be opened, and the clock frequency division can only be configured through ADC0.

25.4.3. ADC enable

The ADCON bit on the ADC_CTL1 register is the enable switch of the ADC module. The ADC module will keep in reset state if this bit is 0. For power saving, when this bit is reset, the analog submodule will be put into powerdown mode. After ADC is enabled, you need delay t_{SU} time for sampling, the value of t_{SU} please refer to the device datasheet.

25.4.4. Single-ended and differential input channels

By writing to bits DIFCTL[21:0] in the ADC_DIFCTL register, the user can configurate channels as differential input or single-ended input, and the ADC must be disabled (ADCON = 0) when the user configurate 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 25-3. ADC differential channel pin matching*.

Table 25-3. ADC differential channel pin matching

Differential channel n number	ADC0		ADC1		ADC2	
	V_{Inn}	V_{Inm}	V_{Inn}	V_{Inm}	V_{Inn}	V_{Inm}
0	PA0_C	PA1_C	PA0_C	PA1_C	PC2_C	PC3_C
1	PA1_C	PA0_C	PA1_C	PA0_C	PC3_C	PC2_C
2	PF11	PF12	PF13	PF14	PF9	PF10
3	PA6	PA7	PA6	PA7	PF7	PF8
4	PC4	PC5	PC4	PC5	PF5	PF6
5	PB1	PB0	PB1	PB0	PF3	PF4
6	PF12	PF11	PF14	PF13	PF10	PF9

7	PA7	PA6	PA7	PA6	PF8	PF7
8	PC5	PC4	PC5	PC4	PF6	PF5
9	PB0	PB1	PB0	PB1	PF4	PF3
10	PC0	PC1	PC0	PC1	PC0	PC1
11	PC1	PC0	PC1	PC0	PC1	PC0
12	PC1	PC2	PC1	PC2	PC1	PC2
13	PC2	PC1	PC2	PC1	PC2	PC1
14	PC2	PC3	PC2	PC3	PH2	PH3
15	PC3	PC2	PC3	PC2	PH3	PH4
16	PA2	PA3	PA2	PA3	PH4	PH5
17	PA3	PA2	PA3	PA2	PH5	PH4
18	PA0	PA1	null	null	null	null
19	PA1	PA0	null	null	null	null
20	PA4	PA5	PA4	PA5	null	null
21	PA5	PA4	PA5	PA4	null	null

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:

$$V_{Inn} - V_{Inm} = V_{REFP} * (2 * D_{out} / 4095 - 1) \quad (25\text{-}1)$$

25.4.5. Routine and inserted sequence

The channel management circuit can organize the sampling conversion channels into two sequences: a routine sequence and an inserted sequence.

The routine sequence supports up to 16 channels, and each channel is called routine channel. The RL[3:0] bits in the ADC_RSQ0 register specify the total conversion sequence length. The ADC_RSQ0~ADC_RSQ8 registers specify the selected channels of the routine sequence.

The inserted sequence supports up to 4 channels, and each channel is called inserted channel. The IL[1:0] bits in the ADC_ISQ0 register specify the total conversion sequence length. The ADC_ISQ0~ADC_ISQ2 register specify the selected channels of the inserted sequence.

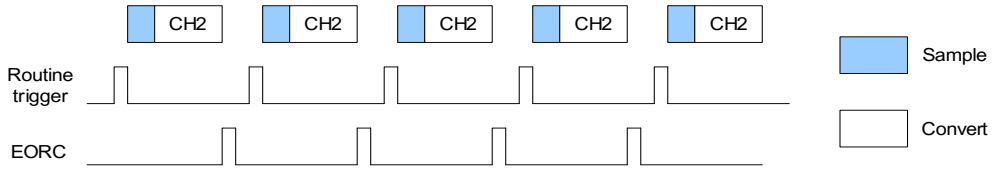
Note: Although the ADC supports 22 multiplexed channels, the maximum length of the routine sequence is only 16,and the maximum length of the inserted sequence is only 4.

25.4.6. Operation modes

Single operation mode

This mode can be running on both routine and inserted sequence. In the single operation mode, the ADC performs conversion on the channel specified in the RSQ0[4:0] bits of ADC_RSQ8 register at a routine trigger or the channel specified in the ISQ3[4:0] bits of ADC_ISQ0 register. 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 25-2. Single operation mode



After conversion of a single routine channel, the conversion data will be stored in the ADC_RDATA register, the EORC flag will be set. An interrupt will be generated if the EORCIE bit is set.

After conversion of a single inserted channel, the conversion data will be stored in the ADC_IDATA register, the EOIC will be set. An interrupt will be generated if the EOICIE bit is set.

Software procedure for 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 a TRIGSEL trigger for the routine sequence.
6. Wait the EORC flag to be set.
7. Read the converted in the ADC_RDATA register.
8. Clear the EORC flag by writing 0 to it.

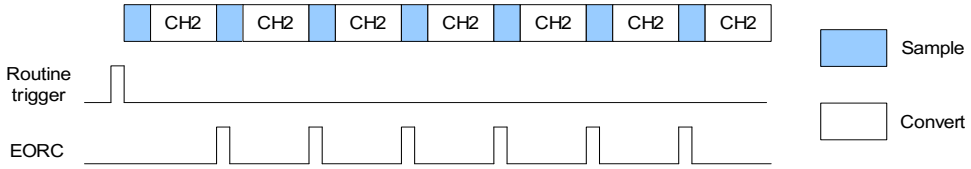
Software procedure for a single operation mode of an inserted channel:

1. Make sure the DISIC, SM in the ADC_CTL0 register are reset.
2. Configure ISQ3 with the analog channel number.
3. Configure ADC_ISQx register.
4. Configure ETMIC[1:0] bits in the ADC_CTL1 register if in need.
5. Set the SWICST bit, or generate a TRIGSEL trigger for the inserted sequence.
6. Wait the EOIC flags to be set.
7. Read the converted in the ADC_IDATA register.
8. Clear the EOIC flags by writing 0 to them.

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 25-3. 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 a TRIGSEL trigger for the routine sequence.
6. Wait the EORC flag to be set.
7. Read the converted in the ADC_RDATA register.
8. Clear the EORC flag by writing 0 to it.
9. Repeat steps 6~8 as soon as the conversion is in need.

To get rid of checking, DMA can be used to transfer the converted data:

1. Set the CTN, RDMA and RDDM bits 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 a TRIGSEL trigger for the routine sequence.

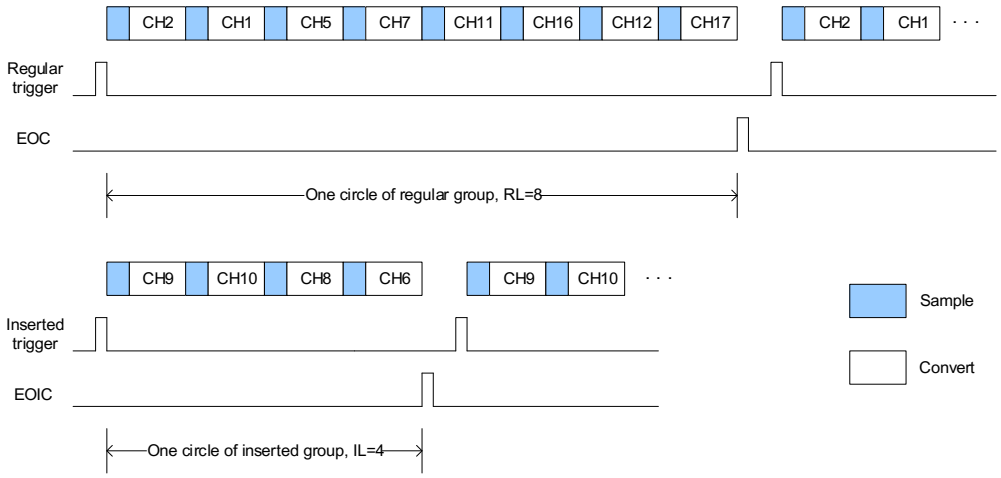
Scan operation mode

The scan operation mode will be enabled when SM bit in the ADC_CTL0 register is set. In this mode, the ADC performs conversion on the channels with a specific sequence specified in the ADC_RSQ0~ADC_RSQ8 registers or ADC_ISQ0~ADC_ISQ2 register. When the ADCON has been set high, the ADC samples and converts specified channels one by one in the routine or inserted sequence till the end of the routine or inserted sequence, once the corresponding software trigger or TRIGSEL trigger is active. The conversion data will be stored in the ADC_RDATA or ADC_IDATA register. After conversion of the routine or inserted sequence, the EORC or EOIC will be set. An interrupt will be generated if the EORCIE or EOICIE bit is set. The RDMA or IDMA bit in ADC_CTL1 register must be set when the routine

or inserted sequence works in scan mode.

After conversion of a routine sequence, the conversion can be restarted automatically if the CTN bit in the ADC_CTL1 register is set.

Figure 25-4. Scan operation mode, continuous disable



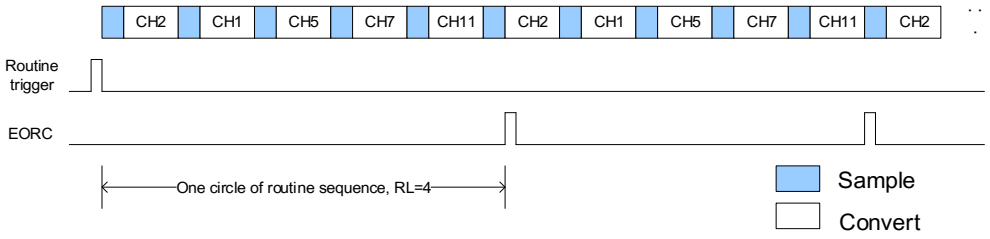
Software procedure for scan operation mode on a routine sequence:

1. Set the SM bit in the ADC_CTL0 register and the RDMA bit in the ADC_CTL1 register
2. Configure ADC_RSQx registers.
3. Configure ETMRC[1:0] bits in the ADC_CTL1 register if in need.
4. Prepare the DMA module to transfer data from the ADC_RDATA.
5. Set the SWRCST bit, or generate a TRIGSEL trigger for the routine sequence.

Software procedure for scan operation mode on an inserted sequence:

1. Set the SM bit in the ADC_CTL0 register.
2. Configure ADC_ISQx registers.
3. Configure ETMIC[1:0] bits in the ADC_CTL1 register if in need.
4. Prepare the DMA module to transfer data from the ADC_IDATA.
5. Set the SWICST bit, or generate a TRIGSEL trigger for the inserted sequence.

Figure 25-5. Scan operation mode, continuous enable



Discontinuous operation mode

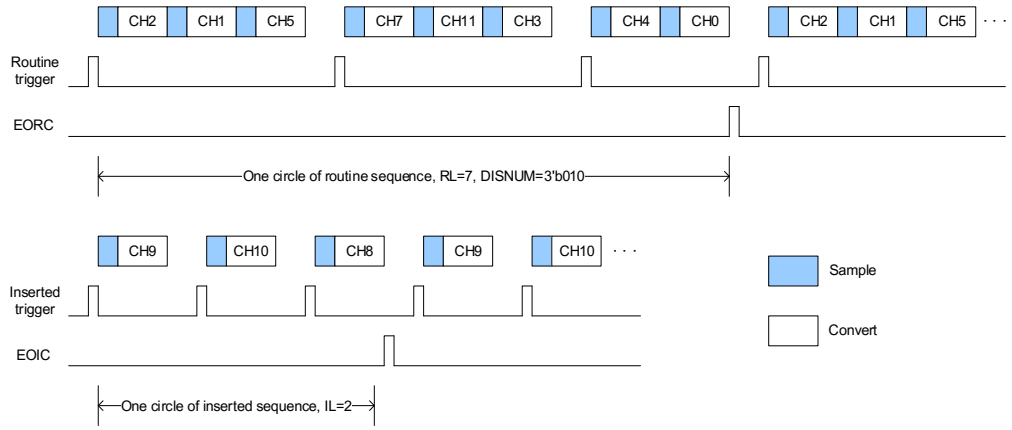
For routine sequence, the discontinuous operation mode will be enabled when DISRC bit in the ADC_CTL0 register is set. In this mode, the ADC performs a short sequence of n conversions (n does not exceed 8) which is a part of the sequence selected in the

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 channels are done. The EORC will be set after every cycle of the routine sequence. An interrupt will be generated if the EORCIE bit is set.

For inserted sequence, the discontinuous operation mode will be enabled when DISIC bit in the ADC_CTL0 register is set. In this mode, the ADC performs one conversion which is a part of the sequence of conversions selected in the ADC_ISQ0~ADC_ISQ2 register. When the corresponding software trigger or TRIGSEL trigger is active, the ADC samples and coverts the next channel configured in the ADC_ISQ0~ADC_ISQ2 register until all the inserted channels are done. The EOIC will be set after every circle of the inserted sequence. An interrupt will be generated if the EOICIE bit is set.

The routine and inserted sequence cannot both work in discontinuous operation mode. Only one sequence conversion can be set in discontinuous operation mode at a time.

Figure 25-6. Discontinuous operation mode



Software procedure for discontinuous operation mode on a routine sequence:

1. Set the DISRC bit in the ADC_CTL0 register and the RDMA bit in the ADC_CTL1 register.
2. Configure DISNUM[2:0] bits in the ADC_CTL0 register.
3. Configure ADC_RSQx 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 a TRIGSEL trigger for the routine sequence.
7. Repeat step6 if in need.

Software procedure for discontinuous operation mode on an inserted sequence:

1. Set the DISIC bit in the ADC_CTL0 register and the IDMA bit in the ADC_CTL1 register.
2. Configure ADC_ISQx registers.
3. Configure ETMIC[1:0] bits in the ADC_CTL1 register if in need.
4. Prepare the DMA module to transfer data from the ADC_IDATA.
5. Set the SWICST bit, or generate a TRIGSEL trigger for the inserted sequence.

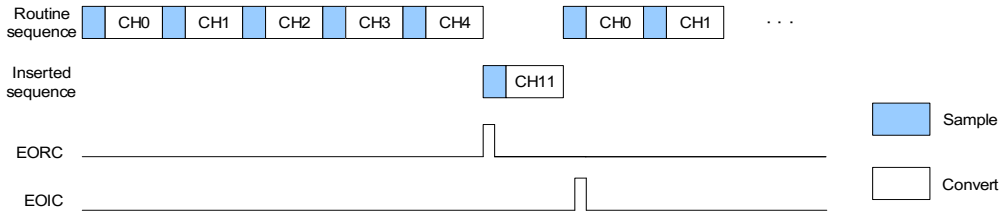
6. Repeat step5 if in need.

25.4.7. Inserted sequence management

■ Auto insertion

The inserted sequence channels are automatically sampled and converted after the routine sequence channels when the ICA bit in ADC_CTL0 register is set. In this mode, TRIGSEL trigger on inserted channels cannot be enabled. A sequence of multiple conversions programmed in the ADC_RSQ0~ADC_RSQ8 and ADC_ISQ0~ADC_ISQ2 registers can be used to convert in this mode. In addition to the ICA bit, if the CTN bit is also set, the routine sequence channels and inserted sequence channels are constantly converted.

Figure 25-7. Auto-insertion, CTN = 1

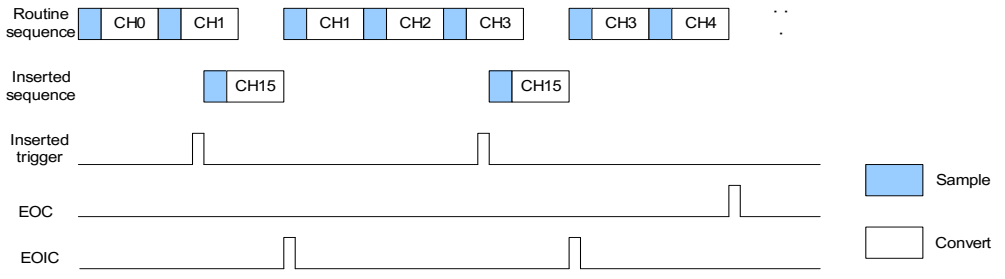


The auto insertion mode cannot be enabled when the discontinuous operation mode is set.

■ Triggered insertion

If the ICA bit is cleared, the triggered insertion occurs if a software or TRIGSEL trigger occurs during the routine sequence channel conversion. In this situation, the ADC aborts from the current conversion and starts the conversion of inserted sequence in scan mode. After the inserted sequence is done, the routine channel conversion is restarted from the aborted conversion.

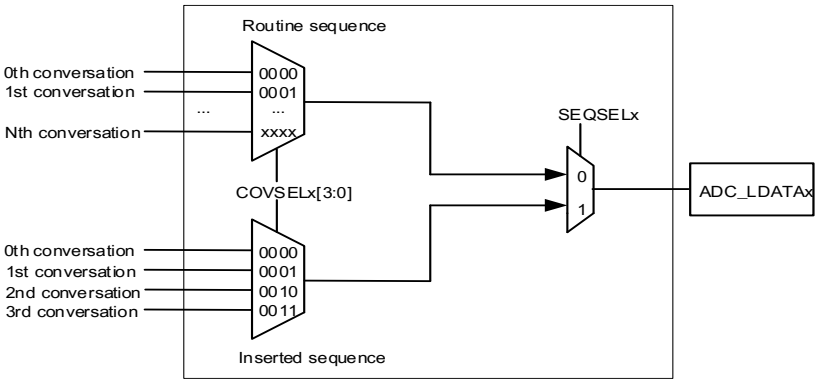
Figure 25-8. Triggered insertion



25.4.8. Convert data latch

The ADC has 4 latch data registers, ADC_LDATAx (x=0…3), which can latch the data from a completed conversion in a routine or inserted sequence. The control logic for the latch data registers is shown in ***Figure 25-9. Data latch***.

Figure 25-9. Data latch

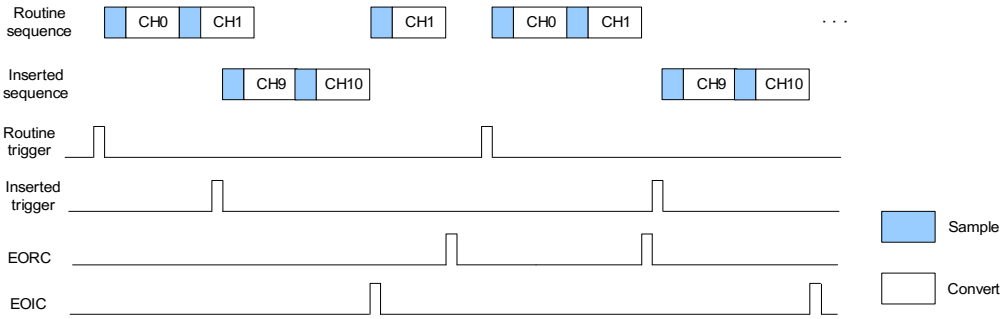


The data to be latched from either the routine or inserted sequence is selected through the SEQSELx (x=0…3) bits in the ADC_LDCTL register, while the COVSELx[3:0] bits selects which conversion result in the sequence to latch.

ADC_LDATAx (x=0…3) registers are set to store the conversion results of the inserted sequence or routine sequence and default store inserted sequence xth conversion results.

Therefore, in addition to using DMA to transfer conversion data from the ADC_RDATA or ADC_IDATA registers, data can also be directly read from the latch data registers, offering greater flexibility for consecutive conversions in a conversion sequence.

Figure 25-10. Sequence data from ADC_LDATAx



Sequence data

ADC_LDATA0	ADC_LDATA1	ADC_LDATA2	ADC_LDATA3
CH9 conversation data	CH10 conversation data	CH0 conversation data	CH1 conversation data

SEQSEL0 = 1'b0 COVSEL0[3:0] = 4'b1001
SEQSEL1 = 1'b0 COVSEL1[3:0] = 4'b1010
SEQSEL2 = 1'b1 COVSEL2[3:0] = 4'b0000
SEQSEL3 = 1'b1 COVSEL3[3:0] = 4'b0001

Software procedure for triggered insertion mode and read the sequence data from ADC_LDATAx:

1. Set the SM bit in the ADC_CTL0 register;
2. Configure ADC_RSQx, ADC_ISQx registers;
3. Configure SEQSEL2, COVSEL2[3:0], SEQSEL3 and COVSEL3[3:0] bits in the ADC_LDCTL register for the routine sequence;
4. Configure SEQSEL0, COVSEL0[3:0], SEQSEL1 and COVSEL1[3:0] bits in the ADC_LDCTL register for the inserted sequence;
5. Configure ETMRC and ETMIC bits in the ADC_CTL1 register if it is needed;

6. Set the SWRCST bit, or generate an external trigger for the routine sequence;
7. Set the SWICST bit, or generate an external trigger for the inserted sequence;
8. Wait for EORC or EOIC flags to be set;
9. If EORC flag is set, read the converted result of routine sequence in the ADC_LDATA2 and ADC_LDATA3 register;
10. Clear the EORC flag by writing 0.
11. If EOIC flag is set, read the converted result of inserted sequence in the ADC_LDATA0 and ADC_LDATA1 register;
12. Clear the EOIC flag by writing 0.

25.4.9. Conversion result threshold monitor function

Analog watchdog 0

The analog watchdog 0 is enabled when the RWD0EN and IWD0EN bits in the ADC_CTL0 register are set for routine and inserted sequence respectively.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, IWD0EN, WD0SC and WD0CHSEL[4:0] bits in ADC_CTL0 register, can be monitored by the analog watchdog 0.

Analog watchdog 1/2

The analog watchdog 1/2 are more flexible, and can configure the watchdog function of single or multiple 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: For ADC0/ADC1, 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.

25.4.10. Data storage mode

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

After being decreased by the user-defined offset written in the ADC_IOFFx registers, the

inserted sequence data value may be a negative value. The sign value is extended.

Figure 25-11. Data storage mode of 14-bit resolution

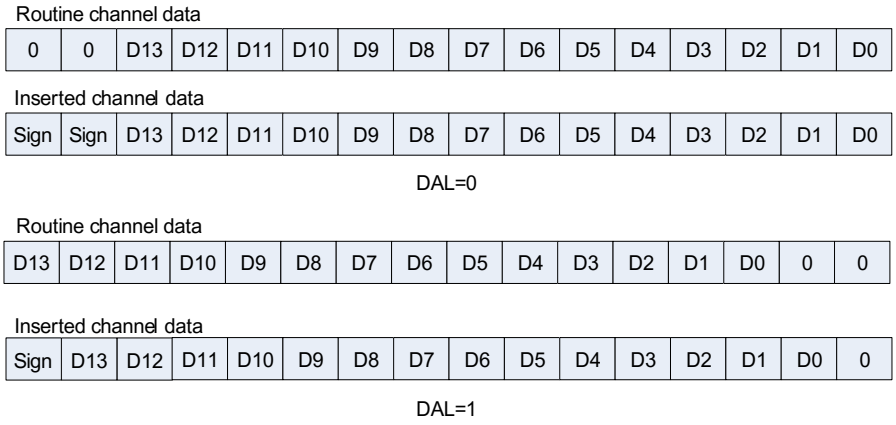


Figure 25-12. Data storage mode of 12-bit resolution

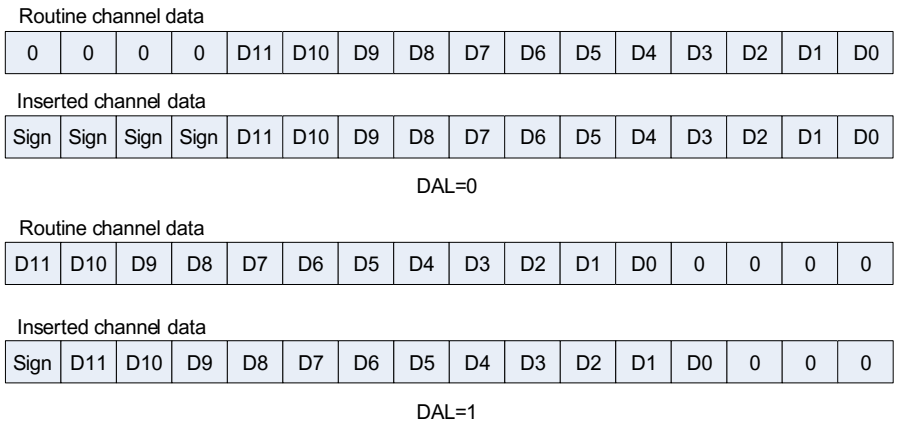


Figure 25-13. Data storage mode of 10-bit resolution

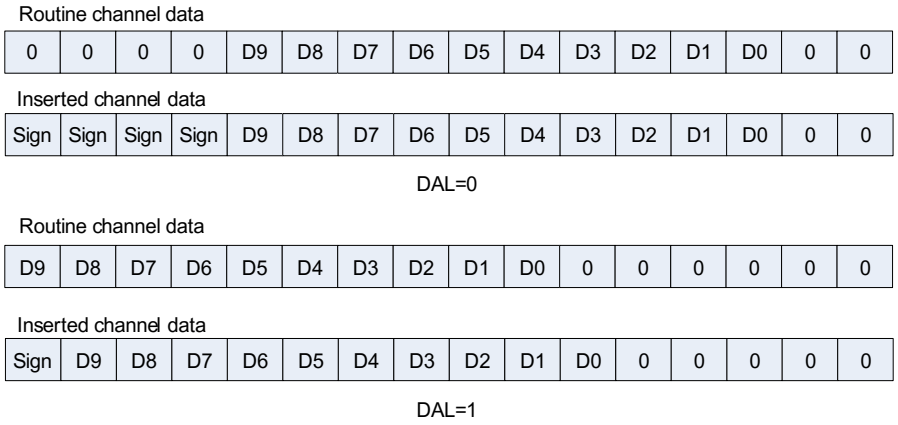
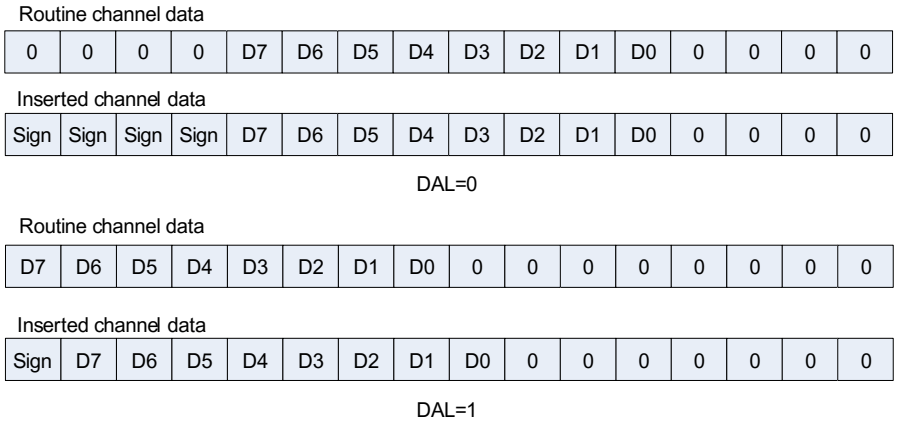
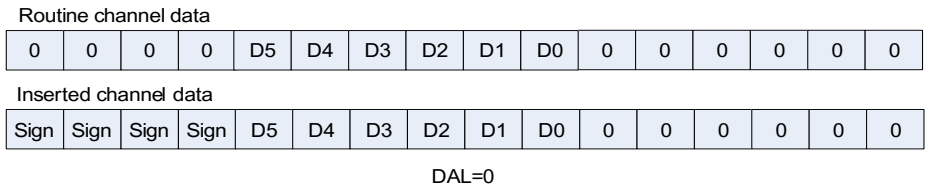


Figure 25-14. Data storage mode of 8-bit resolution



6-bit resolution data alignment is different from 12-bit/10-bit/8-bit resolution data alignment, left alignment (DAL = 1) is not supported in 6-bit resolution, shown as ***Figure 25-15. 6-bit data alignment***.

Figure 25-15. 6-bit data alignment



25.4.11. 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 or ISMPn[9:0] bits in the ADC_ISQ0~ ADC_ISQ2. A different sample time can be specified for each sequence. For 12-bits resolution, the total sampling and conversion time is "sampling time + 14.5" CK_ADC

cycles.

Example:

CK_ADC = 40MHz and sample time is 3.5 cycles, the total conversion time is "3.5+14.5" CK_ADC cycles, that means 0.45 us.

25.4.12. External trigger configuration

The conversion of routine or inserted 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, while the trigger source of inserted sequence is controlled by the ETMIC[1:0] bit in the ADC_CTL1 register.

Table 25-4. Trigger source for routine channels for ADC0/ADC1/ADC2

ETMRC[1:0]	Trigger Source	Trigger Type
01, 10, 11	TRIGSEL	Signal from TRIGSEL
00	SWRCST	Software trigger

Table 25-5. Trigger source for inserted channels for ADC0/ADC1/ADC2

ETMIC[1:0]	Trigger Source	Trigger Type
01, 10, 11	TRIGSEL	Signal from TRIGSEL
00	SWICST	Software trigger

25.4.13. DMA request

The DMA request, which is enabled by the RDMA or IDMA bit of ADC_CTL1 register, is used to transfer data of routine or inserted sequence for conversion of more than one channel. The ADC generates a DMA request at the end of conversion of a routine or inserted channel. When this request is received, the DMA will transfer the converted data from the ADC_RDATA or ADC_IDATA register to the destination location which is specified by the user.

25.4.14. Overflow detection

Overflow detection is enabled when DMA is enabled or EORCM bit in the ADC_CTL1 register is set. An overflow event occurs when a routine or an inserted conversion is done before the prior routine or inserted data has been read out. The ROVF or IOVF bit of the ADC_STAT register is set. Overflow interrupt is generated if the ROVFIE or IOVFIE bit in the ADC_CTL0 register is set.

It is recommended to reinitialize the DMA module to recover the ADC from overflow state. To ensure the routine converted data are transferred correctly, the internal state machine is reset. The ADC conversion will be stalled until the overflow status flag bit is cleared.

Software procedure for recovering the ADC from overflow state:

1. Clear RDMA/IDMA bit of the ADC_CTL1 register to 0
2. Clear ADCON bit of the ADC_CTL1 register to 0

3. Clear CHEN bit of the DMA_CHxCTL register to 0 with reinitializing DMA module
4. Clear ROVF/IOVF bit of the ADC_STAT register to 0
5. Set CHEN bit of the DMA_CHxCTL register to 1
6. Set RDMA/IDMA bit of the ADC_CTL1 register to 1
7. Set ADCON bit of the ADC_CTL1 register to 1
8. Wait T(setup)
9. Start conversion with software or external TRIGSEL trigger.

Note: When RDMA function is used, EORCM should be configured to 0.

25.4.15. ADC internal channels

When the TSVEN1 bit of ADC_CTL1 register is set, the temperature sensor channel (ADC2_CH18) is enabled. When the TSVEN2 bit of ADC_CTL1 register is set, the high-precision temperature sensor channel (ADC2_CH20) is enabled. The VREFINT channel (ADC0_CH21/ADC1_CH17) is enabled by default. The temperature sensor can be used to measure the ambient temperature of the device. The sensor output voltage can be converted into a digital value by ADC. The sampling time for the temperature sensor is recommended to be set to at least t_{s_temp} μs (please refer to the datasheet). When this sensor is not in use, it can be put in power down mode by resetting the TSVEN1 or TSVEN2 bit.

The output voltage of the temperature sensor (only for normal 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_CH21/ADC1_CH17 input channel.

To use the temperature sensor:

1. Configure the conversion sequence (ADC2_IN18) 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.

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

V_{25}: internal temperature sensor output voltage at 25°C, the typical value and factory calibration value address please refer to the datasheet (in the temperature sensor characteristics chapter).

Avg_Slope: Average Slope for curve between Temperature vs. internal temperature

sensor output voltage, the typical value please refer to the datasheet (in the temperature sensor characteristics chapter).

To use the high precision temperature sensor:

1. Configure the ADC clock.
2. Configure the conversion sequence (ADC2_CH20) 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}: internal temperature sensor output voltage at 25°C, the typical value and factory calibration value address please refer to the datasheet (in the high-precision temperature sensor characteristics chapter).

Avg_Slope: Average Slope for curve between Temperature vs. Vtemperature, the typical value please refer to the datasheet (in the high-precision temperature sensor characteristics chapter).

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.

25.4.16. Battery voltage monitoring

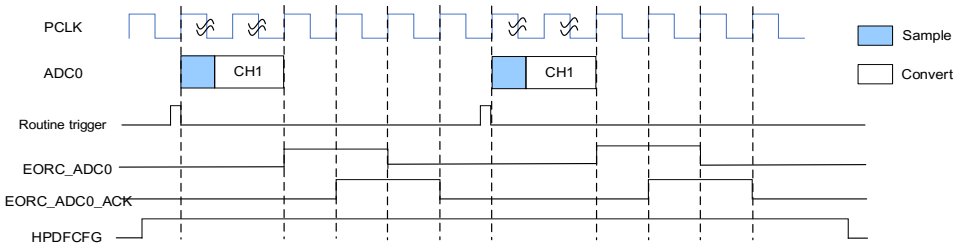
The V_{BAT} channel can be used to measure the backup battery voltage on the V_{BAT} pin. When the VBATEN bit in ADC_CTL1 register is set, V_{BAT} channel (ADC1_IN16 / ADC2_IN17) is enabled and a bridge divider by 4 integrated on the V_{BAT} pin is also enabled automatically with it. As V_{BAT} may be higher than V_{DDA}, this bridge is used to ensure the ADC correct operation. And it connects V_{BAT} / 4 to the ADC1_IN16 / ADC2_IN17 input channel. So, the converted digital value is V_{BAT} / 4. In order to prevent unnecessary battery energy consumption, it is recommended that the bridge will be enabled only when it is required.

25.4.17. 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 RDMA, RDDM, SYNCDMA[1:0], SYNCDDM bit must be cleared to 0. The ADC transfers 16 least significant bits of the routine

data register data to the HPDF, which in turns will reset the EORC flag once the transfer is complete. As shown in *Figure 25-16. Schematic diagram of handshake signal between HPDF and ADC module*.

Figure 25-16. Schematic diagram of handshake signal between HPDF and ADC module



25.4.18. Programmable resolution (DRES)

The resolution can be configured to be either 14, 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 25-6. tCONV timings depending on resolution for ADC0 and ADC1*, *Table 25-7. tCONV timings depending on resolution for ADC2*.

Table 25-6. t_{CONV} timings depending on resolution for ADC0 and ADC1

DRES[1:0] bits	t_{CONV} (ADC clock cycles)	t_{CONV}(ns) at f_{ADC}=80MHz	t_{SMPL}(min) (ADC clock cycles)	t_{ADC} (ADC clock cycles)	t_{ADC}(ns) at f_{ADC}=80MHz
14	16.5	206.25 ns	3.5	20	250 ns
12	14.5	181.25 ns	3.5	18	225 ns
10	12.5	156.25 ns	3.5	16	200 ns
8	10.5	131.25 ns	3.5	14	175 ns

Table 25-7. t_{CONV} timings depending on resolution for ADC2

DRES[1:0] bits	t_{CONV} (ADC clock cycles)	t_{CONV}(ns) at f_{ADC}=85MHz	t_{SMPL}(min) (ADC clock cycles)	t_{ADC} (ADC clock cycles)	t_{ADC}(ns) at f_{ADC}=85MHz
12	14.5	170.59 ns	2.5	17	200 ns
10	12.5	147.06 ns	2.5	15	176.47 ns
8	10.5	123.53 ns	2.5	13	152.94 ns
6	8.5	100 ns	2.5	11	129.41 ns

25.4.19. 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 ADC0 and ADC1, up to 16-bit in ADC2.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:

$$Result = \frac{1}{M} * \sum_{n=0}^{N-1} D_{out}(n) \quad (25\text{-}2)$$

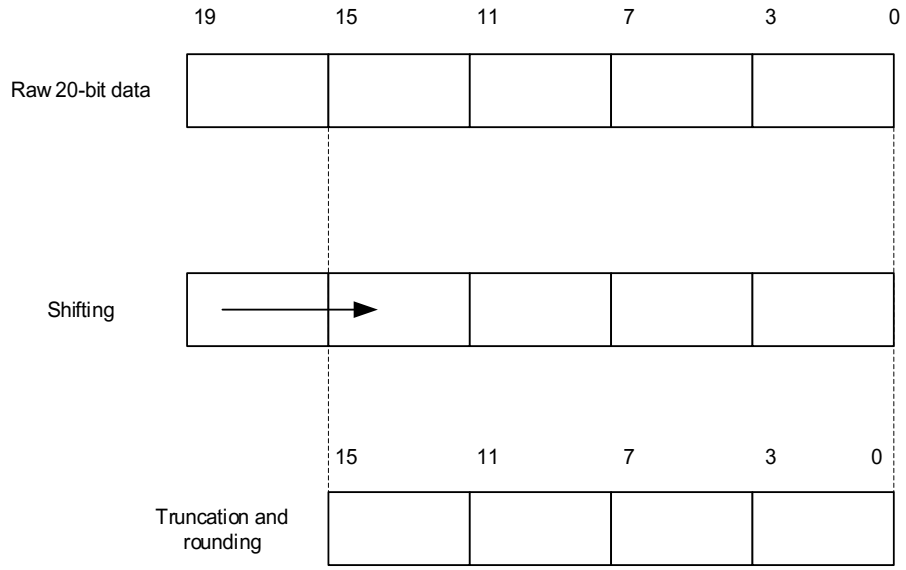
For 14bit-ADC, the on-chip hardware oversampling circuit performs the following functions: summing and bit right shifting. The oversampling ratio N is defined by the OVSR[9:0] bits in the ADC_OVSAMPCTL register. It can range from 2x to1024x. The division coefficient M means bit right shifting up to 11-bit. It is configured through the OVSS[3:0] bits in the ADC_OVSAMPCTL register.

For 14bit-ADC, summation units can produce up to 24 bits (1024 x 14bit), which is first shifted right. Then store the data into register.

For 12bit-ADC, the on-chip hardware oversampling circuit performs the following functions: summing and bit right shifting. The oversampling ratio N is defined by the OVSR[7:0] bits in the ADC_OVSAMPCTL register. It can range from 2x to 256x. The division coefficient M means bit right shifting up to 8-bit. It is configured through the OVSS[3:0] bits in the ADC_OVSAMPCTL register.

For 12bit-ADC, summation units can produce up to 20 bits (256 x 12-bit), which is first shifted right. The upper bits of the result are then truncated, keeping only the 16 least significant bits rounded to the nearest value using the least significant bits left apart by the shifting, before being finally transferred into the data register.

Figure 25-17. 20-bit to 16-bit result truncation (for 12bit ADC)



Note: If the intermediate result after the shifting exceeds 16 bits, the upper bits of the result are simply truncated.

Figure 25-17. 20-bit to 16-bit result truncation (for 12bit ADC) shows a numerical example of the processing, from a raw 20-bit accumulated data to the final 16-bit result.

Figure 25-18. Numerical example with 5-bits shift and rounding (for 12bit ADC)

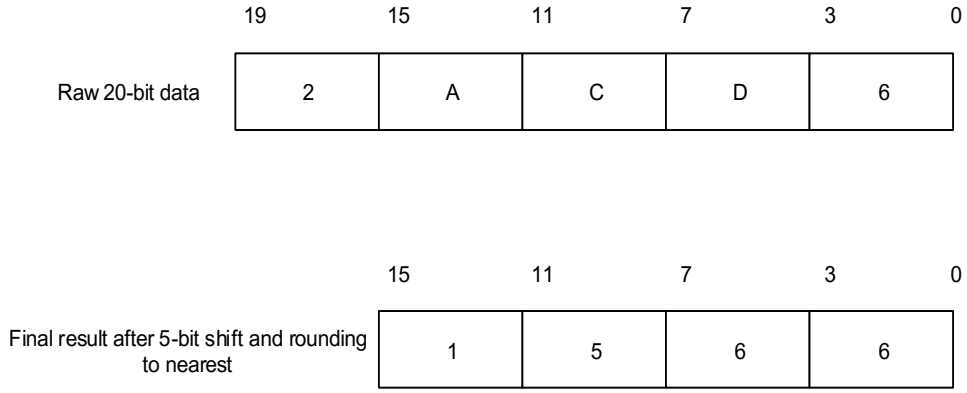


Figure 25-19. 14bit ADC oversampling with 10bits right shift

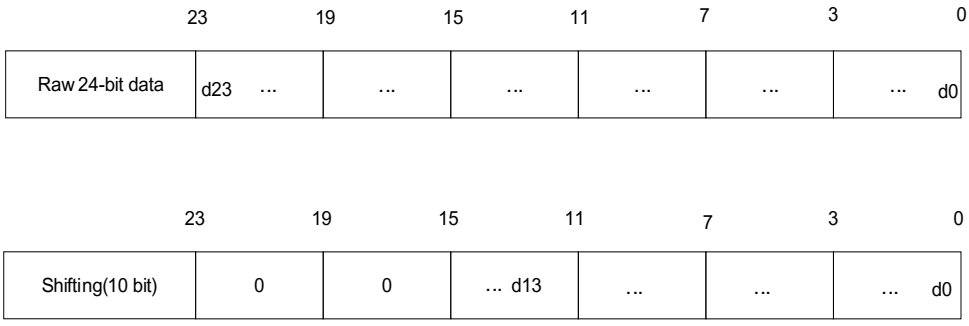
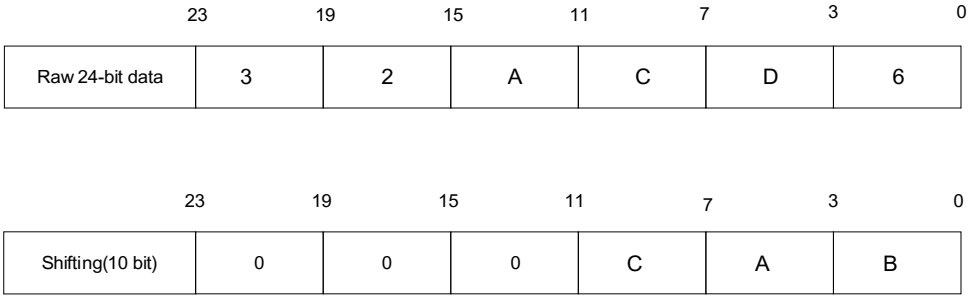


Figure 25-20. Numerical example for 14bit ADC oversampling with 10bits right shift



The *Table 25-8. Some examples show the maximum output results for N and M combinations (grayed values indicates truncation, for 12bit ADC)* below gives the data format for the various N and M combination, for a raw conversion data equal to 0xFFF.

Table 25-8. Some examples show the maximum output results for N and M

combinations (grayed values indicates truncation, for 12bit ADC)

Oversampling ratio	Max Raw data	No-shift OVSS=0000	1-bit shift OVSS=0001	2-bit shift OVSS=0010	3-bit shift OVSS=0011	4-bit shift OVSS=0100	5-bit shift OVSS=0101	6-bit shift OVSS=0110	7-bit shift OVSS=0111	8-bit shift OVSS=1000
2x	0x1FFE	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF	0x00FF	0x007F	0x003F	0x001F
4x	0x3FFC	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF	0x00FF	0x007F	0x003F
8x	0x7FF8	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF	0x00FF	0x007F
16x	0xFFF0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF	0x00FF
32x	0x1FFE0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF
64x	0x3FFC0	0xFFC0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF
128x	0x7FF80	0xFF80	0xFFC0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF
256x	0xFFF00	0xFF00	0xFF80	0xFFC0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF

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:

$$N \times t_{ADC} = N \times (t_{SMPL} + t_{CONV}) \quad (25\text{-}3)$$

25.5. ADC sync mode

In devices with two ADCs, the ADC sync mode can be used.

In ADC sync mode, the conversion of ADC1 are synchronized by the triggers of ADC0. The two ADCs convert parallelly or rotately, according to the mode selected by the SYNCM[3:0] 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. The converted data of routine sequence is stored in the ADC sync routine data register (ADC_SYNCDATA0 or ADC_SYNCDATA1).

The following modes can be configured in *Table 25-9. ADC sync mode table*.

Table 25-9. ADC sync mode table

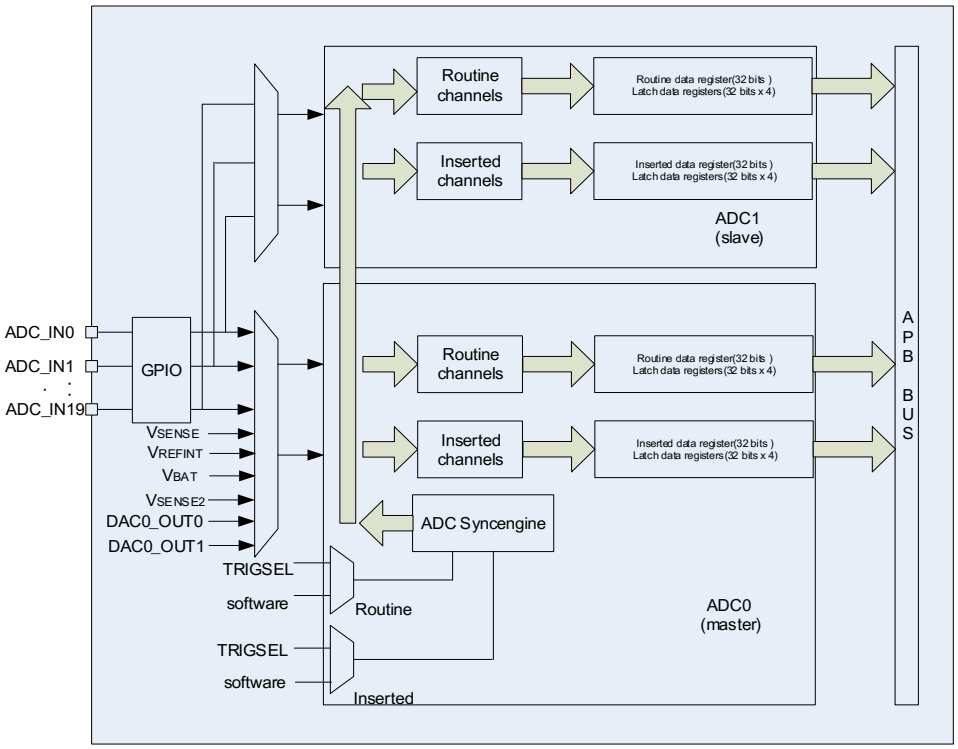
SYNCM[3: 0]	mode
0000	Free mode. All the ADCs work independently
0001	ADC0 and ADC1 work in combined routine parallel & inserted parallel mode
0010	ADC0 and ADC1 work in combined routine parallel & trigger rotation mode
0101	ADC0 and ADC1 work in inserted parallel mode
0110	ADC0 and ADC1 work in routine parallel mode
0111	ADC0 and ADC1 work in routine follow-up mode

SYNCM[3: 0]	mode
1001	ADC0 and ADC1 work in trigger rotation mode

When the ADCs are in a sync mode other than free mode, they should be configured to free mode before being configured to another sync mode.

The ADC sync scheme is shown in *Figure 25-21. ADC sync block diagram* .

Figure 25-21. ADC sync block diagram



25.5.1. Free mode

In this mode, each ADC works independently and does not interfere with each other.

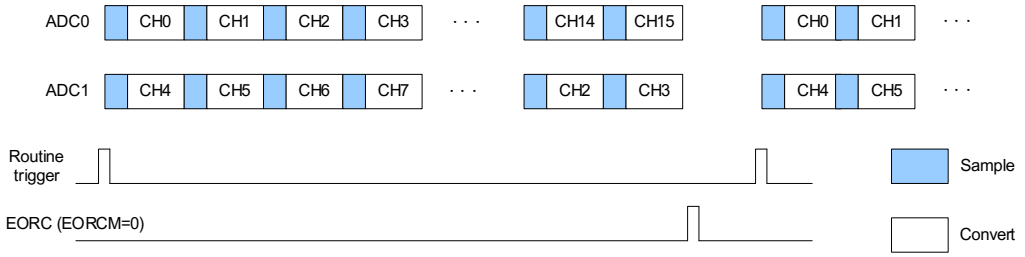
25.5.2. Routine parallel mode

The routine parallel mode is enabled by setting the SYNCM[3:0] bits in the ADC_SYNCCTL register to 4'b0110. 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.

EORC interrupts (if enabled on the ADC interfaces) are generated at the end of conversion events according to the EORCM bit in the ADC_CTL1 register. The behavior of routine

parallel mode is shown in the *Figure 25-22. Routine parallel mode on 16 channels* .

Figure 25-22. Routine parallel mode on 16 channels



Note:

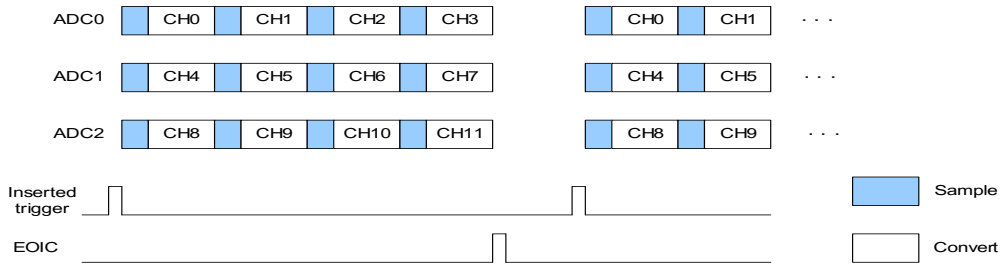
1. Do not convert the same channel on two ADCs at a given time (no overlapping sampling times for the ADCs when converting the same channel).
2. Make sure to trigger the ADCs when none of them is converting (do not trigger ADC0 when some of the conversions are not finished).
3. Make sure to do ADC peripheral reset before ADCON is set to 1.

25.5.3. Inserted parallel mode

The inserted parallel mode is enabled by setting the SYNCM[3:0] bits in the ADC_SYNCCTL register to 4'b0101. In the inserted parallel mode, all of the ADCs convert the inserted sequence parallelly at the selected external trigger of ADC0. The trigger is selected by configuring the ETMIC[1:0] bits in the ADC_CTL1 register of ADC0.

EOIC interrupts (if enabled on the ADC interfaces) are generated at the end of the inserted sequences. The behavior of inserted parallel mode is shown in the *Figure 25-23. Inserted parallel mode on 4 channels* .

Figure 25-23. Inserted parallel mode on 4 channels



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).

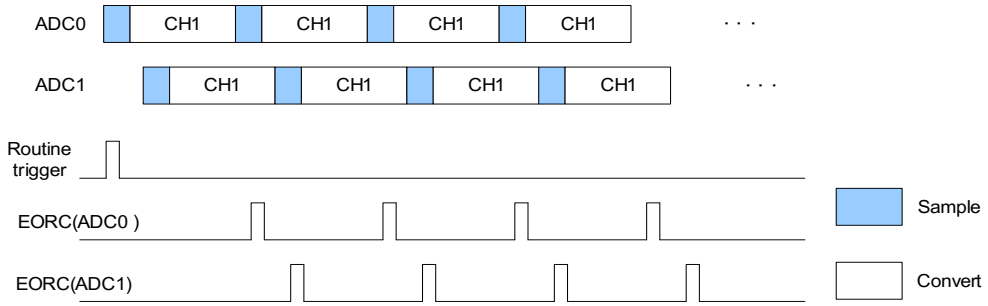
25.5.4. Routine follow-up mode

The routine follow-up mode is enabled by setting the SYNCM[3:0] bits in the ADC_SYNCCTL register to 4'b0111. 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. 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 CTN bit in ADC_CTL1 register is set, the selected routine channels are continuously converted. EORC interrupts (if enabled on the ADC interfaces) are generated at the end of conversion events according to the EORCM bit in the ADC_CTL1 register. The behavior of routine follow-up mode is shown in the *Figure 25-24. Routine follow-up mode on 1 channel in continuous operation mode* .

Figure 25-24. Routine follow-up mode on 1 channel in continuous operation mode



Note:

1. Make sure to trigger the ADCs when none of them is converting (do not trigger ADC0 when some of the conversions are not finished).
2. Make sure to do ADC peripheral reset before ADCON is set to 1.

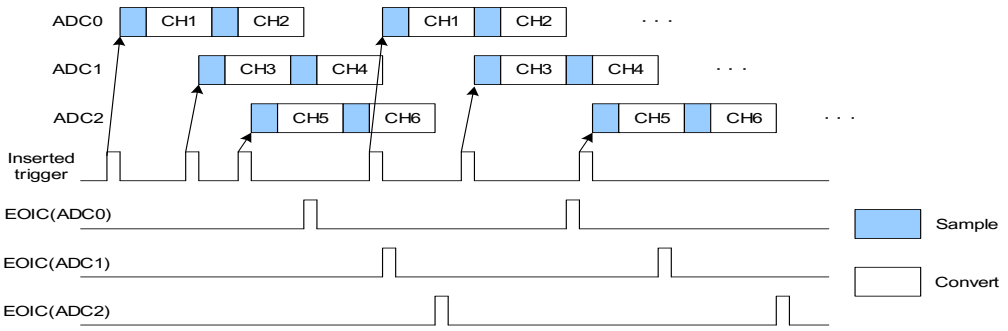
25.5.5. Trigger rotation mode

The trigger rotation mode is enabled by setting the SYNCM[3:0] bits in the ADC_SYNCCTL register to 4'b1001. In the trigger rotation mode, the inserted sequence of the ADCs is triggered in turn by the selected external trigger. If SYNCM=4'b1001, the selected external trigger triggers the ADCs in the sequence: ADC0->ADC1->ADC0->ADC1, and etc. The triggers are selected by configuring the ETMIC[1:0] bits in the ADC_CTL1 register of ADC0.

EOIC interrupts (if enabled on the ADC interfaces) are generated at the end of the inserted sequences. The behavior of trigger rotation mode DISIC bit is 0 is shown in the *Figure 25-25.*

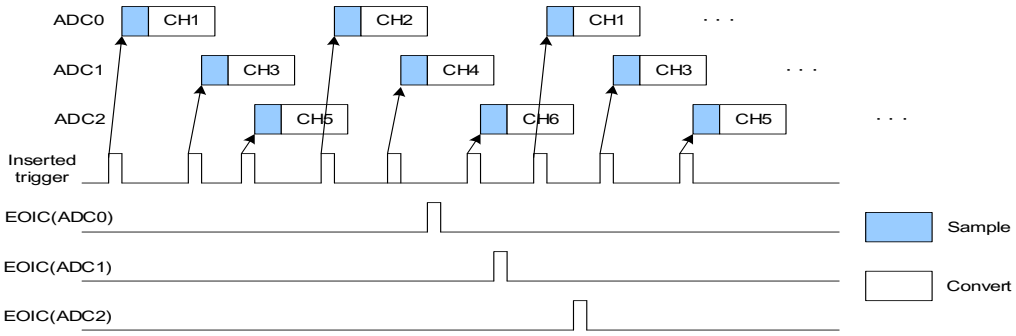
trigger rotation: DISIC=0, IL=1 .

Figure 25-25. trigger rotation: DISIC=0, IL=1



The behavior of trigger rotation mode is shown in the *Figure 25-26. trigger rotation: DISIC=1, IL=1* when DISIC bit is 1.

Figure 25-26. trigger rotation: DISIC=1, IL=1



Note:

1. Do not convert the same channel on two ADCs (no overlapping sampling times for the two ADCs when converting the same channel).
2. Make sure to do ADC peripheral reset before ADCON is set to 1.

25.5.6. Combined routine parallel & inserted parallel mode

The combined routine parallel & inserted parallel mode is enabled by setting the SYNCM[3:0] bits in the ADC_SYNCCTL register to 4'b0001. In the combined routine parallel & inserted parallel sync mode, it is also possible to interrupt parallel conversion of a routine sequence to insert parallel conversion of an inserted sequence.

EORC interrupts (if enabled on the ADC interfaces) are generated at the end of routine conversion events according to the EORCM bit in the ADC_CTL1 register.

EOIC interrupts (if enabled on the ADC interfaces) are generated at the end of the inserted sequences.

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).

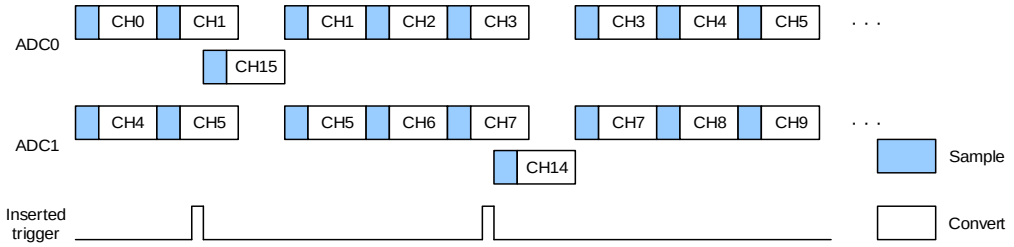
2. The inserted sequence length of the ADCs should be configured the same.
3. Make sure to do ADC peripheral reset before ADCON is set to 1.

25.5.7. Combined routine parallel & trigger rotation mode

The combined routine parallel & trigger rotation mode is enabled by setting the SYNCM[3:0] bits in the ADC_SYNCCTL register to 4'b0010. In the combined routine parallel & trigger rotation mode, it is possible to interrupt routine sequence parallel conversion by the inserted sequence conversion rotately. When the inserted event occurs, the inserted conversion is immediately started. When the routine conversions are interrupted, the routine conversion of the ADCs is stopped at the inserted trigger and resumed parallelly at the end of the inserted conversion. The behavior of routine parallel conversion interrupted by inserted triggers rotately is shown in the *Figure 25-27. Routine parallel & trigger rotation mode: SYNCM = 0010* EORC interrupts (if enabled on the ADC interfaces) are generated at the end of routine conversion events according to the EORCM bit in the ADC_CTL1 register.

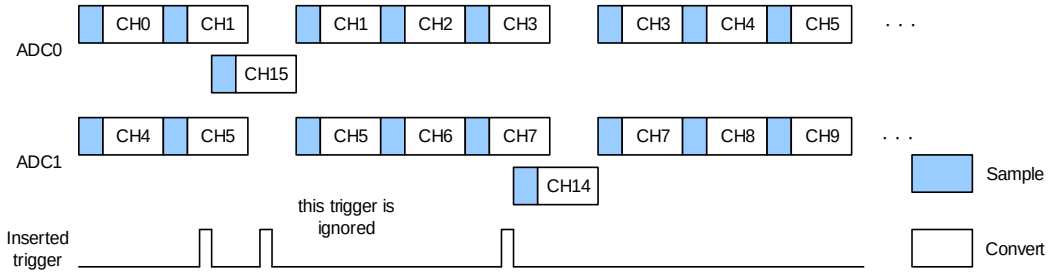
EOIC interrupts (if enabled on the ADC interfaces) are generated at the end of the inserted sequences.

Figure 25-27. Routine parallel & trigger rotation mode: SYNCM = 0010



If another inserted trigger occurs during an inserted conversion, the latter trigger will be ignored, as shown in *Figure 25-28. Trigger occurs during inserted conversion: SYNCM = 0010*.

Figure 25-28. Trigger occurs during inserted conversion: SYNCM = 0010



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. The inserted sequence length of the ADCs should be configured the same.
3. Make sure to do ADC peripheral reset before ADCON is set to 1.

25.5.8. Use DMA in ADC sync mode

In ADC sync mode, the converted data of routine channels are stored in the ADC sync routine data register (ADC_SYNCDATA0 or ADC_SYNCDATA1). DMA can be used to transfer data from ADC_SYNCDATA0 or ADC_SYNCDATA1 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 bitwidth 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_SYNCDATA1 register, while the content of the ADC_SYNCDATA changes to the data that is to be transferred. When ADC0 and ADC1 work in SYNC mode, the transfer sequence is ADC0_RDATA[31:0] -> ADC1_RDATA[31:0] -> ADC0_RDATA[31:0] -> ADC1_RDATA[31:0].

The ADC Sync DMA mode 0 is suitable for:

- ADC0 and ADC1 work in combined routine parallel & inserted parallel mode (SYNCM=4'b0001)
- ADC0 and ADC1 work in combined routine parallel & trigger rotation mode (SYNCM=4'b0010)
- ADC0 and ADC1 work in routine parallel mode (SYNCM=4'b0110)

ADC sync DMA mode 1

In ADC sync DMA mode 1, the bitwidth 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_SYNCDATA0 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]}.

The ADC Sync DMA mode 1 is properly for:

- ADC0 and ADC1 work in combined routine parallel & inserted parallel mode (SYNCM=4'b0001)
- ADC0 and ADC1 work in combined routine parallel & trigger rotation mode (SYNCM=4'b0010)
- ADC0 and ADC1 work in routine parallel mode (SYNCM=4'b0110)
- ADC0 and ADC1 work in follow-up mode (SYNCM=4'b0111)

25.6. ADC interrupts

The interrupt can be produced on one of the events:

- End of conversion for routine and inserted sequences
- The analog watchdog event
- Overflow event

Separate interrupt enable bits are available for flexibility.

The interrupts of ADC0 and ADC1 are mapped into the same interrupt vector IRQ18, ADC2 is mapped into the interrupt vector IRQ127.

25.7. Register definition

ADC0 base address: 0x4001 2400

ADC1 base address: 0x4001 2800

ADC2 base address: 0x4001 2C00

25.7.1. Status register (ADC_STAT)

Address offset: 0x00
Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WDE2	WDE1	Reserved													
rc_w0	rc_w0														
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									IOVF	ROVF	STRC	STIC	EOIC	EORC	WDE0
									rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

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_WDHT2 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_WDHT1 register. Cleared by software writing 0 to it.
29:7	Reserved	Must be kept at reset value.
6	IOVF	Inserted data register overflow flag 0: Inserted data register not overflow 1: Inserted data register overflow Set by hardware when the inserted data register overflow. This flag is only set when IDMA = 1. The recent inserted data is lost when this bit is set. Cleared by software writing 0 to it.
5	ROVF	Routine data register overflow flag 0: Routine data register not overflow 1: Routine data register overflow

		Set by hardware when the routine data register overflow. This flag is only set when RDMA = 1 or EORCM = 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	STIC	Start flag of inserted sequence 0: Conversion is not started 1: Conversion is started Set by hardware when inserted sequence conversion starts. Cleared by software writing 0 to it.
2	EOIC	End of inserted sequence conversion flag 0: No end of inserted sequence conversion 1: End of inserted sequence conversion Set by hardware at the end of all inserted sequence conversion. Cleared by software writing 0 to it.
1	EORC	End of routine sequence conversion flag 0: No end of routine sequence conversion 1: End of routine sequence conversion Set by hardware at the end of a routine sequence conversion. Cleared by software writing 0 to it or by reading the ADC_RDATA register.
0	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_WDLT0 and ADC_WDHT0 registers. Cleared by software writing 0 to it.

25.7.2. 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		IOVFIE	ROVFIE	DRES[1:0]		RWD0EN	IWD0EN	Reserved					
rw	rw			rw	rw	rw		rw	rw						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DISNUM[2:0]			DISIC	DISRC	ICA	WD0SC	SM	EOICIE	WDE0IE	EORCIE	WD0CHSEL[4:0]				
rw			rw	rw	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
29:28	Reserved	Must be kept at reset value.
27	IOVFIE	Interrupt enable for IOVF 0: IOVF interrupt disable 1: IOVF interrupt enable
26	ROVFIE	Interrupt enable for ROVF 0: ROVF interrupt disable 1: ROVF interrupt enable
25:24	DRES[1:0]	ADC data resolution for ADC0/ADC1 00: 14bit 01: 12bit 10: 10bit 11: 8bit ADC data resolution for ADC2 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	IWD0EN	Inserted channel analog watchdog 0 enable 0: Inserted channel analog watchdog 0 disable 1: Inserted channel analog watchdog 0 enable
21:16	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	DISIC	Discontinuous mode on inserted channels 0: Discontinuous operation mode disable 1: Discontinuous operation mode enable
11	DISRC	Discontinuous mode on routine channels

		0: Discontinuous operation mode disable 1: Discontinuous operation mode enable
10	ICA	Inserted sequence convert automatically 0: Inserted sequence convert automatically disable 1: Inserted sequence convert automatically enable
9	WD0SC	When in scan mode, analog watchdog 0 is effective on a single channel 0: 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	EOICIE	Interrupt enable for EOIC 0: Interrupt disable 1: Interrupt enable
6	WDE0IE	Interrupt enable for WDE0 0: Interrupt disable 1: Interrupt enable
5	EORCIE	Interrupt enable for EORC 0: 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 Channel20 and Channel21 are internally connected to the DAC0_OUT0 and V_{REFINT} inputs. ADC1 analog inputs Channel16, Channel17, Channel20 are internally connected to the battery, V_{REFINT} inputs and DAC0_OUT1. ADC2 analog inputs Channel17, Channel18, Channel20 are internally connected to the battery, temperature sensor and high-precision temperature sensor.

25.7.3. 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]		Reserved		VBATEN	Reserved	TSVEN1	SWICST	ETMIC[1:0]		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		IDDM	HPDFCFG	DAL	EORCM	RDDM	RDMA	IDMA	CALNUM[2:0]			RSTCLB	CLB	CTN	ADCON
		rw	rw	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 20(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 10: Falling edge of external trigger for routine channel enable 11: Rising and falling edge of external trigger for routine channel enable
27:26	Reserved	Must be kept at reset value.
25	VBATEN	This bit can be set or cleared by software in ADC2. Channel 16 (1/4 voltage of external battery) enable of ADC1 Channel 17 (1/4 voltage of external battery) enable of ADC2

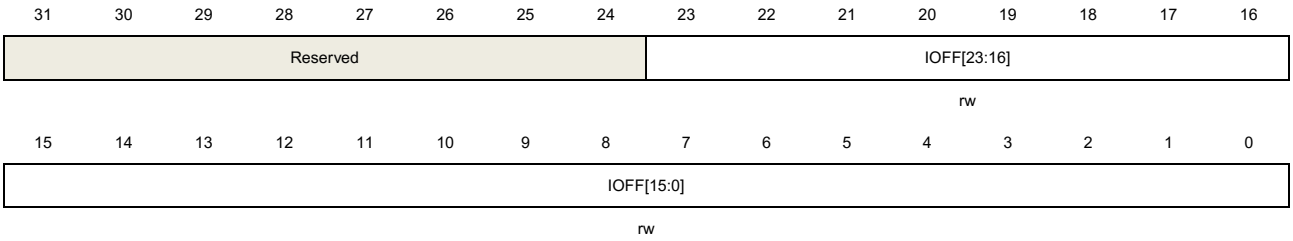
		0: 1/4 voltage of external battery Channel disable 1: 1/4 voltage of external battery Channel enable
24	Reserved	Must be kept at reset value.
23	TSVEN1	This bit can be set or cleared by software in ADC2. Channel 18(temperature sensor) enable of ADC2. 0: temperature sensor Channel disable 1: temperature sensor Channel enable
22	SWICST	Software start on inserted channel. Setting 1 on this bit starts a conversion of a sequence of inserted channels. It is set by software and cleared by software or by hardware immediately after the conversion starts.
21:20	ETMIC[1:0]	External trigger mode for inserted channel 00: External trigger for inserted channel disable 01: Rising edge of external trigger for inserted channel enable 10: Falling edge of external trigger for inserted channel enable 11: Rising and falling edge of external trigger for inserted channel enable
19:14	Reserved	Must be kept at reset value.
13	IDDM	Inserted DMA disable mode This bit configure 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 Inserted conversion.
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, if OVSEN enable,the ADC can only be LSB alignment. 0: LSB alignment 1: MSB alignment
10	EORCM	End of routine sequence conversion mode 0: Only at the end of routine sequence conversion, the EORC bit is set. Overflow detection is disabled unless RDMA=1. 1: At the end of each routine conversion, the EORC bit is set. Overflow is detected automatically
9	RDDM	Routine 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	RDMA	Routine DMA request enable. 0: DMA request disable 1: DMA request enable
7	IDMA	Inserted DMA request enable. 0: DMA request disable 1: DMA request enable
6:4	CALNUM[2:0]	Calibration Times These bits define the calibration times for ADC0/1. 000:1 time 001:3 times 010:7 times 011:15 times 1xx:31 times These bits define the calibration times for ADC2. 000:1 time 010:3 times 100:7 times 110:15 times 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

25.7.4. Inserted channel data offset register x(ADC_IOFFx) (x=0…3)

Address offset: 0x0C-0x18
Reset value: 0x0000 0000

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

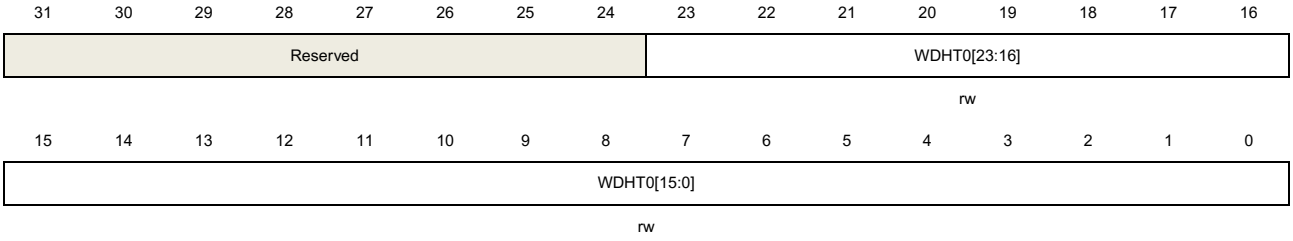


Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:0	IOFF[23:0]	Data offset for inserted channel x, For ADC0/ADC1 are IOFF[23:0], for ADC2 is IOFF[11:0]. And in ADC2, if OVSEN bit enable, offset disable. These bits will be subtracted from the inserted channel converted data, and the offset result are saved in the ADC_IDATA registers.

25.7.5. Watchdog high threshold register0 (ADC_WDHT0)

Address offset: 0x1C
Reset value: 0x00FF FFFF

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

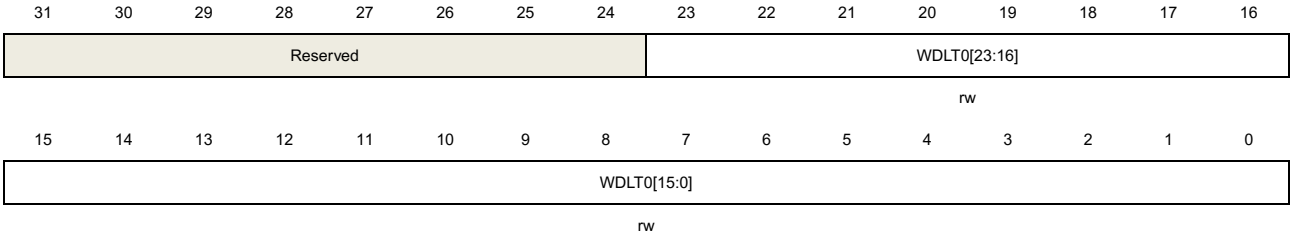


Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:0	WDHT0[23:0]	High threshold for analog watchdog 0, For ADC0/ADC1 are WDHT0[23:0], for ADC2 is WDHT0[11:0]. These bits define the high threshold for the analog watchdog 0.

25.7.6. Watchdog low threshold register0 (ADC_WDLT0)

Address offset: 0x20
Reset value: 0x0000 0000

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



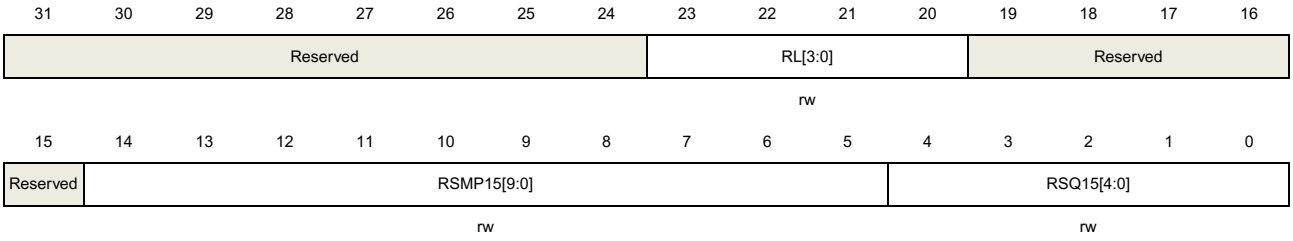
Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:0	WDLT0[23:0]	Low threshold for analog watchdog 0. For ADC0/ADC1 are WDLT0[23:0], for ADC2 is WDLT0[11:0]. These bits define the low threshold for the analog watchdog.

25.7.7. 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 channel 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

25.7.8. Routine sequence register 1 (ADC_RSQ1)

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	RSMP14[9:0]										RSQ14[4:0]				
	rw										rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	RSMP13[9:0]										RSQ13[4:0]				
	rw										rw				

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

25.7.9. Routine sequence register 2 (ADC_RSQ2)

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	RSMP12[9:0]										RSQ12[4:0]				
	rw										rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	RSMP11[9:0]										RSQ11[4:0]				
	rw										rw				

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

25.7.10. Routine sequence register 3 (ADC_RSQ3)

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	RSMP10[9:0]										RSQ10[4:0]				
	rw										rw				

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

25.7.11. 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

25.7.12. 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

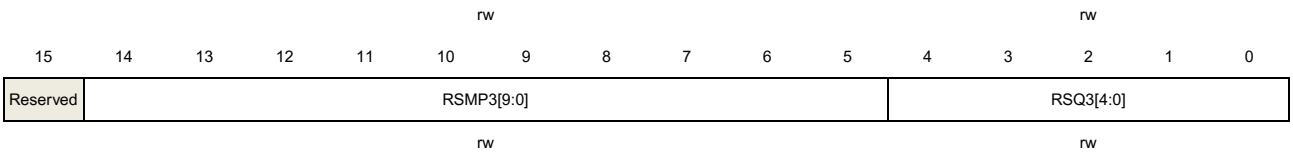
25.7.13. Routine sequence register 6 (ADC_RSQ6)

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]				



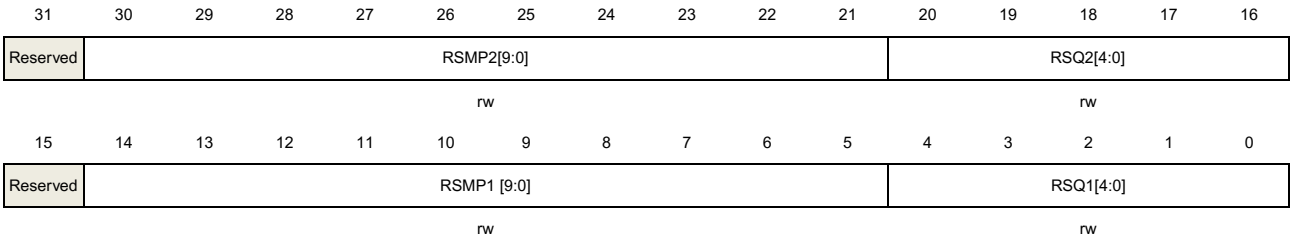
Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30:21	RSMP4[9:0]	Routine channel sample time Refer to RSMP0[9:0] description.
20:16	RSQ4[4:0]	refer to RSQ0[4:0] description
15	Reserved	Must be kept at reset value.
14:5	RSMP3[9:0]	Routine channel sample time Refer to RSMP0[9:0] description.
4:0	RSQ3[4:0]	refer to RSQ0[4:0] description

25.7.14. Routine sequence register 7 (ADC_RSQ7)

Address offset: 0x40

Reset value: 0x0000 0000

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



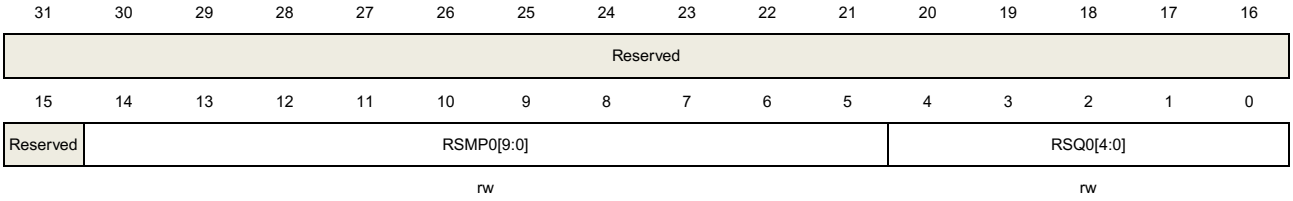
Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30:21	RSMP2[9:0]	Routine channel sample time Refer to RSMP0[9:0] description.
20:16	RSQ2[4:0]	refer to RSQ0[4:0] description
15	Reserved	Must be kept at reset value.
14:5	RSMP1[9:0]	Routine channel sample time Refer to RSMP0[9:0] description.
4:0	RSQ1[4:0]	refer to RSQ0[4:0] description

25.7.15. Routine sequence register 8 (ADC_RSQ8)

Address offset: 0x44

Reset value: 0x0000 0000

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



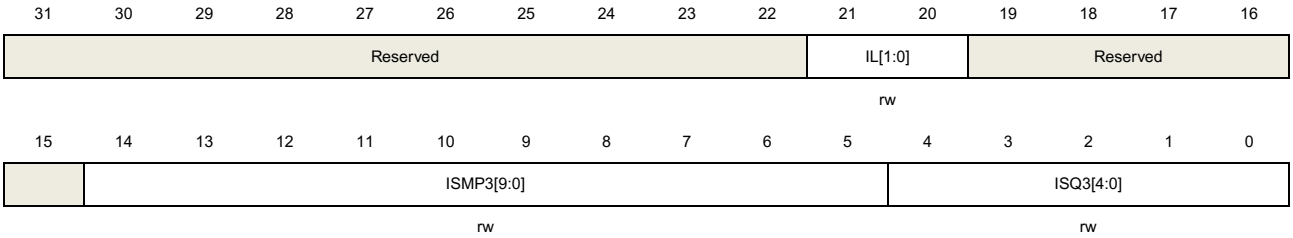
Bits	Fields	Descriptions
31:15	Reserved	Must be kept at reset value.
14:5	RSMP0[9:0]	Routine channel sample time ADC sample time is RSMP0[9:0] + 3.5 cycles. For example, ADC0 RSMP0[9:0] = 100, the sample time is 103.5 cycles. ADC RSMP0[9:0] max is 807, all other values are reserved. **Note**: If ADC RSMPx[9:0]>807 , 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.

25.7.16. Inserted sequence register 0 (ADC_ISQ0)

Address offset: 0x48

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:22	Reserved	Must be kept at reset value.
21:20	IL[1:0]	Inserted sequence length. The total number of conversion in Inserted sequence equals to IL[1:0] + 1.
19:15	Reserved	Must be kept at reset value
14:5	ISMP3[9:0]	Inserted channel sample time

		Refer to ISMP0[9:0] description.
4:0	ISQ3[4:0]	refer to ISQ0[4:0] description

25.7.17. Inserted sequence register 1 (ADC_ISQ1)

Address offset: 0x4C

Reset value: 0x0000 0000

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



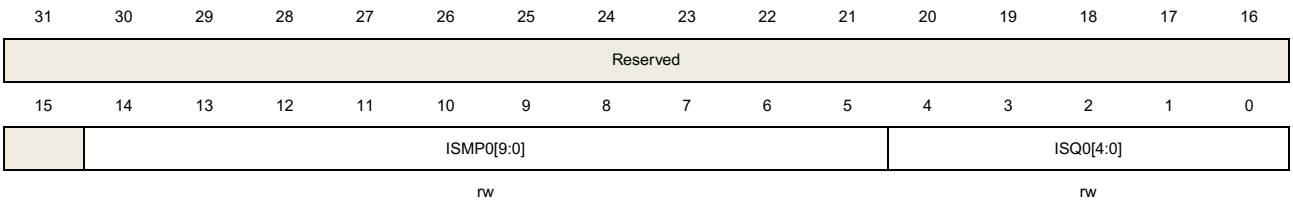
Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30:21	ISMP2[9:0]	Inserted channel sample time Refer to ISMP0[9:0] description.
20:16	ISQ2[4:0]	refer to ISQ0[4:0] description
15	Reserved	Must be kept at reset value.
14:5	ISMP1[9:0]	Inserted channel sample time Refer to ISMP0[9:0] description.
4:0	ISQ1[4:0]	refer to ISQ0[4:0] description.

25.7.18. Inserted sequence register 2 (ADC_ISQ2)

Address offset: 0x50

Reset value: 0x0000 0000

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



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

14:5	ISMP0[9:0]	Inserted channel sample time ADC sample time is ISMP0[9:0] + 3.5 cycles. For example, ADC0 ISMP0[9:0] = 100, the sample time is 103.5 cycles. ADC ISMP0[9:0] max is 807, all other values are reserved. **Note**: If ADC ISMPx[9:0]>807 , the work behavior in adc will not be guaranteed.
4:0	ISQ0[4:0]	The channel number (0..21) is written to these bits to select a channel at the nth conversion in the inserted sequence. Unlike the routine conversion sequence, the inserted channels are converted starting from (4 - IL[1:0] - 1), if IL[1:0] length is less than 4.

IL	Insert channel order
3	ISQ0 >> ISQ1 >> ISQ2 >> ISQ3
2	ISQ1 >> ISQ2 >> ISQ3
1	ISQ2 >> ISQ3
0	ISQ3

25.7.19. Latch data register x (ADC_LDATAx) (x=0...3)

Address offset: 0x54 - 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
LDATAn[31:16]															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LDATAn[15:0]															
r															

Bits	Fields	Descriptions
31:0	LDATAn[31:0]	Inserted or routine number n conversion data. For ADC0/ADC1, LDATAn is [31:0], for ADC2, LDATAn is [15:0]. These bits contain the number n conversion result, which is read only.

25.7.20. Routine data register (ADC_RDATA)

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

RDATA[15:0]
r

Bits	Fields	Descriptions
31:0	RDATA[31:0]	Routine data. For ADC0/ADC1, RDATA is [31:0], for ADC2, RDATA is [15:0]. These bits contain the conversion result from routine sequence, which is read only.

25.7.21. Inserted data register (ADC_IDATA)

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

Bits	Fields	Descriptions
31:0	IDATA[31:0]	Inserted data. For ADC0/ADC1, IDATA is [31:0], for ADC2, IDATA is [15:0]. These bits contain the conversion result from inserted sequence, which is read only.

25.7.22. Latch data control register (ADC_LDCTL)

Address offset: 0x6C
Reset value: 0x0001 0203

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SEQSEL0	Reserved			COVSEL0[3:0]				SEQSEL1	Reserved			COVSEL1[3:0]			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEQSEL2	Reserved			COVSEL2[3:0]				SEQSEL3	Reserved			COVSEL3[3:0]			
rw				rw				rw				rw			

Bits	Fields	Descriptions
31	SEQSEL0	The ADC_LDATA0 register sequence source selection 0: select inserted sequence

		1: select routine sequence **Note:** The software allows this bit to be written only when ADCON=0 (this ensures that no conversion is progress).
30:28	Reserved	Must be kept at reset value.
27:24	COVSEL0[3:0]	The ADC_LDATA0 register conversion source selection When ADC_LDATA0 register is used for sequence conversion, the nth conversion data of the inserted sequence (SEQSEL0 = 0) or the routine sequence (SEQSEL0 = 1) can be selected to be stored in ADC_LDATA0 register. 0000: The 0th conversion of the sequence. 0001: The 1st conversion of the sequence. 0010: The 2nd conversion of the sequence. … 1111: the 15th conversion of the sequence. Other values are reserved. **Note:** When used in inserted sequence (SEQSEL0 = 0), it can only be 0000,0001,0010,0011, and the COVSELn[3:0] default value is n(n=0~3). The software allows this bit to be written only when ADCON=0 (this ensures that no conversion is progress).
23	SEQSEL1	Refer to SEQSEL0 description.
22:20	Reserved	Must be kept at reset value.
19:16	COVSEL1[3:0]	Refer to COVSEL0[3:0] description.
15	SEQSEL2	Refer to SEQSEL0 description.
14:12	Reserved	Must be kept at reset value.
11:8	COVSEL2[3:0]	Refer to COVSEL0[3:0] description.
7	SEQSEL3	Refer to SEQSEL0 description.
6:4	Reserved	Must be kept at reset value.
3:0	COVSEL3[3:0]	Refer to COVSEL0[3:0] description.

25.7.23. Oversample control register (ADC_OVSAMPCTL)

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

Reserved	TOVS	OVSS[3:0]	Reserved	OVSEN
	rw	rw		rw

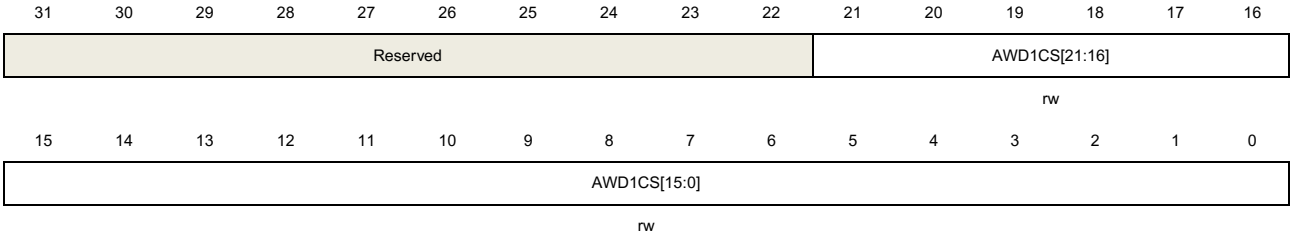
Bits	Fields	Descriptions
31:26	Reserved	Must be kept at reset value.
25:16	OVSR[9:0]	Oversampling ratio This bit filed defines the number of oversampling ratio. ADC0/1 is 1x~1024x. ADC2 is 1X~256X. 10'd0: 1x(no oversampling) 10'd1: 2x 10'd2: 3x …… 10'd1023:1024x **Note:** The software allows this bit 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. For ADC0/1, OVSS change from 0000~1011. For ADC2, OVSS change from 0000~1000. 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	Oversampler Enable This bit is set and cleared by software. 0: Oversampler disabled 1: Oversampler enabled **Note:** The software allows this bit to be written only when ADCON = 0 (this ensures that no conversion is in progress).

25.7.24. Watchdog 1 Channel Selection Register (ADC_WD1SR)

Address offset: 0xA0
Reset value: 0x0000 0000

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

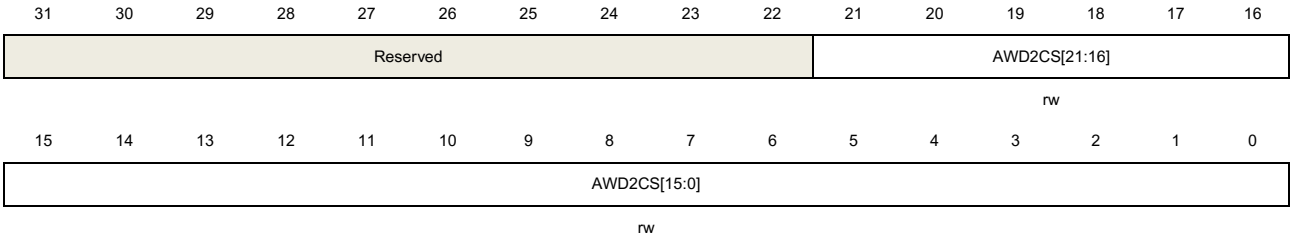


Bits	Fields	Descriptions
31:22	Reserved	Must be kept at reset value.
21:0	AWD1CS[21:0]	Analog watchdog 1 channel selection These bits are set and cleared by software. They enable and select the input channels to be guarded by the analog watchdog 1. AWD1CS[n] = 0: ADC analog input channel n is not monitored 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 or ADC_ISQ registers. 2) Software is allowed to write these bits only when the ADC is disabled (ADCON =0).

25.7.25. Watchdog 2 Channel Selection Register (ADC_WD2SR)

Address offset: 0xA4
Reset value: 0x0000 0000

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

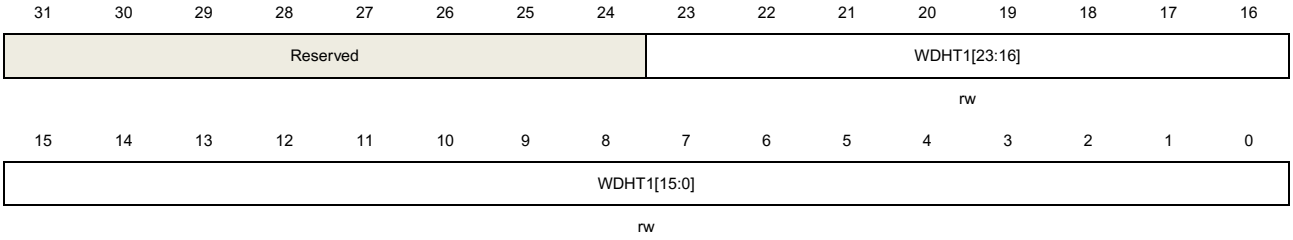


Bits	Fields	Descriptions
31:22	Reserved	Must be kept at reset value.
21:0	AWD2CS[21:0]	Analog watchdog 2 channel selection These bits are set and cleared by software. They enable and select the input channels to be guarded by the analog watchdog 2. AWD2CS[n] = 0: ADC analog input channel n is not monitored 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 or ADC_ISQ registers. 2) Software is allowed to write these bits only when the ADC is disabled (ADCON =0).

25.7.26. Watchdog high threshold register1 (ADC_WDHT1)

Address offset: 0xA8
Reset value: 0x00FF FFFF

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



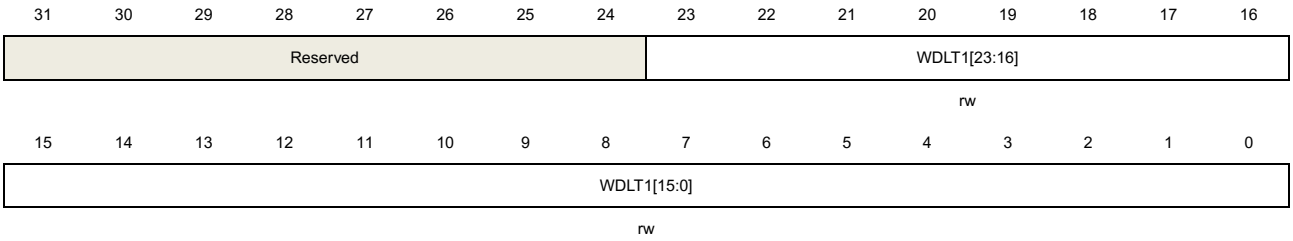
Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:0	WDHT1[23:0]	High threshold for analog watchdog 1. For ADC0/1 are WDHT1[23:0], for ADC2 is WDHT1[7:0]. These bits define the high threshold for the analog watchdog 1. **Note:** Software is allowed to write these bits only when the ADC is disabled (ADCON =0).

25.7.27. Watchdog low threshold register1 (ADC_WDLT1)

Address offset: 0xAC

Reset value: 0x0000 0000

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



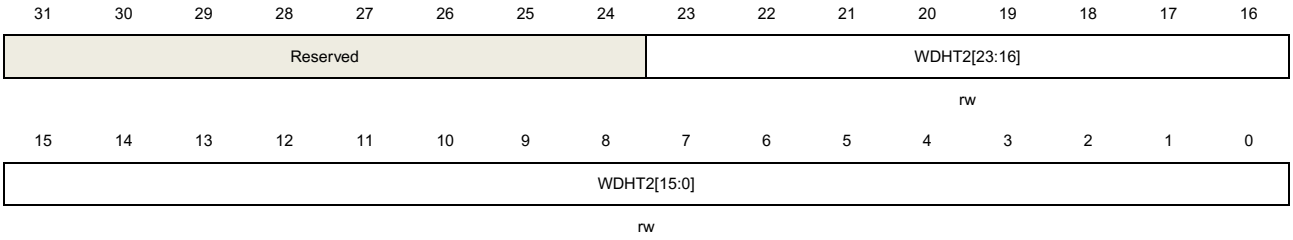
Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:0	WDLT1[23:0]	Low threshold for analog watchdog 1. For ADC0/1 are WDLT1[23:0], for ADC2 is WDLT1[7:0]. These bits define the low threshold for the analog watchdog 1. **Note:** Software is allowed to write these bits only when the ADC is disabled (ADCON =0).

25.7.28. Watchdog high threshold register2 (ADC_WDHT2)

Address offset: 0xB0

Reset value: 0x00FF FFFF

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

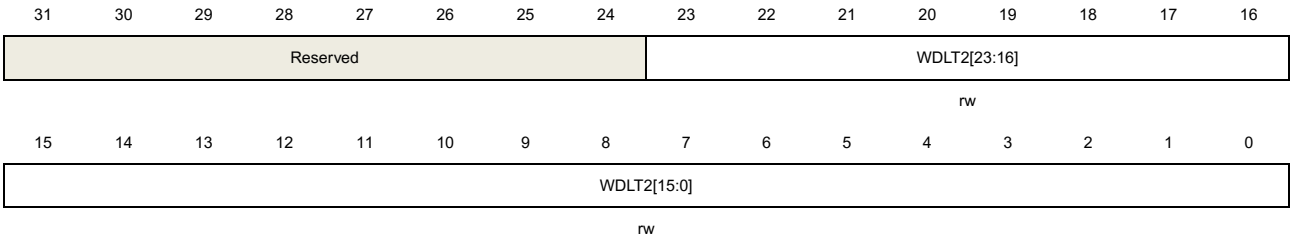


Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:0	WDHT2[23:0]	High threshold for analog watchdog 2. For ADC0/1 are WDHT2[23:0], for ADC2 is WDHT2[7:0]. These bits define the high threshold for the analog watchdog 2. **Note:** Software is allowed to write these bits only when the ADC is disabled (ADCON =0).

25.7.29. Watchdog low threshold register2 (ADC_WDLT2)

Address offset: 0xB4
Reset value: 0x0000 0000

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

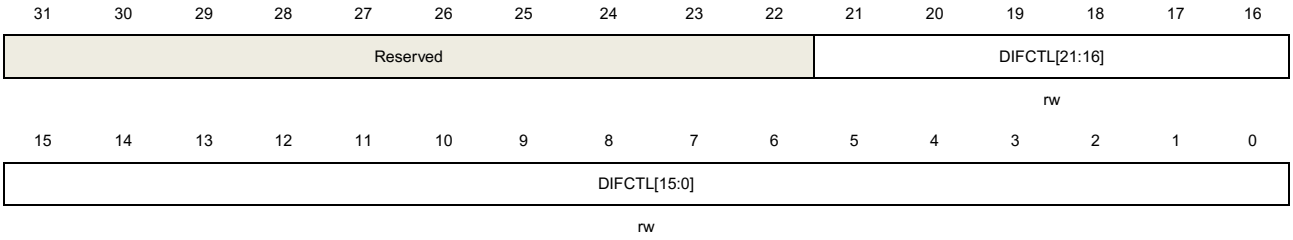


Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:0	WDLT2[23:0]	Low threshold for analog watchdog 2. For ADC0/1 are WDLT2[23:0], for ADC2 is WDLT2[7:0]. These bits define the low threshold for the analog watchdog 2. **Note:** Software is allowed to write these bits only when the ADC is disabled (ADCON =0).

25.7.30. Differential mode control register (ADC_DIFCTL)

Address offset: 0xB8
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.
21:0	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).

25.7.31. 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 and ADC2.

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved					ADC2_IOVF	ADC1_IOVF	ADC0_IOVF	ADC2_ROVF	ADC2_STRC	ADC2_STIC	ADC2_EOIC	ADC2_EORC	ADC2_WDE2	ADC2_WDE1	ADC2_WDE0
								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	ADC1_STIC	ADC1_EOIC	ADC1_EORC	ADC1_WDE2	ADC1_WDE1	ADC1_WDE0	ADC0_ROVF	ADC0_STRC	ADC0_STIC	ADC0_EOIC	ADC0_EORC	ADC0_WDE2	ADC0_WDE1	ADC0_WDE0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

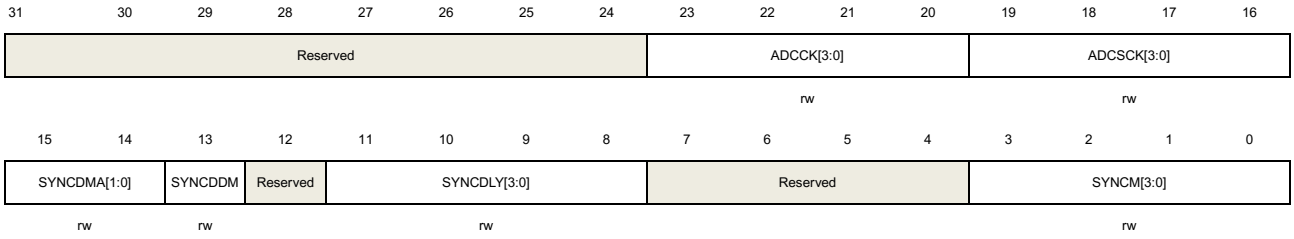
Bits	Fields	Descriptions
31:27	Reserved	Must be kept at reset value.
26	ADC2_IOVF	This bit is the mirror image of the IOVF bit of ADC2
25	ADC1_IOVF	This bit is the mirror image of the IOVF bit of ADC1
24	ADC0_IOVF	This bit is the mirror image of the IOVF bit of ADC0
23	ADC2_ROVF	This bit is the mirror image of the ROVF bit of ADC2
22	ADC2_STRC	This bit is the mirror image of the STRC bit of ADC2
21	ADC2_STIC	This bit is the mirror image of the STIC bit of ADC2
20	ADC2_EOIC	This bit is the mirror image of the EOIC bit of ADC2
19	ADC2_EORC	This bit is the mirror image of the EORC 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	ADC1_STIC	This bit is the mirror image of the STIC bit of ADC1
12	ADC1_EOIC	This bit is the mirror image of the EOIC bit of ADC1
11	ADC1_EORC	This bit is the mirror image of the EORC 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	ADC0_STIC	This bit is the mirror image of the STIC bit of ADC0
4	ADC0_EOIC	This bit is the mirror image of the EOIC bit of ADC0
3	ADC0_EORC	This bit is the mirror image of the EORC 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

25.7.32. Sync control register (ADC_SYNCCTL)

Address offset: 0x304 (for ADC0 / ADC2 base address)
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	ADCCK[3:0]	ADC clock prescaler. These bits are set and cleared by software to select the asynchronous clock frequency division. 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 (only for ADC0) 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 (only for ADC0) 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 (only for ADC0) These bits are used to configure the delay between 2 sampling phases in ADC sync modes to (5+SYNCDLY) ADC clock cycles.
7:4	Reserved	Must be kept at reset value.
3:0	SYNCM[3:0]	ADC sync mode (only for ADC0) When ADC sync mode is enabled these bits should be set to 0000 firstly before change to another value. 0000: Free mode. All the ADCs work independently. 0001: ADC0 and ADC1 work in combined routine parallel & inserted parallel mode. ADC2 works independently.

0010: ADC0 and ADC1 work in combined routine parallel & trigger rotation mode.
0101: ADC0 and ADC1 work in inserted parallel mode.
0110: ADC0 and ADC1 work in routine parallel mode.
0111: ADC0 and ADC1 work in follow-up mode.
1001: ADC0 and ADC1 work in trigger rotation mode.
All other values are reserved.

25.7.33. Sync routine data register0 (ADC_SYNCDATA0)

Address offset: 0x308 (for ADC0 base address)
Reset value: 0x0000 0000

This register 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[15:0]															
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[15:0]	Routine data1(slave adc routine data) in ADC sync mode. SYNCDMA[1:0] must be 2'b10.
15:0	SYNCDATA0[15:0]	Routine data0 (master adc routine data) in ADC sync mode. SYNCDMA[1:0] must be 2'b10,

25.7.34. Sync routine data register1 (ADC_SYNCDATA1)

Address offset: 0x30C (for ADC0 base address)
Reset value: 0x0000 0000

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

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

Bits	Fields	Descriptions
31:0	SYNCDATA[31:0]	SYNCDMA[1:0]=2'b01, which is selected from the routine data (master / slave) of the ADCs in turn.

26. Digital-to-analog converter (DAC)

26.1. 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. Sample and keep mode can reduce power consumption. Sample and keep mode can reduce power consumption.

The sample and keep mode can reduce the power consumption of DAC.

The DAC channels can work independently or concurrently.

26.2. Characteristics

The main features of DAC are as follows:

- 8-bit or 12-bit resolution.
- Left or right data alignment.
- DMA capability for each channel and underrun function.
- Conversion update synchronously.
- Conversion triggered by external triggers.
- Configurable internal buffer.
- Extern voltage reference, V_{REFP}.
- Output buffer calibration.
- Using sample and keep mode to reduce the power consumption.
- Noise wave generation (LFSR noise mode and Triangle noise mode).
- Two DAC channels in concurrent mode.

Figure 26-1. DAC block diagram and *Table 26-1. DAC I/O description* show the block diagram of DAC and the pin description of DAC, respectively.

Figure 26-1. DAC block diagram

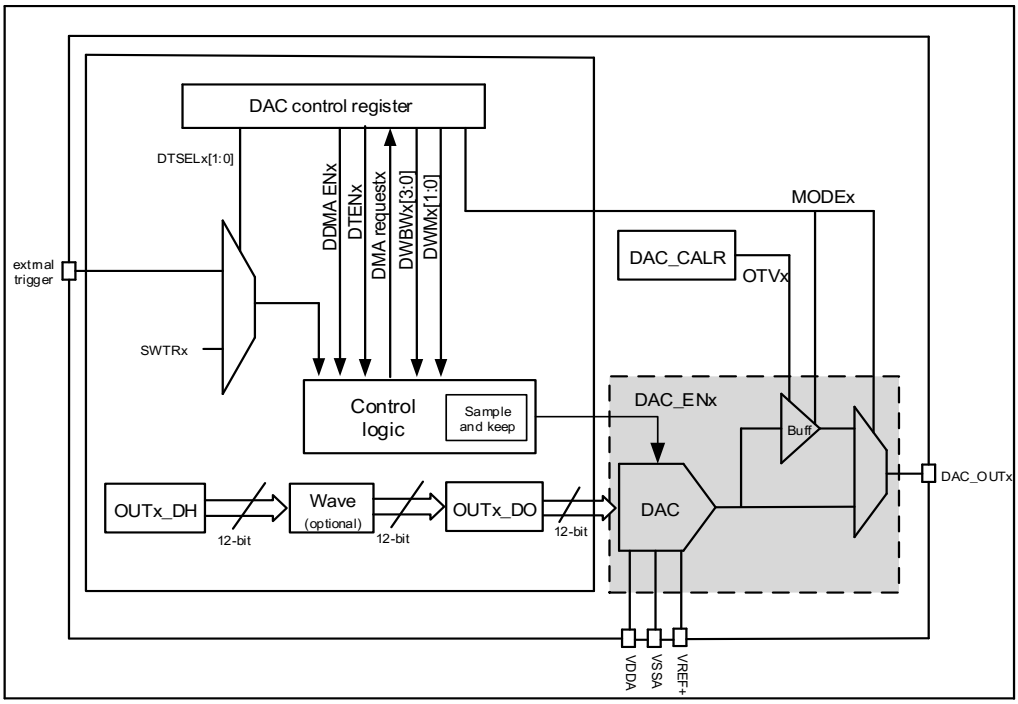


Table 26-1. DAC I/O description

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 26-2. DAC triggers and outputs summary

	DAC0	
Channel	**Channel0**	**Channel1**
DAC outputs connected to I / Os	PA4	PA5
DAC output buffer	•	•
DAC trigger signals from TRIGSEL	•	
DAC software trigger	•	

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

26.3. Function description

26.3.1. 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.

26.3.2. 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[2:0] bit in the DAC_MDCR register.

26.3.3. DAC data configuration

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

26.3.4. 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 26-3. Triggers of DAC***.

Table 26-3. Triggers of DAC

DTSELx[1:0]	Trigger Source	Trigger Type
2b'00	TRIGSEL	Hardware trigger
2b'01	SWTR	Software trigger
2b'10	Reserved	Reserved
2b'11		

The external trigger is generated from the TRIGSEL, while the software trigger can be generated by setting the SWTRx bits in the DAC_SWT register.

26.3.5. DAC conversion

If the external trigger is enabled by setting the DTENx bit in DAC_CTL0 register, the DAC holding data is transferred to the DAC output data (OUTx_DO) register when the selected

trigger event happened. When the external trigger is disabled, the transfer is performed automatically.

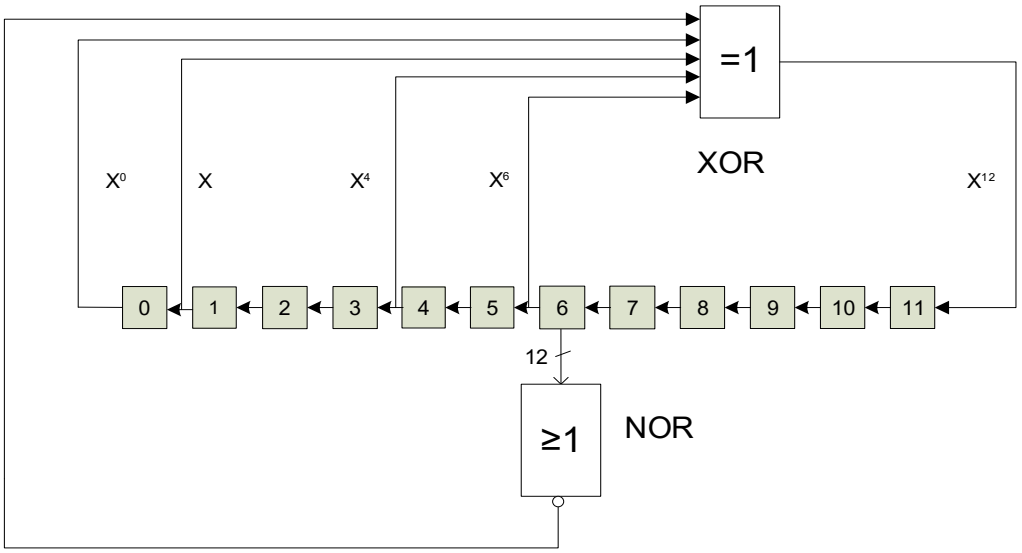
When the DAC holding data (OUTx_DH) is loaded into the OUTx_DO register, after the time $t_{SETTLING}$ which is determined by the analog output load and the power supply voltage, the analog output is valid.

26.3.6. 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.

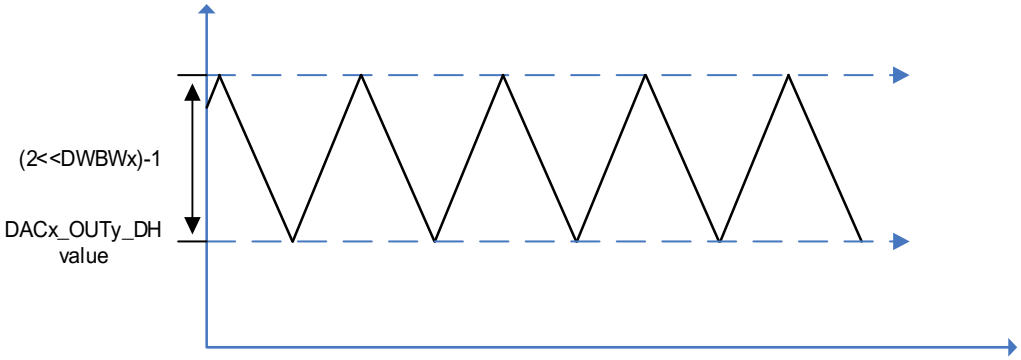
LFSR noise wave mode: there is a Linear Feedback Shift Register (LFSR) in the DAC control logic, it controls the LFSR noise signal which is added to the OUTx_DH value, and then the result is stored into the OUTx_DO register. When the configured DAC noise wave bit width is less than 12, the noise signal equals to the LSB DWBWx bits of the LFSR register, while the MSB bits are masked.

Figure 26-2. DAC LFSR algorithm



Triangle noise mode: a triangle signal is added to the OUTx_DH value, and then the result is stored into the 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 26-3. DAC triangle noise wave



26.3.7. DAC output voltage

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

$$V_{DAC_OUT}=V_{REFP}*OUTx_DO/4096 \quad (26\text{-}1)$$

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

26.3.8. DMA request

When the external trigger is enabled, the DMA request is enabled by setting the DDMAENx bit of the DAC_CTL0 register. A DMA request will be generated when an external hardware trigger (not a software trigger) occurs.

If the second external trigger arrives before confirming the previous request, the new request will not be serviced, and an underrun error event occurs. The DDUDRx bit in the DAC_STAT0 register is set, an interrupt will be generated if the DDUDRIEx bit in the DAC_CTL0 register is set. The DMA request will be stalled until the DDUDRx bit is cleared.

26.3.9. DAC concurrent conversion

When the two output channels work at the same time, for maximum bus bandwidth utilization in specific applications, two output channels can be configured in concurrent mode. In concurrent mode, the OUTx_DH and OUTx_DO value will be updated at the same time.

There are three concurrent registers that can be used to load the OUTx_DH value: DACC_R8DH, DACC_R12DH and DACC_L12DH. User just need to access a unique register to realize driving two DAC channels at the same time.

When external trigger is enabled, please ensure both DTENx bits be set, DTSEL0/DTSEL1 bits be same to guarantee the simultaneous trigger.

When DMA is enabled, please ensure any DDMAENx bit in one DAC be set.

The noise mode and noise bit width can be configured either the same or different, depending

on the application scenario.

26.3.10. DAC output buffer calibration

The output voltage may be offset when DAC use buffer, so it is necessary to compensate output voltage.

The DAC calibration transfer function is:

$$V_{out}=(D/2^{N-1})*G*V_{REFP}+V_{of} \quad (21\text{-}2)$$

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 external pin and on chip 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 OTVx[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} or temperature change, the user can calibrate at any point in the application process through the software.

The contents of OTVx[4:0] bits can be calculated in a faster way by using successive approximation or dichotomy techniques. The following process is successive approximation method:

- Writing 0 to DENx bit in DAC_CTL0 register to disable DAC output.
- Writing 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_{TRIM} 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.

Note: CALENx should write back to 0 after the calibration process, and then write DENx to 1 to use DAC at normal mode. It is forbidden to set DENx and CALENx to 1 at the same time.

26.3.11. DAC modes

DAC can be set to normal mode or sample and keep mode. The DAC out can be connected to external pin or on chip 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 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, APB1 and IRC32K drive 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 and the TSAMPx[9:0] bits can be changed by software during DAC out 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.

Time calculation

The calculation of the time for the three stages above are based on LXTAL/IRC32K clock periods. To configure enough sample and refresh time, refer to the following formula:

Table 26-4. Formula of sample and refresh time

Buffer State	t_{sample}	$t_{refresh}$	t_{keep}
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.

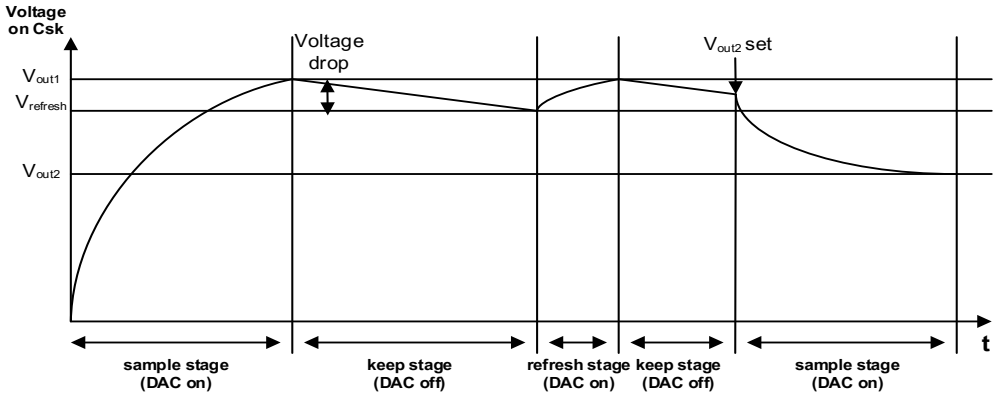
(2) R_{BON}/R_{BOFF} is the output impedence when buffer on or off, C_{SK} is the sample and keep capacitor (internal or external). When MODEx[2:0] bits in DAC_MDCR register are 3'b111, the internal capacitor is used to keep the DAC output voltage for on-chip peripherals.

(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 26-4. DAC sample and keep



26.3.12. 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 exiting from the standby mode, the DAC need to be reinitialized to work again.

26.4. DAC register

DAC0 base address: 0x4000 7400

26.4.1. DACx control register 0 (DAC_CTL0)

Address offset: 0x00
Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	CALEN1	DDUDRIE1	DDMAEN1	DWBW1[3:0]				DWM1[1:0]		Reserved		DTSEL1[1:0]		DTEN1	DEN1
	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	CALEN0	DDUDRIE0	DDMAEN0	DWBW0[3:0]				DWM0[1:0]		Reserved		DTSEL0[1:0]		DTEN0	DEN0
	rw	rw	rw	rw				rw				rw		rw	rw

Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
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 1x: Triangle 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: EXTRIG(external trigger from 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	Reserved	Must be kept at reset value.
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 1x: Triangle 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: EXTRIG(external trigger from 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

26.4.2. DACx software trigger register (DAC_SWT)

Address offset: 0x04

Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														SWTR1	SWTR0
														w	w

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	SWTR1	DACx_OUT1 software trigger, cleared by hardware. 0: Software trigger disabled 1: Software trigger enabled
0	SWTR0	DACx_OUT0 software trigger, cleared by hardware. 0: Software trigger disabled 1: Software trigger enabled

26.4.3. 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.

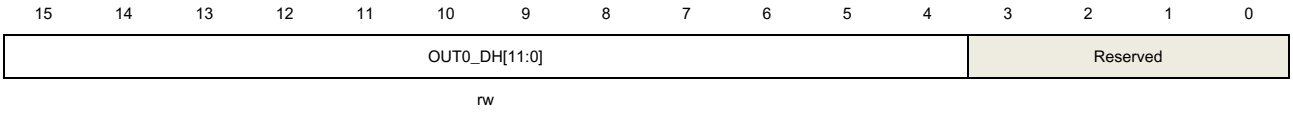
26.4.4. DACx_OUT0 12-bit left-aligned data holding register (DAC_OUT0_L12DH)

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															



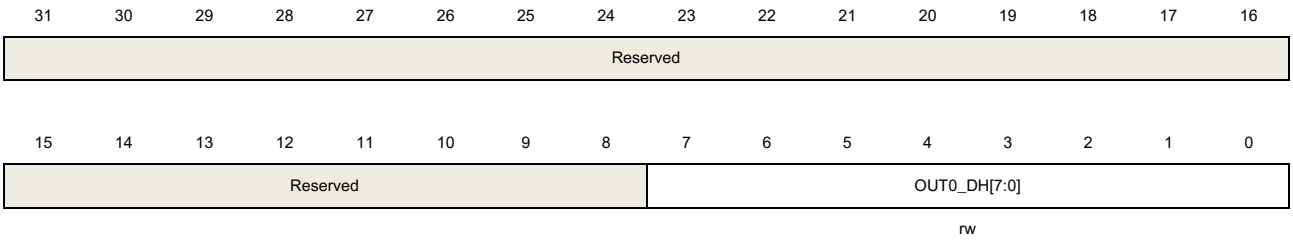
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:4	OUT0_DH[11:0]	DACx_OUT0 12-bit left-aligned data. These bits specify the data that is to be converted by DACx_OUT0.
3:0	Reserved	Must be kept at reset value.

26.4.5. DACx_OUT0 8-bit right-aligned data holding register (DAC_OUT0_R8DH)

Address offset: 0x10

Reset value: 0x0000 0000

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



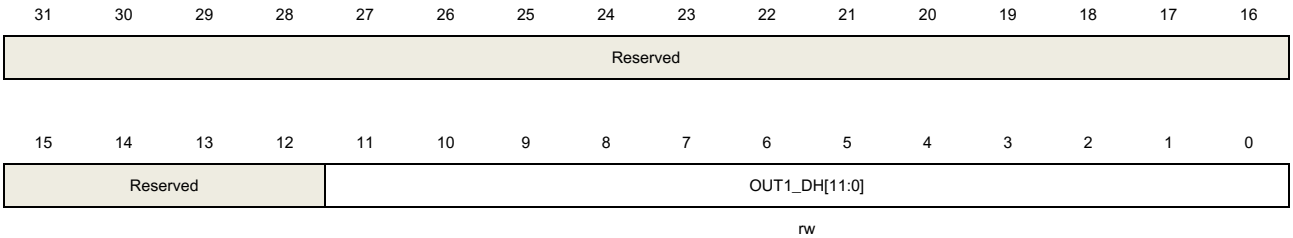
Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	OUT0_DH[7:0]	DACx_OUT0 8-bit right-aligned data. These bits specify the MSB 8-bit of the data that is to be converted by DACx_OUT0.

26.4.6. 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).



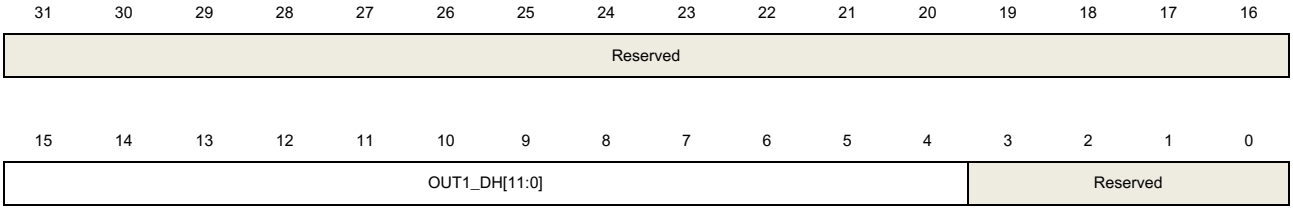
Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11:0	OUT1_DH[11:0]	DACx_OUT1 12-bit right-aligned data. These bits specify the data that is to be converted by DACx_OUT1.

26.4.7. 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).



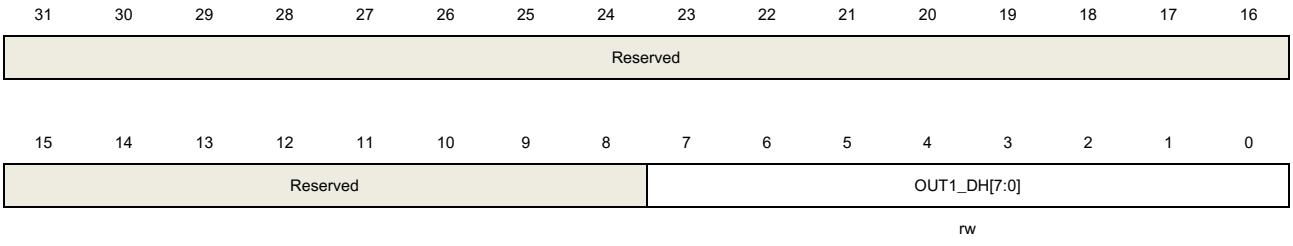
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:4	OUT1_DH[11:0]	DACx_OUT1 12-bit left-aligned data. These bits specify the data that is to be converted by DACx_OUT1.
3:0	Reserved	Must be kept at reset value.

26.4.8. DACx_OUT1 8-bit right-aligned data holding register (DAC_OUT1_R8DH)

Address offset: 0x1C

Reset value: 0x0000 0000

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



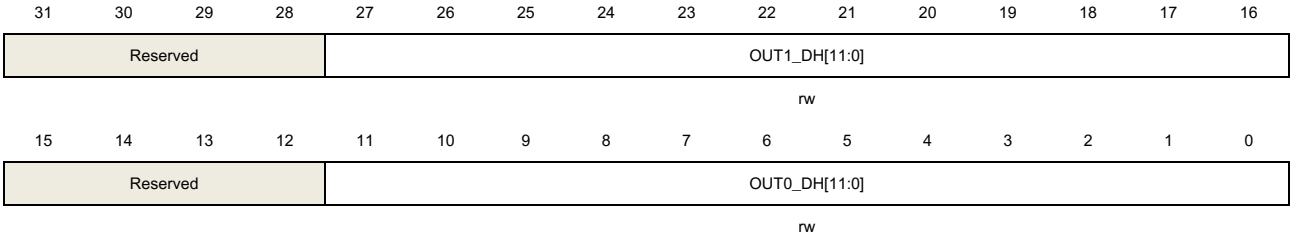
Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	OUT1_DH[7:0]	DACx_OUT1 8-bit right-aligned data These bits specify the MSB 8-bit of the data that is to be converted by DACx_OUT1.

26.4.9. DACx concurrent mode 12-bit right-aligned data holding register (DACC_R12DH)

Address offset: 0x20

Reset value: 0x0000 0000

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



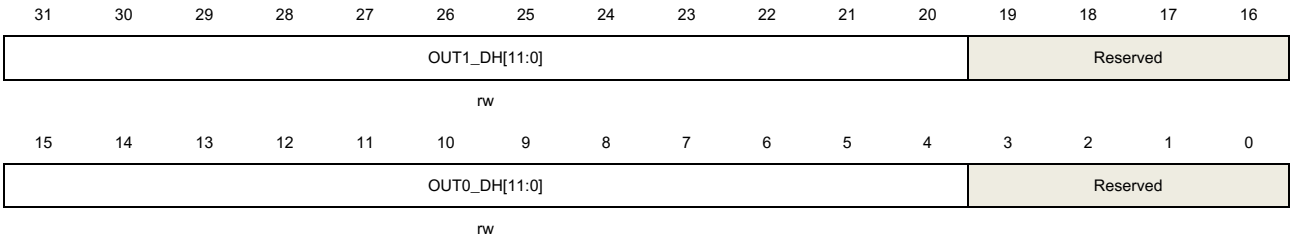
Bits	Fields	Descriptions
31:28	Reserved	Must be kept at reset value.
27:16	OUT1_DH[11:0]	DACx_OUT1 12-bit right-aligned data These bits specify the data that is to be converted by DACx_OUT1.
15:12	Reserved	Must be kept at reset value.
11:0	OUT0_DH[11:0]	DACx_OUT0 12-bit right-aligned data These bits specify the data that is to be converted by DACx_OUT0.

26.4.10. DACx concurrent mode 12-bit left-aligned data holding register (DACC_L12DH)

Address offset: 0x24

Reset value: 0x0000 0000

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



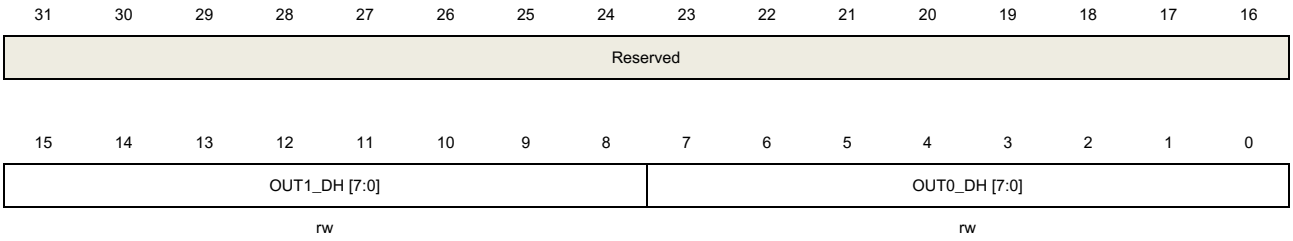
Bits	Fields	Descriptions
31:20	OUT1_DH[11:0]	DACx_OUT1 12-bit left-aligned data These bits specify the data that is to be converted by DACx_OUT1.
19:16	Reserved	Must be kept at reset value.

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.

26.4.11. 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).

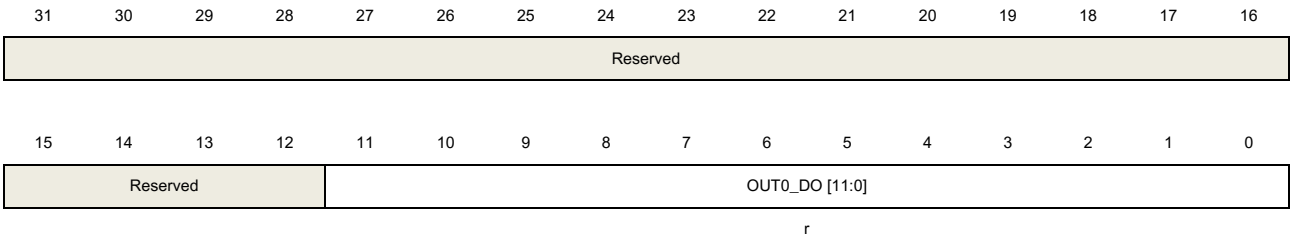


Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:8	OUT1_DH[7:0]	DACx_OUT1 8-bit right-aligned data These bits specify the MSB 8-bit of the data that is to be converted by DACx_OUT1.
7:0	OUT0_DH[7:0]	DACx_OUT0 8-bit right-aligned data These bits specify the MSB 8-bit of the data that is to be converted by DACx_OUT0.

26.4.12. DACx_OUT0 data output register (DAC_OUT0_DO)

Address offset: 0x2C
Reset value: 0x0000 0000

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



Bits	Fields	Descriptions

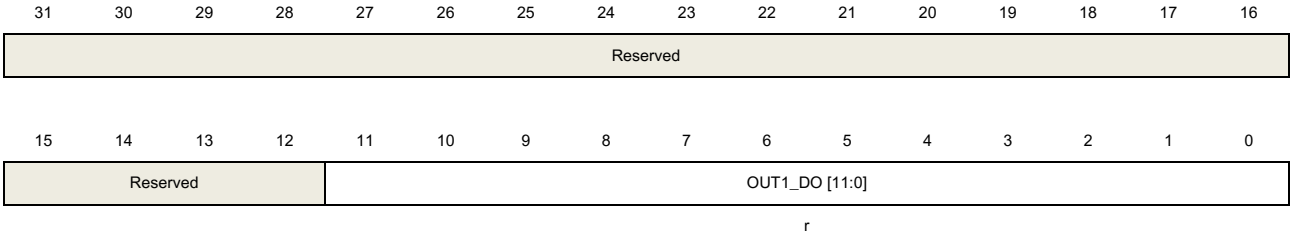
31:12	Reserved	Must be kept at reset value.
11:0	OUT0_DO [11:0]	DACx_OUT0 12-bit output data These bits, which are read only, store the data that is being converted by DACx_OUT0.

26.4.13. DACx_OUT1 data output register (DAC_OUT1_DO)

Address offset: 0x30

Reset value: 0x0000 0000

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



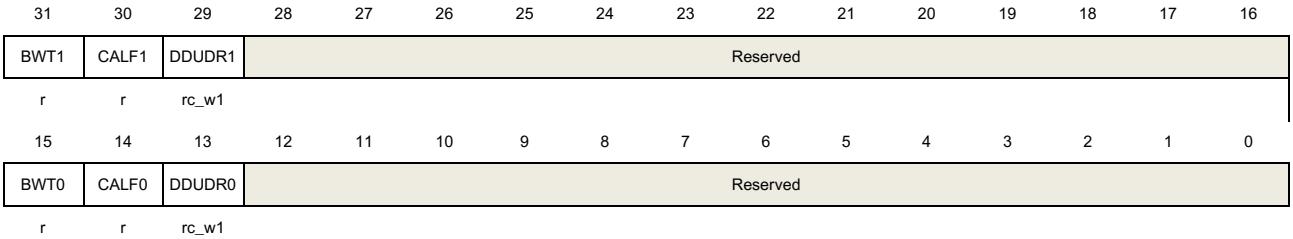
Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11:0	OUT1_DO [11:0]	DACx_OUT1 12-bit output data These bits, which are read only, store the data that is being converted by DACx_OUT1.

26.4.14. DACx status register 0 (DAC_STAT0)

Address offset: 0x34

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31	BWT1	DACx_OUT1 SKSTR1 writing flag. This bit is set by the system when the sample and keep mode is enabled. When the DACx_SKSTR1 is writing,the bit is set, when the write operation is complete, the bit is cleared by hardware. 0: There is no writing operation of DACx_SKSTR1.

		1: There is a writing operation of DACx_SKSTR1.
30	CALF1	DACx_OUT1 calibration offset flag. This bit is set and cleared by hardware. 0: The offset correction value is lower than to the calibration value. 1: The offset correction value is higher than or equal to the calibration value.
29	DDUDR1	DACx_OUT1 DMA underrun flag. This bit is set by hardware and cleared by software (by writing it to 1). 0: no underrun occurred. 1: underrun occurred (Speed of DAC trigger is higher than the DMA transfer).
28:16	Reserved	Must be kept at reset value.
15	BWT0	DACx_OUT0 SKSTR0 writing flag. This bit is set by the system when the sample and keep mode is enabled. When the DACx_SKSTR0 is writing,the bit is set, when the write operation is complete, the bit is cleared by hardware. 0: There is no writing operation of DACx_SKSTR0. 1: There is a writing operation of DACx_SKSTR0.
14	CALF0	DACx_OUT0 calibration offset flag. This bit is set and cleared by hardware. 0: The offset correction value is lower than to the calibration value. 1: The offset correction value is higher than or equal to the calibration value.
13	DDUDR0	DACx_OUT0 DMA underrun flag. This bit is set by hardware and cleared by software (by writing it to 1). 0: no underrun occurred. 1: underrun occurred (Speed of DAC trigger is higher than the DMA transfer).
12:0	Reserved	Must be kept at reset value.

26.4.15. DACx calibration register (DAC_CALR)

Address offset: 0x38

Reset value: 0x000F 000F

This register has 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.

26.4.16. DACx mode control register (DAC_MDCR)

Address offset: 0x3C

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved													MODE1[2:0]		
													rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													MODE0[2:0]		
													rw		

Bits	Fields	Descriptions
31:19	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, It is not recommended to modify this during use when DEN1=1 or CALEN1=1. DACx_OUT1 in normal mode 000: Buffer is enabled and DACx_OUT1 is connected to external pin 001: Buffer is enabled and DACx_OUT1 is connected to on chip peripherals and to external pin. 010: Buffer is disabled and DACx_OUT1 is connected to external pin 011: Buffer is disabled and DACx_OUT1 is connected to on chip peripherals. DACx_OUT1 in sample and keep mode 100: Buffer is enabled and DACx_OUT1 is connected to external pin 101: Buffer is enabled and DACx_OUT1 is connected to on chip peripherals and to external pin. 110: Buffer is disabled and DACx_OUT1 is connected to on chip peripherals and to external pin. 111: Buffer is disabled and DACx_OUT1 is connected to on chip peripherals.
15:3	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 DACx_CTL0 register, It is not recommended to modify this during use when DEN0=1 or CALEN0=1. DACx_OUT0 in normal mode

000: Buffer is enabled and DACx_OUT0 is connected to external pin
001: Buffer is enabled and DACx_OUT0 is connected to on chip peripherals and to external pin.
010: Buffer is disabled and DACx_OUT0 is connected to external pin
011: Buffer is disabled and DACx_OUT0 is connected to on chip peripherals.
DACx_OUT0 in sample and keep mode
100: Buffer is enabled and DACx_OUT0 is connected to external pin
101: Buffer is enabled and DACx_OUT0 is connected to on chip peripherals and to external pin.
110: Buffer is disabled and DACx_OUT0 is connected to on chip peripherals and to external pin.
111: Buffer is disabled and DACx_OUT0 is connected to on chip peripherals.

26.4.17. 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).

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						TSAMP0[9:0]									
						rc_w1									

Bits	Fields	Descriptions
31:10	Reserved	Must be kept at reset value.
9:0	TSAMP0[9:0]	DACx_OUT0 sample time. When BWT0 is 1, this register cannot be written.

26.4.18. 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. When BWT1 is 1, this register cannot be written.

26.4.19. DACx sample and keep keep time register (DAC_SKKTR)

Address offset: 0x48

Reset value: 0x0001 0001

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



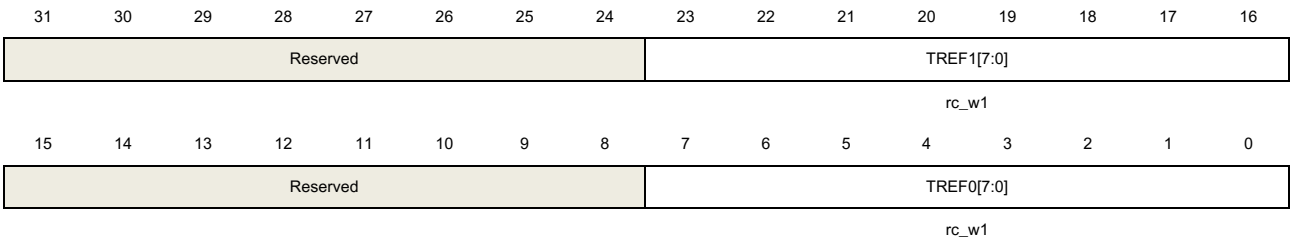
Bits	Fields	Descriptions
31:26	Reserved	Must be kept at reset value.
25:16	TKEEP1[9: 0]	DACx_OUT1 keep time.(valid only in sample and keep mode)
15:10	Reserved	Must be kept at reset value.
9:0	TKEEP0[9: 0]	DACx_OUT0 keep time.(valid only in sample and keep mode)

26.4.20. DACx sample and keep refresh time register (DAC_SKRTR)

Address offset: 0x4C

Reset value: 0x0001 0001

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



Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:16	TREF1[7: 0]	DACx_OUT1 refresh time.

15:8	Reserved	Must be kept at reset value.
7:0	TREF0[7: 0]	DACx_OUT0 refresh time.

27. Comparator (CMP)

27.1. 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.

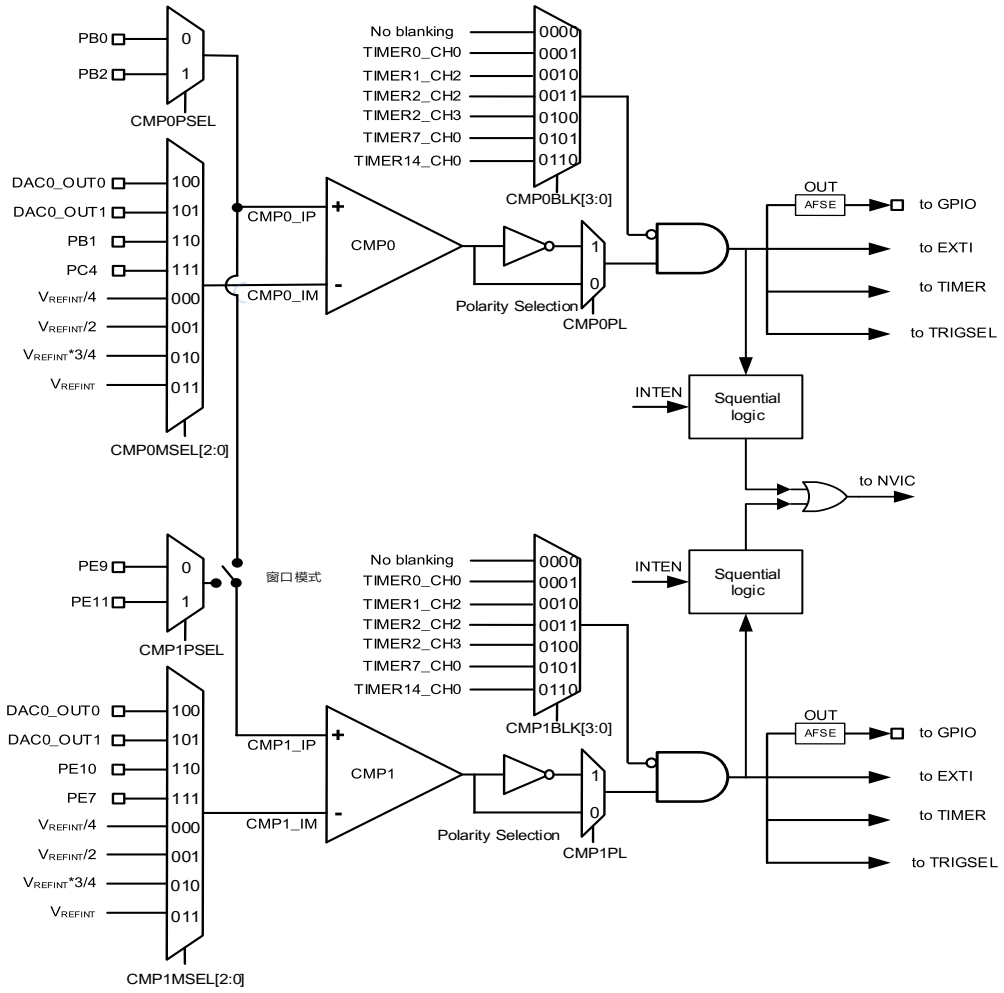
27.2. Characteristic

- Rail-to-rail comparators.
- Configurable hysteresis.
- Configurable speed and consumption.
- Configurable analog input source.
  - DAC output.
  - Multiplexed I / O pins.
  - The whole or sub-multiple values of internal reference voltage.
- Outputs with blanking source.
- Window comparator.
- Outputs to I / O.
- Outputs to timers for triggering.
- Digital filter.
- Outputs to EXTI.
- Outputs to NVIC.
- Outputs to TRIGSEL.

27.3. Function overview

The block diagram of CMP is shown below:

Figure 27-1. CMP block diagram



Note: V_{REFINT} is 1.2V.

27.3.1. CMP clock

The clock of the CMP which is connected to APB bus, is synchronous with PCLK.

27.3.2. 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 input channel.

■ 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 27-1. CMP inputs and outputs summary details the inputs and outputs of the CMP.

Table 27-1. CMP inputs and outputs summary

	CMP0	CMP1
CMP non inverting inputs connected to I/Os	PB0 PB2	PE9 PE11
CMP inverting inputs connected to I/Os	PB1 PC4	PE10 PE7
CMP inverting inputs connected to internal signals	V_{REFINT}/4, V_{REFINT}/2, V_{REFINT}*3/4, V_{REFINT}, DAC0_OUT0, DAC0_OUT1	V_{REFINT}/4, V_{REFINT}/2, V_{REFINT}*3/4, V_{REFINT}, DAC0_OUT0, DAC0_OUT1
CMP outputs connected to I/Os	PC5 (AF13) PE12 (AF13)	PE8 (AF13) PE13 (AF13)
CMP outputs connected to EXTI	•	
CMP outputs connected to TRIGSEL	•	
CMP outputs connected to NVIC	•	
CMP outputs connected to internal signals	•	
CMP outputs connected to TIMER break	•	
CMP_MUX_OUT (controlled by AFSE[x])	PA6 (AF10) PA8 (AF12) PB12 (AF13) PE6 (AF11) PE15 (AF13) PG2 (AF11) PG3 (AF11) PG4 (AF11) PK0 (AF11) PK1 (AF11) PK2 (AF10)	

Note:

1. Refer to the system configuration registers for details on CMP outputs connected to internal signals.

2. Refer to the TIMER module for details on CMP outputs connected to TIMER break.

27.3.3. CMP operating mode

For a given application, there is a trade-off between the CMP power consumption versus propagation delay, which is adjusted by configuring bits CMPxM[1:0] in CMPx_CS register. The CMP works fastest with highest power consumption when CMPxM[1:0] = 2'b00, while works slowest with lowest power consumption when CMPxM [1:0]= 2'b11.

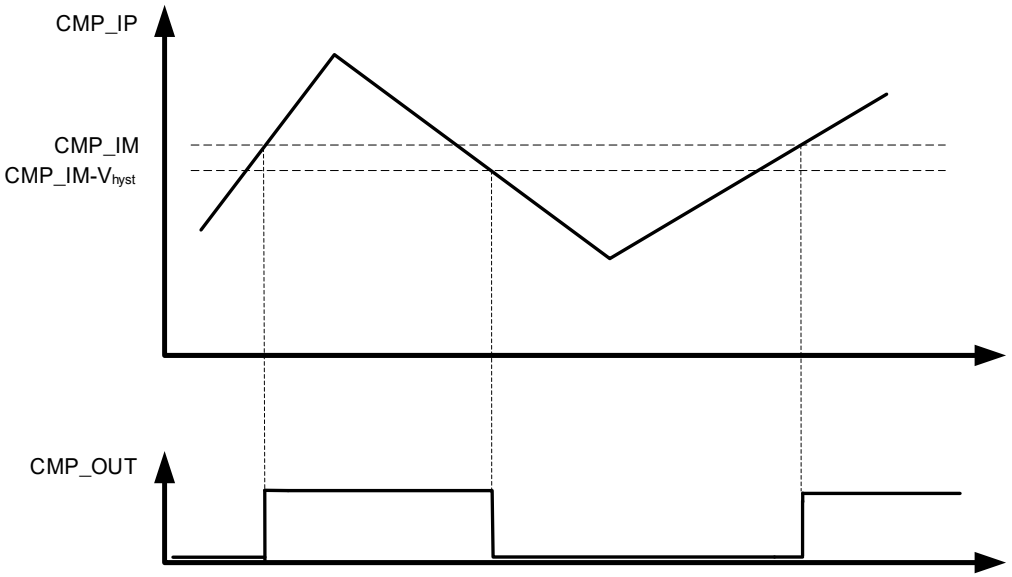
27.3.4. CMP Window mode

If the WNDEN bit in CMP1_CS register is set, comparator windows mode is enabled, input plus of comparator 1 is connected with input plus of comparator 0. If the minus input of CMP0 and CMP1 is connected to different voltage, the voltage range from lower threshold to upper threshold, is monitored by analyzing the comparator 0 and comparator 1 output.

27.3.5. CMP hysteresis

In order to avoid spurious output transitions that caused by the noise signal, a programmable hysteresis is designed to force the hysteresis value by configuring CMPx_CS register. This function could be shut down if it is unnecessary.

Figure 27-2. CMP hysteresis



27.3.6. CMP register write protection

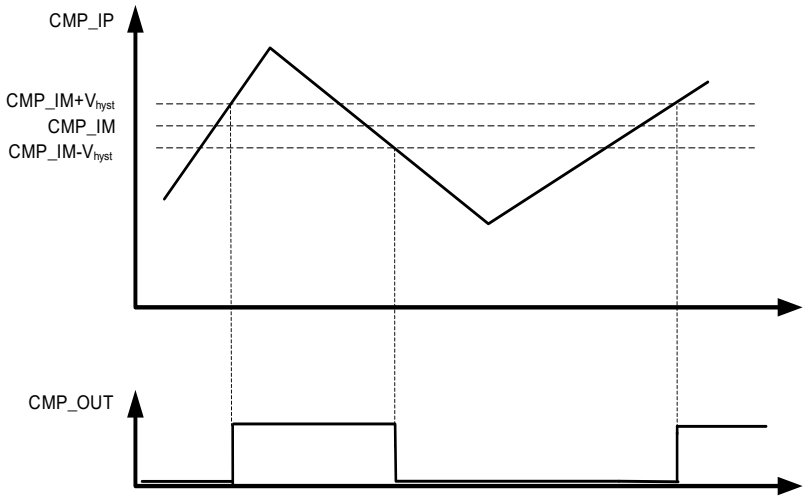
The CMP control and status register (CMPx_CS) and alternate select register (CMP_SR) can be protected from writing by setting CMPxLK bit to 1. The CMPx_CS register, including the CMPxLK bit will be read-only, and can only be reset by the system reset.

27.3.7. CMP output blanking

CMP output blanking function can be used to avoid interference of short pulses in the input signal to CMP output signal. If the CMPxBLK[3:0] bits in the CMPx_CS register are setting to an available value, the CMP output final signal is obtained by ANDing the complementary signal of the selected blanking signal with the raw output of the comparator. The blanking function can be used for false overcurrent detection in motor control applications.

Figure 27-3. The CMP outputs signal blanking shows the comparator output blank function.

Figure 27-3. The CMP outputs signal blanking



27.3.8. 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.

27.3.9. CMP digital filter

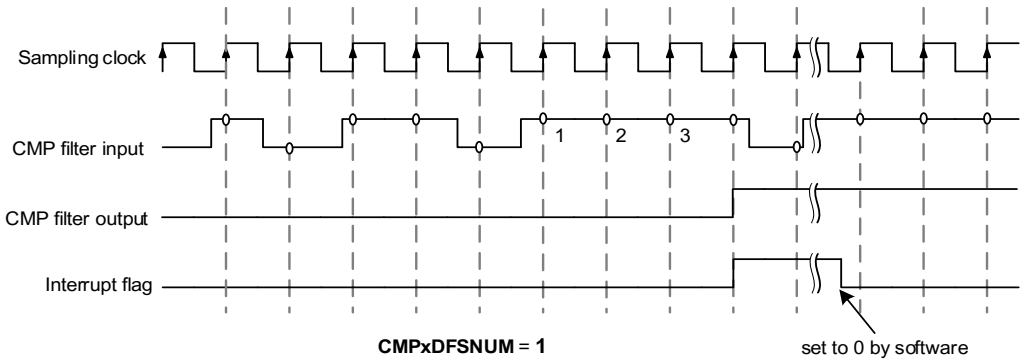
The sampling frequency of digital filter can be determined by the CMPxDFSCDIV[1:0] bits in the CMPx_CS register. The valid sampling times that the same level value is continuously sampled can be selected by the CMPxDFSNUM bit in the CMPx_CS register. The features of digital filter are as follows:

- PCLK/8, PCLK/16 or PCLK/32 can be selected as sampling clock.
- The filter function can also be disabled.
- The valid sampling times that the same level value can be two or three.

The CMP output signal will be sampled at every sampling clock, and when the same level value is sampled two or three times in succession, this level is identified as the noise-filtered valid output at the next sampling clock. *Figure 27-4. The CMP output signal filtering* shows

details of digital filter output function.

Figure 27-4. The CMP output signal filtering



Note: The eliminated noise width can be selected by setting the CMPxDFSCDIV[1:0] bits and CMPxDFSNUM bit in the CMPx_CS register.

27.3.10. 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.

27.4. Register definition

CMP base address: 0x5800 3800

27.4.1. 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														CMP1IF	CMP0IF
														r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														CMP1O	CMP0O
														r	r

Bits	Fields	Descriptions
31:18	Reserved	Must be kept at reset value.
17	CMP1IF	CMP1 interrupt flag 0: No CMP1 output interrupt 1: CMP1 output interrupt Set by hardware when the CMP1 output the rising edge. 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 the rising edge. Cleared by software writing 1 to CMP0IC bit in the CMP_IFC register.
15:2	Reserved	Must be kept at reset value.
1	CMP1O	CMP1 output state This 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

27.4.2. 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														CMP1IC	CMP0IC
														w	w

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

Bits	Fields	Descriptions
31:18	Reserved	Must be kept at reset value.
17	CMP1IC	CMP1 interrupt flag clear 0: No clear CMP1 interrupt flag 1: Clear CMP1 interrupt flag
16	CMP0IC	CMP0 interrupt flag clear 0: No clear CMP0 interrupt flag 1: Clear CMP0 interrupt flag
15:0	Reserved	Must be kept at reset value.

27.4.3. CMP alternate select register (CMP_SR)

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					AFSE[10:0]										
					rw										

Bits	Fields	Descriptions
31:11	Reserved	Must be kept at reset value.
10:0	AFSE[10:0]	CMP selects alternate output ports For each bit, 0 is selected for CMP0_OUT as the alternate function with the corresponding GPIO, and 1 is selected for CMP1_OUT. Bit0: PA6 bit1: PA8

bit2: PB12
bit3: PE6
bit4: PE15
bit5: PG2
bit6: PG3
bit7: PG4
bit8: PK0
bit9: PK1
bit10: PK2

27.4.4. CMP0 control/status register (CMP0_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
CMP0LK	CMP0DFSNUM	CMP0DFSCDIV[1:0]		CMP0BLK[3:0]				Reserved			CMP0PSEL	Reserved	CMP0MSEL[2:0]		
rw	rw	rw		rw							rw		rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		CMP0M[1:0]		Reserved		CMP0HST[1:0]		Reserved	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 and CMP_SR bits are read-only
30	CMP0DFSNUM	CMP0 digital filter sampling number This bit is used to select valid sampling number of the CMP0 digital filter. 0: The same value is continuously sampled 2 times 1: The same value is continuously sampled 3 times
29:28	CMP0DFSCDIV[1:0]	CMP0 digital filter sampling clock division These bits are used to select the sampling frequency of digital filter. 00: Digital filter is not used 01: Sampling frequency is f_{PCLK}/8 10: Sampling frequency is f_{PCLK}/16 11: Sampling frequency is f_{PCLK}/32
27:24	CMP0BLK[3:0]	CMP0 output blanking source

		These bits are used to select which timer output controls the comparator output blanking. 0000: No blanking 0001: Select TIMER0_CH0 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 TIMER2_CH3 output compare signal as blanking source 0101: Select TIMER7_CH0 output compare signal as blanking source 0110: Select TIMER14_CH0 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: PB0 1: PB2
19	Reserved	Must be kept at reset value.
18:16	CMP0MSEL[2:0]	CMP0_IM input selection These bits are used to select the source connected to the CMP0_IM input of the CMP0. 000: V_{REFINT} / 4 001: V_{REFINT} / 2 010: V_{REFINT}*3 / 4 011: V_{REFINT} 100: DAC0_OUT0 101: DAC0_OUT1 110: PB1 111: PC4
15:14	Reserved	Must be kept at reset value.
13:12	CMP0M[1:0]	CMP0 mode These bits are used to control the operating mode of the CMP0 adjust the speed / consumption. 00: High speed / full power 01 / 10: Medium speed / medium power 11: Very-low speed / ultra-low power
11:10	Reserved	Must be kept at reset value.
9:8	CMP0HST[1:0]	CMP0 hysteresis These bits are used to control the hysteresis level. 00: No hysteresis 01: Low hysteresis 10: Medium hysteresis

		11: High hysteresis
7	Reserved	Must be kept at reset value.
6	CMP0INTEN	CMP0 interrupt enable 0: Disabled 1: Enabled
5:4	Reserved	Must be kept at reset value.
3	CMP0PL	Polarity of CMP0 output This bit is used to select the polarity of CMP0 output. 0 : Output is not inverted 1 : Output is inverted
2	CMP0SEN	Voltage scaler enable bit 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
1	CMP0BEN	Scaler bridge enable bit 0: Disable scaler resistor bridge in case that CMP1BEN bit of CMP1_CS is also reset 1: Enable scaler resistor bridge
0	CMP0EN	CMP0 enable 0: CMP0 disabled 1: CMP0 enabled

27.4.5. CMP1 control/status register (CMP1_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
CMP1LK	CMP1DFSNUM	CMP1DFSCDIV[1:0]		CMP1BLK[3:0]				Reserved			CMP1PSEL	Reserved	CMP1MSEL[2:0]		
rw	rw	rw		rw							rw		rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		CMP1M[1:0]		Reserved		CMP1HST[1:0]		Reserved	CMP1INTEN	Reserved	WNDEN	CMP1PL	CMP1SEN	CMP1BEN	CMP1EN
		rw				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 and CMP_SR bits are read-only
30	CMP1DFSNUM	CMP1 digital filter sampling number This bit is used to select valid sampling number of the CMP1 digital filter. 0: The same value is continuously sampled 2 times 1: The same value is continuously sampled 3 times
29:28	CMP1DFSCDIV[1:0]	CMP1 digital filter sampling clock division These bits are used to select the sampling frequency of digital filter. 00: Digital filter is not used 01: Sampling frequency is f_{PCLK}/8 10: Sampling frequency is f_{PCLK}/16 11: Sampling frequency is f_{PCLK}/32
27:24	CMP1BLK[3:0]	CMP1 output blanking source These bits are used to select which timer output controls the comparator output blanking. 0000: No blanking 0001: Select TIMER0_CH0 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 TIMER2_CH3 output compare signal as blanking source 0101: Select TIMER7_CH0 output compare signal as blanking source 0110: Select TIMER14_CH0 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: PE9 1: PE11
19	Reserved	Must be kept at reset value.
18:16	CMP1MSEL[2:0]	CMP1_IM input selection These bits are used to select the source connected to the CMP1_IM input of the CMP1. 000: V_{REFINT} / 4 001: V_{REFINT} / 2 010: V_{REFINT}*3 / 4 011: V_{REFINT} 100: DAC0_OUT0 101: DAC0_OUT1

		110: PE10 111: PE7
15:14	Reserved	Must be kept at reset value.
13:12	CMP1M[1:0]	CMP1 mode These bits are used to control the operating mode of the CMP1 adjust the speed / consumption. 00: High speed / full power 01 / 10: Medium speed / medium power 11: Very-low speed / ultra-low power
11:10	Reserved	Must be kept at reset value.
9:8	CMP1HST[1:0]	CMP1 hysteresis These bits are used to control the hysteresis level. 00: No hysteresis 01: Low hysteresis 10: Medium hysteresis 11: High hysteresis
7	Reserved	Must be kept at reset value.
6	CMP1INTEN	CMP1 interrupt enable 0: Disabled 1: Enabled
5	Reserved	Must be kept at reset value.
4	WNDEN	Window mode enable This bit is used to select CMP1_IP source. 0: CMP1_IP is connected to CMP1 non-inverting input 1: CMP1_IP is connected to CMP0_IP
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
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

28. VREF

28.1. 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.

28.2. Characteristics

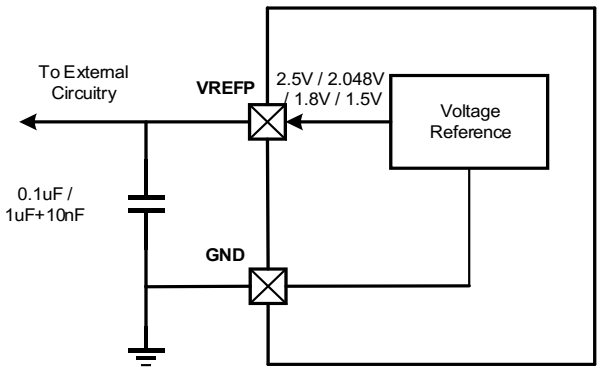
- Stable voltage, and product calibrated.
- Connects to VREFP pin to source off-chip circuits.
- VREFBUF does not provide off-chp power supply functionality.
- 1.5V, 1.8V, 2.048V or 2.5V configurable reference voltage output.

28.3. 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 1.5V, 1.8V, 2.048V, or 2.5V 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 1uF and 10nF connected in parallel) which is recommended to ground is required.

Figure 28-1.Precision reference connection



As shown in *Table 28-1. 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 28-1. VREF modes

VREFEN	HIPM	Mode
0	0	VREF disabled, VREFP pin pulled-down to VSSA.
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.

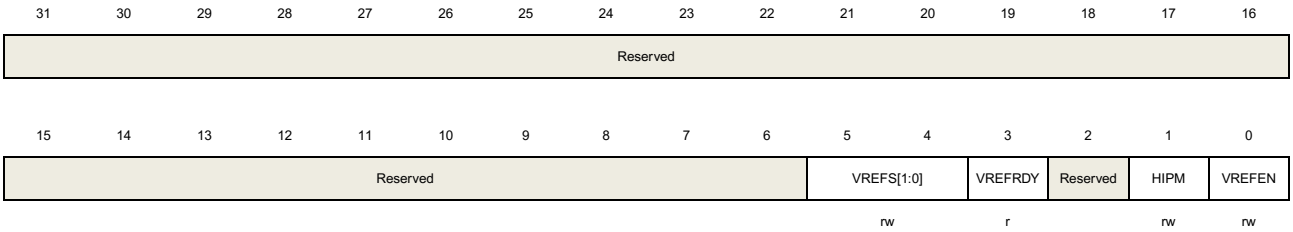
28.4. Register definition

VREF base address: 0x5800 3C00

28.4.1. 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).

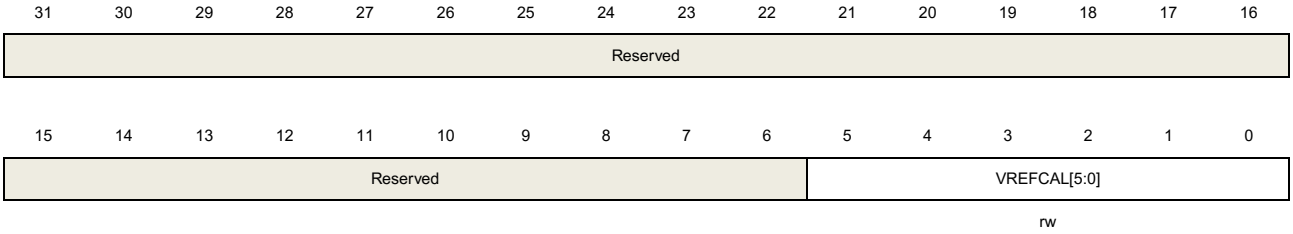


Bits	Fields	Descriptions
31:6	Reserved	Must be kept at reset value.
5:4	VREFS[1:0]	Voltage reference select These bits set the value of voltage reference output by the VREF. 00: The voltage reference is around 2.5 V 01: The voltage reference is around 2.048 V 10: The voltage reference is around 1.8 V 11: The voltage reference is around 1.5 V 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

28.4.2. Calibration register (VREF_CALIB)

Address offset: 0x04
Reset value: 0x0000 0020

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



Bits	Fields	Descriptions
31:6	Reserved	Must be kept at reset value.
5:0	VREFCAL[5:0]	VREF calibration After reset, these bits will be initialized to the calibration value saved in the Flash memory during the course of production test. Writing to these bits can adjust the output of the VREF. **Note:** If the user performs the calibration procedure, the VREF calibration must increase progressively from 0x00 to 0x3F.

29. Low power digital temperature sensor (LPDTS)

29.1. Overview

Low power digital temperature sensor(LPDTS) is used to transmit square wave, which is converted by temperature and the frequency is proportional to the absolute temperature. The frequency measurement is based on the PCLK or the LXTAL clock.

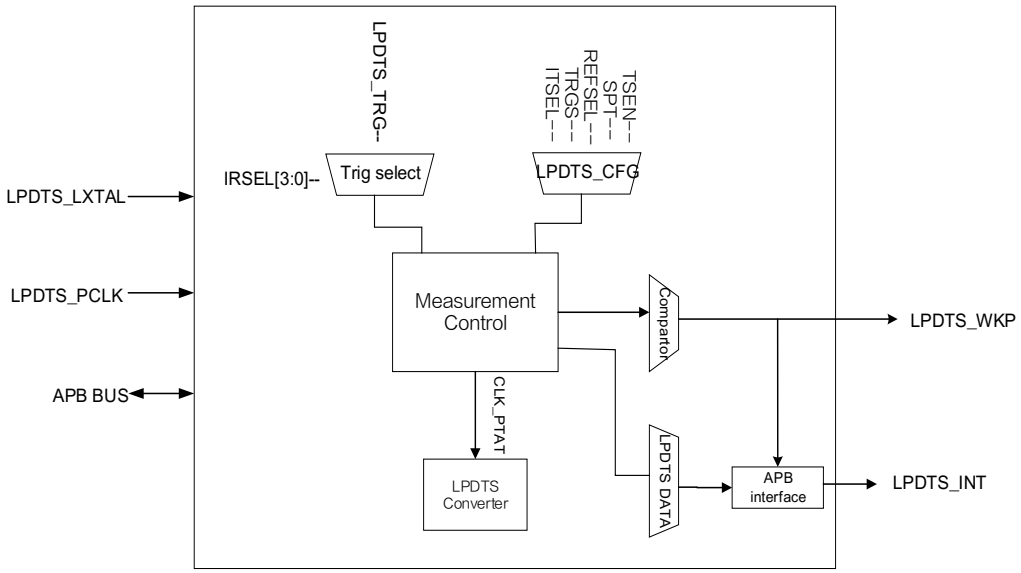
29.2. Characteristics

- The trigger source of measurement can be set to software or hardware.
- Programmable sampling time.
- Counter synchronized on LXTAL or PCLK.
- The interrupt can be generated when the temperature is below a low threshold or above a high threshold and at the end of measurement.
- The generation of asynchronous wakeup signal indicates that the measurement result is higher or lower than the specified threshold when the LXTAL is selected as reference clock.

29.3. Block diagram

Figure 29-1. LPDTS block diagram shows the LPDTS block diagram.

Figure 29-1. LPDTS block diagram



29.4. Function overview

29.4.1. LPDTS internal signals

Table 29-1. LPDTS signals

Signal name	Type	Description
LPDTS_LXTAL	input	LXTAL clock
LPDTS_PCLK	input	APB clock
LPDTS_INT	output	The interrupt of internal temperature sensor
LPDTS_WKP	output	The wakeup of internal temperature sensor

29.4.2. Operating modes

The REFSEL bit in LPDTS_CFG can be set to selected multiple operation modes.

■ PCLK mode (REFSEL = 0)

The register can be wrote or read by software. The REFSEL bit is set to 0 to select the PCLK as reference clock.

■ PCLK and LXTAL mode (REFSEL = 1)

The register can be wrote or read by software. The REFSEL bit is set to 1 to select the LXTAL as reference clock.

■ LXTAL mode (REFSEL = 1 and PCLK OFF)

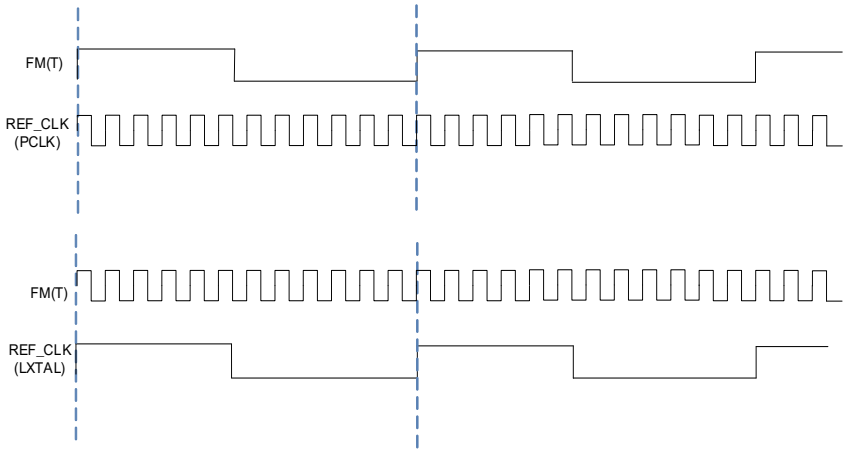
The temperature sensor registers cannot be accessed. The LXTAL is selected as reference clock. This mode can use hardware triggers to exit the Deep-sleep mode.

29.4.3. Temperature measurement principles

A signal is output, which FM(T) frequency (typically 641 kHz) is related to temperature, by the analog part of temperature sensor. Two counters are embedded in the temperature sensor block, which makes the counting mode relevant to the reference clock frequency. The counting result is stored in the LPDTS_DATA register.

- When the reference clock is PCLK, the measurement method is to sample one or multiple FM(T) cycles and count at the rising edge and falling edge of PCLK.
- When the reference clock is LXTAL, the measurement method is to sample one or multiple LXTAL cycles and count at the rising edge and falling edge of FM(T).

Figure 29-2. Method for different REF_CLK



The Temperature calculation formula When PCLK is used:

$$T=T0+((2\times F_{PCLK}/COVAL)\times SPT-100\times FREQ)/RF_CF \quad (29\text{-}1)$$

The Temperature calculation formula When LXTAL is used:

$$T=T0+(((F_{LXTAL}\times COVAL)/(2\times SPT))-(100\times FREQ))/RF_CF \quad (29\text{-}2)$$

where:

- T0 is equal to 25 °C.
- COVAL is the value of the counter output value for temperature sensor which measured and stored in the LPDTS_DATA register.
- SPT is Sampling time for temperature sensor.
- FREQ is engineering value of the frequency measured at T0 for temperature sensor which measured and stored in the LPDTS_SDATA register. It is expressed in hundreds of Hertz.
- RF_CF is the engineering value of the ramp coefficient for the temperature sensor. During the measurement and stored in the LPDTS_RDATA register, the unit is Hz / °C.

29.4.4. Sampling time

Increasing the sampling period helps to improve the measurement accuracy. It is most effective when the reference frequency is set close to the sampling frequency.

The default value of the sampling time should be set as one REF_CLK cycle or one FM(T) cycle, and the corresponding modes are LXTAL mode and PCLK mode.

The sampling time is configured through SPT bits in LPDTS_CFG register (see ***Table 29-2. Sampling time configuration***).

Table 29-2. Sampling time configuration

SPT[3:0]	LXTAL or FM(T) clock cycle(s)
0000	1
0001	1

SPT[3:0]	LXTAL or FM(T) clock cycle(s)
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14
1111	15

29.4.5. Trigger input

The trigger source can be selected to trigger the temperature measurement through ITSEL[3:0] bits in LPDTS_CFG.

- Software trigger
  The software trigger is selected when ITSEL[3:0] is set to '0000' in LPDTS_CFG.
  - Check if TSRF is set to 1. When TSRF is set, start the temperature measurement by setting TRGS bit in LPDTS_CFG register. Otherwise, ignore this step.
  - After completion of measurement, if the TRGS bit remains at 1, the measurement will restart when the TSRF flag changes to 1.
- Hardware trigger
  The temperature sensor can only capture a hardware trigger rising edge when TSRF bit is set, otherwise the trigger is ignored.

Table 29-3. Trigger configuration

Name	ITSEL[3:0]				Comment
NA	0	0	0	0	No hardware trigger
0001	0	0	0	1	reserved
0010	0	0	1	0	
0011	0	0	1	1	
0100	0	1	0	0	LPDTS_TRG
0101	0	1	0	1	reserved
0110	0	1	1	0	
0111	0	1	1	1	
1000	1	0	0	0	
1001	1	0	0	1	
1010	1	0	1	0	

Name	ITSEL[3:0]				Comment
1011	1	0	1	1	
1100	1	1	0	0	
1101	1	1	0	1	
1110	1	1	1	0	
1111	1	1	1	1	

Note: The LPDTS_TRG is the output of the TRIGSEL module. The INSEL0[7:0] of TRIGSEL_LPDTS register in the TRIGSEL module are used to select trigger input source of LPDTS_TRG trigger input.

29.4.6. On-off control and ready flag

The LPDTS block can be enabled by setting TSEN bit in LPDTS_CFG register. The TSRF flag in the Temperature sensor status register (LPDTS_STAT) indicate that the LPDTS block is ready for temperature measurement: when TSRF bit is set to 1, the measurement can be started. Once a measurement has started, TSRF bit is reset. New measurement can not be initiated at this time. If a new measurement is needed, it is necessary to wait for the completion of the last measurement and set the TSRF bit.

29.4.7. LPDTS low-power modes

Table 29-4. Temperature sensor behavior in low-power modes

Mode	Description
Sleep	LPDTS interrupt can be used to exit from Sleep mode when the LXTAL or PCLK as reference clock.
Deep-sleep	LPDTS interrupt can be used to exit from Deep-sleep mode when the LXTAL as reference clock.

29.4.8. LPDTS interrupts

The LPDTS interrupt line can be connected to the CPU NVIC or to the EXTI controller.

- The interrupt can be generated in two situations:
  - At the end of measurement.
  - The measurement result is higher or lower than a specified threshold.
- There are two kinds of interrupt in LPDTS module.
  - Synchronous interrupt: Three interrupt events can be select via 3 bits in LPDTS_INTEN register.
  - Asynchronous wakeup: Three asynchronous wakeup events can be selected via 3 bits in LPDTS_INTEN register.
- All combination of interrupts are allowed.

Note: Asynchronous wakeup is used only when the LXTAL is selected as reference clock.

The following table shows the interrupt bits and their description.

Table 29-5. Temperature sensor behavior in low-power modes

Interrupt event	Interrupt flag	Enable control bit	Interrupt clear bit	Exit from Sleep mode	Exit from Deep-sleep mode	Synchronous/ Asynchronous
When the measurement is done	EMIF in LPDTS_STAT	EMIE in LPDTS_INTEN	EMIC in LPDTS_INTC	YES	NO	Synchronous on PCLK
When the measurement is lower than the specified threshold	LTIF in LPDTS_STAT	LTIE in LPDTS_INTEN	LTIC in LPDTS_INTC	YES	NO	
When the measurement is higher than the specified threshold	HTIF in LPDTS_STAT	HTIE in LPDTS_INTEN	HTIC in LPDTS_INTC	YES	NO	
When the measurement is done	EMAIF in LPDTS_STAT	EMAIE in LPDTS_INTEN	EMAIC in LPDTS_INTC	YES	YES	Asynchronous
When the measurement is lower than the specified threshold	LTAIF in LPDTS_STAT	LTAIFE in LPDTS_INTEN	LTAIC in LPDTS_INTC	YES	YES	
When the measurement is higher than the specified threshold	HTAIF in LPDTS_STAT	HTAIE in LPDTS_INTEN	HTAIC in LPDTS_INTC	YES	YES	

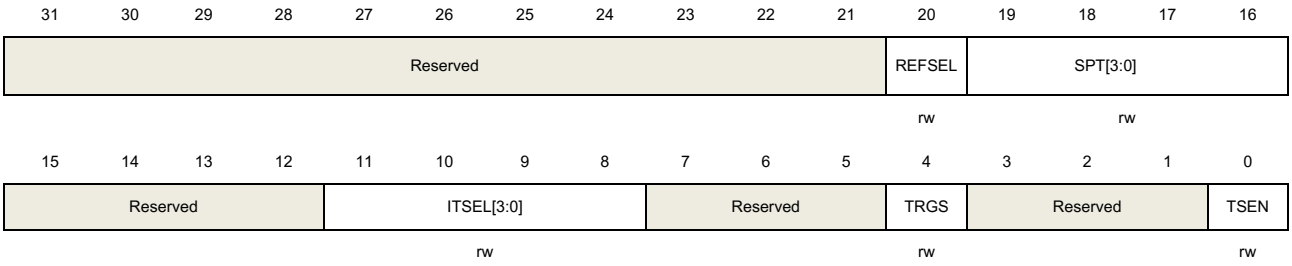
29.5. Register definition

LPDTS base address: 0x5800 6800

29.5.1. Configuration register (LPDTS_CFG)

Address offset: 0x00
Reset value: 0x0000 0000

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



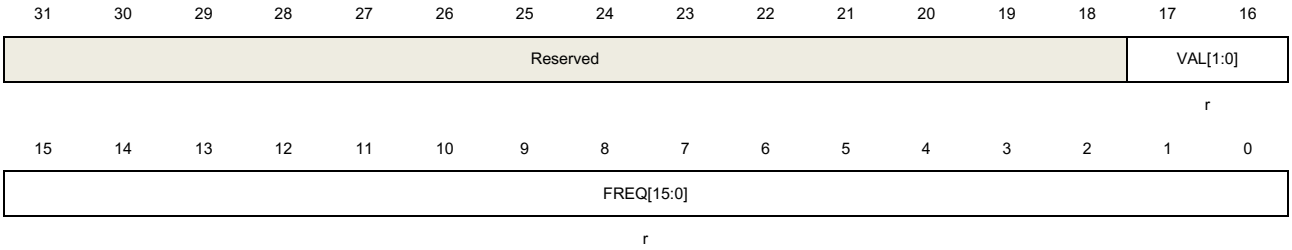
Bits	Fields	Descriptions
31:21	Reserved	Must be kept at reset value.
20	REFSEL	Reference clock selection 0: High speed reference clock (PCLK) 1: Low speed reference clock (LXTAL)
19:16	SPT[3:0]	Sampling time These bits allow increasing the sampling time to improve measurement precision.
15:12	Reserved	Must be kept at reset value.
11:8	ITSEL[3:0]	Input trigger selection These bits select which input triggers a temperature measurement.
7:5	Reserved	Must be kept at reset value.
4	TRGS	Trigger selection for frequency measurement 0: No software trigger. 1: Software trigger for a frequency measurement when temperature sensor is ready.
3:1	Reserved	Must be kept at reset value.
0	TSEN	Enable temperature sensor 0: Disable temperature sensor 1: Enable temperature sensor

29.5.2. Sensor T0 data register 1 (LPDTS_SDATA)

Address offset: 0x08

System reset value: 0x000X XXXX

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



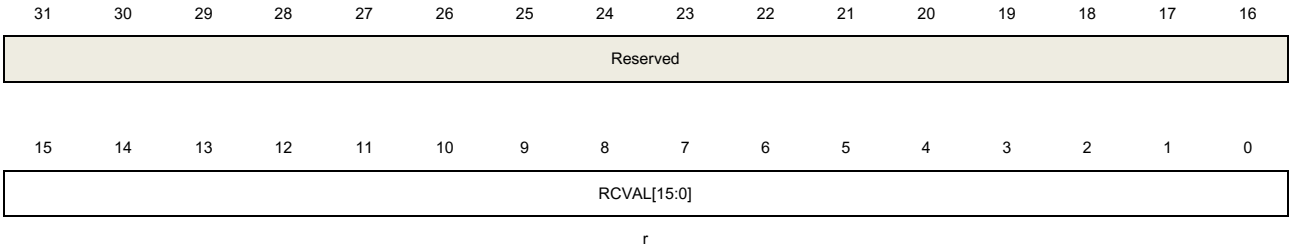
Bits	Fields	Descriptions
31:18	Reserved	Must be kept at reset value.
17:16	VAL[1:0]	The collected temperature value These bits are the value of T0 temperature. 00: 25 °C 01: -40 °C Others: Reserved
15:0	FREQ[15:0]	Frequency value These bits are the value of the frequency measured when the temperature is T0. This value is set by 0.1 kHz step.

29.5.3. Ramp data register (LPDTS_RDATA)

Address offset: 0x10

System reset value: 0xXXXX XXXX

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



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	RCVAL[15:0]	Ramp coefficient These bits are the value of the ramp coefficient. This value is set by 1 Hz/°C step.

29.5.4. Interrupt threshold register (LPDTS_IT)

Address offset: 0x14
System reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:16	INTHT[15:0]	Interrupt high threshold These bits indicate the highest value than can be reached before the interrupt is generated.
15:0	INTLT[15:0]	Interrupt low threshold These bits indicate the lowest value than can be reached before the interrupt is generated.

29.5.5. Temperature data register (LPDTS_DATA)

Address offset: 0x1C
System reset value: 0x0000 0000

This register contains the number of REF_CLK cycles used to compute the FM(T) frequency.

This register has 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															

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

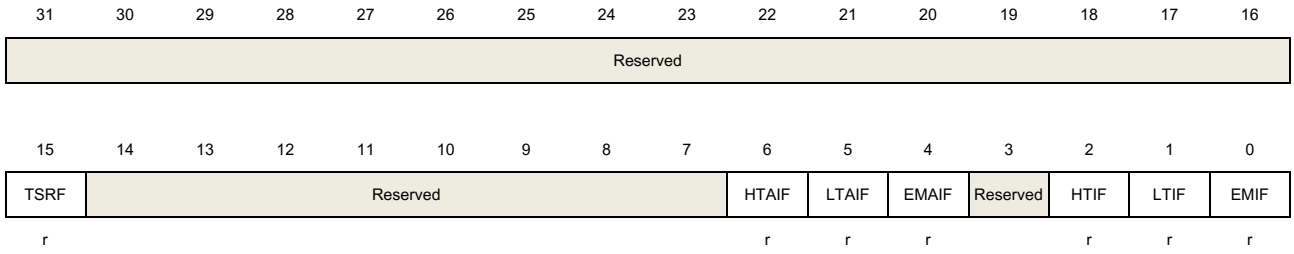
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	COVAL[15:0]	Value of the counter output

29.5.6. Temperature sensor status register (LPDTS_STAT)

Address offset: 0x20

System 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	TSRF	Temperature sensor ready flag 0: Temperature sensor not ready 1: Temperature sensor ready
14:7	Reserved	Must be kept at reset value.
6	HTAIF	High threshold asynchronous interrupt flag Set by hardware when the high threshold is reached and the HTAIE bit is set. Reset by software by setting the HTAIC bit in the LPDTS_INTC register. 0: No high threshold asynchronous interrupt generated 1: High threshold interrupt asynchronous generated
5	LTAIF	Low threshold asynchronous interrupt flag Set by hardware when the low threshold is reached and the LTAIE bit is set. Reset by software by setting the LTAIC bit in the LPDTS_INTC register. 0: No low threshold asynchronous interrupt generated 1: Low threshold asynchronous interrupt generated
4	EMAIF	End of measurement asynchronous interrupt flag Set by hardware when the temperature measure is finished and the EMAIE bit is set. Reset by software by setting the EMAIC bit in the LPDTS_INTC register. 0: No end of measurement asynchronous interrupt generated 1: End of measurement asynchronous interrupt generated
3	Reserved	Must be kept at reset value.
2	HTIF	High threshold interrupt flag Set by hardware when the high threshold is reached and the HTIE bit is set (synchronized on PCLK). Reset by software by setting the HTIC bit in the LPDTS_INTC register. 0: No high threshold interrupt generated 1: High threshold interrupt generated
1	LTIF	Low threshold interrupt flag

		Set by hardware when the low threshold is reached and the LTIE bit is set (synchronized on PCLK). Reset by software by setting the LTIC bit in the LPDTS_INTC register. 0: No low threshold interrupt generated 1: Low threshold interrupt generated
0	EMIF	End of measurement interrupt flag Set by hardware when the temperature measure is finished and the EMIE bit is set (synchronized on PCLK). Reset by software by setting the EMIC bit in the LPDTS_INTC register. 0: No end of measurement interrupt generated 1: End of measurement interrupt generated

29.5.7. Interrupt enable register (LPDTS_INTEN)

Address offset: 0x24

System reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									HTAIE	LTAIE	EMAIE	Reserved	HTIE	LTIE	EMIE
									rw	rw	rw		rw	rw	rw

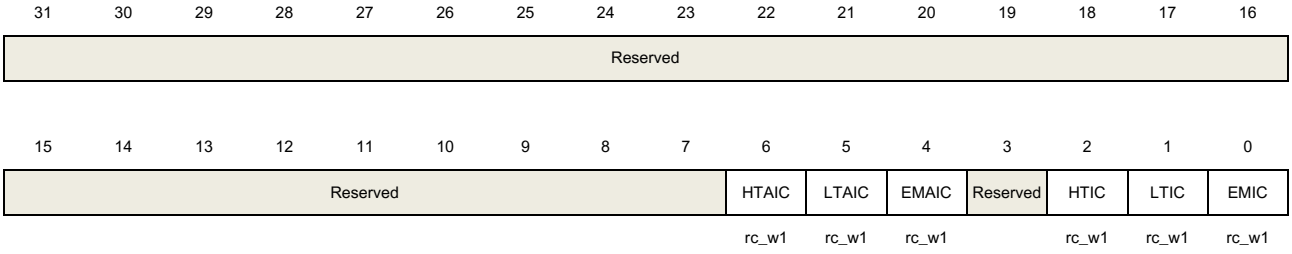
Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value.
6	HTAIE	High threshold asynchronous interrupt enable Set and reset by software to enable/disable the high threshold asynchronous interrupt (only when REFSEL = 1). 0: Disable the high threshold asynchronous interrupt 1: Enable the high threshold asynchronous interrupt
5	LTAIE	Low threshold asynchronous interrupt enable Set and reset by software to enable/disable the low threshold asynchronous interrupt (only when REFSEL = 1). 0: Disable the low threshold asynchronous interrupt 1: Enable the low threshold asynchronous interrupt
4	EMAIE	End of measurement asynchronous interrupt enable Set and reset by software to enable/disable the end of measurement asynchronous interrupt (only when REFSEL = 1). 0: Disable the end of measurement asynchronous interrupt

		1: Enable the end of measurement asynchronous interrupt
3	Reserved	Must be kept at reset value.
2	HTIE	High threshold interrupt enable Set and reset by software to enable/disable the high threshold interrupt which is synchronized on PCLK. 0: Disable the high threshold interrupt 1: Enable the high threshold interrupt
1	LTIE	Low threshold interrupt enable Set and reset by software to enable/disable the low threshold interrupt which is synchronized on PCLK. 0: Disable the low threshold interrupt 1: Enable the low threshold interrupt
0	EMIE	End of measurement interrupt enable Set and reset by software to enable/disable the end of measurement interrupt which is synchronized on PCLK. 0: Disable the end of measurement interrupt 1: Enable the end of measurement interrupt

29.5.8. Interrupt clear flag register (LPDTS_INTC)

Address offset: 0x28
System 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	HTAIC	High threshold asynchronous interrupt clear Write 1 by software to clear the HTAIF flag in the LPDTS_STAT register.
5	LTAIC	Low threshold asynchronous interrupt clear Write 1 by software to clear the LTAIF flag in the LPDTS_STAT register.
4	EMAIC	End of Measure asynchronous interrupt clear Write 1 by software to clear the EMAIF flag of the LPDTS_STAT register.

3	Reserved	Must be kept at reset value.
2	HTIC	High threshold interrupt clear Write 1 by software to clear the HTIF flag in the LPDTS_STAT register.
1	LTIC	Low threshold interrupt clear Write 1 by software to clear the LTIF flag in the LPDTS_STAT register.
0	EMIC	End of measurement interrupt clear Write 1 by software to clear the EMIF flag in the LPDTS_STAT register.

29.5.9. Option register (LPDTS_OP)

Address offset: 0x2C

System reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:0	OP[31:0]	general purpose option bits

30. TFT-LCD interface (TLI)

30.1. Overview

The TLI (TFT-LCD Interface) module handles the synchronous LCD interface and provides pixel data, clock and timing signals for passive LCD display. It supports a wide variety of displays with fully programmable timing parameters. A built-in DMA engine continuously move data from system memory to TLI and then, output to an external LCD display. Two separate layers are supported in TLI, as well as layer window and blending function.

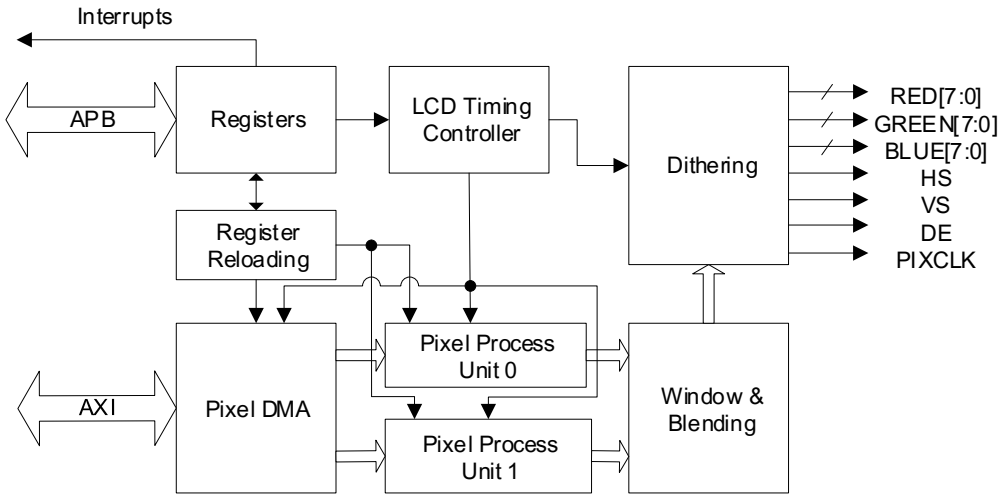
30.2. Characteristics

- Supports up to 24 bits data output per pixel
- Supports up to 2048 x 2048 resolution
- Timing parameters is fully programmable
- Built-in DMA engine to handle frame data copy
- 2 separate frame layers with window and blending function
- Support various pixel formats: ARGB8888, RGB888, RGB565, etc
- Support CLUT (Color Look-Up-Table) and Color-Keying format
- Dithering operation to low bits of a pixel
- AXI master interface with burst of 16 double-words
- 2 display layers with dedicated FIFO (64x64-bit)

30.3. Block diagram

Figure 30-1. TLI module block diagram shows the block diagram of the TLI module. There are three clock domains in TLI. The register works in APB clock and is visited by system APB bus. The Pixel DMA module works in AXI clock and fetches pixel data from system memory using AXI bus. The remaining modules work in TLI clock. The TLI clock is divided from PLL2 clock. The parameters of PLL2 and dividing factor are configured in RCU module.

Figure 30-1. TLI module block diagram



30.4. Signal description

TLI provides a 24-bit RGB Parallel display interface, which is shown in *Table 30-1. Pins of display interface provided by TLI*.

Table 30-1. Pins of display interface provided by TLI

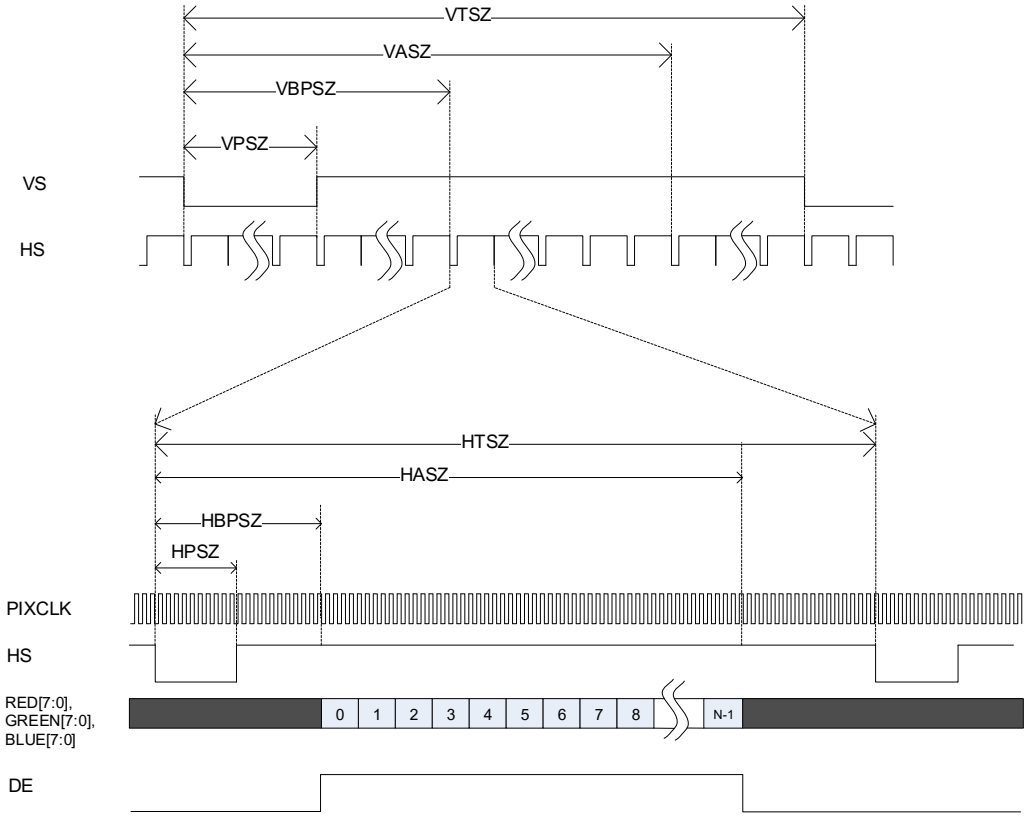
Direction	Name	Pin	Width	Description
Output	HS	TLI_HSYNC	1	Horizontal synchronous
Output	VS	TLI_VSYNC	1	Vertical synchronous
Output	DE	TLI_DE	1	Data enable
Output	PIXCLK	TLI_PIXCLK	1	Pixel clock
Output	RED[7:0]	TLI_Rx	8	Pixel red data
Output	GREEN[7:0]	TLI_Gx	8	Pixel green data
Output	BLUE[7:0]	TLI_Bx	8	Pixel blue data

30.5. Function overview

30.5.1. LCD display timing

LCD interface is a synchronous data interface with pixel clock, pixel data and horizontal and vertical synchronous signals. The *Figure 30-2. Display timing diagram* shows the signal timing of HS and VS for a whole frame. The timing parameters are configured in TLI_SPSZ, TLI_BPSZ, TLI_ASZ and TLI_TSZ registers. The timing values in these registers assume that the position of the first point is (0, 0).

Figure 30-2. Display timing diagram



30.5.2. Pixel DMA function

Following the configuration of register module, the Pixel DMA reads pixel data from memory to the pixel buffer in internal PPU (Pixel Process Unit) continuously.

After enabled, the Pixel DMA begins to fetch pixel data from system and push these data into the pixel buffer in PPU as long as the pixel buffer is not full. It always tries to use BURST16 AXI transaction to fetch pixel data.

TLI supports 2 separate frame layers and each layer has a separate frame buffer address in system. The Pixel DMA has only one AXI access interface, so it will perform round-robin arbitration between the 2 layers during pixels fetching, if both layers are enabled.

FBADD in TLI_LxFBADDR register define the frame buffer address or fetching address of each layer.

FLL in TLI_LxFLLEN defines the line length in bytes of a frame. If the length of a frame line in bytes is N, program FLL with N+7.

There may be some spacing between two frame lines in system memory and the spacing information is defined by STDOFF in TLI_LxFLLEN register. For example if the address of the first pixel in a frame line is M, and the address of the first pixel in the next frame line will be M+STDOFF. If there is no memory spacing between frame lines, just program STDOFF with FLL-7.

FTLN in TLI_LxFTLN register defines the number of lines in a frame.

30.5.3. Pixel formats

The Pixel DMA pushes pixel data into PPU in double words format and PPU (Pixel Process Unit) is responsible for converting various pixel formats into an internal ARGB8888 format. TLI supports up to eight pixel formats as shown in *Table 30-2. Supported pixel formats*. The PPF[2:0] in TLI_LxPPF register defines the pixel format.

ARGB8888 format needs 8-bits data in each channel (Alpha, Red, Green and Blue), while ARGB1555 and ARGB4444 formats have fewer bits than 8 in some channels. PPU converts these formats into ARGB8888 by filling LSBs with MSBs for each channel. When processing RGB888 and RGB565 formats, PPU assumes that Alpha=255 and also fill filling LSBs with MSBs if the channel bit number less than 8.

AL88, AL44 and L8 formats are LUT (Look-Up-Table) formats. In these channels, L is the address of the look-up table. TLI has 2 internal look-up tables: one for each layer. The internal look-up table size is 256x24bits (256 entries and each entry stores a 24-bits RGB value). When processing LUT format pixel, PPU reads out an entry from the look-up table and uses this entry as the RGB value. Because the address of look-up table is 8-bit, PPU also fill LSBs with MSBs if L channel has bits less than 8. The entries in the look-up tables are uninitialized after reset, so the application should initialize the look-up table with proper value using TLI_LxLUT register before display a look-up table format layer. The TLI_LxLUT is a write-only register and a write operation to this register will write an entry to the look-up table.

Each layer is able to be configured into color keying mode. The register LxCKEY defines a RGB value. When color keying mode is enabled for a layer, PPU will compare each RGB value of each pixel in this layer with the LxCKEY and force the pixel's ARGB value to 0 if the value matches.

Table 30-2. Supported pixel formats

PPF[2:0]	Pixel Format
000	ARGB8888
001	RGB888
010	RGB565
011	ARGB1555
100	ARGB4444
111	AL88
101	L8
110	AL44

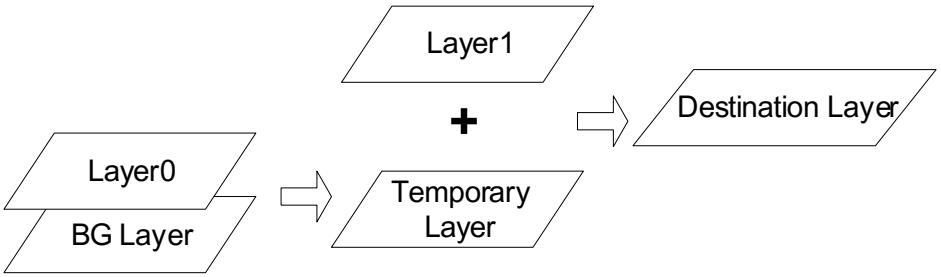
30.5.4. Layer window and blending function

TLI supports window function for each layer and blending function between two layers. TLI first perform window operation to each layer and then blend two layers into a frame.

The window function defines a display window, and each layer has separate window parameters defined by TLI_LxHPOS and TLI_LxVPOS registers. These window parameters define a window inside the layer. The pixel inside the window will keep its original value, while the pixel outside will be replaced with a default pixel defined in TLI_LxDC register.

The blending units first blends Layer0 and BG Layer into a temporary layer, and then blends Layer1 and the temporary layer into destination layer. BG Layer's ARGB value is defined TLI_BGC register. If a layer is disabled, blending function uses the layer's default color.

Figure 30-3. Block diagram of Blending



Blending formula

The general blending formula is:

$$BC=BF_1*C+BF_2*C_S \quad (30\text{-}1)$$

- BC is blended color
- BF_1 is alpha calculation factor 1 of blending method
- C is current layer color
- BF_2 is alpha calculation factor 2 of blending method
- C_S is subjacent layers blended color

The blend factor of current pixel is either normalization Pixel Alpha x normalization Specified Alpha or normalization Specified Alpha which is decided by register configuration.

30.5.5. Layer configuration reload

As is described above, each layer has its own frame buffer, pixel format, window, default color configuration registers and each register has a shadow register. A shadow register share the same address with the real register. Each time when the application writes to a layer-related register address, the corresponding shadow registers is updated immediately, while the real register will not change until a reload operation and only the real register has effect to the TLI function.

There are two methods for application to trigger a reload operation: request reload and frame blank reload. For request reload mode, TLI begins to load the shadow registers into real registers immediately after application set RQR bit in TLI_RL register. For frame blank reload mode, after setting FBR bit in TLI_RL register, the TLI waits for a frame vertical blanking and load the shadow registers. In both modes, hardware automatically clears the RQR or FBR bit

after successfully reload.

30.5.6. Dithering function

The dithering module adds a 2-bit pseudo-random value to each pixel channel. This function is able to make the image smoother when 18-bits interface is used to display a 24-bit data. Application may switch on this function using DFEN bit in TLI_CTL register.

30.6. Interrupts

There are several status and error flags in TLI, and interrupt may be asserted from these flags. The status flags will assert global interrupt, while the error flags will assert error interrupt.

Table 30-3. Status flags

Status Flag Name	Description
LMF	Line mark flag
LCRF	Layer configuration reloaded flag

Table 30-4. Error flags

Error Flag Name	Description
TEF	Transaction error flag
FEF	FIFO error flag

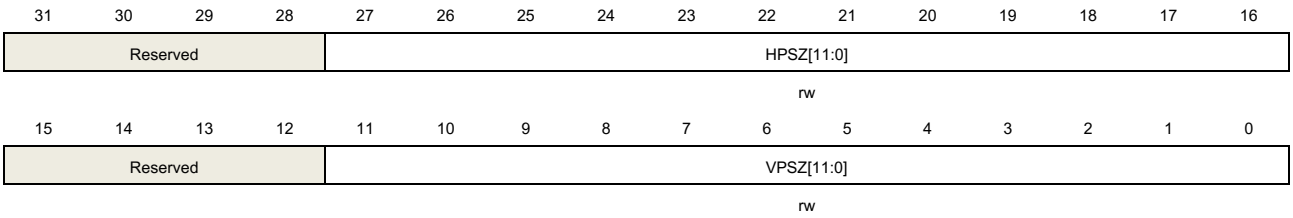
30.7. Register definition

TLI base address: 0x5000 1000

30.7.1. Synchronous pulse size register (TLI_SPSZ)

Address offset: 0x08
Reset value: 0x0000 0000

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

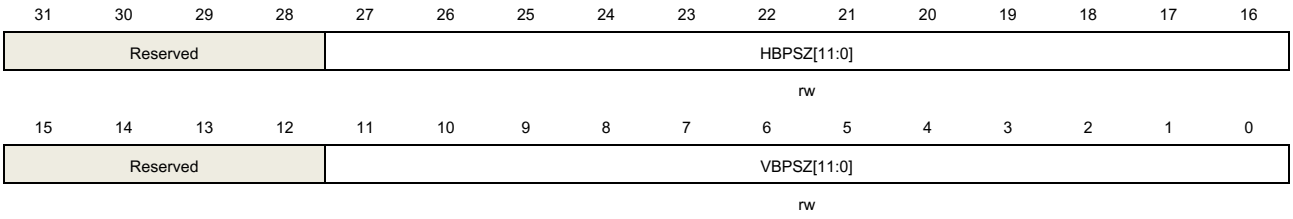


Bits	Fields	Descriptions
31:28	Reserved	Must be kept at reset value.
27:16	HPSZ[11:0]	Size of the horizontal synchronous pulse The HPSZ value should be configured to the pixels number of horizontal synchronous pulse minus 1.
15:12	Reserved	Must be kept at reset value.
11:0	VPSZ[11:0]	Size of the vertical synchronous pulse The VPSZ value should be configured to the pixels number of vertical synchronous pulse minus 1.

30.7.2. Back-porch size register (TLI_BPSZ)

Address offset: 0x0C
Reset value: 0x0000 0000

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



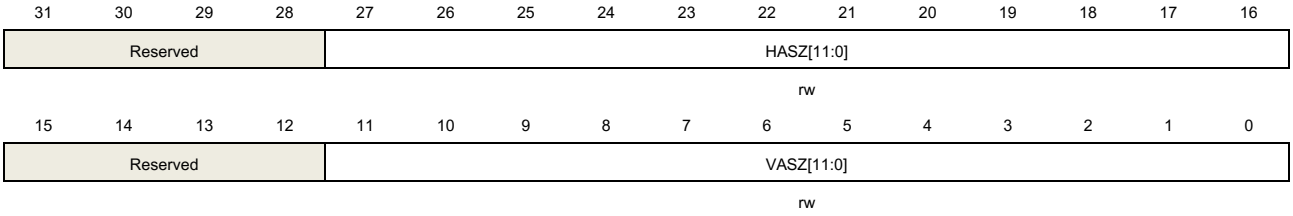
Bits	Fields	Descriptions
31:28	Reserved	Must be kept at reset value.
27:16	HBPSZ[11:0]	Size of the horizontal back porch plus synchronous pulse

		The HBPSZ value should be configured to the pixels number of horizontal back porch and synchronous pulse minus 1.
15:12	Reserved	Must be kept at reset value.
11:0	VBPSZ[11:0]	Size of the vertical back porch plus synchronous pulse The VBPSZ value should be configured to the pixels number of vertical back porch and synchronous pulse minus 1.

30.7.3. Active size register (TLI_ASZ)

Address offset: 0x10
Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:28	Reserved	Must be kept at reset value.
27:16	HASZ[11:0]	Size of the horizontal active area width plus back porch and synchronous pulse The HASZ value should be configured to the pixels number of horizontal active area width plus back porch and synchronous pulse minus 1.
15:12	Reserved	Must be kept at reset value.
11:0	VASZ[11:0]	Size of the vertical active area width plus back porch and synchronous pulse The VASZ value should be configured to the pixels number of vertical active area height plus back porch and synchronous pulse minus 1.

30.7.4. Total size register (TLI_TSZ)

Address offset: 0x14
Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:28	Reserved	Must be kept at reset value.
27:16	HTSZ[11:0]	Horizontal total size of the display, including active area, back porch, synchronous pulse and front porch The HTSZ value should be configured to the pixels number of horizontal active area width plus back porch, front porch and synchronous pulse minus 1.
15:12	Reserved	Must be kept at reset value.
11:0	VTSZ[11:0]	Vertical total size of the display, including active area, back porch, synchronous pulse and front porch The VTSZ value should be configured to the pixels number of vertical active area height plus back porch, front porch and synchronous pulse minus 1.

30.7.5. Control register (TLI_CTL)

Address offset: 0x18

Reset value: 0x0000 2220

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HPPS	VPPS	DEPS	CLKPS	Reserved											DFEN
rw	rw	rw	rw												rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	RDB[2:0]			Reserved	GDB[2:0]			Reserved	BDB[2:0]			Reserved			TLIEN
	r				r				r						rw

Bits	Fields	Descriptions
31	HPPS	Horizontal pulse polarity selection 0: Horizontal synchronous pulse active low 1: Horizontal synchronous pulse active high
30	VPPS	Vertical pulse polarity selection 0: Vertical synchronous pulse active low 1: Vertical synchronous pulse active high
29	DEPS	Not data enable polarity selection 0: Not data enable active low 1: Not data enable active high
28	CLKPS	Pixel clock polarity selection 0: Pixel clock is TLI clock 1: Pixel clock is inverted TLI clock
27:17	Reserved	Must be kept at reset value.
16	DFEN	Dither function enable

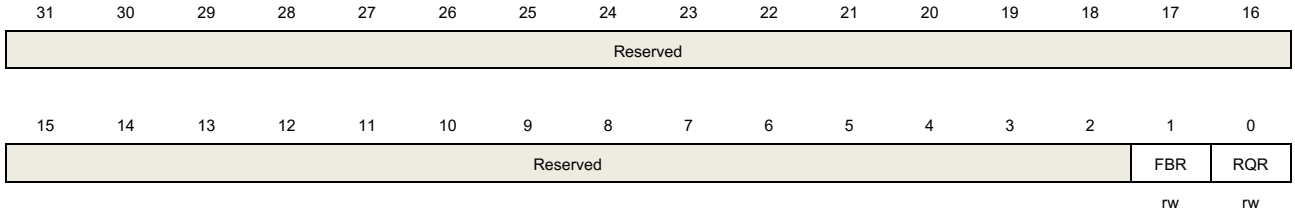
		0: Dither function disable 1: Dither function enable
15	Reserved	Must be kept at reset value.
14:12	RDB[2:0]	Red channel dither bits number Fixed to 2, read only
11	Reserved	Must be kept at reset value.
10:8	GDB[2:0]	Green channel dither bits number Fixed to 2, read only
7	Reserved	Must be kept at reset value.
6:4	BDB[2:0]	Blue channel dither bits number Fixed to 2, read only
3:1	Reserved	Must be kept at reset value.
0	TLIEN	TLI enable bit 0: TLI disable 1: TLI enable

30.7.6. Reload layer register (TLI_RL)

Address offset: 0x24

Reset value: 0x0000 0000

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



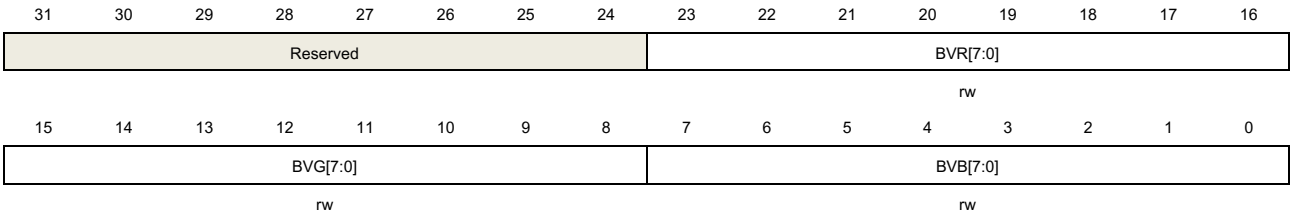
Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	FBR	Frame blank reload This bit is set by software and cleared by hardware after reloading 0: Reload disable 1: The layer configuration will be reloaded into core at frame blank
0	RQR	Request reload This bit is set by software and cleared by hardware after reloading 0: Reload disable 1: The layer configuration will be reloaded into core after this bit sets

30.7.7. Background color register (TLI_BGC)

Address offset: 0x2C

Reset value: 0x0000 0000

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



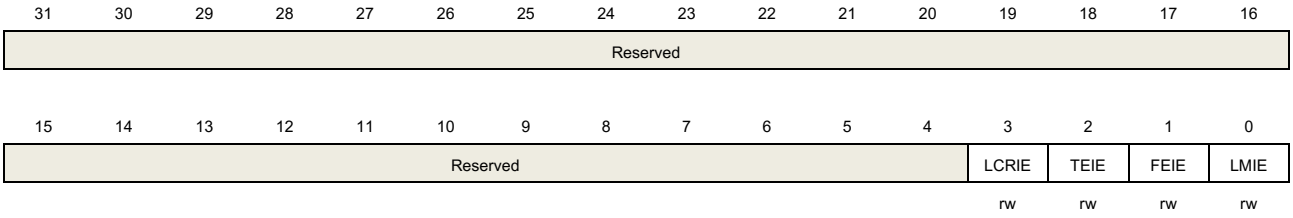
Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:16	BVR[7:0]	Background value red
15:8	BVG[7:0]	Background value green
7:0	BVB[7:0]	Background value blue

30.7.8. Interrupt enable register (TLI_INTEN)

Address offset: 0x34

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	LCRIE	Layer configuration reloaded interrupt enable 0: Layer configuration reloaded flag won't generate an interrupt 1: Layer configuration reloaded flag will generate an interrupt
2	TEIE	Transaction error interrupt enable 0: Transaction error flag won't generate an interrupt 1: Transaction error flag will generate an interrupt
1	FEIE	FIFO error interrupt enable 0: FIFO error flag won't generate an interrupt

		1: FIFO error flag will generate an interrupt
0	LMIE	Line mark interrupt enable 0: Line mark flag won't generate an interrupt 1: Line mark flag will generate an interrupt

30.7.9. Interrupt flag register (TLI_INTF)

Address offset: 0x38

Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												LCRF	TEF	FEF	LMF
												r	r	r	r

Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value.
3	LCRF	Layer configuration reloaded flag 0: No layer configuration reloaded flag 1: Layer configuration is reloaded triggered by FBR bit in TLI_RL
2	TEF	Transaction error flag 0: No transaction error flag 1: A transaction error on AXI bus occurs
1	FEF	FIFO error flag 0: No FIFO error flag 1: A FIFO under-run error occurs The under-run error occurs when the value written in TLI_LxFLLEN and TLI_LxFTLN is less than required.
0	LMF	Line mark flag 0: No line mark flag 1: Line number reaches the specified value in TLI_LM

30.7.10. Interrupt flag clear register (TLI_INTC)

Address offset: 0x3C

Reset value: 0x0000 0000

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

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

Reserved

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												LCRC	TEC	FEC	LMC
												w	w	w	w

Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value.
3	LCRC	Layer configuration reloaded flag clear Write 1 to clear layer configuration reloaded flag
2	TEC	Transaction error flag clear Write 1 to clear transaction error flag
1	FEC	FIFO error flag clear Write 1 to clear FIFO error flag
0	LMC	Line mark flag clear Write 1 to clear line mark flag

30.7.11. Line mark register (TLI_LM)

Address offset: 0x40

Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:11	Reserved	Must be kept at reset value.
10:0	LM[10:0]	Line mark value The LMF bit in TLI_INTF will be set after the line number reaches this value

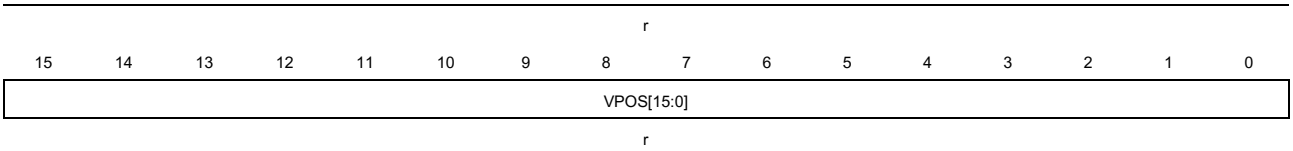
30.7.12. Current pixel position register (TLI_CPPOS)

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
HPOS[15:0]															



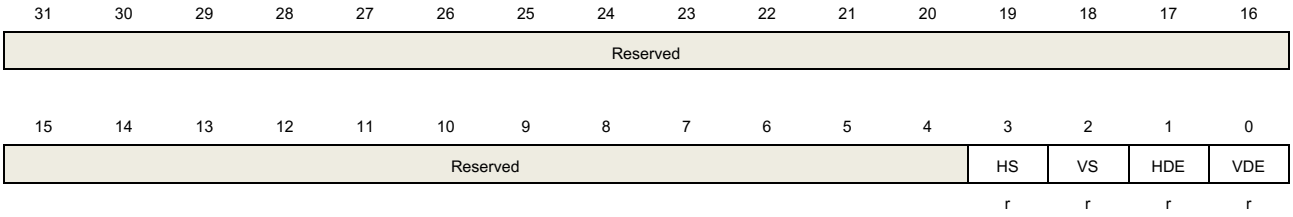
Bits	Fields	Descriptions
31:16	HPOS[15:0]	Horizontal position Horizontal position of the current displayed pixel
15:0	VPOS[15:0]	Vertical position Vertical position of the current displayed pixel

30.7.13. Status register (TLI_STAT)

Address offset: 0x48

Reset value: 0x0000 000F

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



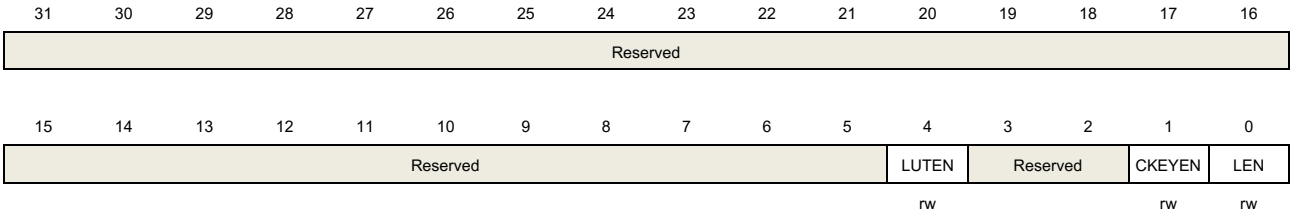
Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value.
3	HS	Current HS status of the TLI
2	VS	Current VS status of the TLI
1	HDE	Current HDE status 0: HPOS in TLI_CPPOS register is not between the HBPSZ in TLI_BPSZ register and HASZ in TLI_ASZ register. 1: HPOS in TLI_CPPOS register is between the HBPSZ in TLI_BPSZ register and HASZ in TLI_ASZ register.
0	VDE	Current VDE status 0: VPOS in TLI_CPPOS register is not between the VBPSZ in TLI_BPSZ register and VASZ in TLI_ASZ register. 1: VPOS in TLI_CPPOS register is between the VBPSZ in TLI_BPSZ register and VASZ in TLI_ASZ register.

30.7.14. Layer x control register (TLI_LxCTL) (x = 0,1)

Address offset: 0x84+0x80 * x

Reset value: 0x0000 0000

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



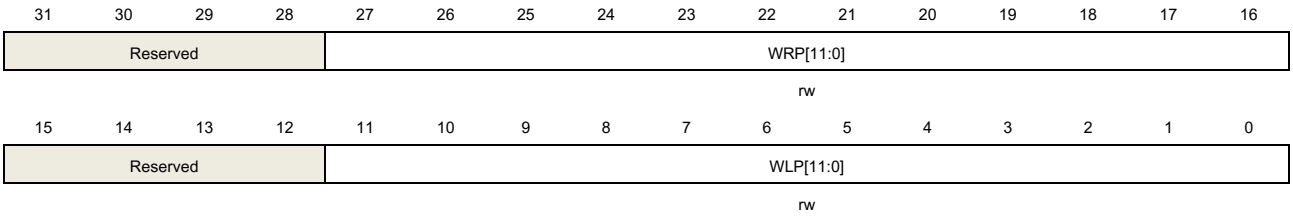
Bits	Fields	Descriptions
31:5	Reserved	Must be kept at reset value.
4	LUTEN	LUT enable 0: LUT is disabled 1: LUT is enabled
3:2	Reserved	Must be kept at reset value.
1	CKEYEN	Color keying enable 0: Color keying is disabled 1: Color keying is enabled
0	LEN	Layer enable 0: This layer is disabled 1: This layer is enabled

30.7.15. Layer x horizontal position parameters register (TLI_LxHPOS) (x = 0,1)

Address offset: 0x88+0x80 * x

Reset value: 0x0000 0000

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



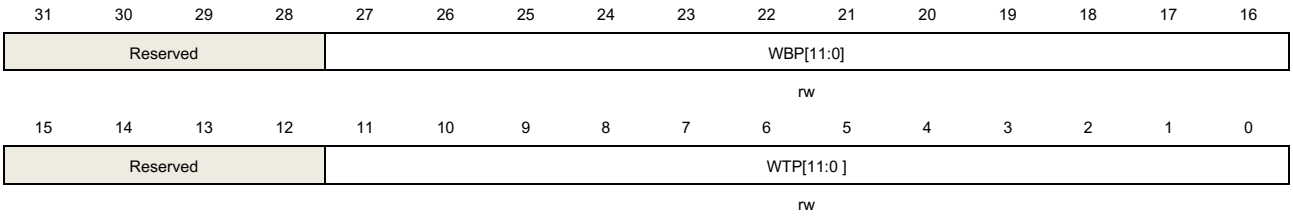
Bits	Fields	Descriptions
31:28	Reserved	Must be kept at reset value.
27:16	WRP[11:0]	Window right position
15:12	Reserved	Must be kept at reset value.
11:0	WLP[11:0]	Window left position

30.7.16. Layer x vertical position parameters register (TLI_LxVPOS) (x = 0,1)

Address offset: 0x8C+0x80 * x

Reset value: 0x0000 0000

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



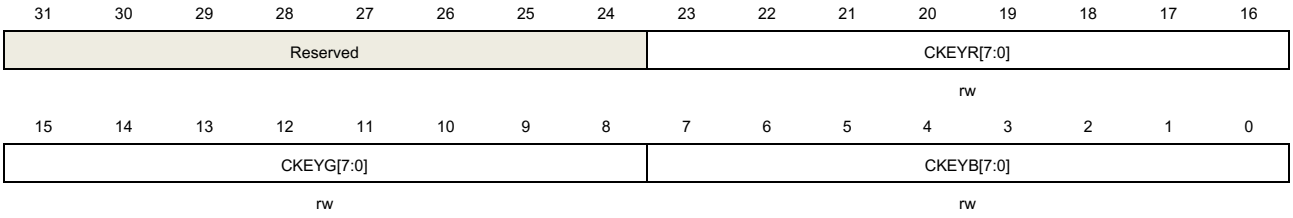
Bits	Fields	Descriptions
31:28	Reserved	Must be kept at reset value.
27:16	WBP[11:0]	Window bottom position
15:12	Reserved	Must be kept at reset value.
11:0	WTP[11:0]	Window top position

30.7.17. Layer x color key register (TLI_LxCKEY) (x = 0,1)

Address offset: 0x90+0x80 * 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.
23:16	CKEYR[7:0]	Color key red
15:8	CKEYG[7:0]	Color key green
7:0	CKEYB[7:0]	Color key blue

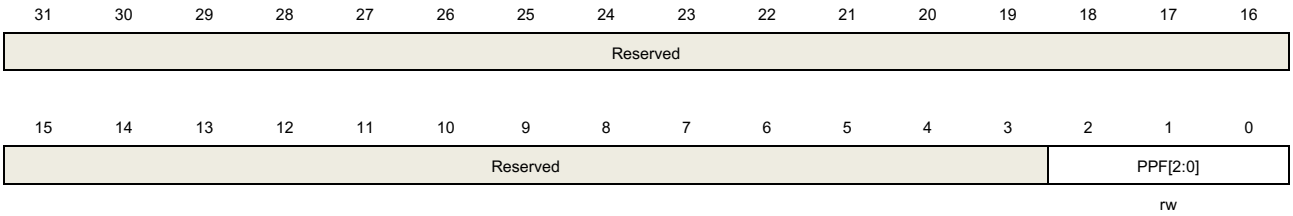
If the pixel RGB value in a layer equals the value in TLI_LxCKEY, the pixel RGB value is reset to 0. That means these pixels is transparent to other layers.

30.7.18. Layer x packeted pixel format register (TLI_LxPPF) (x = 0,1)

Address offset: 0x94+0x80 * x

Reset value: 0x0000 0000

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



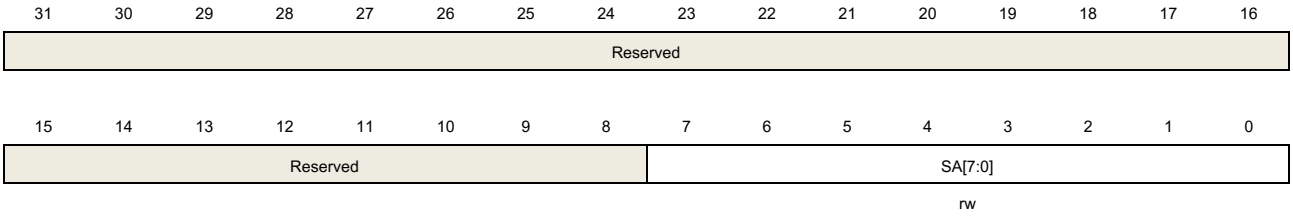
Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value.
2:0	PPF[2:0]	Packeted pixel format These bits configures the packeted pixel format 000: ARGB8888 001: RGB888 010: RGB565 011: ARGB1555 100: ARGB4444 101: L8 110: AL44 111: AL88

30.7.19. Layer x specified alpha register (TLI_LxSA) (x = 0,1)

Address offset: 0x98+0x80 * x

Reset value: 0x0000 00FF

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



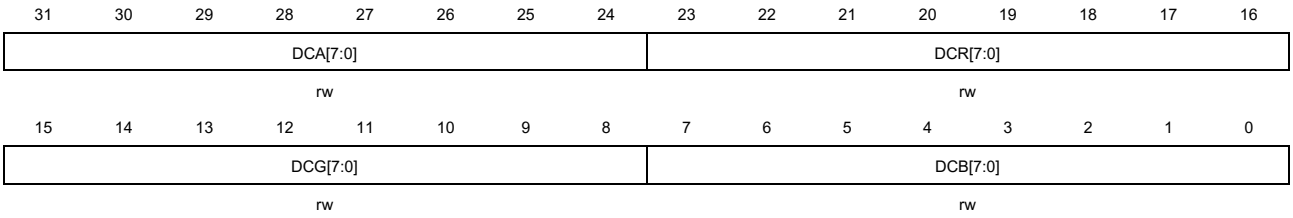
Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	SA [7:0]	Specified alpha The alpha value used for blending

30.7.20. Layer x default color register (TLI_LxDC) (x = 0,1)

Address offset: 0x9C+0x80 * x

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:24	DCA[7:0]	The default color alpha
23:16	DCR[7:0]	The default color red
15:8	DCG[7:0]	The default color green
7:0	DCB[7:0]	The default color blue

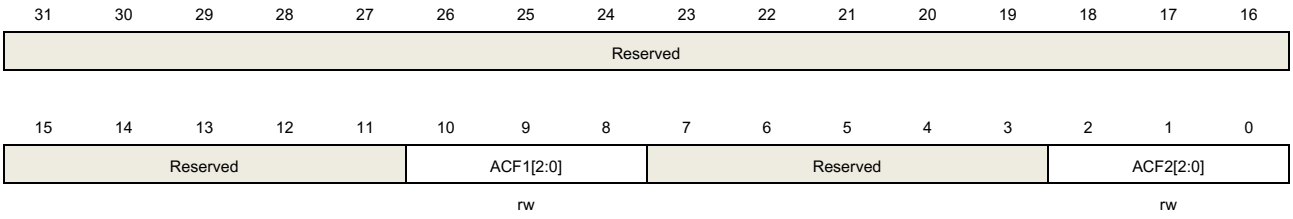
The default color of a layer takes effect when the layer is disabled or outside the window defined in TLI_LxHPOS and TLI_LxVPOS.

30.7.21. Layer x blending register (TLI_LxBLEND) (x = 0,1)

Address offset: 0xA0+0x80 * x

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:11	Reserved	Must be kept at reset value.
10:8	ACF1[2:0]	Alpha calculation factor 1 of blending method 000: Reserved 001: Reserved 010: Reserved 011: Reserved 100: normalization Specified Alpha 101: Reserved

		110: normalization Pixel Alpha x normalization Specified Alpha 111:Reserved
7:3	Reserved	Must be kept at reset value.
2:0	ACF2[2:0]	Alpha calculation factor 2 of blending method 000: Reserved 001: Reserved 010: Reserved 011: Reserved 100: Reserved 101:1- normalization specified alpha 110: Reserved 111: 1-normalization pixel alpha x normalization specified alpha

30.7.22. Layer x frame base address register (TLI_LxFBADDR) (x = 0,1)

Address offset: 0xAC+0x80 * 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
FBADD[31:16]															
rw															

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

Bits	Fields	Descriptions
31:0	FBADD[[31:0]	Frame buffer base address The base address of frame buffer

30.7.23. Layer x frame line length register (TLI_LxFLLEN) (x = 0,1)

Address offset: 0xB0+0x80 * 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		STDOFF[13:0]													
		rw													

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

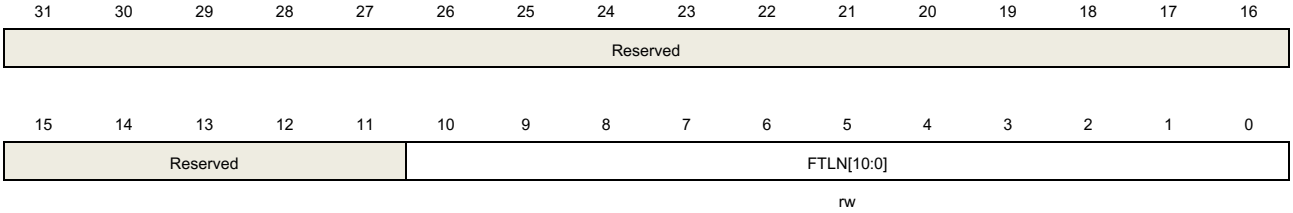
Bits	Fields	Descriptions

31:30	Reserved	Must be kept at reset value.
29:16	STDOFF[13:0]	Frame buffer stride offset This value defines the bytes number from start of a line to the start of next line
15:14	Reserved	Must be kept at reset value.
13:0	FLL[13:0]	Frame line length This value defines the bytes number of a line plus 7

30.7.24. Layer x frame total line number register (TLI_LxFTLN) (x = 0,1)

Address offset: 0xB4+0x80 * x
Reset value: 0x0000 0000

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

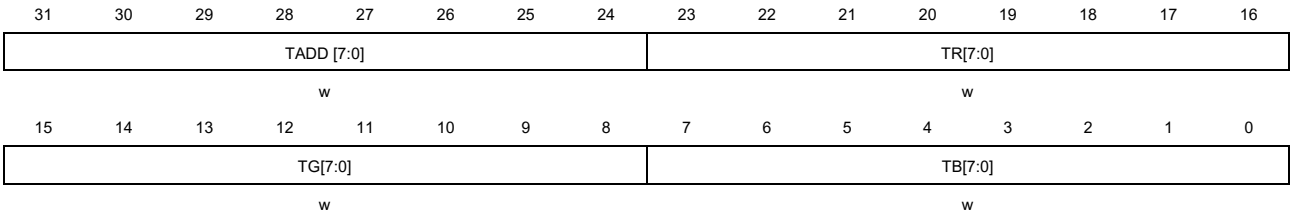


Bits	Fields	Descriptions
31:11	Reserved	Must be kept at reset value.
10:0	FTLN[10:0]	Frame total line number This value defines the line number in a frame

30.7.25. Layer x look up table register (TLI_LxLUT) (x = 0,1)

Address offset: 0xC4+0x80 * x
Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:24	TADD[7:0]	Look up table write address The entry at this address in LUT will be updated with the value of red, green and bule written

23:16	TR[7:0]	Red channel of a LUT entry
15:8	TG[7:0]	Green channel of a LUT entry
7:0	TB[7:0]	Blue channel of a LUT entry

31. DSI host controller (DSI)

31.1. Overview

The Display Serial Interface (DSI) is a component of a suite of communication protocols established by the MIPI® Alliance. The MIPI® DSI Host Controller is a digital module that encompasses the full range of protocol functionalities as outlined in the MIPI® DSI Specification. This controller facilitates a connection between the system and the MIPI® D-PHY, enabling users to interact with displays that adhere to the DSI standards.

Standard and references:

MIPI D-PHY specification version 1.2

Compliant with DSI MIPI specification for Display Serial Interface (Version1.3.1)

Compliant with D-PHY standard MIPI specification (Version1.2)

Compliant with MIPI Alliance Standard for Display Pixel Interface (DPI-2) Version 2.00

31.2. Characteristics

DSI controller:

- Compliant with DSI MIPI specification for Display Serial Interface (Version1.3.1).
- Compliant with MIPI Alliance Standard for Display Pixel Interface (DPI-2) Version 2.00.
- Supports Display resolutions - QCIF(resolution 176x144 pixels or 144x176 pixels), QVGA(resolution 320x240 pixels or 240x320 pixels), VGA(resolution 640x480 pixels or 480x640 pixels), WVGA(resolution 800x480 pixels or 480x800 pixels), XVGA(1024x768 pixels).
- Can be programmed to support either video mode or command mode in single channel mode.
- Supports generic read and write commands.
- Supports pixel formats of type 16bpp [RGB565], 18bpp [RGB666], 18bpp [Loosely packed RGB666] and 24bpp [RGB888].
- Supports recovery due to contention.
- Supports timers and recovery schemes to come out of mode fault errors.
- Has watchdog timers to monitor D-PHY activity at low power mode, high speed mode and during turn-around.
- Supports interrupt to report protocol errors and expiry of timers.
- Supports programmable device initialization timers.
- Supports programmable maximum return packet size command.
- Supports DPI interface signals with different polarities.
- Supports PHY data lane programmability up-to two.

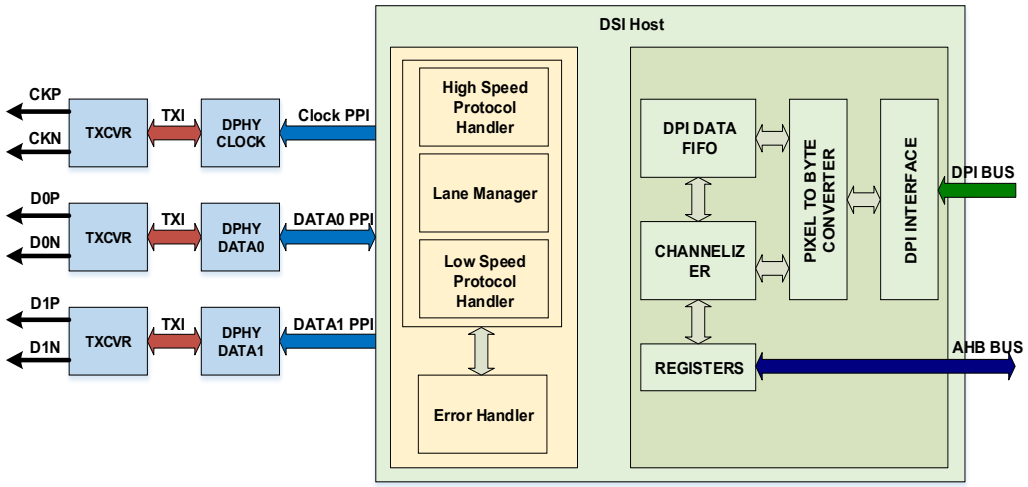
- Supports DPI interface for pixel stream transfer.
- Supports data lane switching to low power mode during idle time
- Supports tearing effect.
- Supports ultra low power mode switching.
- Supports Bus turn around.
- Supports EOT disabling capacity to suit backward compatible Displays.
- Supports Clock stop enabling feature during idle time.
- Supports burst, non-burst video modes with sync pulse and sync events types.
- Added support for deskew calibration.

DPHY:

- Compliant with D-PHY standard MIPI specification (Version1.2).
- Supports synchronous transfer at high speed mode with a bit rate of 80-1500 Mb/s without deskew calibration and up to 2000 Mb/s with deskew calibration.
- Supports asynchronous transfer at low power mode with a bit rate of 10 Mb/s.
- Spaced one hot encoding for Low power [LP] data.
- Supports one clock lane and two data lanes.
- Supports error detection mechanism for sequence errors and contentions.
- Data lanes support transfer of data in high speed mode.
- Supports ultra-low power mode, high speed mode and control mode.
- Has clock divider unit to generate clock for parallel data reception from the PPI.
- Supports 8-bit data PPI to be interoperable with DSI protocol.
- Optional TxClkEsc divide by 2 support.
- Clock lane supports unidirectional communication.
- On-chip clock generation is configurable for the transmitter.
- PHY can be configured only as a master.
- Core is structured to increase the number of data lanes.
- Supports High speed mode in Forward communication.

31.3. Block diagram

Figure 31-1. Functional block diagram of DSI Host



31.4. Function overview

31.4.1. High Speed Protocol Handler

This module handles the HS transmission. Depending on the packet information available in the control FIFO for DCS/Generic transfers, frames and transmits the short packets and packet headers. If the control information is for a long packet, then the data from the Data FIFO is padded as referenced by the number of bytes information in the control FIFO. It has an ECC/CRC generator block to handle ECC and CRC generation for packet header, short packet and long packet respectively. Depending on the signalling in the DPI interface and the programmed parameters from the registers, short packets/long packets are generated and transmitted. Pixels are buffered in a DPI data FIFO.

31.4.2. Low Speed Protocol Handler

This module handles the forward LP data transfer and LP data reception. It completely removes the wrapper and segregates the data bytes in the packet format. Packet header, short packet, packet footer, payload data is extracted from the data stream and fed to the ECC and CRC checkers for ensuring the correctness of the data stream. Reverse data reception at low speed is handled by this block. Bus turnaround is also handled by this unit.

31.4.3. Channelizer

This module has a state machine that grants the DCS transfers or Generic transfers on the DSI traffic whenever the Hsync event, Vsync event or RGB data is not transmitted by the DPI

stream. The BLLP timings are fully occupied for DCS/Generic transfer or Generic transfer or DCS commands transfer via DSI packets.

31.4.4. Lane Manager

It has a lane distributor that distributes its operation along all the supported data lanes. Number of data lanes can be programmed via registers in AHB target interface. It also has logics to drive the clock and data lanes of the D-PHY.

31.4.5. Error Handler

It has logics to predict errors that are happening on the low power mode due to contention. It has timers like high speed transmit timeout, low speed reception timers, turnaround timeout counters to recover itself from fault mode condition and device reset timers for recovery operations from contention situation. Recorded protocol errors are passed on to processor as interrupts.

31.4.6. DPI Interface

After power-up, when all the operational registers are programmed, the DPI interface starts recording the data from the DPI interface. It has a state machine to capture the data in a FIFO when Data enable signal is asserted. It records shut down and turn on events. It triggers the packetiser state machine after the current frame is transmitted. It automatically generates Hsync start, end events, blanking packets for horizontal back and front porch area based on the inputs from the target registers in the AHB. The DSI Tx transfers the SD and CM (color mode on or off) packets during Vertical Back Porch (VBP).

31.4.7. AHB slave Interface

This block includes all DSI Tx operational registers. It connects DSI Host core to external processor using an AHB slave interface and drives the 32-bit data and 32-bit address buses. This interface mainly handles the DSI Tx for initial operations and parse the DCS/generic commands.

Supports DCS/Generic transfers in both high speed and low power mode. It has High speed/low power control and data port registers.

- For short write,

  The control data port register is written with short packet information.

- For long packet,

  The payload word count value is written into payload word count register.

  Payload data is written into data port register.

Once payload data write is completed, the header information need to be written into control port register.

31.4.8. DCS Generic FIFO logics

- For short packets ECC is calculated and will be written into control FIFO
- For long packets CRC is calculated and will be written into data FIFO
- FIFO status information are updated in FIFO status register. Based on that DCS/Generic write and read transfers are controlled
  - FIFO full is asserted when FIFO has location to accept 4 write transfers
    - Only 4 write cycles are allowed when FIFO full status is updated in interrupt status register (DSI_INTSTAT1)
- Control RAM of 256x32-bit is used for storing control information.
  - This RAM is split up into two equal parts
    - Lower half of the RAM (address - 0x00 to 0x7f) is used to store high speed generic/DCS commands
    - Upper half of the RAM (address - 0x80 to 0xff) is used to store low power DCS/generic commands
- Data RAM of 256x32-bit is used for storing data payload information.
  - This RAM is split up into two equal parts
    - Lower half of the RAM (address - 0x00 to 0x7f) is used to store high speed generic/DCS long packet payload data
    - Upper half of the RAM (address - 0x80 to 0xff) is used to store low power DCS/generic long packet payload data

Note: Maximum data payload for DCS/generic long write is limited to 492 bytes.

31.4.9. DPI Data FIFO

Data FIFO simply buffers the pixel data that is streaming from the DPI interface in 24-bit, based on the programmed color format. 2048 x32-bit DPRAM is used for line buffering.

Pixel to byte converter - Convert the pixels to byte format and stores in data FIFO.

31.4.10. D-PHY

Physical Layer has a de-serialiser unit to dialogue with the PPI / lane management unit. Also, has clock divider unit to receive clock during different modes of operation.

31.4.11. RAM

All the RAMs in DSI-Tx need to be Two Port RAMs. The different RAMs and sizes are listed below.

- DPI RAM: 2048 x 32
- DCS/Generic Control RAM: 256 x 32
- DCS/Generic Data RAM: 256 x 32
- DCS/Generic Read RAM: 256 x 32

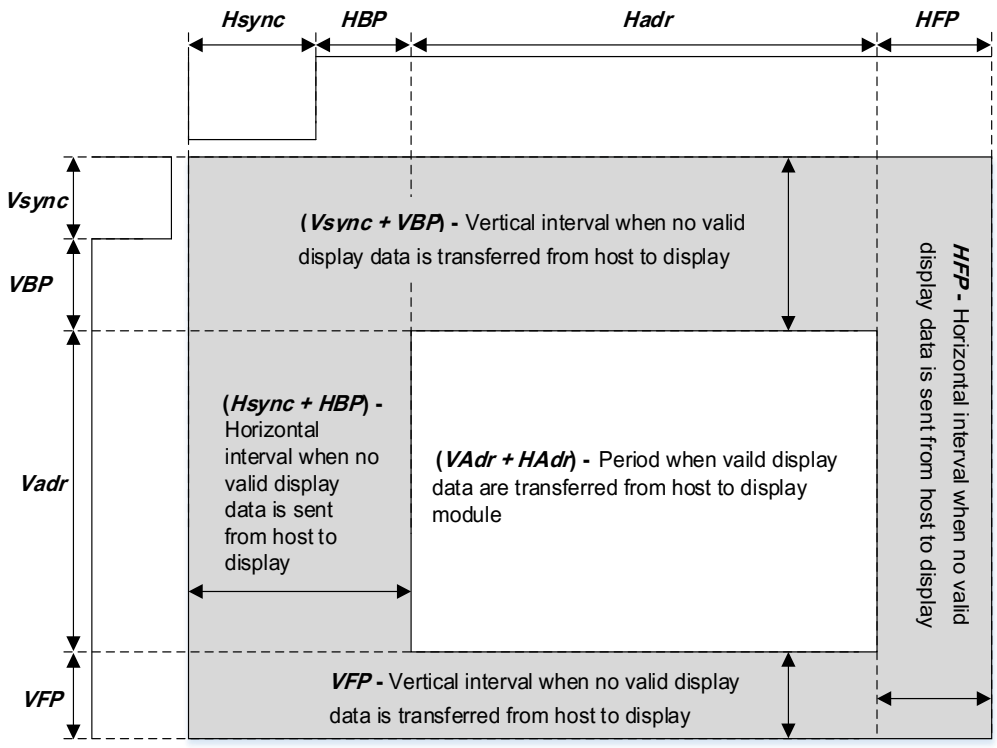
Table 31-1. RAM Details

RAM	Width	Depth	Size (Bytes)	Write clock	Read clock
DCS/Generic control RAM	32	256	1024	ahb_clk	TxByteClkHS
DCS/Generic data RAM	32	256	1024	ahb_clk	TxByteClkHS
DCS/Generic read RAM	32	256	1024	TxByteClkHS	ahb_clk
DPI RAM	32	2048	8192	pclk	TxByteClkHS

31.5. DPI stream to DSI packets

DSI packet flow diagrams are depicted in this section. DPI timing is conveyed over the DSI serial link as stated below.

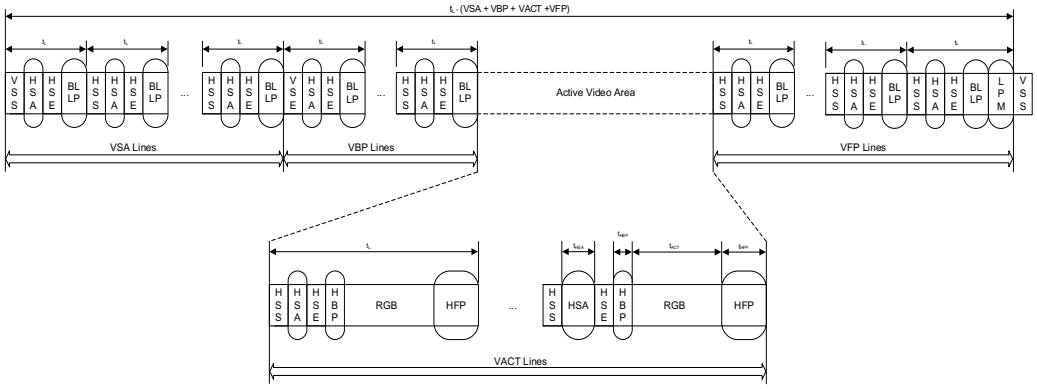
Figure 31-2. DPI Timing conveyed over DSI Link



The following diagram depicts the DSI packet flow for Non burst communication with start

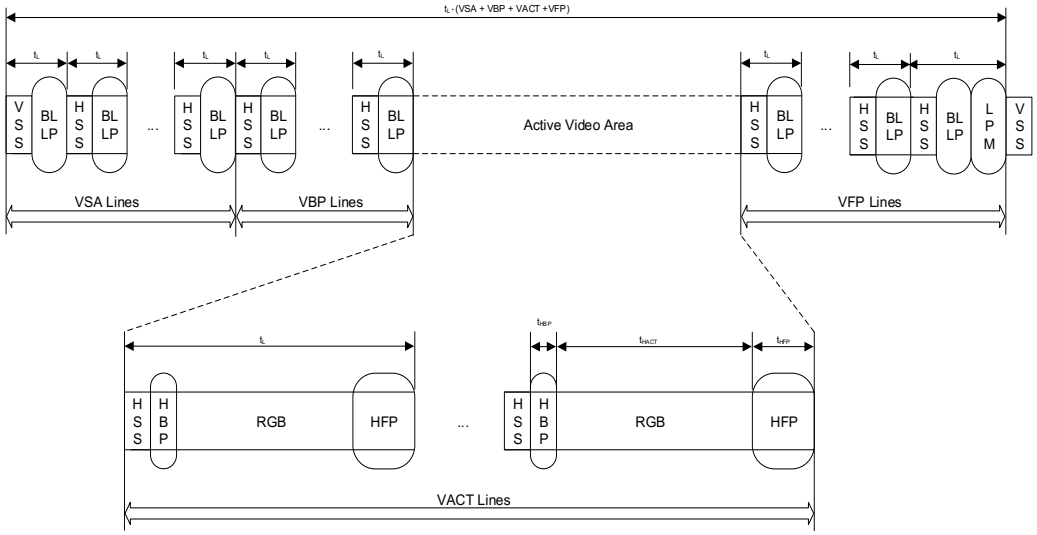
and end.

Figure 31-3. Non Burst Transfer with Start and End



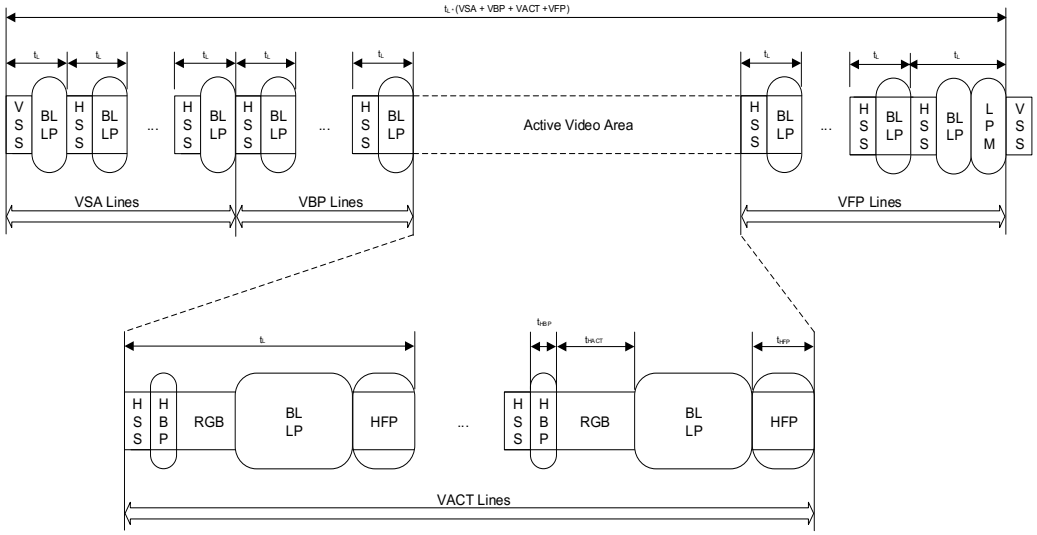
The following diagram depicts the DSI packet flow for Non burst communication with sync event.

Figure 31-4. Non Burst with Sync Event



Corresponding DSI packet flow for burst communication is shown below:

Figure 31-5. DSI Packet flow for Burst communication



31.6. Contention and Recovery Schemes

Contention and recovery schemes are listed below, DSI host and the peripheral initially detect the contention.

Table 31-2. DSI host and peripheral initially detect contention

Host Protocol	Host PHY	Device PHY	Device Protocol
	Detect LP high Fault or LP low fault	Detect LP High Fault or LP Low Fault	
	Transition to Stop state [LP-11]	Transition to LP-RX	
Host Processor wait Timeout		Device waits until it observes stop state before responding	
		Observe stop state	
Request Reset Entry Command to PHY	Send Reset Entry Command	Observe Reset entry Command	
		Flag protocol about Reset Command	Observe Reset entry Command
			Reset Device
	Return to Stop state	Remain in LP-Rx state	
Device reset timeout. Wait until Device completes reset before resuming normal operations	Continue normal operation		Reset completes

31.7. Fault mode recovery schemes

10.1.2 High speed recovery

High speed HS_TX_timeout reg is programmed with a maximum count value that a DSI host may remain at high speed mode. If the timer expires the processor terminates HS transmission and enters EOT sequence and then drives LP-11 stop state.

10.1.3 Low power mode recovery

Low speed LP_TX_timeout reg is programmed with a maximum count value that a DSI host has to wait for low power mode reception. If the timer expires, the DSI host can safely drive stop state LP-11.

10.1.4 Turn around recovery

After a bus turn around event, if there is no turnaround acknowledgement sequence from the peripheral, this timer starts counting. If the timer expires, the DSI host returns to stop state.

31.8. Pixel to Byte encoding formats

Pixel to Byte encoding formats are described below.

31.8.1. RGB666 data Transmission and storage

For a RGB666 stream from DPI is stored in the FIFO in terms of 32-bit data as follows: pattern is repeated for every 9 storage location. A small state machine is used to make up a 32-bit stream from 18-bit stream.

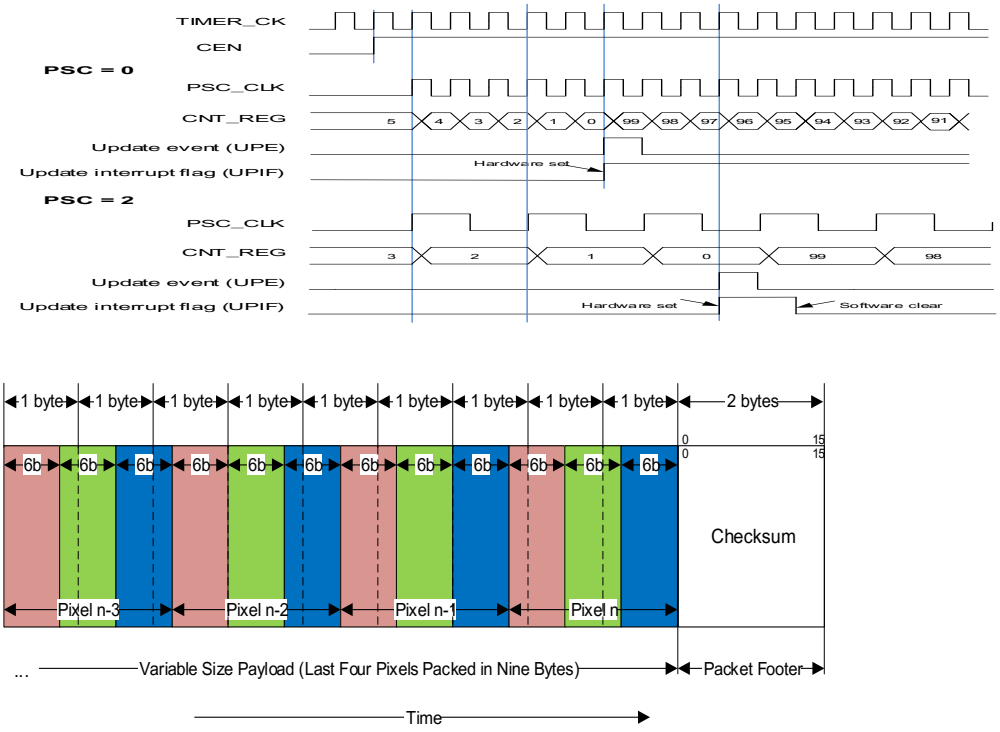
32 Bit Data FIFO Storage

Table 31-3. 32-bit data storage in RGB666

B2[1:0] G2[5:0] R2[5:0] B1[5:0] G1[5:0] R1[5:0]
G4[3:0]R4[5:0] B3[5:0] G3[5:0] R3[5:0] B2[5:2]
R6[5:0] B5[5:0] G5[5:0] R5[5:0] B4[5:0] G4[5:4]
R8[1:0] B7[5:0] G7[5:0] R7[5:0] B6[5:0] G6[5:0]
B9[3:0] G9[5:0] R9[5:0] B8[5:0] G8[5:0] R8[5:2]
. . . . .
. . . . .
. . . . .
. . . . .
Bn[5:0] Gn[5:0] Rn[5:0] Bn-1[5:0] Gn-1[5:0] Rn-1 [1:0]

Transmission of RGB666 stream in DSI lanes.

Figure 31-6. Transmission of RGB666 stream



31.8.2. RGB565 data transmission and storage

Data stored in FIFO

For a RGB565 stream from DPI is stored in the FIFO in terms of 32-bit data as follows: A small state machine is used to make up a 32-bit stream from 16-bit stream.

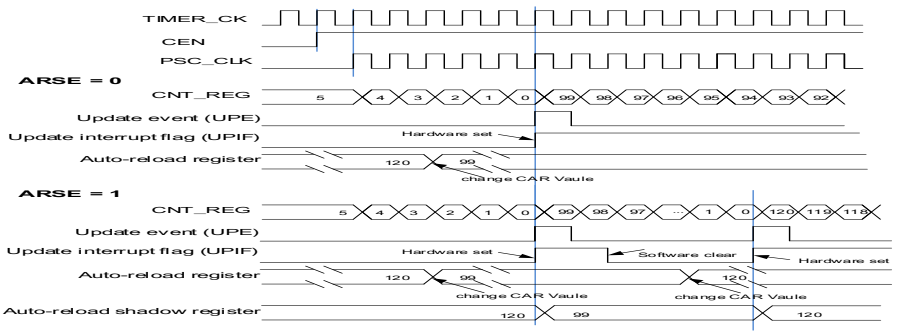
32 Bit Data FIFO Storage

Table 31-4. 32-bit data storage in RGB565

B2[1:0] G2[5:0] R2[5:0] B1[5:0] G1[5:0] R1[5:0]
G4[3:0]R4[5:0] B3[5:0] G3[5:0] R3[5:0] B2[5:2]
R6[5:0] B5[5:0] G5[5:0] R5[5:0] B4[5:0] G4[5:4]
R8[1:0] B7[5:0] G7[5:0] R7[5:0] B6[5:0] G6[5:0]
B9[3:0] G9[5:0] R9[5:0] B8[5:0] G8[5:0] R8[5:2]
. . . . .
. . . . .
. . . . .
. . . . .
Bn[5:0] Gn[5:0] Rn[5:0] Bn-1[5:0] Gn-1[5:0] Rn-1 [1:0]

Transmission of RGB565 stream in DSI lanes.

Figure 31-7. Transmission of RGB565 stream



31.8.3. RGB888 or RGB666 loosely packed data transmission and storage

Data stored in FIFO

RGB888 stream from DPI is stored in the FIFO in terms of 32-bits. A small state machine is used to convert 24-bit stream to 32-bit stream. Similar storage pattern is repeated for every 3 storage location.

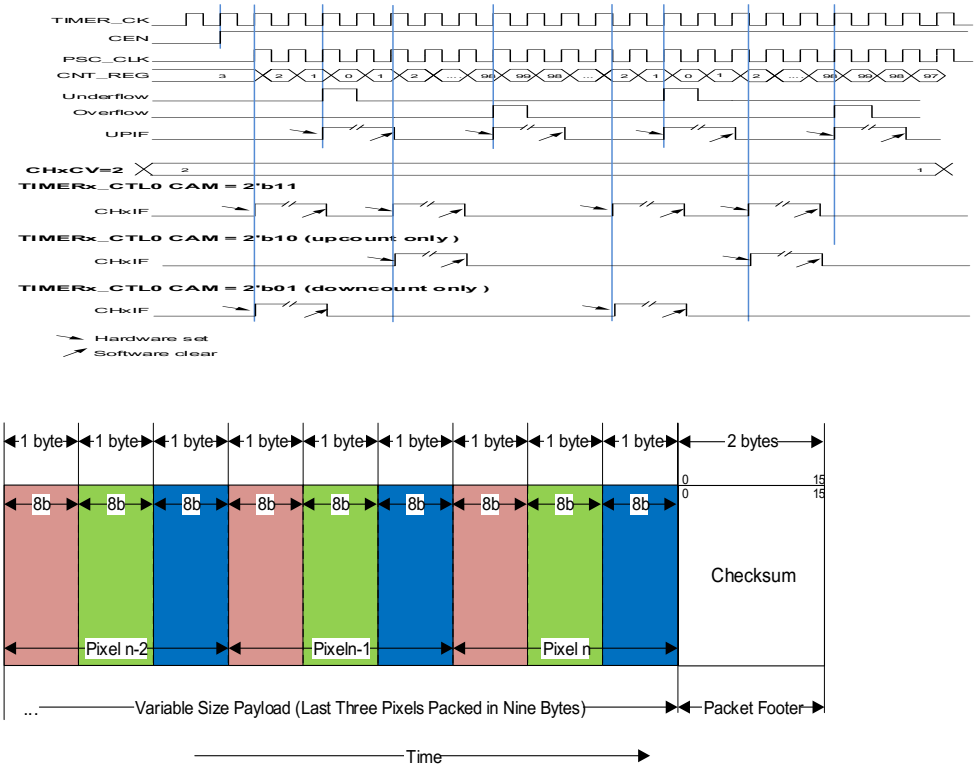
32 Bit Data FIFO Storage

Table 31-5. 32-bit data storage in RGB565

R2[7:0], B1[7:0] G1[7:0] R1[7:0]
G3[7:0] R3[7:0] B2[7:0] G2[7:0]
B4[7:0] G4[7:0] R4[7:0] B3[7:0]
R6[7:0], B5[7:0] G5[7:0] R5[7:0]
G7[7:0] R7[7:0] B6[7:0] G6[7:0]
B8[7:0] G8[7:0] R8[7:0] B7[7:0]
. . . .
. . . .
. . . .
. . . .
Gn-1[7:0] Rn-1[7:0] Bn-2[7:0]Gn-2[7:0]
Bn[7:0] Gn[7:0] Rn[7:0] Bn-1[7:0]

Transmission of RGB888 or RGB666 loosely packed stream in DSI lanes.

Figure 31-8. Transmission of RGB888/RGB666 loosely packed stream



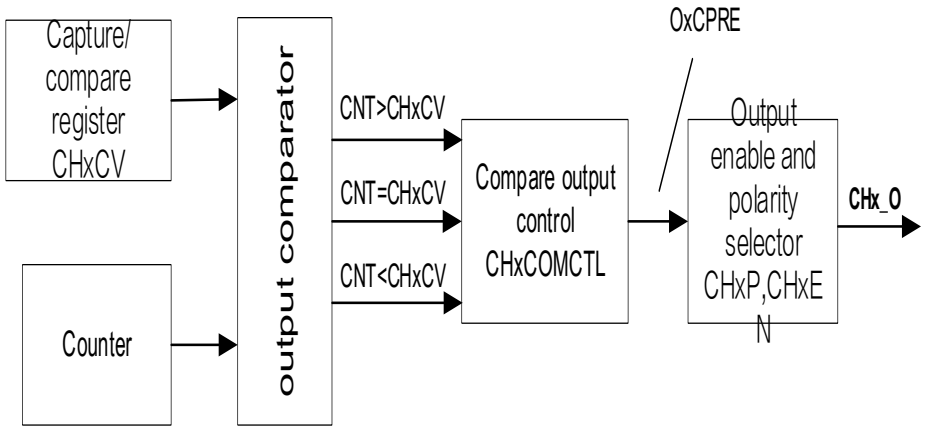
31.9. DSI TX reset and clock generation

31.9.1. Reset Generation

Table 31-6. Reset Table

Reset name	Clock name
txbyte_rst_n	txbyteclkhs
txclkesc_rst_n	txclkesc
rxclkesc_rst_n	rxclkesc
pclk_rst_n	pclk
ahb_hclk_rst_n	ahb_hclk

Figure 31-9. Reset Block Diagram



All are Asynchronous resets. Assertion happens asynchronously and de-assertion of reset happens synchronous to clock.

Resets for all other clock domains are generated based on "pwr_on_rst_n" using the asynchronous reset generation circuit.

The "ahb_hclk_rst_n" signal is generated based on "hreset_n" and "pwr_on_rst_n".

31.9.2. Clocks

Table 31-7. Clocks

Clock name	Source	Max Freq(Hz)
txclkesc	External	20MHz
txbyteclkhs	From DPHY	312.5MHz
dln_rx_RxClkEsc_0	From DPHY	10MHz
pclk(pixel clock)	External DPI interface	416.66MHz
ahb_hclk	External AHB interface	300MHz

31.10. DPHY

31.10.1. Overview of MIPI D-PHY Devices

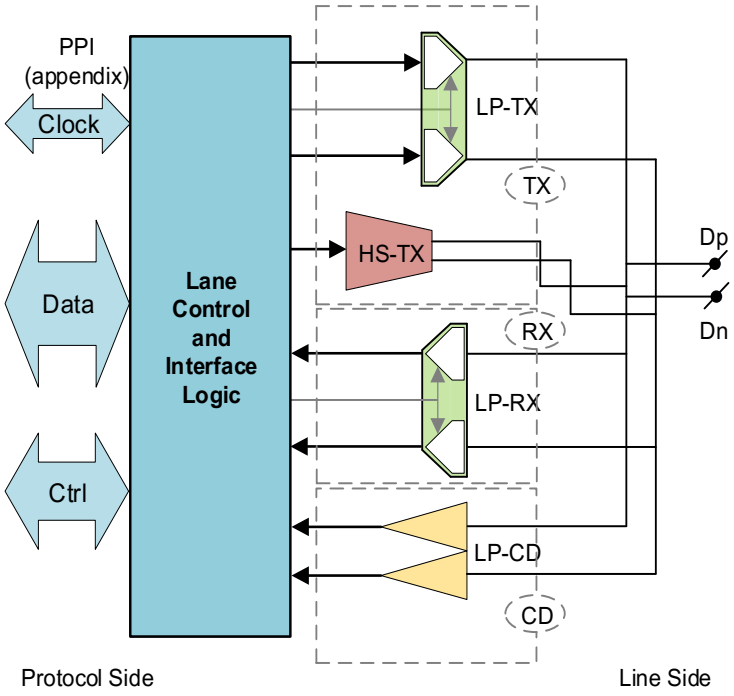
A DPHY configuration has a Clock Lane Module and one or more Data Lane Modules. Each of these DPHY Lane Modules communicates via two Lines to a complementary part on the other side of the Lane Interconnect.

Lanes

Each Lane Module has a control and interface logic unit and a transceiver portion to handle differential High-Speed functions utilizing both interconnect wires simultaneously, single-

ended Low-Power functions operating on each of the interconnect wires individually and a low power contention detector.The I/O functions are controlled by a Lane Control and Interface Logic block.

Figure 31-10. Overview of all functions



Signaling

High-Speed signals have a low voltage swing of 200 mV, while Low-Power signals have a large swing of 1.2V. High-Speed functions are used for High-Speed Data traffic. The Low-Power functions are mainly used for control and can have data transfer support.

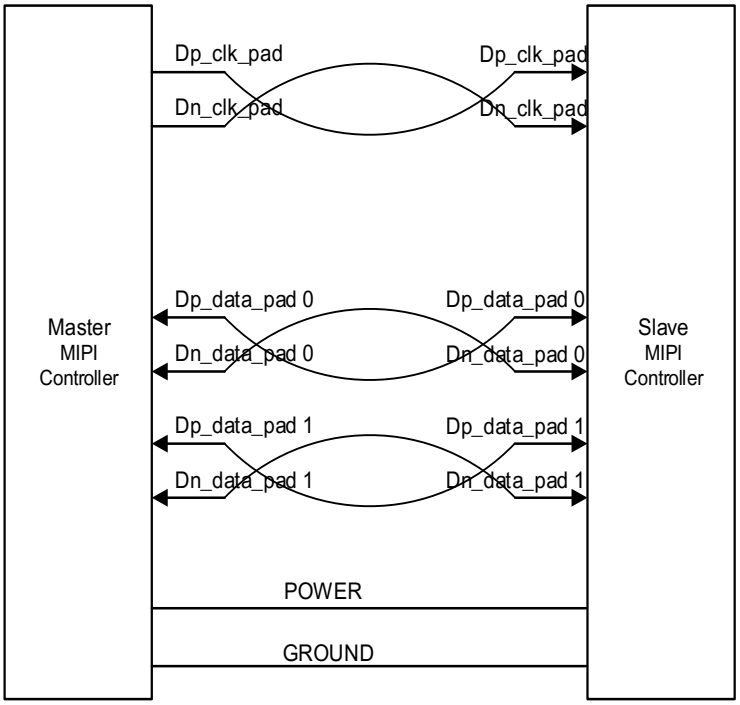
Link

Each Link has a Master and a Slave side. The Master provides the High-Speed DDR Clock signal to Clock Lane and is the main data source. The Slave receives the clock signal at the Clock Lane and is the main data sink. This main direction of communication is denoted as the Forward direction. Communication in the opposite direction is called Reverse traffic. Only bi-directional Data Lanes support both forward and reverse communications.

Lane Control and Interface Logic

It sends and detects start of packet signalling and end of packet signalling on the data lanes. It has a serializer and de-serializer unit to dialogue with the PPI / PHY adapter unit. Also it has clock divider unit to source and receive data during parallel data transfers from and to the PPI.

Figure 31-11. MIPI Link diagram for four data lanes



31.10.2. Overview of D-PHY TX

The D-PHY TX pins are compliant to MIPI's specification. The lane control and interface logic unit operates with the clock provided by internal block of the PHY during the high speed and the external reference clock is used for the low power mode of operation whereas, a separate low power clock that is generated from the data is used in slave mode for low power operations.

D-PHY TX has the state machines to generate sequences for switching to high speed, control mode and ultra-low power modes. They have serialiser unit to convert parallel to serial data.

D-PHY TX has the deskew logic. Based on the assertion of "dln_TxSkewCalHS", the DPHY TX will send out the deskew pattern with proper LP to HS transition.

31.10.3. DPHY TX features

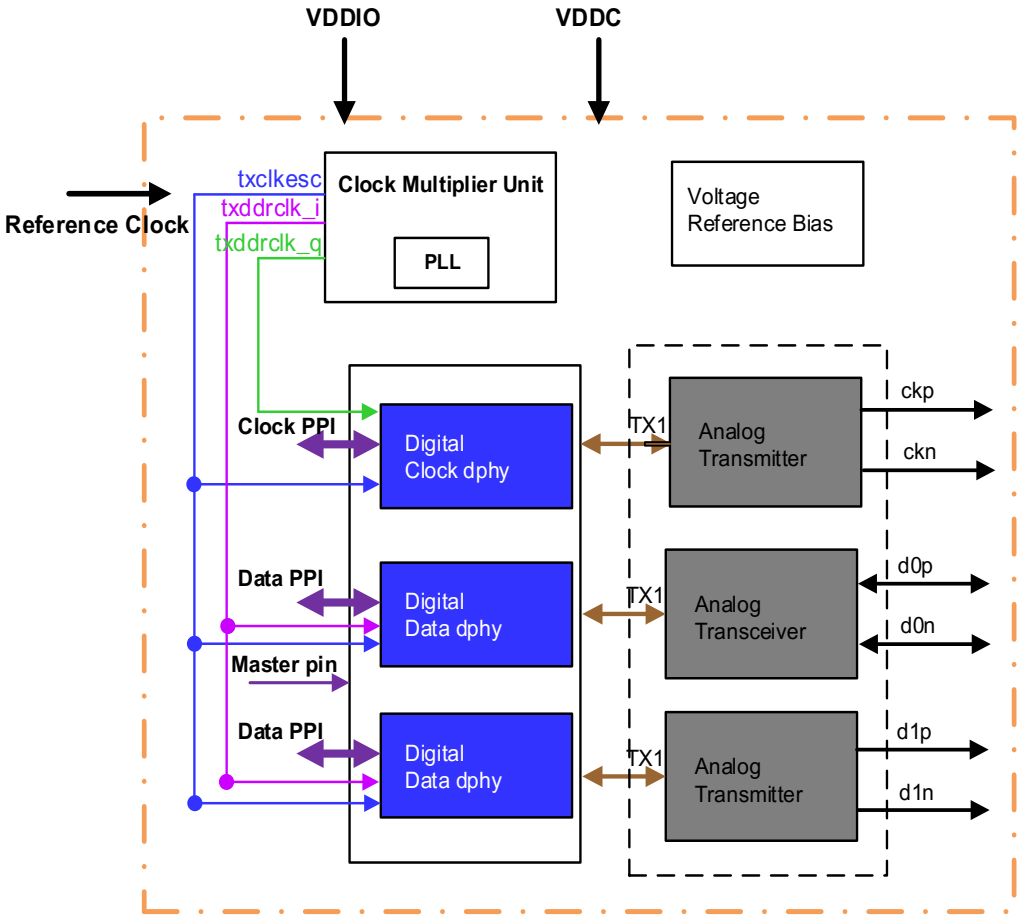
MIPI D-PHY describes a source synchronous, high speed, low power, low cost physical layer. The DPHY TX solution is compliant with MIPI specification. D-PHY Transmitter unit is responsible for transmission of data in High speed (HS) or Low Power (LP) mode. High speed mode is used for high-speed data transmission while the low power mode used for the control purpose. Point-to-point lane interconnect can be used for either data or clock signal transmission. Low-Power receiver is an un-terminated, single-ended receiver circuit. Adjoining figure shows a single lane module.

31.10.4. ACS lane swap feature

The data lane swap is possible using the trim register programming. The trim_2[20] is used to swap the data lane. DPHY clock lane cannot be swapped.

The polarity of the data and clock lanes are possible using the trim register programming. The trim_2[23:21] are used to swap the polarity of clock and data lanes..

Figure 31-12. Block diagram showing Analog and Digital portion of D-PHY TX



NOTE:

In the diagram above,

- Reference Clock - The pin TxClkEsc, input to the PHY. This clock can be provided from a crystal or digitally divided clock ranges from 10.0 MHz to 40MHz (PHY has a divide by 2 option to operate LP mode <=20Mbps as per the MIPI specification).
- VDDIO-Refers power supply to all LDOs and biasing circuits.
- VDDC-Refers power supply to all core voltage operations including digital circuits.

31.10.5. Functional Description of DPHY Pads

Table 31-8. Description of DPHY pads

Pin	Type	Direction	Description
d0p	SIGNAL	INOUT	Positive polarity low voltage differential data signal of transmitter and LP receiver for lane0
d0n	SIGNAL	INOUT	Negative polarity low voltage differential data signal of transmitter and LP receiver for lane0
d1p	SIGNAL	INOUT	Positive polarity low voltage differential data signal of transmitter for lane1
d1n	SIGNAL	INOUT	Negative polarity low voltage differential data signal of transmitter for lane1
ckp	SIGNAL	INOUT	Positive polarity low voltage differential clock signal of transmitter for clock lane
ckn	SIGNAL	INOUT	Negative polarity low voltage differential clock signal of transmitter for clock lane

31.10.6. DPHY PLL

PLL Specifications

Table 31-9. PLL Specifications

Description	Symbol	Min.	Typ.	Max	Unit
Power Supply Voltage(Core)	VDDC	0.72	0.8	0.88	V
Power Supply Voltage(IO)	VDDIO	1.62	1.8	1.98	V
Input Reference Frequency	TxClkEsc	10		40	MHz
Output Frequency		40		1250	MHz
Output Duty Cycle		48	50	52	%
Frequency lock-in time				100	us

The PLL uses a Dual-modulus divider. The equation to compute the divider ratio is N = 2*A + B, where A is the cnta value (6 bits) and B is the cntb value (1 bit). These variables are assigned by trim registers

Table 31-10. Counter settings for different division ratios shows the different counter settings and the corresponding output frequencies for an input reference clock of 20 MHz.

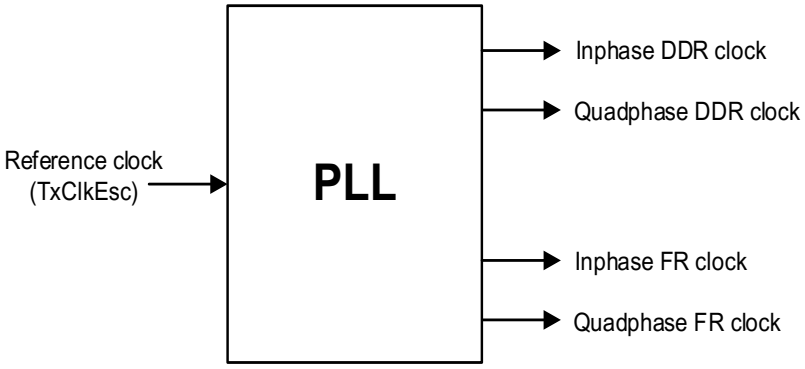
Table 31-10. Counter settings for different division ratios

Freq (MHz)	CNTA<5>	CNTA<4>	CNTA<3>	CNTA<2>	CNTA<1>	CNTA<0>	CNTB	Div Ratio
1260	0	1	1	1	1	1	1	63
1240	0	1	1	1	1	1	0	62
1220	0	1	1	1	1	0	1	61
1200	0	1	1	1	1	0	0	60
1180	0	1	1	1	0	1	1	59
1160	0	1	1	1	0	1	0	58
1140	0	1	1	1	0	0	1	57
1120	0	1	1	1	0	0	0	56
1100	0	1	1	0	1	1	1	55
1080	0	1	1	0	1	1	0	54
1060	0	1	1	0	1	0	1	53
1040	0	1	1	0	1	0	0	52
1020	0	1	1	0	0	1	1	51
1000	0	1	1	0	0	1	0	50
980	0	1	1	0	0	0	1	49
960	0	1	1	0	0	0	0	48
940	0	1	0	1	1	1	1	47
920	0	1	0	1	1	1	0	46
900	0	1	0	1	1	0	1	45
880	0	1	0	1	1	0	0	44
860	0	1	0	1	0	1	1	43
840	0	1	0	1	0	1	0	42
820	0	1	0	1	0	0	1	41
800	0	1	0	1	0	0	0	40
780	0	1	0	0	1	1	1	39
760	0	1	0	0	1	1	0	38
740	0	1	0	0	1	0	1	37
720	0	1	0	0	1	0	0	36
700	0	1	0	0	0	1	1	35
680	0	1	0	0	0	1	0	34
660	0	1	0	0	0	0	1	33
640	0	1	0	0	0	0	0	32
620	0	0	1	1	1	1	1	31
600	0	0	1	1	1	1	0	30
580	0	0	1	1	1	0	1	29
560	0	0	1	1	1	0	0	28
540	0	0	1	1	0	1	1	27
520	0	0	1	1	0	1	0	26
500	0	0	1	1	0	0	1	25

Freq (MHz)	CNTA<5>	CNTA<4>	CNTA<3>	CNTA<2>	CNTA<1>	CNTA<0>	CNTB	Div Ratio
480	0	0	1	1	0	0	0	24
460	0	0	1	0	1	1	1	23
440	0	0	1	0	1	1	0	22
420	0	0	1	0	1	0	1	21
400	0	0	1	0	1	0	0	20
380	0	0	1	0	0	1	1	19
360	0	0	1	0	0	1	0	18
340	0	0	1	0	0	0	1	17
320	0	0	1	0	0	0	0	16
300	0	0	0	1	1	1	1	15
280	0	0	0	1	1	1	0	14
260	0	0	0	1	1	0	1	13
240	0	0	0	1	1	0	0	12
220	0	0	0	1	0	1	1	11
200	0	0	0	1	0	1	0	10
180	0	0	0	1	0	0	1	9
160	0	0	0	1	0	0	0	8
120	0	0	0	0	1	1	0	6
100	0	0	0	0	1	0	1	5
80	0	0	0	0	1	0	0	4
60	0	0	0	0	0	1	1	3
40	0	0	0	0	0	1	0	2

Clock Generation Scheme

Figure 31-13. PLL Block diagram



1. Reference clock can be an external crystal or system clock minimum of 10.0MHz.

2. The Phase Noise of the crystal used for generating input reference clock should be -144dBc/Hz or better at 1 MHz. The deterministic jitter of the input reference clock should be within +/- 75ps (150ps total) or better. Input reference clock jitter +/-150ps is allowed only

when the input jitter frequency is greater than PLL loop bandwidth. PLL loop bandwidth is less than the reference clock frequency divided by 10.

3. Inphase and Quadphase DDR clock output ranging from 40MHz to 1250MHz.

4. Inphase and Quadphase FR clock output ranging from 80MHz to 2500MHz.

5. Quadphase clock is 90 degrees' phase shifted with Inphase clock.

6. Parallel clock - DDR clock/4 is used to sample DFE data.

Note: The ACS DPHY macro does not include any clock generation like crystal oscillators or their associated pads. Clock generation blocks external to the macro need to generate reference clocks appropriately during the HS and LP modes.

Clock Tolerance Requirement

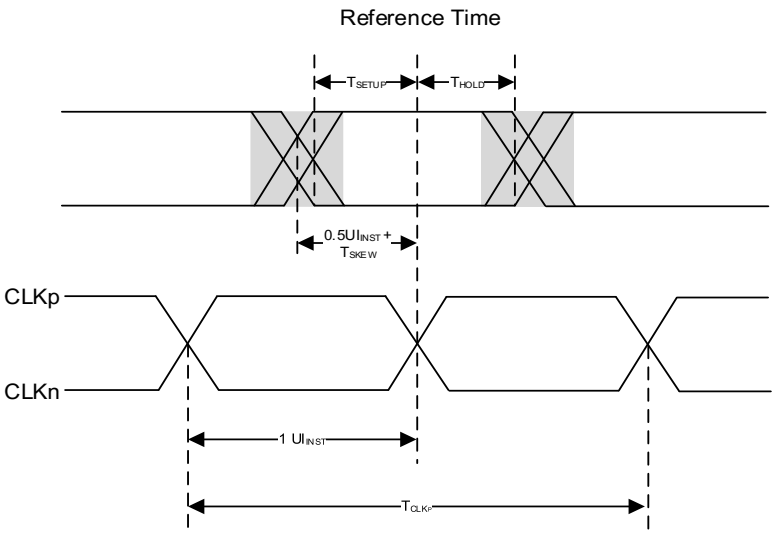
Tolerance factor for high speed clocks

1. Output clock on and off time can vary +-10% for <=1GHz and +/-5% for >1GHz.

2. Reference clock duty cycle can vary from 45% to 55%.

Clock Skews

Figure 31-14. DPHY Data to Clock Timing Definitions



Note: UI is any DDR half cycle.

T_{SKEW} is the allowed deviation of the data launch time to the ideal ½UI_{INST} displaced quadrature clock edge. For data rates above 1.5 Gbps, data to clock skew budget is partitioned in two categories, static and dynamic. The static skew is a skew, which is calibrated out during calibration. The second category is dynamic skew and it is the variation of maximum skew between data to clock.

Table 31-11. DPHY Data to Clock Timing Specifications

Parameter	Description	Min	Nom	Max	Units
$T_{SKEW[TX]}$	Data to clock skew for 80Mbps to 1Gbps	-0.15		0.15	UI
	Data to clock skew for above 1Gbps to 1.5Gbps	-0.20		0.20	UI
$T_{SKEW[TX]static}$	Static data to clock skew for above 1.5Gbps			0.20	UI
$T_{SKEW[TX]dynamic}$	Dynamic data to clock skew for above 1.5Gbps			0.15	UI

31.11. Programing procedure

31.11.1. DSI Host Initializations by the processor

Step 1: Interrupts are enabled by writing in the DSI_INTEN0/DSI_INTEN1.

Step 2: Number of data lanes supported, color formats supported in video mode, command mode, channel number for video and command mode and the data width supported in command mode are programmed in the DSI_FUNCPRG.

Step 3: DSI_HSTXTO, LPRXTO, DSI_TATO, DSI_DEVRSTTM register are programmed for recovery actions

Step 4: Resolution register for DPI and the supported parameters for a specific panel are programmed into the horizontal and vertical sync, porch count registers.

Step 5: The freq mode register (DSI_FMCTL) needs to be programmed like below

bit[0] -> 0 - DDR freq < 200MHz, 1 - DDR freq >= 200MHz

Step 6: DPHY related registers are needs to be programmed for both data lane and clock lane

Step 7: DSI_DPHY_RSTENDFE register needs to be programmed after programming all DPHY registers

Step 8: DSI_INITCFG and DSI_DEVRDY register are programmed once the initial programming operations are completed by the processor.

31.11.2. DPI

DSI Host Operations for DPI mode

Once programming is completed, the DSI host will drive a sustained Tx-stop state [LP-11] on all the lanes of the link for duration as specified in the DSI Host initialization count configuration register (DSI_INITCFG) register.

a) After the DSI host is initialized and if Vsync or Hsync signal is recorded by the DPI interface, the DPI state machine automatically generates Vsync events, Hsync events based on the programmed values in the AHB target register and starts gathering the data on the DPI FIFO until the FIFO gets full.

b) It starts transmitting as DSI packets as soon as the events are generated internally based on the already programmed timing parameters

c) It switches to low power mode for the rest of the timings during line blanking

d) Then continues to transmit hsync events in series as per the parameter inputs from the registers and switches to low power mode until the blanking period is completed.

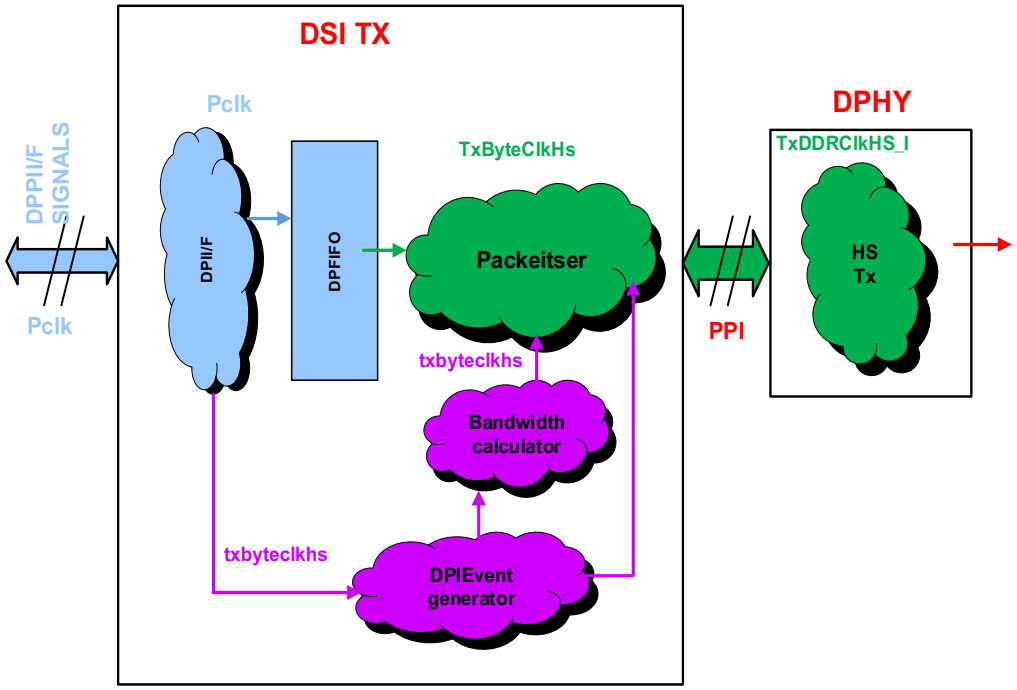
e) RGB packets recorded in the DPI's data FIFO are transmitted after the blanking lines durations is completed. In burst mode, DSI host switches to low power mode after RGB packet transmission till a Hsync event is to be transmitted. This continues until a frame is transmitted

f) Short packets are sent to mark the event happening like Vsync start, Vsync end, Hsync start, Hsync end, shut down, color mode on, color mode off and turn on

g) Shut down, color mode on, color mode off, turn on, events are programmed from AHB target interface. This information is sent as short packets at the immediately following VBP.

DPI transfer flow

Figure 31-15. DPI transfer flow



31.11.3. DCS/Generic through AHB interface

Step 1: Program all the functional PHY and DSI configuration registers

Step 2: Program the reset enable to DFE register as '1'

Step 3: Program the DSI_DEVRDY register bit[0] with value '1'

Step 4: Wait for INIT_DONE interrupt in interrupt status register (DSI_INTSTAT0)

For High-speed Generic/DCS short write

To transmit generic /Manufacturer DCS short write packets in high-speed mode,

Generic write with zero parameter (data_type 03h)

Generic write with one parameter (data_type 13h)

Generic write with two parameter (data_type 23h)

Manufacturer DCS with zero parameter (data_type 05h)

Manufacturer DCS with one parameter (data_type 15h)

Step 5: Processor checks for high-speed generic control FIFO full is not set in the DSI_FIFOSTAT register.

Step 6: Processor then writes into DSI_HSCTL register.

Step 7: If interrupt is received, processor checks for HSCTLFIFOFIF bit in DSI_INTSTAT1.If it is set, then processor checks for high speed generic control FIFO full HSCTLFIFOF is being set in the DSI_FIFOSTAT register, processor can write up to 4 dwords of generic control data into DSI_HSCTL.

Step 8. Otherwise repeat step 6.

For Low power Generic/DCS short write

To transmit generic /Manufacturer DCS short write packets in high-speed mode,

Generic write with zero parameter (data_type 03h)

Generic write with one parameter (data_type 13h)

Generic write with two parameter (data_type 23h)

Manufacturer DCS with zero parameter (data_type 05h)

Manufacturer DCS with one parameter (data_type 15h)

Step 5: Processor checks for low power generic control FIFO full is not set in the DSI_FIFOSTAT register.

Step 6: Processor then writes into DSI_LPCTL.

Step 7: If interrupt is received, processor checks for LPCTLFIFOFIF bit in DSI_INTSTAT1.If it is set, then processor checks for low power generic control FIFO full LPCTLFIFOF is being set in the DSI_FIFOSTAT register, processor can write up to 4 dwords of generic control data into DSI_LPCTL.

Step 8. Otherwise repeat step 6.

For High-speed Generic/DCS Long write

Step 5: Processor checks for high-speed generic data FIFO full is not set in the FIFO status register (DSI_FIFOSTAT).

a) If FIFO is not full, processor writes into the high-speed generic data port register "HS data register (DSI_HSDATA)" until the interrupt received.

b) Processor checks for HSDTFIFOFIF bit in interrupt_status_register1 (DSI_INTSTAT1). If it is set, then processor checks for high speed generic data FIFO full [HS_data_fifo_full] is being set in the FIFO status register (DSI_FIFOSTAT), processor can do 4 writes of payload data into HS gen data register (DSI_HSDATA)". Otherwise, processor writes into the high-speed generic data port register HS gen data register (DSI_HSDATA)" until the interrupt received

c) Wait for FIFO not full flag to be set in the "FIFO status register (DSI_FIFOSTAT)", to continue with the remaining payload.

Steps a, b and c are repeated to transmit the remaining generic data bytes.

Step 6: Processor then checks if the high-speed control FIFO is not full by reading the HS_control_fifo_full of FIFO status register (DSI_FIFOSTAT) and writes the Generic/DCS long write packet header into HS control register (DSI_HSCTL).

10.1.5 Steps 5 and 6 are repeated for the next

For Low power Generic/DCS Long write

Step 5: Processor checks for low power generic data FIFO full is not set in the FIFO status register (DSI_FIFOSTAT).

a) If FIFO is not full, processor writes into the high-speed generic data port register "LP data register (DSI_LPDATA)" until the interrupt received.

b) Processor checks for LPDTFIFOFIF bit in interrupt_status_register1(DSI_INTSTAT1).If it is set, then processor checks for high speed generic data FIFO full LPDTFIFOF is being set in the FIFO status register (DSI_FIFOSTAT), processor can do 4 writes of payload data into LP data register (DSI_LPDATA)". Otherwise, processor writes into the high-speed generic data port register LP data register (DSI_LPDATA)" until the interrupt received

c) Wait for FIFO not full flag to be set in the "FIFO status register (DSI_FIFOSTAT)", to continue with the remaining payload.

Steps a, b and c are repeated to transmit the remaining generic data bytes.

Step 6: Processor then checks if the low power control FIFO is not full by reading the LP_control_fifo_full of FIFO status register (DSI_FIFOSTAT) and writes the Generic/DCS long write packet header into LP control register (DSI_LPCTL).

10.1.6 Steps 5 and 6 are repeated for the next

For High speed Generic/DCS read

To transmit High Speed Generic/DCS read request packets,

Generic read with zero parameter (data_type - 04h),

Generic read with one parameter (data_type - 14h),

Generic read with two parameter (data_type - 24h),

DCS read with no parameter (data_type - 06h)

Step 5: Processor checks for high-speed generic control FIFO is not full (bit 16) by reading the fifo_stat_reg(DSI_FIFOSTAT) register.

Step 6: Processor then writes generic read request short packet into HS_ctrl_reg (DSI_HSCTL)

Step7: Processor waits for "generic_read_data_avail" (bit29) interrupt and clears the same by writing into in "interrupt status register(DSI_INITSTAT0)".

Step 8: Processor reads data from HS_data_reg(DSI_HSDATA).

For Low power Generic/DCS read

To transmit High Speed Generic/DCS read request packets,

Generic read with zero parameter (data_type - 04h),

Generic read with one parameter (data_type - 14h),

Generic read with two parameter (data_type - 24h),

DCS read with no parameter (data_type - 06h)

Step 5: Processor checks for Low power generic control FIFO is not full (bit 24) by reading the fifo_stat_reg(DSI_FIFOSTAT) register.

Step 6: Processor then writes generic read request short packet into LP_ctrl_reg (DSI_LPCTL)

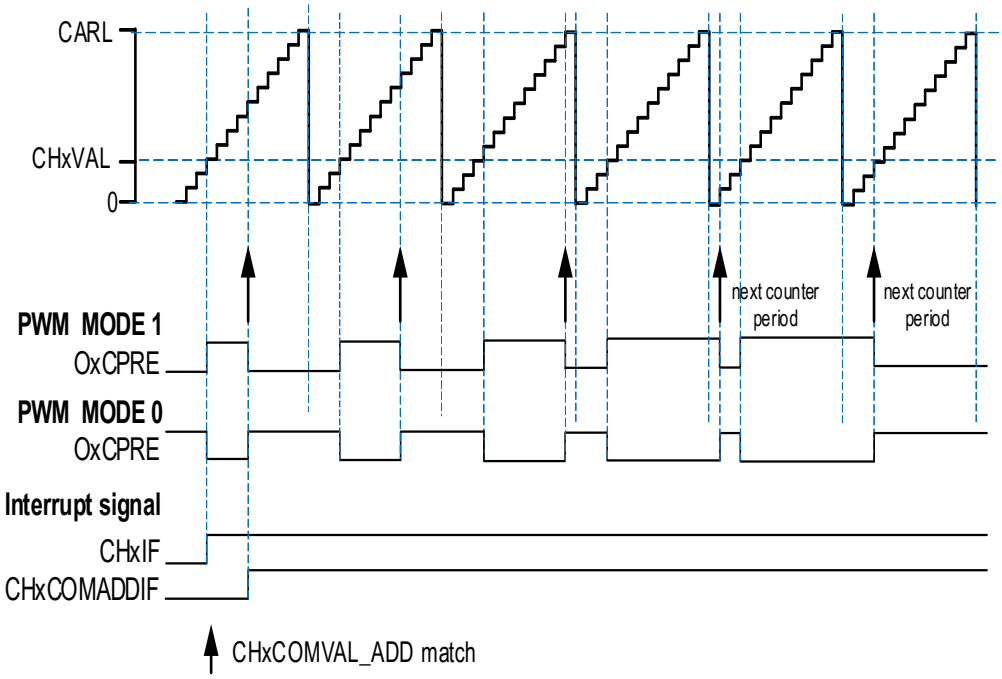
Step7: Processor waits for "generic_read_data_avail" (bit29) interrupt and clears the same by writing into in "interrupt status register(DSI_INITSTAT0)".

Step 8: Processor reads data from LP_data_reg(DSI_LPDATA).

DCS/Generic transfer flow

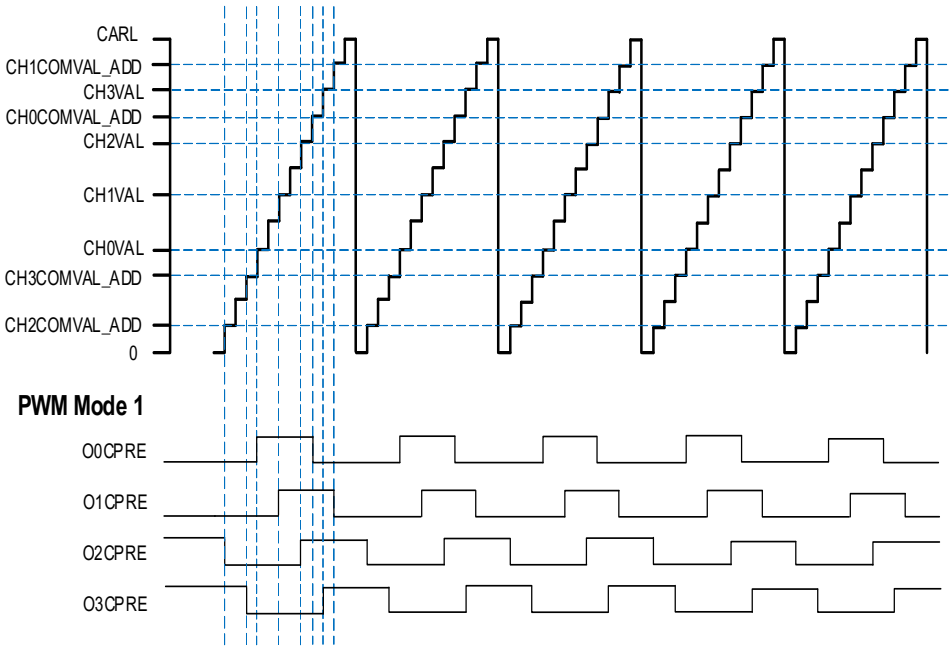
DCS/Generic HS write

Figure 31-16. DCS/Generic transfer flow



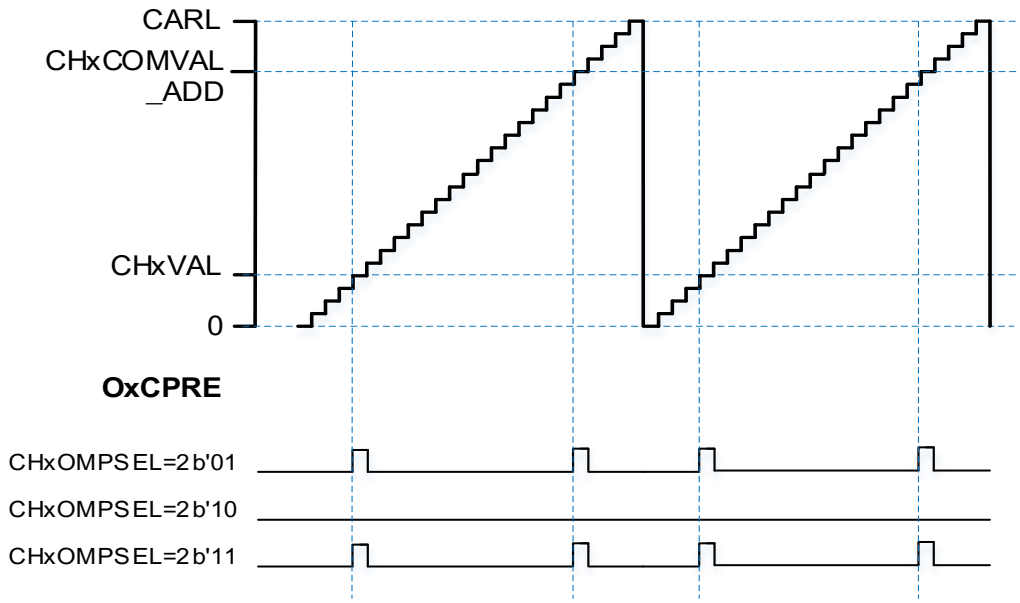
DCS/Generic LP Write

Figure 31-17. DCS/Generic LP Write



DCS/Generic read

Figure 31-18. DCS/Generic read



31.11.4. DSI Host Operations in dual channel mode

Once programming is completed, the DSI host will drive a sustained Tx-stop state [LP-11] on all the lanes of the link for duration as specified in the 1255pprox.1255er1255ion timer register

a) After the DSI host is initialized and if vsync event is recorded, the DPI state machine automatically generates Vsync events, Hsync events based on the programmed values in the AHB register and starts gathering the data on the DPI FIFO until the FIFO gets full

b) The channelizer stream lines the DSI traffic for DCS/Generic if any event is present in the control FIFO and switches to low power mode if there is no recorded DCS/Generic event during BLLP period

c) Then continues to transmit hysnc events in series as per the parameter input from the registers and switches to low power mode or transmits DCS/Generic events until the blanking period is completed

d) RGB packets recorded in the DPI's data FIFO are transmitted after the blanking lines durations is completed

e) RGB packets are sent in a burst mode and again the DSI host switches to low power mode or transmits DCS/Generic events till a Hsync event is to be transmitted, this continues until a frame is transmitted

f) Short packets are sent to mark the event happening like Vsync start, Vsync end, Hsync start, Hsync end, shut down, color mode on, color mode off and turn on

g) Shut down, color mode on, color mode off, turn on events are programmed from AHB

processor, this information is sent as short packets at the immediately following VBP

31.11.5. Blanking Time management

Hsync count, horizontal back porch area timing, horizontal front porch area timings are packed with [blanking packets] in a line. During blanking lines, the entire active video line area is meant for BLLP.

Based on the above programmed values in the register, the DSI-tx controller decides the BLLP packaging in the DSI link.

1) If the count value for a particular area is greater than twice the High speed to low switch count value plus two blanking packet with no payloads , then the DSI-tx controller plans for a low power mode switching and comes back to High speed towards the end of the line. [*]

2) If the count value for a particular area is less than twice the high speed to low switch count value plus two blanking packet with no payloads, then the DSI-tx controller plans for a high speed blanking packet for the blanking time.[**]

3) During blanking lines BTA bus turnaround is initiated if a DCS/Generic transaction is carried with read data phase [***]

4) Otherwise, during blanking lines too only low power mode entry is done by the DSI-tx controller. [*]

5) For the last blanking line of every frame, the DSI-tx controller forces a bus turn around [***]

6) If the DSI-receiver does not respond to this BTA, the DSI-tx recovers itself and places it drivers in low power mode and a BTA timeout interrupt is sent to the processor.

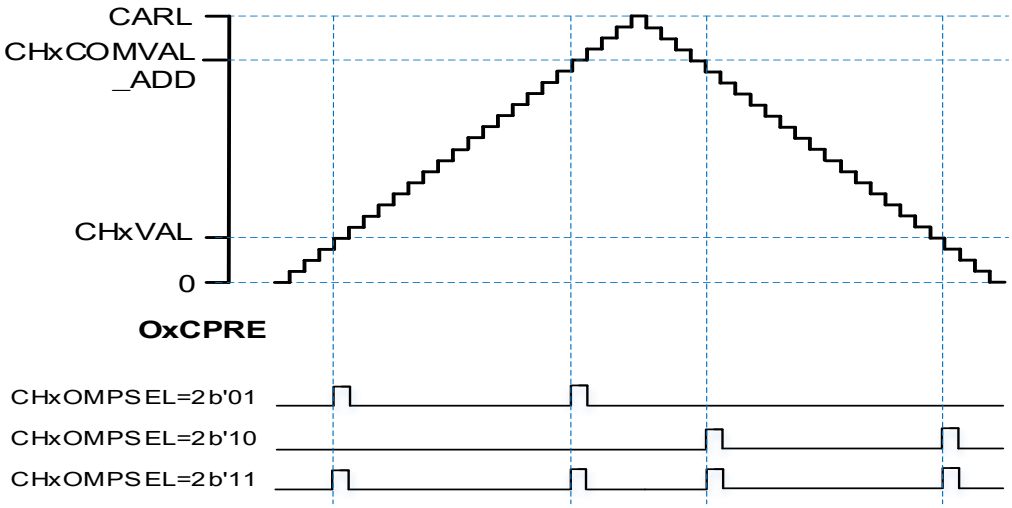
Note: * Remain in Idle Mode with the host processor in LP-11 state and the peripheral in LP-RX

**Transmit one or more non-video packets from the host processor to the peripheral using HS Mode.

***If the previous processor-to-peripheral transmission ended with BTA, transmit one or more packets from the peripheral to the host processor using Escape Mode

31.12. TLI DMA

Figure 31-19. Diagram of TLI DMA and DSI controller



The DSI controller can support the use of TLI DMA to acquire pixel data for DSI in command mode. This mode is referred to as WMS DMA mode, which supports memory_write_start (WMS) and memory_write_continue (WMC) DCS commands. The WMS_DMA mode supports pixel data rate control and the tearing effect report mechanism.

The DSI controller can be configured to enable the WMS_DMA (TLI_DMA) feature. Once enabled, it will occupy the DMA function of TLI, meaning TLI will receive DMA data and will not send DPI data to the DSI controller. The configuration information such as address and data amount for DMA transfer can be set via TLI control registers of DMA.

In order to transmit image data in adapted command mode: Configure the WMS_DMA mode and set TLIDMA of the DSI Wrapper control register to allow DSI to use the TLI DMA function.

The DSI wrapper synchronizes the transfer process through the following methods:

1. Controlling the start/stop of WMS_DMA.
2. Managing data flow control between WMS_DMA and the DSI host.

The transfer for refreshing the display frame buffer can be triggered in the following ways:

1. Manual trigger by setting the WMS_DMA enable (WMS_EN) bit in the DSI wrapper WMS write control register (DSI_WWMS_WCTL).
2. Automatic trigger when a Tear Effect Interrupt (TEIF) event occurs and Auto Refresh (AR) is enabled.

The choice between manual and automatic modes is done by the Auto Refresh (AR) bit

in the DSI wrapper control register (DSI_WCTL). In automatic refresh mode, the WMS_DMA enable (WMS_EN) bit in the DSI wrapper WMS write control register (DSI_WWMS_WCTL) is automatically set by a TEIF event.

When the transmission of a frame is completed, whether in manual or automatic refresh mode, the DSI wrapper terminates TFT DMA by resetting the WMS_DMA enable (WMS_EN) bit in the DSI wrapper WMS write control register (DSI_WWMS_WCTL) and setting the WMS_DMA end of interrupt flag (WMS_END). An interrupt is generated if the WMS_DMA end mask interrupt enable (WMS_ENDMSK) bit in the DSI wrapper WMS interrupt mask register (DSI_WWMS_INTMSK) is set.

31.13. Tearing effect

The DSI specification supports the tearing effect feature in command mode display.

When the TESRC bit in the DSI Wrapper Control Register (DSI_WCTL) is set, the tearing effect is controlled via a GPIO signal. The polarity of the input signal can be configured using the TEPOL bit in the DSI Wrapper Control Register (DSI_WCTL).

When the TESRC bit in the DSI Wrapper Control Register (DSI_WCTL) is reset, the tearing effect is managed through the DSI link.

31.14. WMS hardware unpacking

31.14.1. WMS automatic unpacking: Using HS command mode to send WMS as an example

Step 1: Program all the functional PHY and DSI configuration registers

Step 2: Program the reset enable to DFE register as '1'

Step 3: Program the device ready register bit[0] with value '1'

Step 4: Wait for "init_done" interrupt in interrupt status register

Step 5: Configure the DSI_WCMDDS register, set HSCMDDS=1, the DSI_WCTL register, set TLIDMA=1; and configure the other bits in DSI_WCTL as needed.

Configure DSI_WDMARCTL, setting DMASRC to the appropriate request source, with the default being half-empty.

Read DSI_WWMS_WCTL to check for any pending WMS_END interrupts and clear them if present.

Configure DSI_WWMS_INTMSK to enable interrupts.

Configure DSI_WWMS_HEAD and DSI_WWMS_SIZE.

Step 6: Configure TLI settings to ensure that the DMA data size transmitted in one frame matches the packet length configured in DSI.

Step 7: Set the WMS_EN bit in DSI_WWMS_WCTL.

Step 8: Wait for interrupts, read DSI_WWMS_WCTL, and check the WMS_END flag. When the WMS_END flag is asserted, the WMS_EN bit will be automatically cleared by hardware.

31.14.2. WMS automatic unpacking: Using LP command mode to send WMS as an example

Step 1: Program all the functional PHY and DSI configuration registers

Step 2: Program the reset enable to DFE register as '1'

Step 3: Program the device ready register bit[0] with value '1'

Step 4: Wait for "init_done" interrupt in interrupt status register

Step 5: Configure the DSI_WCMDDS register, set HSCMDDS=1 and LPCMDDS=1, configure the DSI_WCTL register, set TLIDMA=1, and configure the remaining bits in DSI_WCTL as needed.

Configure DSI_WDMARCTL, setting DMASRC to the appropriate request source, with the default being half-empty.

Read DSI_WWMS_WCTL to check for any pending WMS_END interrupts and clear them if present.

Configure DSI_WWMS_INTMSK to enable interrupts.

Configure DSI_WWMS_HEAD and DSI_WWMS_SIZE.

Step 6: Configure the TLI settings to ensure that the DMA data size transmitted in one frame matches the packet length configured in DSI.

Step 7: Set the WMS_EN bit in DSI_WWMS_WCTL.

Step 8: Wait for interrupts, read DSI_WWMS_WCTL, and check the WMS_END flag. When the wms_end_flag is asserted, WMS_EN will be automatically cleared by hardware.

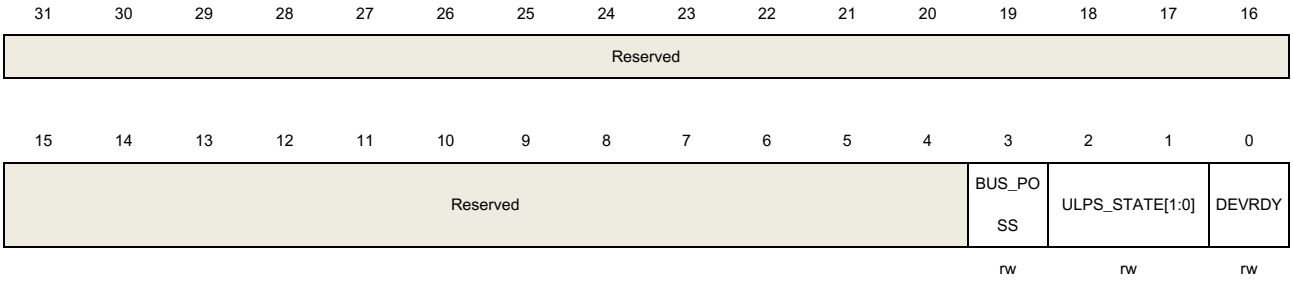
31.15. Register definition

DSI_HOST base address: 0x5200 3000

31.15.1. DSI Host device ready register (DSI_DEVRDY)

Address offset: 0x00
Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value.
3	BUS_POSS	Display bus possession 0: Stop the bus possession of the DSI receiver 1: Inform that DSI receiver has to be given bus possession for receiving the tearing effect trigger message
2:1	ULPS_STATE[1:0]	Inform entire DSI host to enter ultra low power state(ULPS). Software needs to ensure that there is a minimum of 1ms time available before clearing the ULPS exit State. 00: Normal operation. 01: Ultra low power EXIT mode. 10: Ultra low power[POWER SAVING] mode. 11: Reserved
0	DEVRDY	Device ready. Set by the processor to inform that device is ready for transmission. This register should be set to "1" after dphy_parameter register, all the count registers, and timeout and interrupt enable registers are being programmed. **Note:** Reprogramming the registers by resetting the device_ready bit results in re-enumeration of the DSI controller from the power up sequence. DEVRDY should remain set while the device is subjected to ULPS states. 0: Not ready 1: Ready

31.15.2. DSI Host interrupt status register0 (DSI_INTSTAT0)

Address offset: 0x04

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DPIPE	DPILTO	RXCONTDET	INIT_DONE	SPKTCMDS	RXDSIPV	RXINVTXCE	ACKNE	TAACKTO	LPRXTO	HSTXTO	FIFOE	LCONT	HCONT	TXDSIVCIDINV	TXDSIDTNR
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
TXCHKE	TXECCME	TXECCSE	TXFCTLE	RXDSIVCIDINV	RXDSIDTNR	RXCHKE	RXECCME	RXECCSE	RXFCTLE	RXPTOE	RXLPTXSYNCE	RXESCMEE	RXEOTSYNCE	RXSOTSYNCE	RXSOTE
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	DPIPE	DPI program error 0: No error 1: Error
30	DPILTO	DPI line timeout 0: Not timeout 1: Timeout
29	RXCONTDET	Rx Contention Detected. 0: No error 1: Error
28	INIT_DONE	DSI 1261pprox.1261er1261ion is done. 0: Init is not done 1: Init is done
27	SPKTCMDS	Special packet command sent. Set to confirm the transmission of the DPI event specific commands set in the DSI_DPICTL register. 0: Not Sent 1: Sent
26	RXDSIPV	Rx DSI protocol violation. Set if acknowledge short packet shows DSI protocol violation error 0: No error 1: Error
25	RXINVTXCE	Rx Invalid transmission count error. Set if acknowledge short packet shows an invalid transmission count error. 0: No error

		1: Error
24	ACKNE	ACK with No error 0: one or more error 1: No error
23	TAACKTO	Turn around ack timeout 0: Not timeout 1: Timeout
22	LPRXTO	LP Rx timeout 0: Not timeout 1: Timeout
21	HSTXTO	HS Tx timeout 0: Not timeout 1: Timeout
20	FIFOE	FIFO empty. 0: Not all FIFO is empty 1: All FIFO is empty
19	LCONT	Low contention Set to '1' if a LP low fault is registered by at the D-PHY contention detector. If this interrupt is set device should be re-enumerated. 0: No error 1: Error
18	HCONT	High contention. Set to '1' if a LP high fault is registered by at the D-PHY contention detector. If this interrupt is set device should be re-enumerated. 0: No error 1: Error
17	TXDSIVCIDINV	Tx DSI virtual channel ID invalid 0: No error 1: Error
16	TXDSIDTNR	Tx DSI data type not recognized 0: No error 1: Error
15	TXCHKE	Tx checksum(CRC) error 0: No error 1: Error
14	TXECCME	Tx ECC multibit error 0: No error 1: Error
13	TXECCSE	Tx ECC single bit error 0: No error

		1: Error
12	TXFCTLE	Tx False Control Error 0: No error 1: Error
11	RXDSIVCIDINV	Rx DSI virtual channel ID invalid 0: No error 1: Error
10	RXDSIDTNR	Rx DSI data type not recognized 0: No error 1: Error
9	RXCHKE	Rx checksum error 0: No error 1: Error
8	RXECCME	Rx ECC multibit error 0: No error 1: Error
7	RXECCSE	Rx ECC single bit error 0: No error 1: Error
6	RXFCTLE	RxFalse Control Error. 0: No error 1: Error
5	RXPTOE	Rx Peripheral timeout Error. 0: No error 1: Error
4	RXLPTXSYNCE	Rx LP tx sync error. 0: No error 1: Error
3	RXESCMEE	Rx Escape Mode Entry Error. 0: No error 1: Error
2	RXEOTSYNCE	Rx Eot sync Error. 0: No error 1: Error
1	RXSOTSYNCE	Rx Sot Sync Error. 0: No error 1: Error

0	RXSOTE	Rx Sot Error. 0: No error 1: Error

31.15.3. DSI Host interrupt enable register 0 (DSI_INTEN0)

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
DPIPEIE	DPILTOIE	RXCONTDETIE	INIT_DONEIE	SPKTCMDSIE	RXDSIPVIE	RXINVTXCEIE	ACKNEIE	TAACKTOIE	LPRXTOIE	HSTXTOIE	FIFOEIE	LCONTIE	HCONTIE	TXDSIVCIDINVIE	TXDSIDTNRIE
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
TXCHKEIE	TXECCMEIE	TXECCSEIE	TXFCTLEIE	RXDSIVCIDINVIE	RXDSIDTNRIE	RXCHKEIE	RXECCMEIE	RXECCSEIE	RXFCTLEIE	RXPTOEIE	RXLPTXSYNCEIE	RXESCMEEIE	RXEOTSYNCEIE	RXSOTSYNCEIE	RXSOTEIE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31	DPIPEIE	DPI program error interrupt enable. 0: Disable 1: Enable
30	DPILTOIE	Enable DPI line timeout interrupt enable. 0: Disable 1: Enable
29	RXCONTDETIE	Rx Contention Detected interrupt enable. 0: Disable 1: Enable
28	INIT_DONEIE	DSI initialization is done interrupt enable. 0: Disable 1: Enable
27	SPKTCMDSIE	Special packet command sent interrupt enable. 0: Disable 1: Enable
26	RXDSIPVIE	Rx DSI protocol violation interrupt enable. 0: Disable 1: Enable
25	RXINVTXCEIE	Rx Invalid transmission count error interrupt enable. 0: Disable

		1: Enable
24	ACKNEIE	ACK with No error interrupt enable. 0: Disable 1: Enable
23	TAACKTOIE	Turn around ack timeout interrupt enable. 0: Disable 1: Enable
22	LPRXTOIE	LP Rx timeout interrupt enable. 0: Disable 1: Enable
21	HSTXTOIE	HS Tx timeout interrupt enable. 0: Disable 1: Enable
20	FIFOEIE	FIFO empty interrupt enable. 0: Disable 1: Enable
19	LCONTIE	Low contention interrupt enable. 0: Disable 1: Enable
18	HCONTIE	High contention interrupt enable. 0: Disable 1: Enable
17	TXDSIVCIDINVIE	Tx DSI virtual channel ID invalid interrupt enable. 0: Disable 1: Enable
16	TXDSIDTNRIE	Tx DSI data type not recognized interrupt enable. 0: Disable 1: Enable
15	TXCHKEIE	Tx checksum(CRC) error interrupt enable. 0: Disable 1: Enable
14	TXECCMEIE	Tx ECC multibit error interrupt enable. 0: Disable 1: Enable
13	TXECCSEIE	Tx ECC single bit error interrupt enable. 0: Disable 1: Enable

12	TXFCTLEIE	Tx False Control Error interrupt enable. 0: Disable 1: Enable
11	RXDSIVCIDINVIE	Rx DSI virtual channel ID invalid interrupt enable. 0: Disable 1: Enable
10	RXDSIDTNRIE	Rx DSI data type not recognized interrupt enable. 0: Disable 1: Enable
9	RXCHKEIE	Rx checksum error interrupt enable. 0: Disable 1: Enable
8	RXECCMEIE	Rx ECC multibit error interrupt enable. 0: Disable 1: Enable
7	RXECCSEIE	Rx ECC single bit error interrupt enable. 0: Disable 1: Enable
6	RXFCTLEIE	RxFalse Control Error interrupt enable. 0: Disable 1: Enable
5	RXPTOEIE	Rx Peripheral timeout Error interrupt enable. 0: Disable 1: Enable
4	RXLPTXSYNCEIE	Rx LP tx sync error interrupt enable. 0: Disable 1: Enable
3	RXESCMEEIE	Rx Escape Mode Entry Error interrupt enable. 0: Disable 1: Enable
2	RXEOTSYNCEIE	Rx Eot sync Error interrupt enable. 0: Disable 1: Enable
1	RXSOTSYNCEIE	Rx Sot Sync Error interrupt enable. 0: Disable 1: Enable
0	RXSOTEIE	Rx Sot Error interrupt enable. 0: Disable

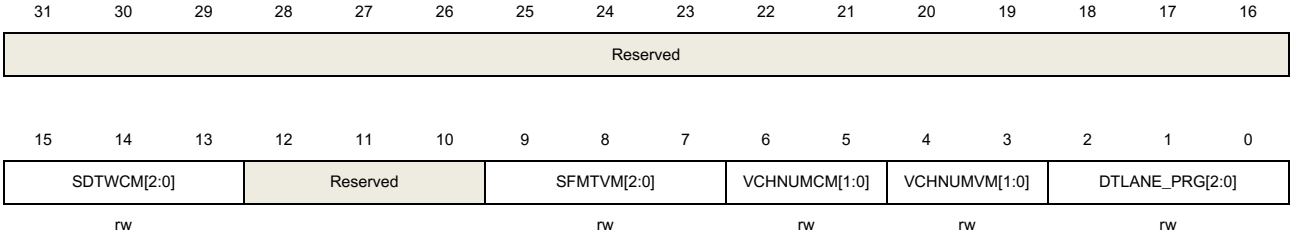
1: Enable

31.15.4. DSI Host function program register (DSI_FUNCPRG)

Address offset: 0x0C

Reset value: 0x0000 0001

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



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:13	SDTWCM[2:0]	Supported data width in command mode. 000: Command mode is not supported 001/010/011: Either of these value need to be set for DCS/Generic transfers 100 to 111: Reserved
12:10	Reserved	Must be kept at reset value.
9:7	SFMTVM[2:0]	Supported color format for video mode. 000: Video mode is not supported. 001: RGB565 or 16-bit format 010: RGB666 or 18-bit format 011: RGB666 loosely packed format 100: RGB888 or 24-bit format 101 to 111: Reserved
6:5	VCHNUMCM[1:0]	Virtual channel number for command mode. 00: Virtual channel 0 01: Virtual channel 1 10: Virtual channel 2 11: Virtual channel 3
4:3	VCHNUMVM[1:0]	Virtual channel number for video mode. 00: Virtual channel 0 01: Virtual channel 1 10: Virtual channel 2 11: Virtual channel 3
2:0	DTLANE_PRG[2:0]	Number of data lanes. 000: Reserved

001: One data lane

010: Two data lane

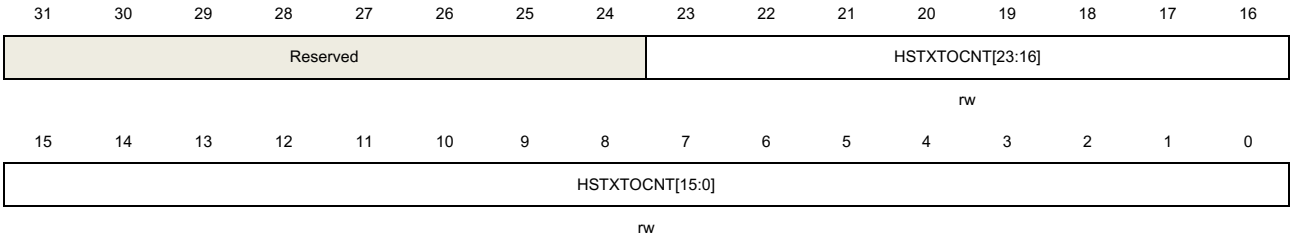
011 to 111: Reserved

31.15.5. DSI Host high speed transmission timeout register (DSI_HSTXTO)

Address offset: 0x10

Reset value: 0x0000 FFFF

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



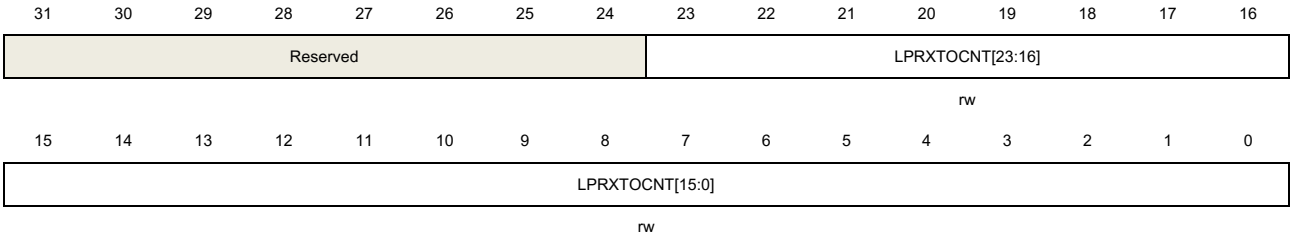
Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:0	HSTXTOCNT[23:0]	High speed transmission timeout counter. The timeout value is protocol specific. Time out value is calculated from txbyteclkhs.

31.15.6. DSI Host low power reception timeout register (DSI_LPRXTO)

Address offset: 0x14

Reset value: 0x0000 FFFF

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



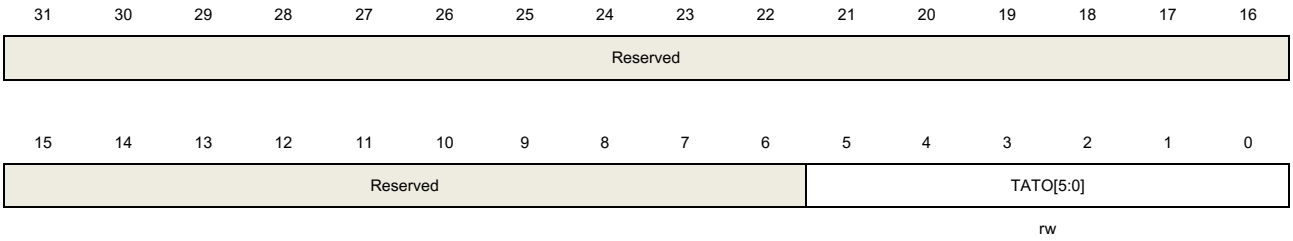
Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:0	LPRXTOCNT[23:0]	Low power reception timeout counter. The timeout value is protocol specific. Time out value is calculated from txclkesc(50ns).

31.15.7. DSI Host turn around timeout register (DSI_TATO)

Address offset: 0x18

Reset value: 0x0000 0017

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



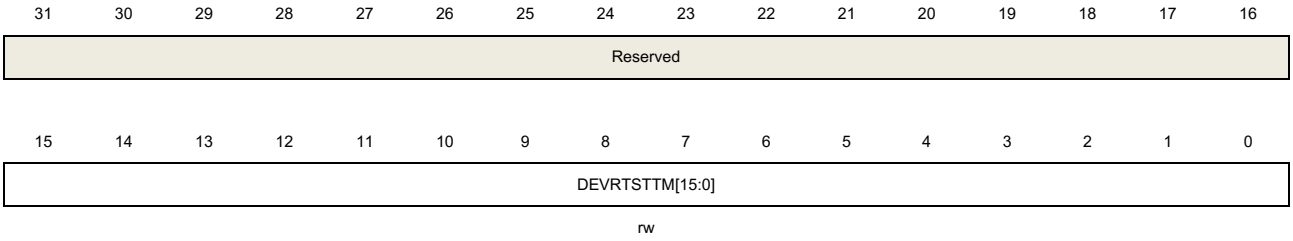
Bits	Fields	Descriptions
31:6	Reserved	Must be kept at reset value.
5:0	TATO[5:0]	Turn around timeout.

31.15.8. DSI Host device reset timer register (DSI_DEVRSTTM)

Address offset: 0x1C

Reset value: 0x0000 00FF

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



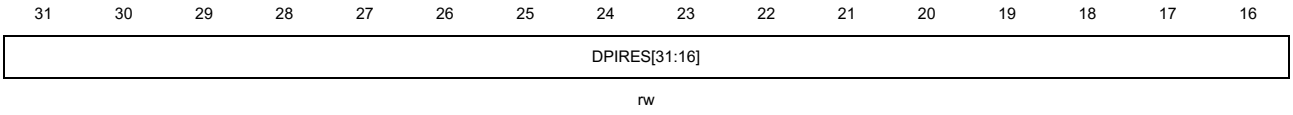
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	DEVRSTTM[15:0]	Device reset timer.

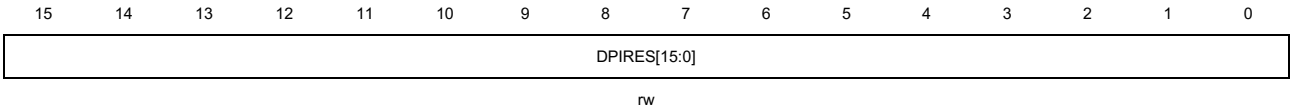
31.15.9. DSI Host DPI resolution register (DSI_DPIRES)

Address offset: 0x20

Reset value: 0x0000 0000

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





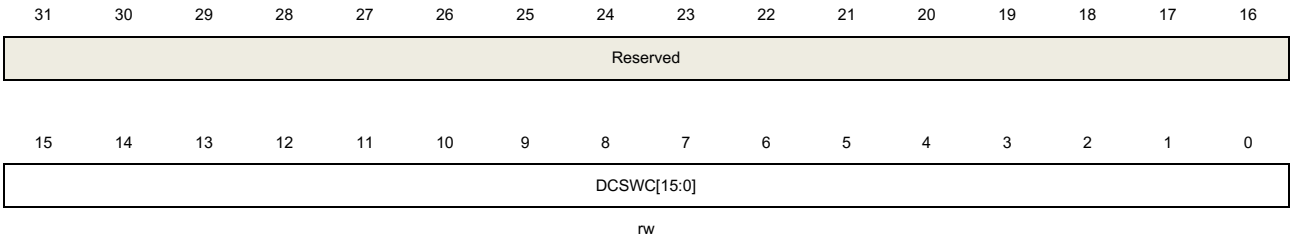
Bits	Fields	Descriptions
31:0	DPIRES[31:0]	DPI resolution [15:0]: horizontal address(Hadr) count in pixels [31:16]: vertical address(Vadr) count in pixels

31.15.10. DSI Host DCS word count register (DSI_DCSWC)

Address offset: 0x24

Reset value: 0x0000 0000

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



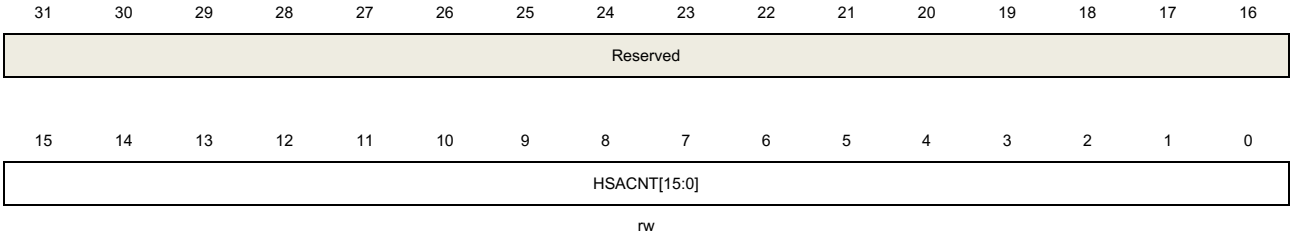
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	DCSWC[15:0]	DCS/Generic word count values need to be written before DCS or Generic long write.

31.15.11. DSI Host horizontal sync active count configuration register (DSI_HSACFG)

Address offset: 0x28

Reset value: 0x0000 0000

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



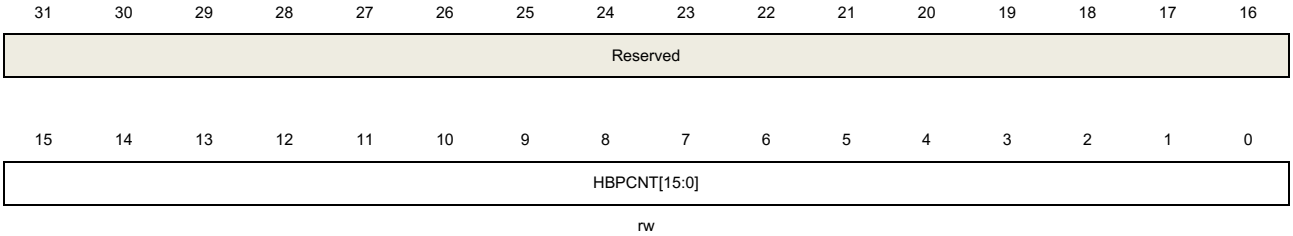
Bits	Fields	Descriptions

31:16	Reserved	Must be kept at reset value.
15:0	HSACNT[15:0]	Horizontal sync active count value

31.15.12. DSI Host horizontal back porch count configuration register (DSI_HBPCFG)

Address offset: 0x2C
Reset value: 0x0000 0000

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

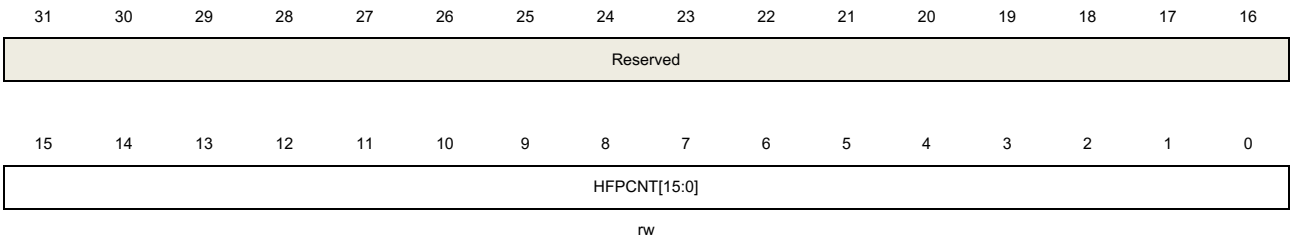


Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	HBPCNT[15:0]	Horizontal back porch count value

31.15.13. DSI Host horizontal front porch count configuration register (DSI_HFPCFG)

Address offset: 0x30
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.
15:0	HFPCNT[15:0]	Horizontal front porch count value

31.15.14. DSI Host horizontal active area count configuration register (DSI_HACFG)

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
HACNT[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	HACNT[15:0]	Horizontal active area count, that is the time for active Horizontal Address. Shows the horizontal active area value in terms of txbyteclkhs.

31.15.15. DSI Host vertical sync active count configuration register (DSI_VSACFG)

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
Reserved															

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

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	VSACNT[15:0]	Vertical sync active count value.

31.15.16. DSI Host vertical back porch count configuration register (DSI_VBPCFG)

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

Reserved

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

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	VBPCNT[15:0]	Vertical back porch count value.

31.15.17. DSI Host vertical front porch count configuration register (DSI_VFPCFG)

Address offset: 0x40

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
VFPCNT[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	VFPCNT[15:0]	Vertical front porch count value.

31.15.18. DSI Host data lane high low switch count configuration register (DSI_DLCFG)

Address offset: 0x44

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
DLCFG[15:0]															
rw															

Bits	Fields	Descriptions

31:16	Reserved	Must be kept at reset value.
15:0	DLCFG[15:0]	Switch count of HS to LP or LP to HS for Data lane. The count is in terms txbyteclkhs.

31.15.19. DSI Host DPI control register (DSI_DPICTL)

Address offset: 0x48

Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												CMOFF	CMON	TON	SD
												rw	rw	rw	rw

Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value.
3	CMOFF	Color Mode OFF(0x08) command is 1274pprox.1274er for the DPI's virtual channel.
2	CMON	Color Mode ON(0x04) command is 1274pprox.1274er for the DPI's virtual channel.
1	TON	Turn ON (0x02) command is 1274pprox.1274er for the DPI's virtual channel.
0	SD	Shut down(0x01) command is 1274pprox.1274er for the DPI's virtual channel.

31.15.20. DSI Host Dphy PLL lock count register (DSI_DPHY_PLLLK)

Address offset: 0x4C

Reset value: 0x0000 07D0

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
PLLCNTVAL[15:0]															
rw															

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

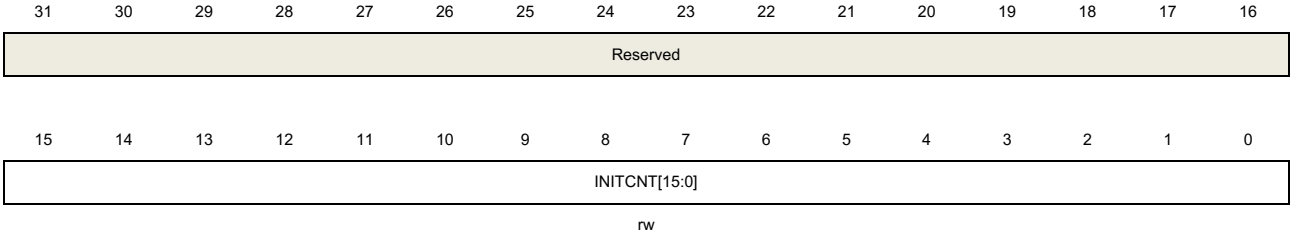
15:0	PLLCNTVAL[15:0]	PLL counter value in terms of low power clock.

31.15.21. DSI Host initialization count configuration register (DSI_INITCFG)

Address offset: 0x50

Reset value: 0x0000 07D0

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



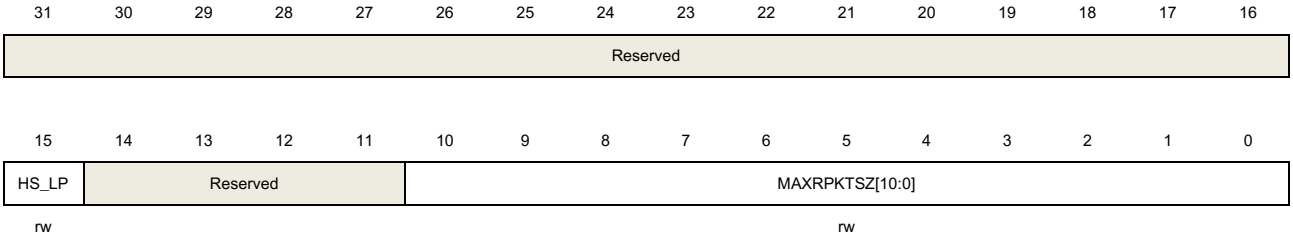
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	INITCNT[15:0]	Counter value in terms of low power clock to initialize the DSI Host IP that drives a stop state on the mipi's D-PHY bus. DPHY Initialization period min 100μs(0x7D0 txclkesc).

31.15.22. DSI Host maximum return packet control register(DSI_MAXRPKTCTL)

Address offset: 0x54

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	HS_LP	Indicates the data transfer type 0: Maximum return packet size packet will transfer in HS 1: Maximum return packet size packet will transfer in LPDT
14:11	Reserved	Must be kept at reset value.
10:0	MAXRPKTSZ[10:0]	Maximum return packet size in bytes.

31.15.23. DSI Host Video mode format register (DSI_VMFMT)

Address offset: 0x58

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														VMFMT[1:0]	
														rw	

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1:0	VMFMT[1:0]	Video mode format 00: Reserved 01: Non Burst Mode with Sync Pulse 10: Non Burst Mode with Sync events 11: Burst Mode

31.15.24. DSI Host clock EOT control register (DSI_CKEOTCTL)

Address offset: 0x5C

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													VBTADIS	CKSTOP	EOTDIS
													rw	rw	rw

Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value.
2	VBTADIS	Disable video BTA 0: BTA sending at the last blanking line of VFP is enabled. 1: BTA sending at the last blanking line of VFP is disabled.
1	CKSTOP	Enable clock stopping feature. 0: Disable 1: Enable

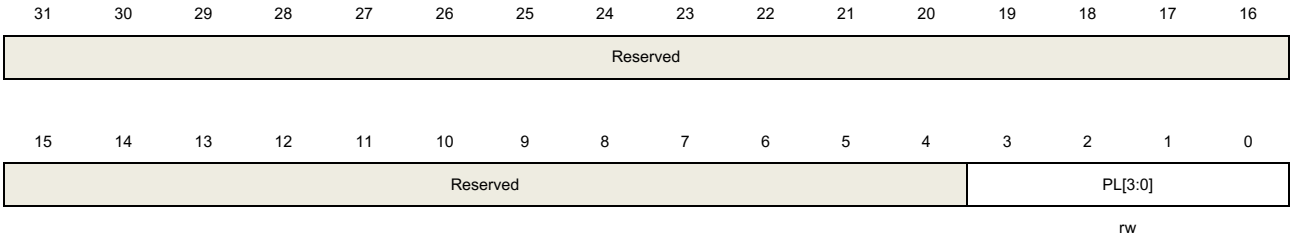
0	EOTDIS	Disable EOT packet(EoTp) transmission 0: Enable 1: Disable

31.15.25. DSI Host polarity control register (DSI_PLCTL)

Address offset: 0x60

Reset value: 0x0000 0000

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



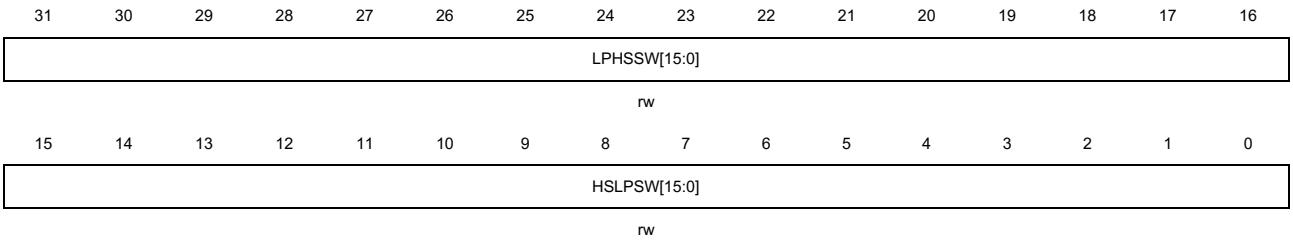
Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value.
3:0	PL[3:0]	Polarity 0000: Polarity for Vsync 0001: Polarity for Hsync 0010: Polarity for shut down 0011: Polarity for Color mode Others: Reserved

31.15.26. DSI Host clock lane switching configuration register (DSI_CKLSWCFG)

Address offset: 0x64

Reset value: 0x0000 0000

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



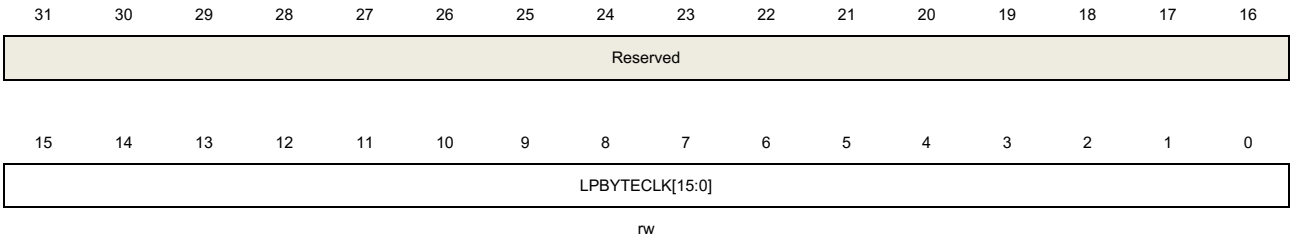
Bits	Fields	Descriptions
31:16	LPHSSW[15:0]	LP to HS switching counter for clock Lane
15:0	HSLPSW[15:0]	HS to LP switching counter for clock Lane

31.15.27. DSI Host low power byte clock configuration register (DSI_LPBCLKCFG)

Address offset: 0x68

Reset value: 0x0000 0001

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



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	LPBYTECLK[15:0]	LP clock equivalence in terms of byte clock.

31.15.28. DSI Host Dphy parameter configuration register (DSI_DPHY_PCFG)

Address offset: 0x6C

Reset value: 0x0B06 1A04

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



Bits	Fields	Descriptions
31:24	HS_EXIT[7:0]	HS exit time($T_{HS\text{-}EXIT}$).
23:16	HS_TRAIL[7:0]	HS trail time($T_{HS\text{-}TRAIL}$).
15:8	HS_ZERO[7:0]	HS drive zero time($T_{HS\text{-}ZERO}$).
7:0	HS_PREP[7:0]	HS preparation time($T_{HS\text{-}PREPARE}$).

31.15.29. DSI Host Dphy clock lane timing parameter configuration register (DSI_DPHY_CLTMPCFG)

Address offset: 0x70

Reset value: 0x0B06 1A04

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
CLK_EXIT[7:0]								CLK_TRAIL[7:0]							
rw								rw							

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

Bits	Fields	Descriptions
31:24	CLK_EXIT[7:0]	CLK exit time($T_{CLK-EXIT}$).
23:16	CLK_TRAIL[7:0]	CLK trail time($T_{CLK-TRAIL}$).
15:8	CLK_ZERO[7:0]	CLK drive zero time($T_{CLK-ZERO}$).
7:0	CLK_PREP[7:0]	CLK preparation time($T_{CLK-PREPARE}$).

31.15.30. DSI Host Dphy reset enable to DFE register (DSI_DPHY_RSTENDFE)

Address offset: 0x74

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															RSTENDFE
															rw

Bits	Fields	Descriptions
31:1	Reserved	Must be kept at reset value.
0	RSTENDFE	Reset (enable) to the Digital Front End(DFE) 0: Disable 1: Enable

31.15.31. DSI Host Dphy trim control register 0 (DSI_DPHY_TRIMCTL0)

Address offset: 0x78

Reset value: 0x2AAA A067

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

LPLDOPD_SEL	LPLDOPD_IN	TX_VMD_REF_400MV_TRIM[2:0]	TX_PREDRV_REF_400MV_TRIM[1:0]	PLL_CP_LDO_900MV_TRIM[1:0]	LPRX_REF_650MV_TRIM[1:0]	LP_LDO_REF_725MV_TRIM[1:0]	CD_LOW_325MV_TRIM[1:0]	CD_HIGH_750MV_TRIM[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

CD_HIGH_750MV_TRIM[0]	BG_TRIM[1:0]	VBG_SEL	DC_TEST_DATA	HS_DC_TEST_EN	LP_DC_TEST_EN	TX_BCLK_POL_SEL	TX_HS_SLEW[1:0]	TX_HS_DRV[1:0]	TRIM_LP_SLEWRATE[1:0]	TRIM_FF_BYPASSC	TRIM_FF_BYPASS
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31	LPLDOPD_SEL	LP LDO power down selection
30	LPLDOPD_IN	External LP LDO power down
29:27	TX_VMD_REF_400MV_TRIM[2:0]	Adjust HS TX common mode voltage Default value should be '101'.
26:25	TX_PREDRV_REF_400MV_TRIM[1:0]	Adjust HS TX pre-driver LDO reference voltage Default value should be '01'.
24:23	PLL_CP_LDO_900MV_TRIM[1:0]	Adjust PLL CP LDO reference voltage Default value should be '01'.
22:21	LPRX_REF_650MV_TRIM[1:0]	Adjust LP RX reference voltage Default value should be '01'.
20:19	LP_LDO_REF_725MV_TRIM[1:0]	Adjust LP LDO reference voltage Default value should be '01'.
18:17	CD_LOW_325MV_TRIM[1:0]	Adjust CD reference Low voltage Default value should be '01'.
16:15	CD_HIGH_750MV_TRIM[1:0]	Adjust CD reference High voltage Default value should be '01'.
14:13	BG_TRIM[1:0]	Adjust bandgap reference voltage Default value should be '01'.
12	VBG_SEL	Select the PVT independent Bandgap voltage
11	DC_TEST_DATA	LP or HS data to transmit out when anyone of the DC test is enabled
10	HS_DC_TEST_EN	Enable HS transmitter DC test 0: Disable 1: Enable
9	LP_DC_TEST_EN	Enable LP transmitter DC test 0: Disable

		1: Enable
8	TX_BCLK_POL_SEL	Select the TX byte clock polarity.
7:6	TX_HS_SLEW[1:0]	Adjust the HS transmit signal slew rate Default value should be '01'.
5:4	TX_HS_DRV[1:0]	Adjust the HS driver impedance Default value should be '01'.
3:2	TRIM_LP_SLEWRATE[1:0]	Adjust the LP transmit signal slew rate Default value should be '01'.
1	TRIM_FF_BYPASSC	Used to bypass the LPTX signals sampling done by the reference clock 0: Disable 1: Enable
0	TRIM_FF_BYPASS	Used to bypass the LPTX signals sampling done by the reference clock 0: Lane0 1: Lane1

31.15.32. DSI Host Dphy trim control register 1 (DSI_DPHY_TRIMCTL1)

Address offset: 0x7C

Reset value: 0x0C03 003F

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
ESCAPE_CLK_SEL	TRIM_AFE_OFF_ULPS	TRIM_DIS_ULPS_LP00_CK	TRIM_DIS_ULPS_LP00	PLL_SUPPLY_SEL	PLL_CP_CT_TRIM[1:0]		LPXORCLK_DLY[1:0]		LPRX_GLITCH_FIL_T_BYPASS	RXLANE_POLARITY_SWAP0	CDPD0	BYPASS_LPTX_PU_LOGIC_CK	BYPASS_LPTX_PU_LOGIC	REFCLK_DLY[1:0]	
rw	rw	rw	rw	rw	rw		rw		rw	rw	rw	rw	rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSTODBY4	PLL_STCKSEL[1:0]		BW_RES_SEL	RES[2:0]			EXT_RSTB	EXT_RSTB_EN	CNTA[5:0]						CNTB
rw	rw		rw	rw			rw	rw	rw						rw

Bits	Fields	Descriptions
31	ESCAPE_CLK_SEL	It is used to select the reference clock 0: Reference clock without division 1: Reference clock divide by 2
30	TRIM_AFE_OFF_ULPS	Enable to power off all AFE blocks in ULPS
29	TRIM_DIS_ULPS_LP00_CK	Disable weak pull-down resistor added in mipi clock lane

28	TRIM_DIS_ULPS_LP00	Disable weak pull-down resistor added in mipi data lanes
27	PLL_SUPPLY_SEL	PLL CP supply voltage selection 0: if the IO power supply is 2.5V and above 1: if the IO power supply is below 2.5V
26:25	PLL_CP_CT_TRIM[1:0]	Programmable PLL charge pump current Default value should be '01'.
24:23	LPXORCLK_DLY[1:0]	Programmable delay in LP XOR clock Default value should be '00'.
22	LPRX_GLITCH_FILT_BYPASS	LPXOR clock glitch filter bypass enable
21	RXLANE_POLARITY_SWAP0	Control to swap the polarity of the RX output
20	CDPD0	Contention detection block power down signal
19	BYPASS_LPTX_PU_LOGIC_CK	LP11 initialization bypass signal for clock lane
18	BYPASS_LPTX_PU_LOGIC	LP11 initialization bypass signal for data lanes
17:16	REFCLK_DLY[1:0]	Programmable escape clock delay in AFE to capture the LP data driven by the DFE Default value should be '11'.
15	TSTODBY4	Control to select Clock Divide by 4 or Clock Divide by 8 as output.
14:13	PLL_STCKSEL[1:0]	Control to select a clock from PLL to test
12	BW_RES_SEL	PLL LPF resistor selection
11:9	RES[2:0]	LPF resistor selection
8	EXT_RSTB	External active-low reset
7	EXT_RSTB_EN	Enable external active-low reset
6:1	CNTA[5:0]	PLL Divider ratio used in counter.
0	CNTB	Reference CNTA[5:0]

31.15.33. DSI Host Dphy trim control register 2 (DSI_DPHY_TRIMCTL2)

Address offset: 0x80
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		PDR	TRIM_PLL_LOCK	PLLPD	HSTX_LDO_PD	BGPD	V2IPD	CPSWP	PSWP1	PSWP0	DPHY_MUX_CFG	LANE1_EN	LANE0_EN	SYS_RESET_N_INT	Reserved
		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	TRIM_LPTX_D1	TRIM_LPTX_D0	TRIM_LPTX_CK	TRIM_LPTX_BYPASS_EN	TRIM_HSTX_D1	TRIM_HSTX_D0	TRIM_HSTX_CK	TRIM_HSTX_BYPASS_EN	TRIM_LPRX_D0	TRIM_LPRX_BYPASS_EN
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29	PDR	Power down control
28	TRIM_PLL_LOCK	Force PLL lock detect high
27	PLLPD	Force PLL circuits power down if PDR is set 1
26	HSTX_LDO_PD	Force high speed driver LDO circuits power down
25	BGPD	Force bandgap circuits power down if PDR is set 1
24	V2IPD	Force bias circuits power down if PDR is set 1
23	CPSWP	Clock lane polarity swap enable
22	PSWP1	Data lane 1 polarity swap enable
21	PSWP0	Data lane 0 polarity swap enable
20	DPHY_MUX_CFG	Lane swap configuration bit
19	LANE1_EN	lane enable control for lane1
18	LANE0_EN	lane enable control for lane0
17	SYS_RESET_N_INT	Optional system reset signal
16:10	Reserved	Must be kept at reset value.
9	TRIM_LPTX_D1	Control to overwrite LP TX data of data lane1
8	TRIM_LPTX_D0	Control to overwrite LP TX data of data lane0
7	TRIM_LPTX_CK	Control to overwrite LP TX data of clock lane
6	TRIM_LPTX_BYPASS_EN	Control to overwrite LP TX enable
5	TRIM_HSTX_D1	Control to overwrite HS TX data of data lane1
4	TRIM_HSTX_D0	Control to overwrite HS TX data of data lane0

3	TRIM_HSTX_CK	Control to overwrite HS TX data of clock lane
2	TRIM_HSTX_BYPASS_EN	Control to overwrite HS TX enable
1	TRIM_LPRX_D0	Control to overwrite LP RX data
0	TRIM_LPRX_BYPASS_EN	Control to overwrite LP RX enable

31.15.34. DSI Host deskew control register (DSI_DSKCTL)

Address offset: 0x88

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														DSKDUR[15:14]	
														rw	

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

Bits	Fields	Descriptions
31:18	Reserved	Must be kept at reset value.
17:2	DSKDUR[15:0]	deskew duration Minimum deskew duration should be 16UI. 2 UI = 1DDR, 4 DDR = 1Byte clock. This counter runs in byteclk domain hence 16/8 = 2 TxByteClkHS. And maximum value is max 10us.
1:0	DSKCALIB[1:0]	deskew calibration 00: No deskew enabled 01: Enables the init deskew 10: Enables the periodic deskew 11: Reserved

31.15.35. DSI Host periodic deskew timer register(DSI_DSKTM)

Address offset: 0x8C

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

DSKTM[15:0]
rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	DSKTM[15:0]	Periodic deskew timer The periodic deskew will be enabled with this specified interval.

31.15.36. DSI Host deskew status register (DSI_DSKSTAT)

Address offset: 0x90

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															DSKIP
															r

Bits	Fields	Descriptions
31:1	Reserved	Must be kept at reset value.
0	DSKIP	Deskew calibration is in progress. 0: Not in progress 1: In progress

31.15.37. DSI Host error auto recovery control register (DSI_EARCTL)

Address offset: 0x98

Reset value: 0x0000 003F

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										LP_RX_TO_CLR	HS_RX_TO_CLR	LO_CONT_CLR	HI_CONT_CLR	INVLD_DT_CLR	ECC_MUL_ERR_CLR
										rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions

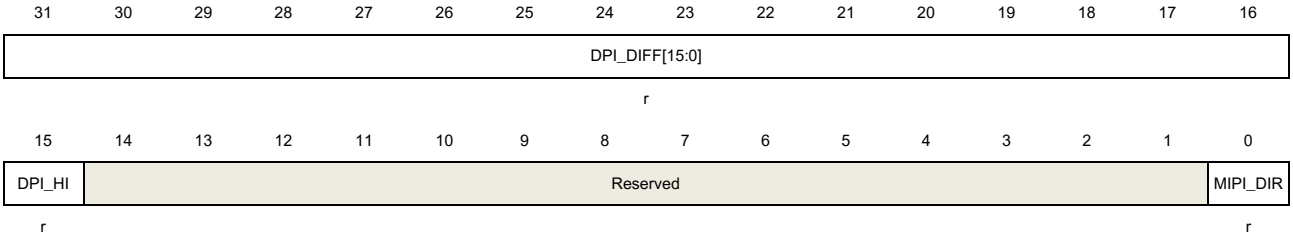
31:6	Reserved	Must be kept at reset value.
5	LP_RX_TO_CLR	lp_rx_timeout_clr error recovery action is taken immediately by DSI TX 0: Disable 1: Enable
4	HS_RX_TO_CLR	Hs_rx_timeout_clr error recovery action is taken immediately by DSI TX 0: Disable 1: Enable
3	LO_CONT_CLR	lo_cont_clr error recovery action is taken immediately by DSI TX 0: Disable 1: Enable
2	HI_CONT_CLR	Hi_cont_clr error recovery action is taken immediately by DSI TX 0: Disable 1: Enable
1	INVLD_DT_CLR	Invld_dt_clr error recovery action is taken immediately by DSI TX 0: Disable 1: Enable
0	ECC_MUL_ERR_CLR	Ecc_mul_err_clr error recovery action is taken immediately by DSI TX 0: Disable 1: Enable

31.15.38. DSI Host MIPI direction and DPI status register (DSI_MIPIDIRDPISTAT)

Address offset: 0x9C

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:16	DPI_DIFF[15:0]	The difference in one line time between DPI and DSI.
15	DPI_HI	DPI line time is greater or DSI line time is greater. 0: DPI is less than to DSI. 1: DPI is greater than or equal to DSI.
14:1	Reserved	Must be kept at reset value.

0	MIPI_DIR	MIPI BUS direction 0: DSI Host has the control over MIPI bus. 1: DSI Host is in Receive mode.

31.15.39. DSI Host Polarity Swap register (DSI_PLSWAP)

Address offset: 0xA0

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												CMPL	SDPL	HSPL	VSPL
												rw	rw	rw	rw

Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value.
3	CMPL	Color mode polarity 0: Active low 1: Active high
2	SDPL	Shutdown polarity 0: Active low 1: Active high
1	HSPL	Hsync polarity 0: Active low 1: Active high
0	VSPL	Vsync polarity 0: Active low 1: Active high

31.15.40. DSI Host frequency mode control register (DSI_FMCTL)

Address offset: 0xA4

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	FREQMOD
	rw

Bits	Fields	Descriptions
31:1	Reserved	Must be kept at reset value.
0	FREQMOD	frequency mode 0: DDR freq < 200MHz 1: DDR freq >= 200MHz

31.15.41. DSI Host Tx data width configuration register (DSI_TXDTWCFG)

Address offset: 0xB0

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								TXDTWIDTH[7:0]							
								rw							

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	TXDTWIDTH[7:0]	Tx data width hs Default value should be 0, 8 bit data width.

31.15.42. DSI Host LP control register (DSI_LPCTL)

Address offset: 0xF8

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								WORDCNT[16:8]							
								rw							

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

Bits	Fields	Descriptions

31:24	Reserved	Must be kept at reset value.
23:8	WORDCNT[16:0]	Word count **Note**: 1. Maximum word count is 492 Bytes 2. Invalid parameters must be set to "00h". 3. For data type "05h" and "06h" this field is programmed with opcode and 00h 4. For data type "15h" this field is programmed with opcode and parameter
7:6	VCH[1:0]	Virtual channel
5:0	DTTYPE[5:0]	Data type

31.15.43. DSI Host LP data register (DSI_LPDATA)

Address offset: 0xFC

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
LPDATA[31:16]															
rw															

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

Bits	Fields	Descriptions
31:0	LPDATA[31:0]	Data Used for Generic/ DCS data transfers in low power mode.

31.15.44. DSI Host HS control register (DSI_HSCTL)

Address offset: 0x100

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								WORDCNT[16:8]							
								rw							

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

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:8	WORDCNT[16:0]	Word count

Note:

1. Maximum word count is 492 Bytes
2. Invalid parameters must be set to "00h".
3. For data type "05h" and "06h" this field is programmed with opcode and 00h
4. For data type "15h" this field is programmed with opcode and parameter

7:6	VCH[1:0]	Virtual channel
5:0	DTTYPE[5:0]	Data type

31.15.45. DSI Host HS data register (DSI_HSDATA)

Address offset: 0x104

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
HSDATA[31:16]															
rw															

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

Bits	Fields	Descriptions
31:0	HSDATA[31:0]	Data Used for Generic/ DCS data transfers in high-speed mode.

31.15.46. DSI Host FIFO status register (DSI_FIFOSTAT)

Address offset: 0x108

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					LPCTLFIFOE	LPCTLFIFOHE	LPCTLFIFOF	Reserved					HSCTLFIFOE	HSCTLFIFOHE	HSCTLFIFOF
					r	r	r						r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					LPDTFIFOE	LPDTFIFOHE	LPDTFIFOF	Reserved					HSDTFIFOE	HSDTFIFOHE	HSDTFIFOF
					r	r	r						r	r	r

Bits	Fields	Descriptions
31:27	Reserved	Must be kept at reset value.
26	LPCTLFIFOE	LP Generic/DCS control FIFO empty

		0: FIFO is not empty 1: FIFO is empty
25	LPCTLFIFOHE	LP Generic/DCS control FIFO half empty 0: FIFO is not half empty 1: FIFO is half empty
24	LPCTLFIFOF	LP Generic/DCS control FIFO full 0: FIFO is not full 1: FIFO is full
23:19	Reserved	Must be kept at reset value.
18	HSCTLFIFOE	HS Generic/DCS control FIFO empty 0: FIFO is not empty 1: FIFO is empty
17	HSCTLFIFOHE	HS Generic/DCS control FIFO half empty 0: FIFO is not half empty 1: FIFO is half empty
16	HSCTLFIFOF	HS Generic/DCS control FIFO full 0: FIFO is not full 1: FIFO is full
15:11	Reserved	Must be kept at reset value.
10	LPDTFIFOE	LP Generic/DCS data FIFO empty 0: FIFO is not empty 1: FIFO is empty
9	LPDTFIFOHE	LP Generic/DCS data FIFO half empty 0: FIFO is not half empty 1: FIFO is half empty
8	LPDTFIFOF	LP Generic/DCS data FIFO full 0: FIFO is not full 1: FIFO is full
7:3	Reserved	Must be kept at reset value.
2	HSDTFIFOE	HS Generic/DCS data FIFO empty 0: FIFO is not empty 1: FIFO is empty
1	HSDTFIFOHE	HS Generic/DCS data FIFO half empty 0: FIFO is not half empty 1: FIFO is half empty
0	HSDTFIFOF	HS Generic/DCS data FIFO full 0: FIFO is not full

1: FIFO is full

31.15.47. DSI Host interrupt status register1 (DSI_INTSTAT1)

Address offset: 0x10C

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										TETGIF	RDDTAIF	HSCTLFIFOFIF	LPCTLFIFOFIF	LPDTFIFOFIF	HSDTFIFOFIF
										rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

Bits	Fields	Descriptions
31:6	Reserved	Must be kept at reset value.
5	TETGIF	Tearing effect(TE) trigger message received 0: Not received 1: Received
4	RDDTAIF	LP Generic/DCS read data available. 0: Not available 1: Available
3	HSCTLFIFOFIF	HS Generic/DCS control FIFO full 0: FIFO is not full 1: FIFO is full
2	LPCTLFIFOFIF	LP Generic/DCS control FIFO full 0: FIFO is not full 1: FIFO is full
1	LPDTFIFOFIF	LP Generic/DCS data FIFO full 0: FIFO is not full 1: FIFO is full
0	HSDTFIFOFIF	HS Generic/DCS data FIFO full 0: FIFO is not full 1: FIFO is full

31.15.48. DSI Wrapper control register (DSI_WCTL)

Address offset: 0x200

Reset value: 0x0000 0007

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							ARBUF	RBUF	ARM	TESRC	TLIDMA	TEPOL	Reserved		
							rw	w	rw	rw	rw	rw			

Bits	Fields	Descriptions
31:9	Reserved	Must be kept at reset value.
8	ARBUF	Hardware auto refresh cmd fifo buffer enable 0: No auto-refresh of cmd FIFO 1: Auto-refresh cmd FIFO when software set or hardware set WMS_EN
7	RBUF	Refresh cmd fifo buffer Write 1 to refresh cmd FIFO, not readable.
6	ARM	Automatic refresh mode 0: Automatic refresh mode disable 1: Automatic refresh mode enable
5	TESRC	Tearing effect (TE) source 0: DSI Link. 1: External pin
4	TLIDMA	TLI DMA mode 0: DMA mode disable, DMA used by TLI 1: DMA mode enable, DMA used by DSI
3	TEPOL	TE pin polarity This bit selects the polarity of the external pin tearing effect (TE) source 0: Rising edge 1: Falling edge.
2:0	Reserved	Must be kept at reset value.

31.15.49. DSI Wrapper interrupt enable register (DSI_WINTEN)

Address offset: 0x204

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	PLLULKIE	PLLLKIE	TEIE
	rw	rw	rw

Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value.
2	PLLULKIE	PLL unlock interrupt enable This bit enables the PLL unlock interrupt 0: Interrupt disabled 1: Interrupt enabled
1	PLLLKIE	PLL lock interrupt enable This bit enables the PLL lock interrupt 0: Interrupt disabled 1: Interrupt enabled
0	TEIE	Tearing effect interrupt enable This bit enables the tearing effect interrupt 0: Interrupt disabled 1: Interrupt enabled

31.15.50. DSI Wrapper interrupt and status register (DSI_WINTSTAT)

Address offset: 0x208

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												PLLLKS	PLLULKIF	PLLLKIF	TEIF
												r	rw	rw	rw

Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value.
3	PLLLKS	PLL lock status This bit is set when the PLL is locked and cleared when it is unlocked. 0: PLL is unlocked 1: PLL is locked
2	PLLULKIF	PLL unlock interrupt flag This bit is set when the PLL is unlocked.

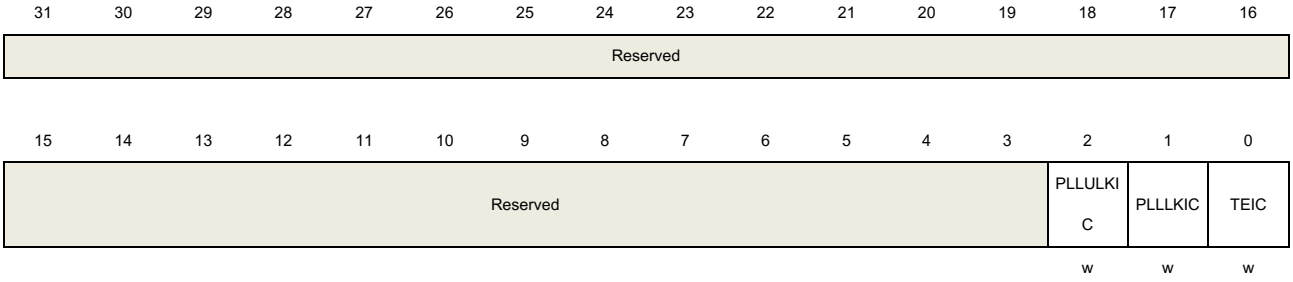
		0: No PLL unlock occurred 1: Occurred PLL unlock
1	PLLLKIF	PLL lock interrupt flag This bit is set when the PLL is locked. 0: No PLL lock occurred 1: Occurred PLL lock
0	TEIF	Tearing effect interrupt flag This bit is set when the tearing effect occurs. 0: No tearing effect occurred 1: Tearing effect lock

31.15.51. DSI Wrapper interrupt flag clear register (DSI_WINTC)

Address offset: 0x20C

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value.
2	PLLULKIC	PLL unlock interrupt flag clear Write 1 to clear the PLLULKIF flag in the DSI_WINTSTAT register
1	PLLLKIC	PLL lock interrupt flag clear Write 1 to clear the PLLLKIF flag in the DSI_WINTSTAT register
0	TEIC	Tearing effect interrupt flag clear Write 1 to clear the TEIF flag in the DSI_WINTSTAT register

31.15.52. DSI Wrapper RGB data format register (DSI_WRGBDF)

Address offset: 0x210

Reset value: 0x0000 0000

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

Reserved

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

Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value.
2:0	TLI_DF[2:0]	TLI data format select 000: RGB888(Default) 001: RGB565 010: RGB666(RGB888 to RGB666) 011: RGB666(RGB565 to RGB666) Others: Reserved

31.15.53. DSI Wrapper command mode data source register (DSI_WCMDDS)

Address offset: 0x220

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														LPCMDDS	HSCMDDS
														rw	rw

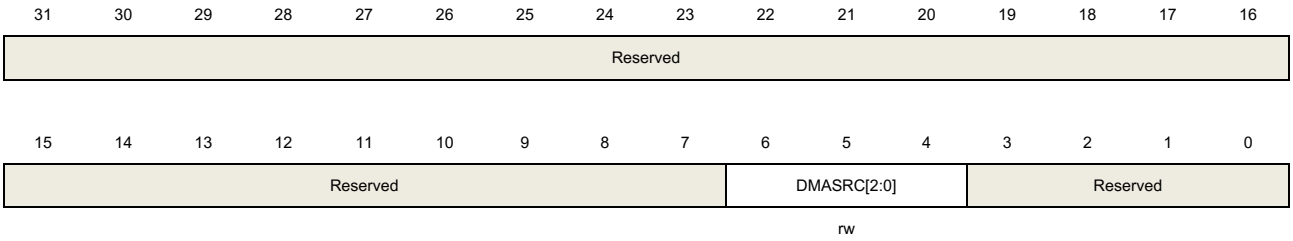
Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	LPCMDDS	Command mode low power fifo write data source **Note:** If the DMA function in LP mode is used, not only need to set this bit to 1, but also set HSCMDDS to 1. 0: From DSI_LPDATA(0xFC) 1: From TLI DMA
0	HSCMDDS	Command mode high speed fifo write data source 0: From DSI_HSDATA(0x104) 1: From TLI DMA

31.15.54. DSI Wrapper DMA request control register (DSI_WDMARCTL)

Address offset: 0x224

Reset value: 0x0000 0000

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



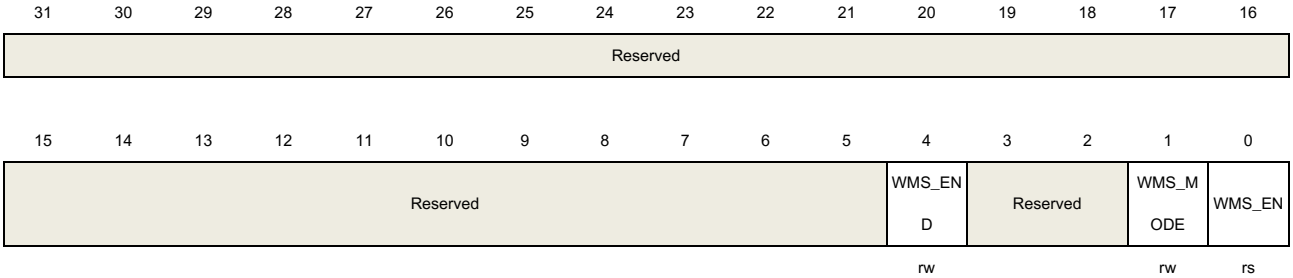
Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value.
6:4	DMASRC[2:0]	DMA request source select 000: HSDTFIFOHE 001: HSDTFIFOE 010: ~HSDTFIFOF 100: LPDTFIFOHE 101: LPDTFIFOE 110: ~LPDTFIFOF 111: Reserved
3:0	Reserved	Must be kept at reset value.

31.15.55. DSI Wrapper WMS write control register (DSI_WWMS_WCTL)

Address offset: 0x230

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:5	Reserved	Must be kept at reset value.
4	WMS_END	Write memory split packet end flag

		Hardware generates this END state signal, and the software can write 1 to clear it.
3:2	Reserved	Must be kept at reset value.
1	WMS_MODE	WMS function used for different modes 0: HS mode 1: LP mode
0	WMS_EN	Write memory split packet enable Use TLI for transmission. When enabled, it can activate DMA requests to TLI. This bit enables the function of automatically splitting DMA into WMS and WMC. The software writes 1 to enable the trigger (or the TE signal can be used; hardware enables it when the ARM bit is set, upon receiving the TE signal), and the hardware clears it after completing a frame of DMA transmission and unpacking. **Note:** Software can only write 1, and hardware clears the bit.

31.15.56. DSI Wrapper WMS interrupt mask register (DSI_WWMS_INTMSK)

Address offset: 0x234

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											WMS_ENDMSK	Reserved			
											rw				

Bits	Fields	Descriptions
31:5	Reserved	Must be kept at reset value.
4	WMS_ENDMSK	WMS_END flag interrupt signal mask 0: No interrupt generated 1: Interrupt generated
3:0	Reserved	Must be kept at reset value.

31.15.57. DSI Wrapper WMS header message register (DSI_WWMS_HEAD)

Address offset: 0x238

Reset value: 0x0039 3C2C

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	LONG_DCS_CMD[7:0]
	rw

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

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:16	LONG_DCS_CMD[7:0]	Long packet DCS command Automatically filled into the command FIFO, the default is DCS long write 0x39.
15:8	SECOND_HEAD[7:0]	Secondary split packet head message Header for subsequent packets after splitting DMA into multiple packets, with the default value being WMS code=0x3C.
7:0	FIRST_HEAD[7:0]	First split packet head message Header for the first packet after splitting DMA into multiple packets, with the default value being WMS code=0x2C.

31.15.58. DSI Wrapper WMS size control register (DSI_WWMS_SIZE)

Address offset: 0x23C

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
SUM_LEN[23:8]															
rw															

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

Bits	Fields	Descriptions
31:8	SUM_LEN[23:0]	The total number of words to be transferred by a single DMA trigger, which can generally be set to the data size of one frame.
7:0	PKT_LEN[7:0]	Divide the DMA into the length of each packet, in the number of words.

31.15.59. DSI Wrapper WMS status register 0 (DSI_WWMS_STAT0)

Address offset: 0x240

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
PKT_NUM[7:0]															
r															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	PKT_NUM[7:0]	The number of packets into which the current DMA transfer has been split.

31.15.60. DSI Wrapper WMS status register 1 (DSI_WWMS_STAT1)

Address offset: 0x244

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								SUM_CNT[23:16]							
								r							

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

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:0	SUM_CNT[23:0]	The total amount of data that has been transferred for the current DMA operation.

31.15.61. DSI Host interrupt enable register 1 (DSI_INTEN1)

Address offset: 0x250

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									TETGIE	RDDTAIE	HSCTLFIFOFIE	LPCTLFIFOFIE	LPDTFIFOFIE	HSDTFIFOFIE	Reserved
									rw	rw	rw	rw	rw	rw	

Bits	Fields	Descriptions

31:7	Reserved	Must be kept at reset value.
6	TETGIE	Tearing effect(TE) trigger message received interrupt enable. 0: Disable 1: Enable
5	RDDTAIE	LP Generic/DCS read data available interrupt enable. 0: Disable 1: Enable
4	HSCTLFIFOFIE	HS Generic/DCS control FIFO full interrupt enable. 0: Disable 1: Enable
3	LPCTLFIFOFIE	LP Generic/DCS control FIFO full interrupt enable. 0: Disable 1: Enable
2	LPDTFIFOFIE	LP Generic/DCS data FIFO full interrupt enable. 0: Disable 1: Enable
1	HSDTFIFOFIE	HS Generic/DCS data FIFO full interrupt enable. 0: Disable 1: Enables
0	Reserved	Must be kept at reset value.

32. Fast Fourier Transform (FFT)

32.1. 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.

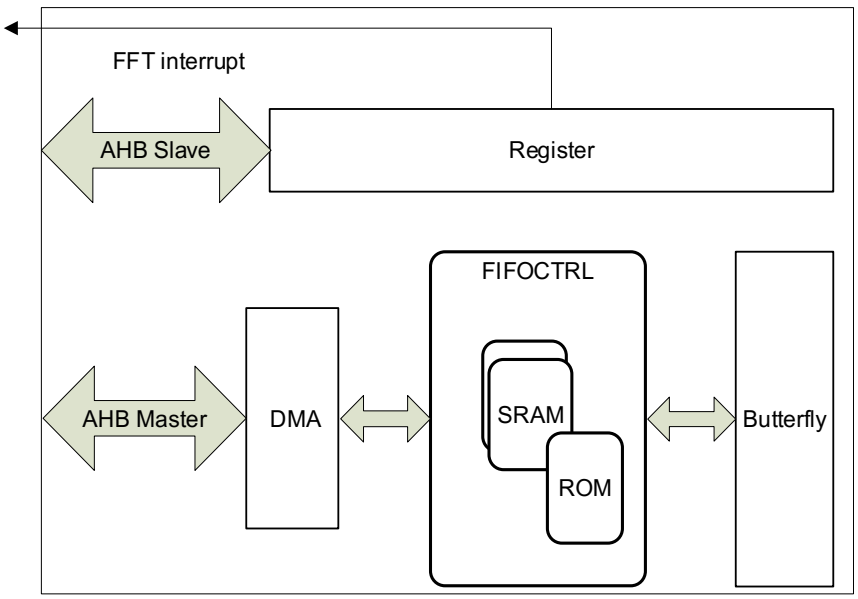
32.2. Characteristics

- 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

32.3. Functional description

Figure 32-1. FFT module block diagram provides details of the internal configuration of the FFT module.

Figure 32-1. 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.

32.4. Data format

The operation data and calculation result data format is given in ***Table 32-1. IEEE 32-Bit Single Precision Floating-Point Format***. They must meet IEEE 32-bit Single Precision Floating-Point.

Table 32-1. 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)
10.1.7 1 -ve	0	non zero	De-normalized (V=$(-1)^S$*$2^{(-126)}$* (0.M))
10.1.8 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.

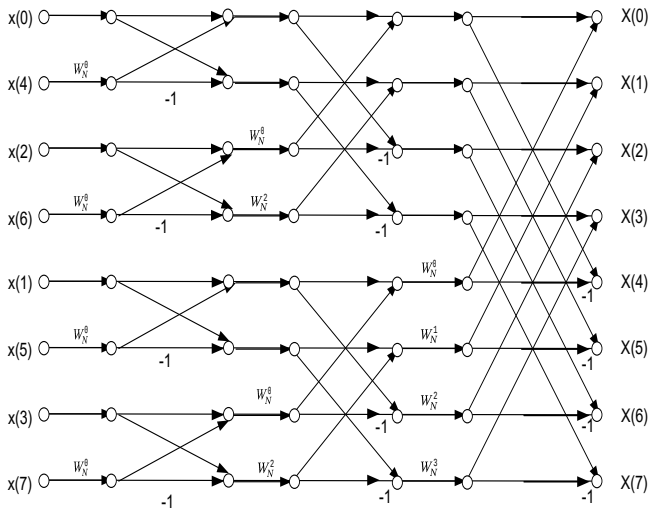
Not a Number (NaN): An NaN operand (E=max, M!=0) input is treated as Infinity (E=max,

M=0).

32.5. 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 32-2 8 point DIT FFT flow diagram***.

Figure 32-2 8 point DIT FFT flow diagram



The 8 points input sequence occurs in bit-reversed order with respect to the output, as shown in ***Table 32-2. 8 points input sequence bits reverse operation***.

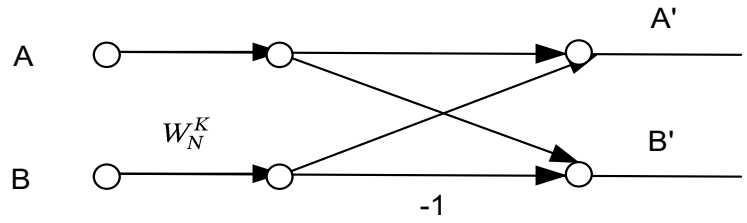
Table 32-2. 8 points input sequence bits reverse operation

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

32.6. Butterfly unit

The butterfly unit is the basic element of FFT calculation, and the progress are shown in the ***Figure 32-3. 8 points DIT FFT flow diagram***.

Figure 32-3. 8 points DIT FFT flow diagram



$$A' = A + BW = Re(A) + Re(BW) + j[Im(A) + Im(BW)]$$

$$B' = A - BW = Re(A) - Re(BW) + j[Im(A) - Im(BW)]$$

Where $W = W_N^k = e^{-j\frac{2\pi}{N}k}$

32.7. 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.

32.8. FFT SRAM

The 8KB space of FFT SRAM(0x3000 C000 - 0x3000 DFFF) 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.

32.9. 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 operation done.

The real part and image part start address increase and point to next input datas after every FFT operation done. That is, the INDEX[15:0] in FFT_LOOPLEN register increase 1 after every FFT operation 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.

32.10. 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.

32.11. 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 IRQ155.

32.12. Register definition

FFT start address: 0x4802 4000

32.12.1. 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

32.12.2. 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'.

32.12.3. 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'.

32.12.4. 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'.

32.12.5. 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'.

32.12.6. 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'.

33. High-Performance Digital Filter (HPDF)

33.1. Overview

A high performance digital filter module (HPDF) for external sigma delta (∑-Δ) modulator is integrated in GD32H77x_78x. 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.

33.2. Characteristics

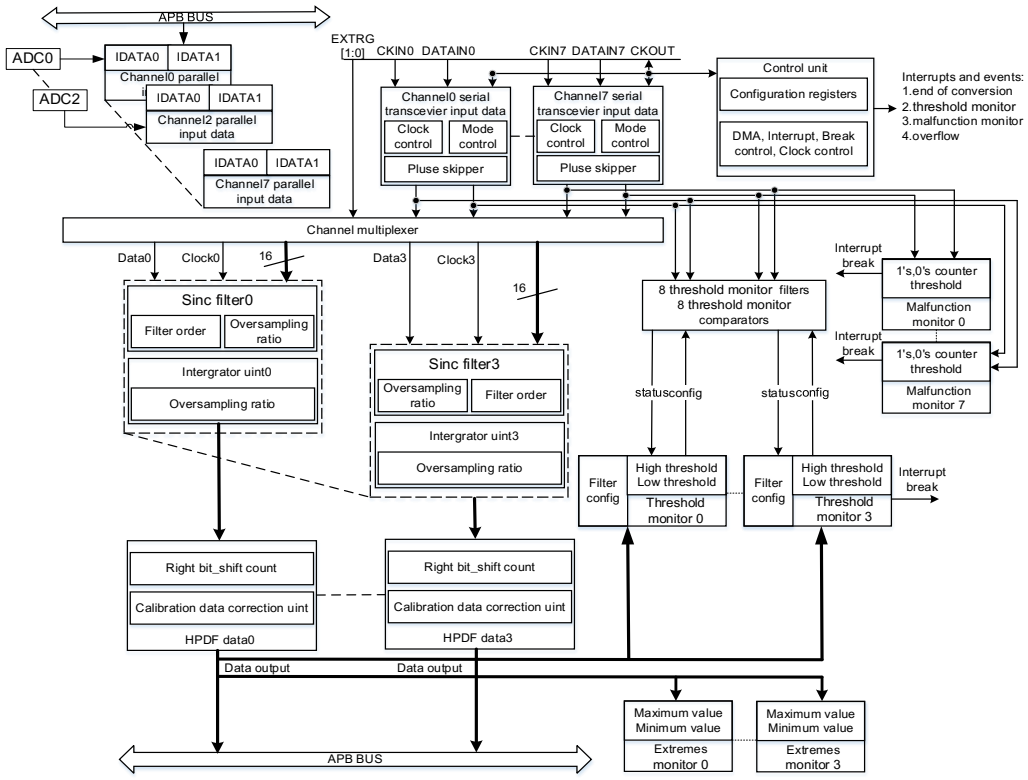
- 8 multiplex digital serial input channels
  - configurable SPI and Manchester interfaces
- 8 internal digital parallel input channels
  - input with up to 16-bit resolution
  - internal source: ADC data or memory (CPU/DMA write) data stream
- Configurable Sinc filter and integrator
  - the order and 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
- The third-phase data is obtained by adding the two phases of filtered data and taking the inverse, and it supports threshold monitoring
- The number of break signals can be configured
  - The TIMER is triggered after the break signal occurs the configured number of times

33.3. Function overview

33.3.1. HPDF Block Diagram

The structural block diagram of HPDF is shown in *Figure 33-1. HPDF block diagram*.

Figure 33-1. HPDF block diagram



The HPDF interface communicates with the external ∑-Δ modulator by the pins and internal signal in *Table 33-1. HPDF pins definition*.

Table 33-1. 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 ∑-Δ modulator.
CKINx	Clock input	External ∑-Δ modulator provides clock signal to serial interface.

PINs	Type	Description
DATAINx	Data input	The external ∑-Δ modulator transmits 1 bit data stream to the serial channel by this pin.

Table 33-2. HPDF internal signal

Break name	Break destination
HPDF_BREAK[0]	TIMER0 break0 / TIMER14 break0 / TIMER41 break0
HPDF_BREAK[1]	TIMER0 break1 / TIMER15 break0 / TIMER42 break0
HPDF_BREAK[2]	TIMER7 break0 / TIMER16 break0 / TIMER43 break0
HPDF_BREAK[3]	TIMER7 break1 / TIMER40 break0 / TIMER44 break0

33.3.2. 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.

33.3.3. 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 $f_{HPDFCLK}$ >= $4f_{CKIN}$. If the Manchester coding interface is used, the system clock $f_{HPDFCLK}$ >= $6f_{CKIN}$ 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 SAI0SEL[2:0] bit field in the configuration register 2 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.

33.3.4. 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

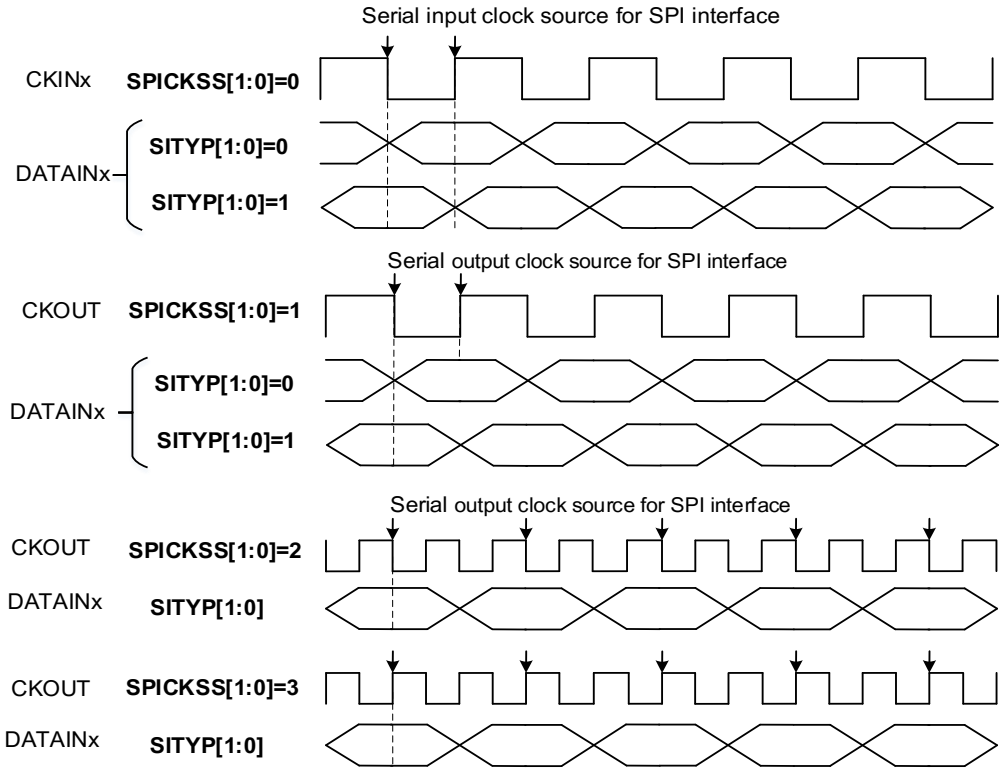
SPICKSS[1:0] bit field in HPDF_CHxCTL register. The data sampling points in SPI communication are shown in the table.

Table 33-3. 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 33-3. SPI interface clock configuration*, the sequence diagram of SPI data transmission is shown in the figure below.

Figure 33-2. 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 $f_{HPDFCLK}/4$.

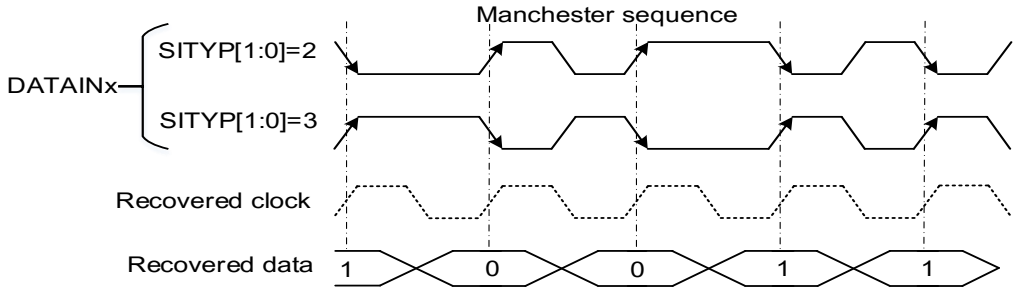
Manchester interface

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 33-3. 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:

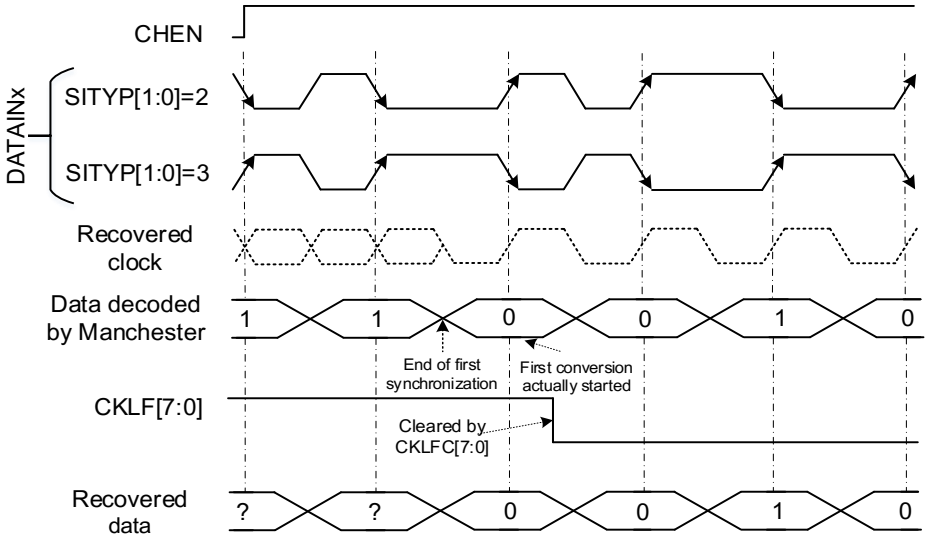
$$((CKOUTDIV+1)xT_{SYSCLK})<T_{Manchester_clock}<(2xCKOUTDIVxT_{SYSCLK}) \quad (33\text{-}1)$$

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 33-4. 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 ∑Δ 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 inserted) 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 CTCNT[27:0] in register HPDF_FLTyCT 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, CTCNT[27:0]=0 in HPDF_FLTyCT 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 CTCNT[27:0]. Therefore it is recommended to not interrupt the conversion for correct conversion duration result.

Conversion times:

inserted conversion or regular conversion with FAST = 0 (or first conversion if FAST=1):

for Sincx filters:

$$T = CTCNT / f_{HPDFCLK} = [SFOR * (IOR-1 +SFO) + SFO] / f_{CKIN}$$

for FastSinc filter:

$$T = CTCNT / f_{HPDFCLK} = [SFOR * (IOR-1 +4) + 2] / f_{CKIN}$$

regular conversion with FAST = 1 (except first conversion):

for Sincx and FastSinc filters:

$$T = CTCNT / f_{HPDFCLK} = [SFOR * IOR] / f_{CKIN}$$

in case if SFOR = SFOR[9:0]+1 = 1 (filter bypassed, active only integrator):

$$T = IOR / f_{CKIN} \text{ (but CTCNT=0)}$$

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.

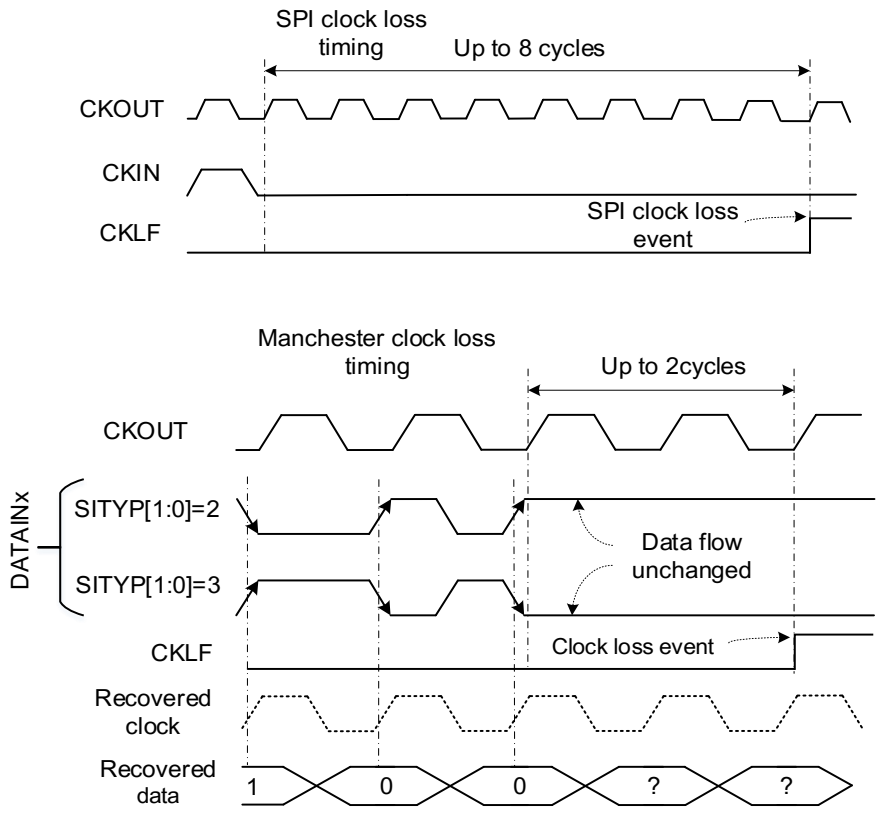
When the transceiver of the serial interface has not been synchronized, the clock loss flag bit is set to 1 and cannot be cleared by the corresponding CKLFC[7:0]. Therefore, the correct steps to use the clock loss function are as follows:

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 33-5. 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.

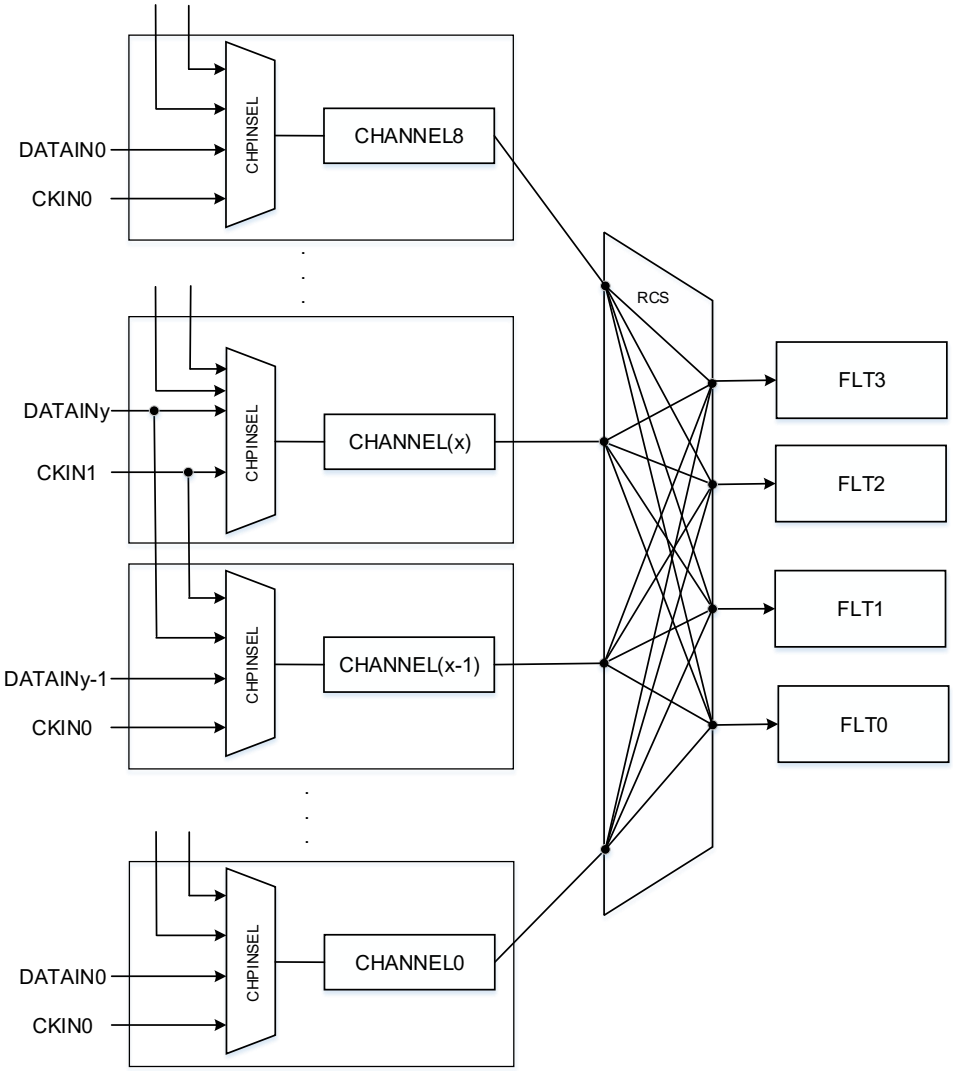
When PDM microphone data stream is input into serial channel, its configuration process is as follows:

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 1323pprox. 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 33-6. Channel pins redirection.***

Figure 33-6. 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_CHxCFG0 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.

33.3.5. 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 (channel0/2/4/6) supports dual channel mode. If odd channel (channel1/3/5/7) 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 33-4. 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.

33.3.6. Regular group conversion

HPDF module has 8 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 8 channels, which is determined by the RCS[2:0] 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.

33.3.7. Inserted group conversion

The conversion channel of the inserted group must select at least one of the 8 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, th filter 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 and ICTEEN[1:0] not 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 33-5. 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	TIMER22_TRGO0
HPDF_ITRG12	TIMER23_TRGO0
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.

33.3.8. 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 33-6. 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	+/-x^2	+/-2x^2	+/-x^3	+/-x^4	+/-x^5
4	+/-4	+/-16	+/-32	+/-64	+/-256	+/-1024
8	+/-8	+/-64	+/-128	+/-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.

33.3.9. 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 33-7. 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)

33.3.10. 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 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 33-8. 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 regular or inserted 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[1:0] 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'1, it means that channel 0 occurred 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 ((TMFOR[4:0] + 1)×4+ 2+1).

   $Sinc^X$ filter (x=1..3): Number of samples is equal to ((TMFOR[4:0] + 1)×TMSFO[1:0] + TMSFO[1:0] +1).

2. Subsequent conversions other than the first conversion:

   FastSinc and $Sinc^X$ filter (x=1..3): Number of samples is equal to (TMFOR[4:0] + 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[7:0] bit field indicates whether the maximum 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 SHTFC[7:0] or SLTFC[7:0] bit in the HPDF_FLTyTMFC register.

As is shown in ***Table 33-2. 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.

33.3.11. Sum threshold monitor

The purpose of the two filter sum threshold monitors is to monitor the third phase data, which

is obtained by adding the filtered data of the two phases and then inverting the result. The following equation determines the third phase data:

$$i_C = -(i_A + i_B) \quad (33\text{-}2)$$

If the value overflows after addition and inversion, the third phase data value will be the maximum value (8388607) or the minimum value (-8388608). When the data reaches the threshold set by the sum threshold monitor (maximum or minimum threshold), an interrupt or break event will be generated. The maximum threshold is determined by the SHTVAL[23:0] bits in HPDF_FLTSUMTMHT register, and the minimum threshold is determined by the SLTVAL[23:0] bits in HPDF_FLTSUMTMLT register.

Sum threshold monitor working mode

The working mode of the sum threshold monitor is divided into standard mode and fast mode. In fast mode, the input data for addition A and B is taken from the serial data stream. The data source for addition A/B is selected by the TMFDASEL[2:0]/TMFDBSEL[2:0] bit field in the HPDF_FLTSUMTMCTL register. In standard mode, the input data for addition A and B is taken from the final data. The data source for addition A/B is selected by the CDASEL[1:0]/CDBSEL[1:0] bit field in the HPDF_FLTSUMTMCTL register. The fast mode of the sum threshold monitor can be enabled by the STMFM bit in HPDF_FLTSUMTMCTL register. The characteristics in both cases are as follows:

Table 33-9. Features of sum threshold monitor working mode

Mode	Enable Bit	Channel Data Source	Analog Input Data Source	Input Data Resolution	Detailed Description
Standard mode	STMFM=0	Serial data stream, Parallel data	HPDF final data output	24bit	The final data is only available after the channel conversion. Slow response time, not suitable for overcurrent/overvoltage detection
Fast mode	STMFM=1	Serial data stream	Serial data stream	16bit	The input data is provided in continuous mode, regardless of regular or inserted conversion. Fast response time, suitable for overcurrent/overvoltage detection.

Note: When using the sum threshold monitor, it is necessary to maintain synchronization of the two conversion results.

For STMFM = 0, to ensure that the conversions of the selected main filters are completed synchronously, it is necessary to ensure that the two parallel (or serial) data inputs are

synchronized after the two conversions are started (if parallel input is used, Dual mode (DPM[1:0] = 2) can be used to achieve this);

For STMFM = 1, to ensure that the conversions of the selected channel filters are completed synchronously, it is necessary to ensure that the serial data input is synchronized after the two channels are enabled.

Sum threshold monitor configuration

The configuration requirements for the sum threshold monitor are as follows:

1. The filter order and oversampling rate of the data sources for addition A and B, as well as the integrator sampling rate, must be configured consistently.
2. When STMFM = 0, addition A and B cannot both be started by general software simultaneously; they can only be started using trigger/software synchronization. When using software synchronization, either addition A or addition B must select HPDF_FLT0 data.

Sum threshold monitor flag

The global state of the sum threshold monitor is the STMEOF flag in HPDF_FLTSUMTMSTAT register. When STMEOF=1, it indicates that sum threshold monitor event has occurred, that is, an event that exceeds the (upper/lower limit) threshold is generated. If the sum threshold monitor event interrupt STMIE=1 in HPDF_FLTSUMTMCTL register is enabled, an sum threshold monitor interrupt can be generated. When SHTF and SLTF are cleared, the STMEOF bit is cleared.

The HPDF_FLTSUMTMSTAT register defines the error event flag of the channel exceeding the threshold. The SHTF bit indicates whether the maximum sum threshold SHTVAL[23:0] has been exceeded. The SLTF bit indicates whether the minimum sum threshold SLTVAL[23:0] value has been exceeded. Clear the sum threshold event flag by writing "1" to the corresponding HTFC or LTFC bit in the HPDF_FLTSUMTMFC register.

As is shown in ***Table 33-2. HPDF internal signal***, there are 4 break output signals in the HPDF module. The break output signals are assigned to sum threshold monitor threshold event by setting the SHTBSD[3:0] and SLTBSD[3:0] bit fields in the HPDF_FLTSUMTMHT register and the HPDF_FLTSUMTMLT register.

33.3.12. 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.

33.3.13. 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.

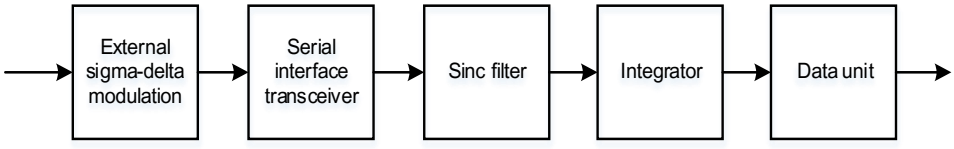
If the sampling final output data word is higher than the value in the maximum value register of the extremes monitor (MAXVAL[23:0] bits in the HPDF_FLTyEMMAX register), the value of this register is updated to the current final output data. If the sampling final output data word is smaller than the 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.

33.3.14. 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 33-7. 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 33-10. 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, $f_{DATA}<=f_{HPDFCLK}$ must be satisfied.

Signed data format

Signed data in HPDF module: parallel data register, regular and inserted group data register, threshold monitor value, 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): $(SFOR^{SFO})\times IOR <= 2^{31}$.
2. When using FastSinc filter: $2\times(SFOR^2)\times IOR <= 2^{31}$.

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.

33.3.15. HPDF interrupt

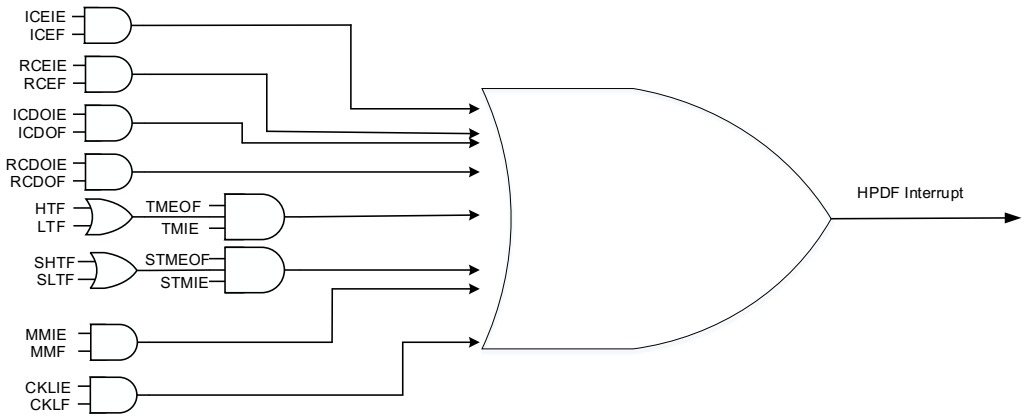
HPDF interrupt events can be divided into channel conversion interrupt events, threshold monitor interrupt events, malfunction monitor interrupt events, and channel clock loss interrupt events. The specific interrupt event description is as *Table 33-11. HPDF interrupt event*.

Table 33-11. 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
STMEOF SHTF SLTF	sum threshold monitor events	Write 1 to SHTFC bit Write 1 to SLTFC bit	STMIE
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 33-8. HPDF interrupt logic diagram* shown.

Figure 33-8. HPDF interrupt logic diagram



33.4. Register definition

HPDF base address: 0x4001 7000

33.4.1. 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

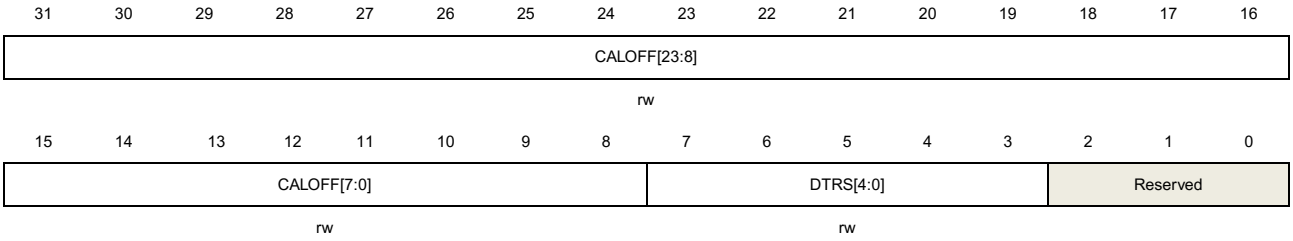
		00: External CKINx input is selected for SPI clock source, sampling point is determined by SITYP[1:0] 01: Internal CKOUT output is selected for SPI clock source, sampling point is determined by SITYP[1:0] 10: Internal CKOUT is selected for SPI clock source, sampling point on each second CKOUT falling 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).



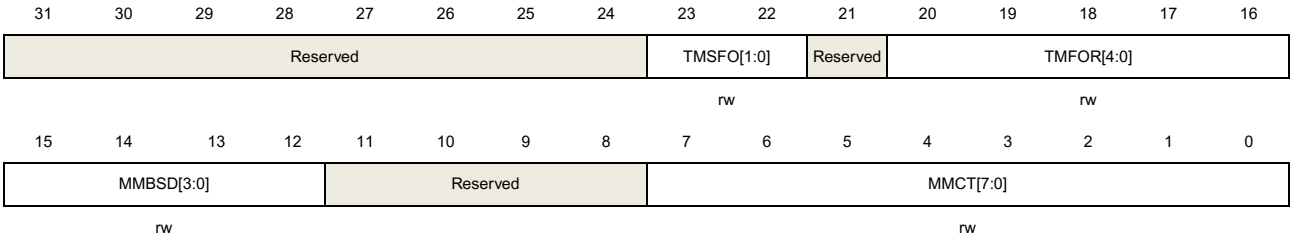
Bits	Fields	Descriptions
31:8	CALOFF[23:0]	24-bit calibration offset Calibration offset must be performed for each conversion result of the channel. These bits can be set by software.
7:3	DTRS[4:0]	Data right bit-shift 0-31: The number of bits that determine the right shift of data Bit-shift is performed 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).

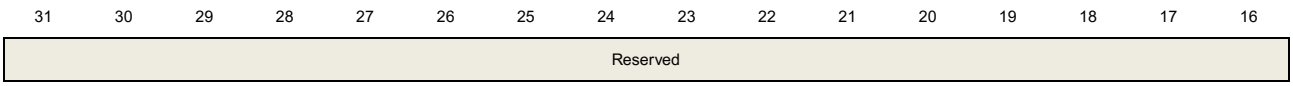


Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:22	TMSFO[1:0]	Threshold monitor Sinc filter order selection 00: FastSinc filter 01: Sinc^1 filter 10: Sinc^2 filter 11: Sinc^3 filter These bits can be configured only when CHEN=0 (in HPDF_CHxCTL register).
21	Reserved	Must be kept 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 0/2/4/6: sample in DATAIN1[15:0] is automatically copied into DATAIN0[15:0] of channel 1. For channel 1/3/5/7: 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 0/2/4/6: Channel x data sample is stored into DATAIN0[15:0].

For channel 1/3/5/7: 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.

33.4.2. 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 ICSYN=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: 0x0100 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				BRKNUM[3:0]				TMCHEN[7:0]							
rw								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
31:28	Reserved	Must be kept at reset value.
27:24	BRKNUM[3:0]	Break signal number configuration The TIMER is triggered after the break signal occurs the configured number of times. 0: Break signal of the HPDF filter y is set to 0.
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.

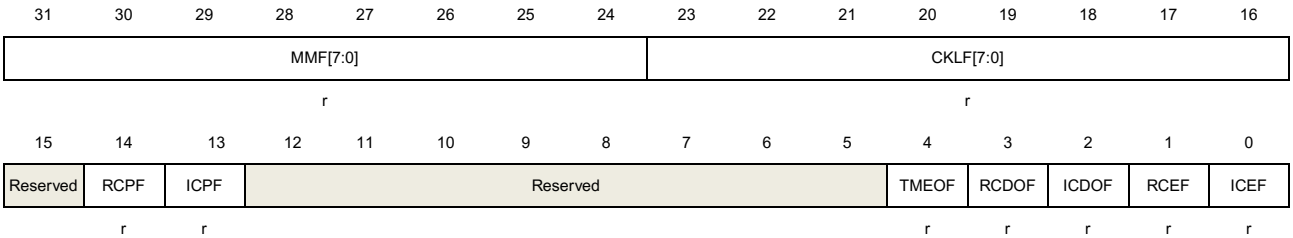
		EMCS[x] = 0: Extremes monitor y does not monitor data from channel x EMCS[x] = 1: Extremes monitor y monitor data from channel x
7	Reserved	Must be kept at reset value
6	CKLIE	Clock loss interrupt enable 0: Detection of channel input clock loss interrupt is disabled 1: Detection of channel input clock 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: 0x00FF 0000

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

All the bits of HPDF_FLTySTAT are automatically reset when FLTEN=0.



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 ICDOF=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'.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MMFC[7:0]								CKLFC[7:0]							
rc_w1								rc_w1							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												RCDOFC	ICDOFC	Reserved	
												rc_w1	rc_w1		

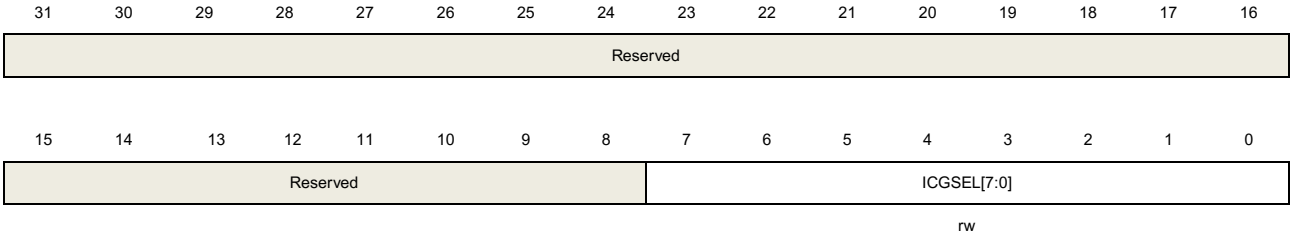
Bits	Fields	Descriptions
31:24	MMFC[7:0]	Clear the malfunction monitor flag

		MMFC[x]=0: No effect MMFC[x]=1: Clear the malfunction monitor flag on channel x This bit is only available in HPDF_FLT0INTC 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_FLT0INTC 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]	Inserted 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.

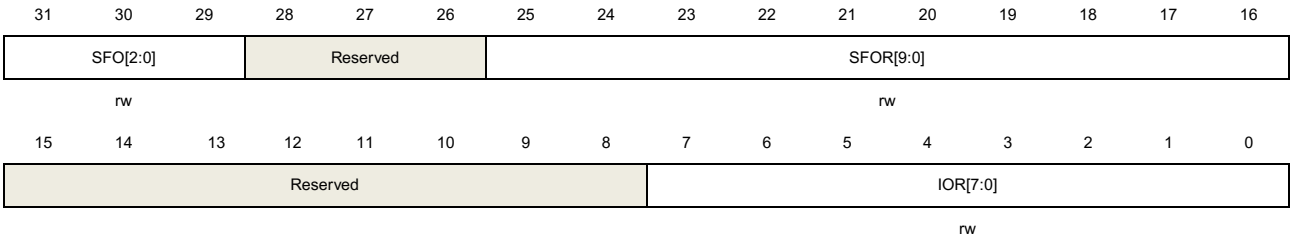
At least one channel must always be selected for the inserted group. All writes that make ICGSEL[7:0]=0 are ignored.

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.

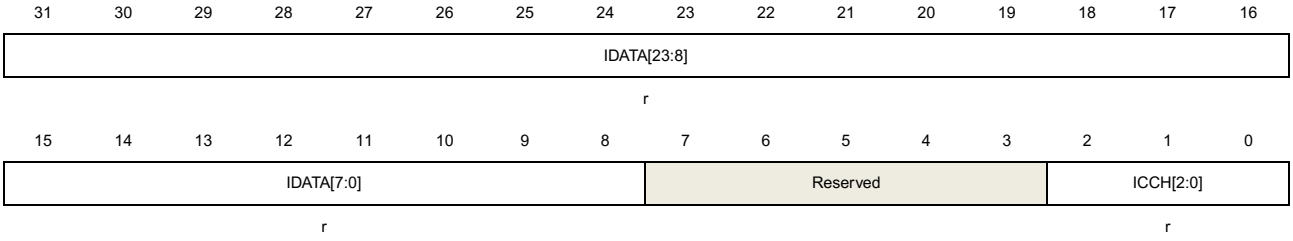
Filter y inserted group conversion data register (HPDF_FLTyIDATA)

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.



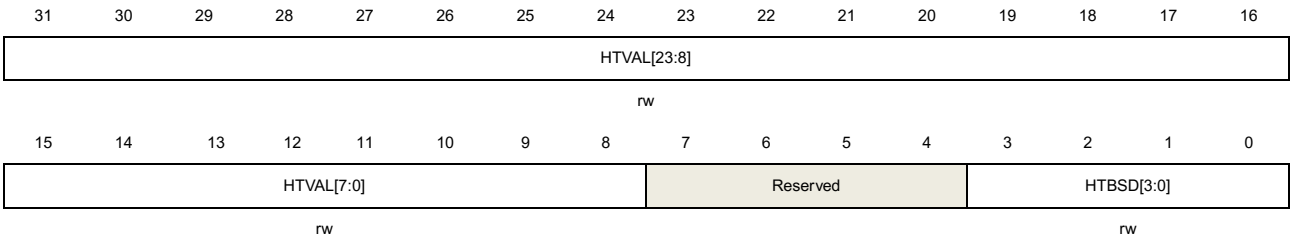
Bits	Fields	Descriptions
31:8	RDATA[23:0]	Regular channel conversion data When each regular conversion finishes, its data is stored in these bits. The data is

		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).

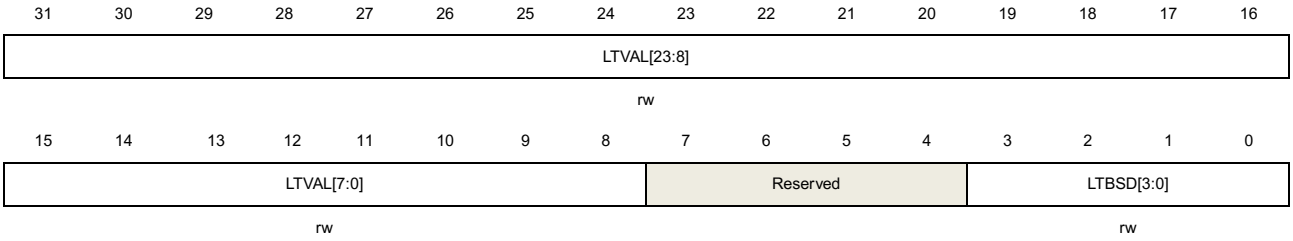


Bits	Fields	Descriptions
31:8	HTVAL[23:0]	Threshold monitor high threshold value These bits are written by software to determine the high threshold for the threshold monitor. If TMFM=1, the higher 16 bits determine the 16-bit threshold as compared with the threshold monitor filter output. Bits HTVAL[7:0] 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 x 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).



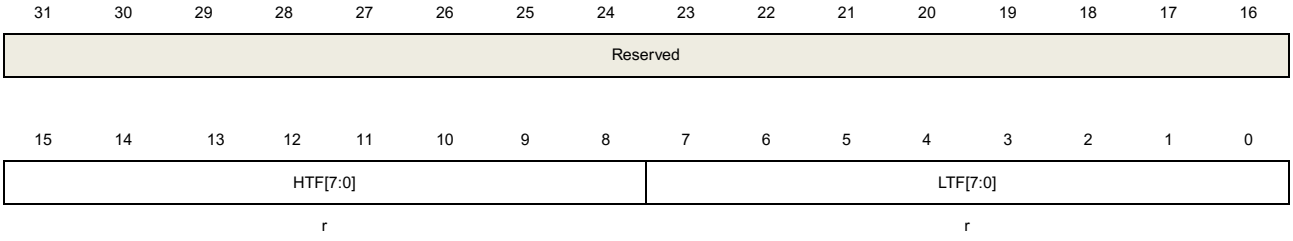
Bits	Fields	Descriptions
31:8	LTVAL[23:0]	Threshold monitor low threshold value. These bits are written by software to determine the low threshold for the threshold monitor. If TMFM=1, the higher 16 bits determine the 16-bit threshold as compared with the threshold monitor filter output. Bits LTVAL[7:0] 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]=0: No low threshold error on channel x

LTF[x]=1: Low threshold error on channel x

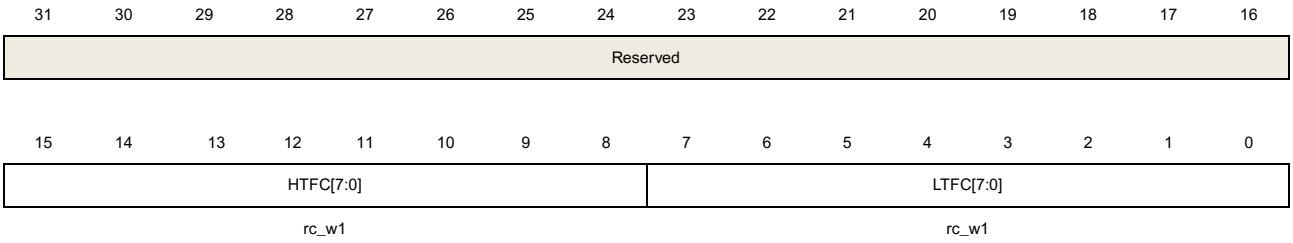
It is set by hardware. It can be cleared by software setting the corresponding LTFC[7:0] bit in the HPDF_FLTyTMFC register.

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).



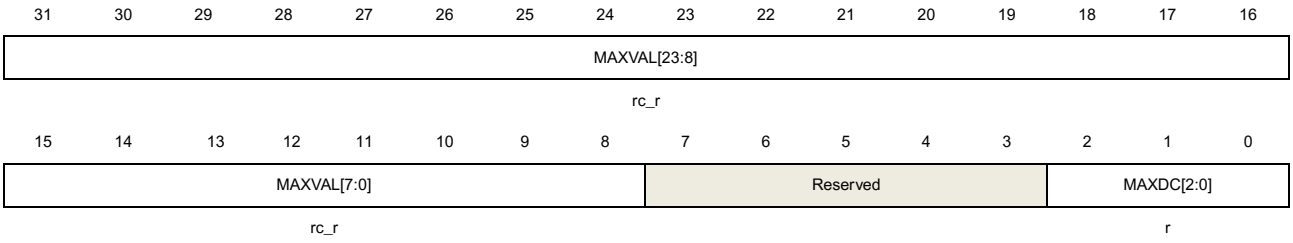
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:8	HTFC[7:0]	Clear the threshold monitor high threshold flag HTFC[x]=0: No effect HTFC[x]=1: Clear the threshold monitor high threshold flag on channel x
7:0	LTFC[7:0]	Clear the threshold monitor low threshold flag LTFC[x]=0: No effect LTFC[x]=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).



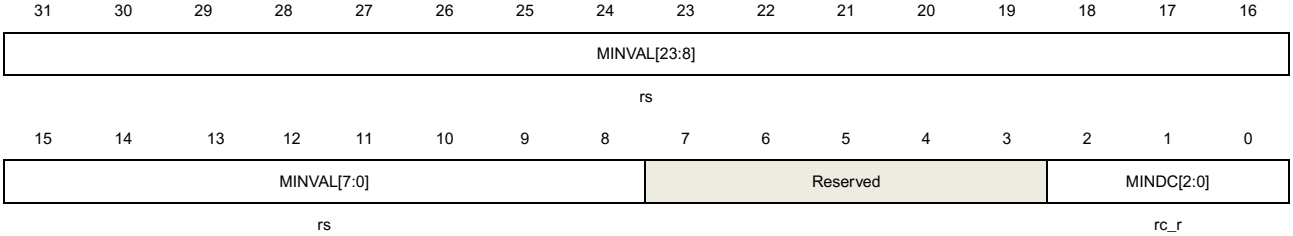
Bits	Fields	Descriptions
31:8	MAXVAL[23:0]	Extremes monitor maximum value These bits are set by hardware and indicate the highest value of channel converted by HPDF_FLTy.

		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).

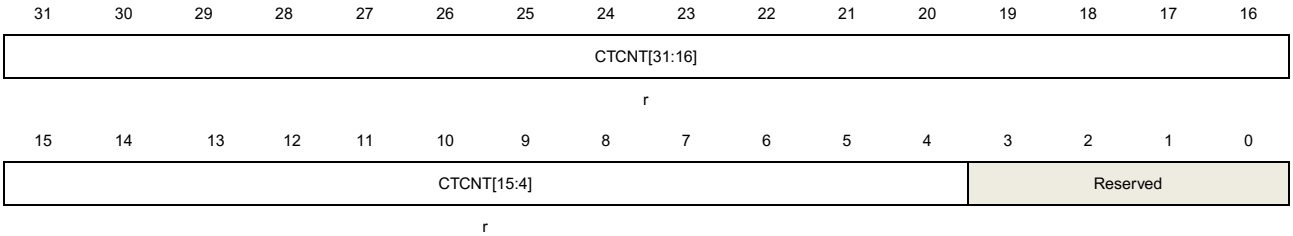


Bits	Fields	Descriptions
31:8	MINVAL[23:0]	Extremes monitor minimum value These bits are set by hardware and indicate the lowest value converted by HPDF_FLTy. These bits can be reset by reading of this register.
7:3	Reserved	Must be kept at reset value.
2:0	MINDC[2:0]	Extremes monitor minimum data channel 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 = CTCNT[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 CTCNT[27:0] = 0 only when the filter is bypassed (SFOR[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 + FORD) + 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] / f_{CKIN}.
		in case if SFOR = SFOR [9:0]+1 = 1: CTCNT = 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).

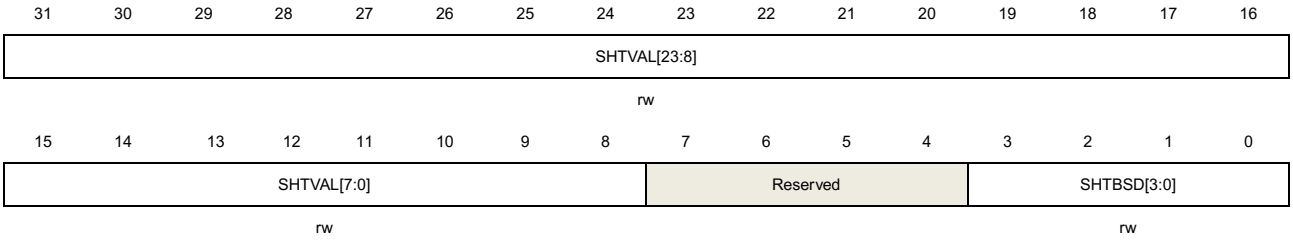
3:0	Reserved	Must be kept at reset value.

Filter sum threshold monitor high threshold register (HPDF_FLTSUMTMHT)

Address offset: 0x13C

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:8	SHTVAL[23:0]	Filter sum threshold monitor high threshold value These bits are written by software to determine the high threshold for the threshold monitor that the sum of the two filters. If STMFM=1, the higher 16 bits determine the 16-bit threshold as compared with the threshold monitor filter output. Bits SHTVAL[7:0] are ignored.
7:4	Reserved	Must be kept at reset value.
3:0	SHTBSD[3:0]	High sum threshold event break signal distribution SHTBSD[i] = 0: Break signal is not distributed to high threshold event

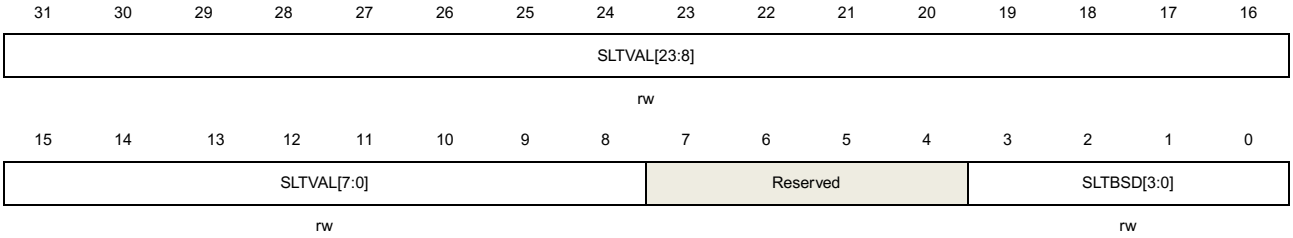
SHTBSD[i] = 1: Break signal x is distributed to high threshold event

Filter sum threshold monitor low threshold register (HPDF_FLTSUMTMLT)

Address offset: 0x140

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:8	SLTVAL[23:0]	Filter sum threshold monitor low threshold value These bits are written by software to determine the low threshold for the threshold monitor that the sum of the two filters. If STMFM=1, the higher 16 bits determine the 16-bit threshold as compared with the threshold monitor filter output. Bits SLTVAL[7:0] are ignored.
7:4	Reserved	Must be kept at reset value.
3:0	SLTBSD[3:0]	Low sum threshold event break signal distribution SLTBSD[i] = 0: Break signal is not distributed to low threshold event SLTBSD[i] = 1: Break signal x is distributed to low threshold event

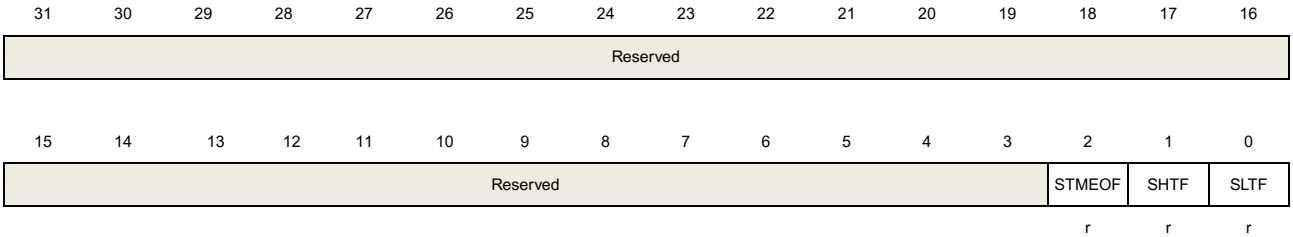
Filter sum threshold monitor status register (HPDF_FLTSUMTMSTAT)

Address offset: 0x144

Reset value: 0x0000 0000

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

Note: All the bits of HPDF_FLTSUMTMSTAT are automatically reset when FLTEN =0



Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value.
2	STMEOF	Sum threshold monitor event occurred flag

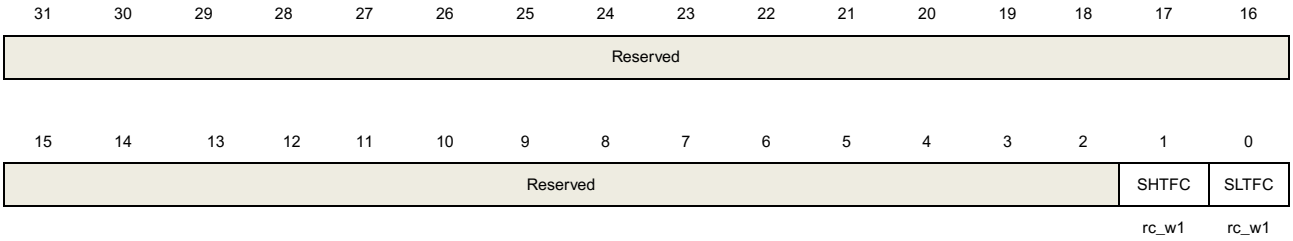
		0: No sum threshold monitor event occurred. 1: Sum threshold monitor event occurred which detected data crosses the threshold This bit is set by hardware. It is cleared by clearing SHTF and SLTF in HPDF_FLTSUMTMSTAT register (by writing '1' into the clear bits in HPDF_FLTSUMTMFC register).
1	SHTF	Sum threshold monitor high threshold flag SHTF=1: No high threshold error SHTF=1: high threshold error It is set by hardware. It can be cleared by software setting the corresponding SHTFC bit in the HPDF_FLTSUMTMFC register.
0	SLTF	Sum threshold monitor low threshold flag SLTF=1: No low threshold error SLTF=1: Low threshold error It is set by hardware. It can be cleared by software setting the corresponding SLTFC bit in the HPDF_FLTSUMTMFC register.

Filter sum threshold monitor flag clear register (HPDF_FLTSUMTMFC)

Address offset: 0x148

Reset value: 0x0000 0000

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



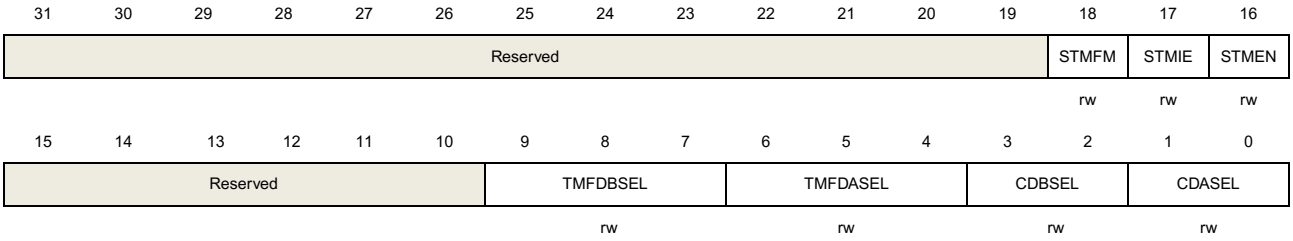
Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	SHTFC	Clear the sum threshold monitor high threshold flag SHTFC=0: No effect SHTFC=1: Clear the sum threshold monitor high threshold flag (SHTF bit in the HPDF_FLTSUMTMSTAT)
0	SLTFC	Clear the sum threshold monitor low threshold flag SLTFC=0: No effect SLTFC=1: Clear the sum threshold monitor low threshold flag (SLTF bit in the HPDF_FLTSUMTMSTAT)

Filter sum threshold monitor control register (HPDF_FLTSUMTMCTL)

Address offset: 0x14C

Reset value: 0x0000 0084

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



Bits	Fields	Descriptions
31:19	Reserved	Must be kept at reset value.
18	STMFM	Sum threshold monitor fast mode 0: Sum threshold monitor watch on the final data after performing offset correction and right shift 1: Sum threshold monitor watch on serial input data stream
17	STMIE	Sum threshold monitor interrupt enable 0: Sum threshold monitor interrupt is disabled 1: Sum threshold monitor interrupt is enabled
16	STMEN	Sum threshold monitor enable 0: Sum threshold monitor is disabled 1: Sum threshold monitor is enabled
15:10	Reserved	Must be kept at reset value.
9:7	TMFDBSEL	The data for the input phase B of the addition is selected from the serial data stream 000: addition B is threshold monitor filter0 data 001: addition B is threshold monitor filter1 data 010: addition B is threshold monitor filter2 data 011: addition B is threshold monitor filter3 data 100: addition B is threshold monitor filter4 data 101: addition B is threshold monitor filter5 data 110: addition B is threshold monitor filter6 data 111: addition B is threshold monitor filter7 data
6:4	TMFDASEL	The data for the input phase A of the addition is selected from the serial data stream 000: addition A is threshold monitor filter0 data 001: addition A is threshold monitor filter1 data 010: addition A is threshold monitor filter2 data 011: addition A is threshold monitor filter3 data 100: addition A is threshold monitor filter4 data 101: addition A is threshold monitor filter5 data 110: addition A is threshold monitor filter6 data

		111: addition A is threshold monitor filter7 data
3:2	CDBSEL	The data for the input phase B of the addition is selected from the final output data 00: addition B is HPDF data0 01: addition B is HPDF data1 10: addition B is HPDF data2 11: addition B is HPDF data3
1:0	CDASEL	The data for the input phase A of the addition is selected from the final output data 00: addition A is HPDF data0 01: addition A is HPDF data1 10: addition A is HPDF data2 11: addition A is HPDF data3

34. Trigonometric Math Unit (TMU)

34.1. 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.

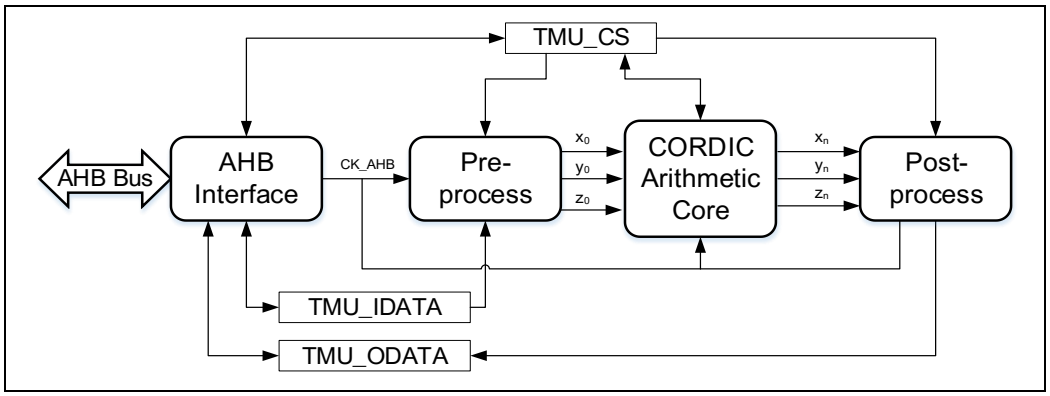
34.2. Characteristics

- 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.

34.3. Block diagram

Figure 34-1. TMU block diagram provides details of the internal configuration of the TMU.

Figure 34-1. 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.

34.4. Function overview

34.4.1. 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 34-3 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 34-1. 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 34-1. 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 34-2. 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 34-2. 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

34.4.2. 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 34-3 IEEE754 32-bit single precision floating-point format***.

Table 34-3 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.

34.4.3. 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 34-4. TMU mode configuration***.

Table 34-4. 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 34-5. Mode 0 description, when IFLTEN = 1 and OFLTEN = 1***.

Table 34-5. 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 34-6. Mode 0 description, when IFLTEN = 0 or OFLTEN = 0***.

Table 34-6. 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 $-\pi$ 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.
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.
3. 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.
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 34-7. Mode 1 description, when IFLTEN = 1 and OFLTEN = 1***.

Table 34-7. 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 34-8. Mode 1 description, when IFLTEN = 0 or OFLTEN = 0***.

Table 34-8. 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 $-\pi$ 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* 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 $100 * \sin(\frac{\pi}{2})$. 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 1368pprox.1368e. 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.
4. 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: $y_1 = \frac{100}{128} * \sin(\frac{\pi}{2})$, and reading the TMU_ODATA register again can get the second output data:$y_2 = \frac{100}{128} * \cos(\frac{\pi}{2})$.
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: $100 * \sin(\frac{\pi}{2})$ =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 $\mathrm{atan2}(y, x)$ 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 34-9. Mode 2 description, when IFLTEN = 1 and OFLTEN = 1***.

Table 34-9. 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 34-10. Mode 2***

description, when IFLTEN = 0 or OFLTEN = 0.

Table 34-10. 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 $[-\pi, \pi]$. 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 $\sqrt{x^2+y^2}$>=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 $\theta = \text{atan}(5,80)$. 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 34-11. Mode 3 description, when IFLTEN = 1 and OFLTEN = 1*.

Table 34-11. 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 34-12. Mode 3 description, when IFLTEN = 0 or OFLTEN = 0***.

Table 34-12. 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 $[-\pi,\pi]$. 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 34-13. Mode 4 description, when IFLTEN = 1 and OFLTEN = 1***.

Table 34-13. 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 34-14. Mode 4 description, when IFLTEN = 0 or OFLTEN = 0***.

Table 34-14. Mode 4 description, when IFLTEN = 0 or OFLTEN = 0

Parameter	Range	Description
Input data	$\frac{x}{2^f}\in$[-1,1]	If $x\in$[-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 $-1<=\frac{x}{2^f}<1$. 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 $[-\pi,\pi]$. The output data is multiplied by π and 2^f to get the real angle value θ.
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 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 34-15. Mode 5 description, when IFLTEN = 1 and OFLTEN = 1***.

Table 34-15. Mode 5 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	$\cosh(x) \in [1,1.692]$	--
Second output data	$\sinh(x) \in [-1.366,1.366]$	--
FACTOR[2:0]	Not available	keep at reset value 3'b000

When IFLTEN = 0 or OFLTEN = 0, the detailed information refer to ***Table 34-16. Mode 5 description, when IFLTEN = 0 or OFLTEN = 0.***

Table 34-16. Mode 5 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, q1.15 or float format.
First 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.
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 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 34-17. Mode 6 description, when IFLTEN = 1 and OFLTEN = 1***.

Table 34-17. Mode 6 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	sinh (x) ∈[-1.366,1.366]	--
Second output data	cosh (x) ∈[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 34-18. Mode 6 description, when IFLTEN = 0 or OFLTEN = 0.***

Table 34-18. Mode 6 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	$\frac{\sinh(x)}{2}$∈[-0.683,0.683]	The output data is multiplied 2 to get the real hyperbolic sine of a hyperbolic angle x.
Second output data	$\frac{\cosh(x)}{2}$∈[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 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 34-19. Mode 7 description, when IFLTEN = 1 and OFLTEN = 1***.

Table 34-19. Mode 7 description, when IFLTEN = 1 and OFLTEN = 1

Parameter	Range	Description
Input data	x∈[-0.806,0.806]	Overflow when exceed the range.
Output data	$\tanh^{-1}(x)$ ∈[-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 34-20. Mode 7 description, when IFLTEN = 0 or OFLTEN = 0***.

Table 34-20. Mode 7 description, when IFLTEN = 0 or OFLTEN = 0

Parameter	Range	Description
Input data	$\frac{x}{2}$∈[-0.403,0.403]	x∈[-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, q1.15 or float format.
Output data	$\frac{\tanh^{-1}(x)}{2}$∈[-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 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 34-21. Mode 8 description, when IFLTEN = 1 and OFLTEN = 1***.

Table 34-21. Mode 8 description, when IFLTEN = 1 and OFLTEN = 1

Parameter	Range	Description
Input data	x∈[0.107,9.35]	Overflow when exceed the range.
Output data	ln (x) ∈(-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 34-22. Mode 8 description, when IFLTEN = 0 or OFLTEN = 0.***

Table 34-22. Mode 8 description, when IFLTEN = 0 or OFLTEN = 0

Parameter	Range	Description
Input data	$\frac{x}{2^f} \in [0.0535,0.875]$	x ∈ [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.
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∈[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 34-23. Recommended scaling factors in mode 8* for different inputs.

Table 34-23. 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 34-24. Mode 9 description, when IFLTEN = 1 and OFLTEN = 1*.

Table 34-24. 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 34-25. Mode 9 description, when IFLTEN = 0 or OFLTEN = 0*.

Table 34-25. Mode 9 description, when IFLTEN = 0 or OFLTEN = 0

Parameter	Range	Description
Input data	$\frac{x}{2^f} \in [0.027,0.875]$	$x \in [0.027,2.34]$, 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]	$f \in [0,2]$	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 34-26. Recommended scaling factors in mode 9* for different inputs

Table 34-26. 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]

34.4.4. 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.

34.4.5. 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.

34.4.6. 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.

34.5. Registers definition

TMU base address: 0x4802 4400

34.5.1. 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]			Reserved				MODE[3:0]			
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 lower half-word of TMU_IDATA register contains the first output data, and 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 written 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.

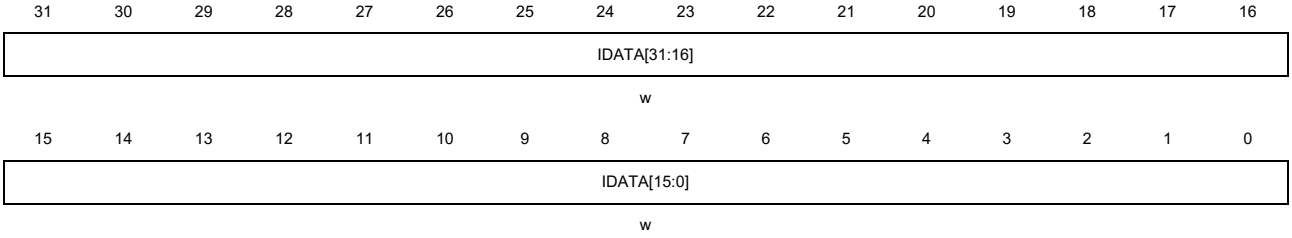
		1: enabled. When ENDF is set, the interrupt request is generated.
15	IFLTEN	Input floating point format enable 0: Input data support fixed point signed integer format q1.15 or q1.31 1: Input data support 32-bit single precision floating point format
14	OFLTEN	Output floating point format enable 0: Output 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	Reserved	Must be kept at reset value.
3:0	MODE[3:0]	Mode of TMU operation 0000: mode0,$m * \cos(\theta)$ 0001: mode1, $m * \sin(\theta)$ 0010: mode2, $\text{phase} = \text{atan2}(y,x)$ 0011: mode3, $\text{modulus} = \sqrt{x^2+y^2}$ 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

34.5.2. Input data register (TMU_IDATA)

Address offset: 0x04
Reset value: 0xXXXX XXXX

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

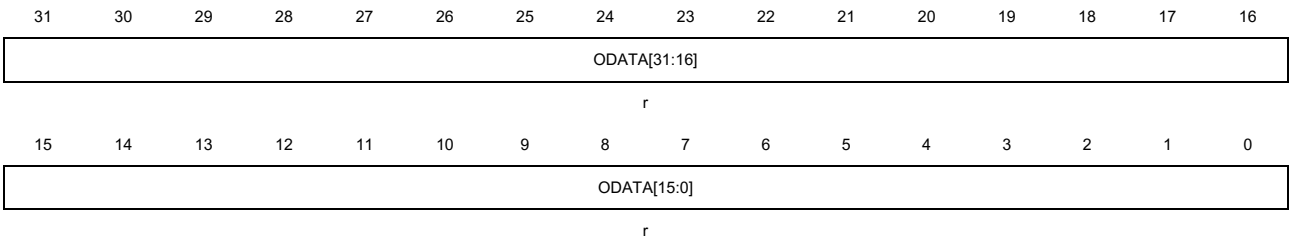


Bits	Fields	Descriptions
31:0	IDATA[31:0]	The input data The input data is written into this register. For details, refer to *Table 34-1. Input data configuration*. **Note:** 1. When no TMU operation is ongoing and the required number of arguments has been written, a new operation will be started automatically. 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.

34.5.3. Output data register (TMU_ODATA)

Address offset: 0x08
Reset value: 0x0000 0000

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



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 34-2. 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.

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

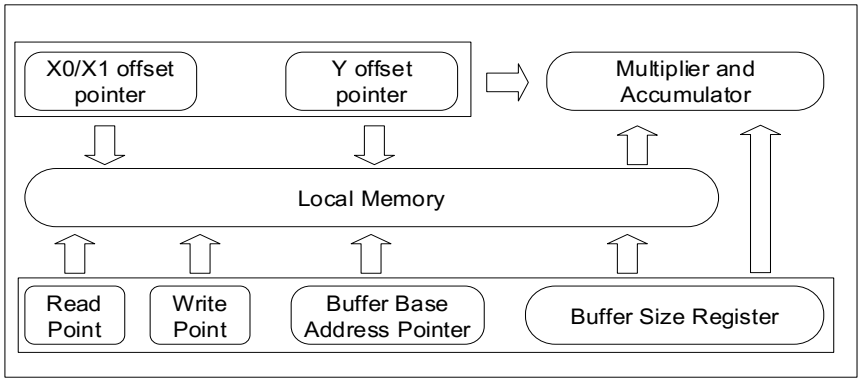
- Fixed or float multiplier and accumulator.
- 256 x 32-bit local memory.
- 16-bit fixed-point or 32-bit float point input and output.
- Up to three buffers, two input buffers and one output buffer.
- Buffer can be circular.
- FIR and IIR can be realized.
- Vector functions support convolution, Dot product, 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 float point multiplier and accumulator. Two 16-bit input signed data or 32-bit input float data 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 calculates 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.

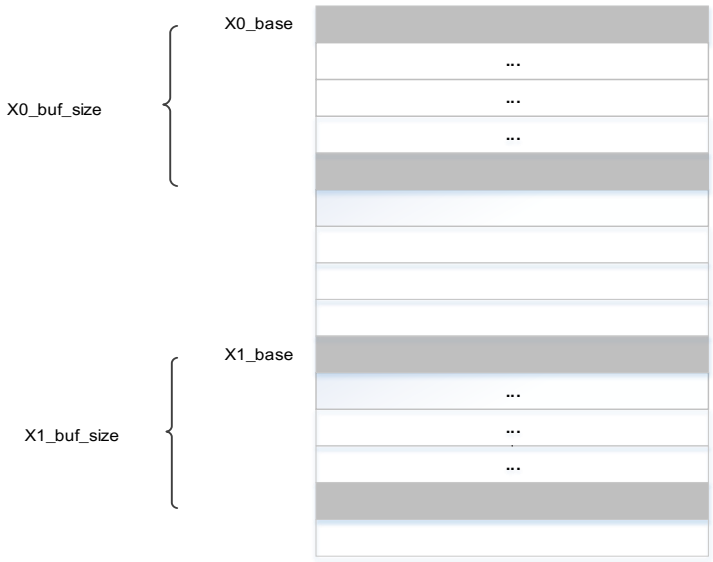
An optional headroom is added to the buffer size if circular buffer operation is required, Moreover, for regulating the DMA or CPU activity, it is necessary to set watermark level. The headroom value and watermark level should be selected according to the actual application.

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

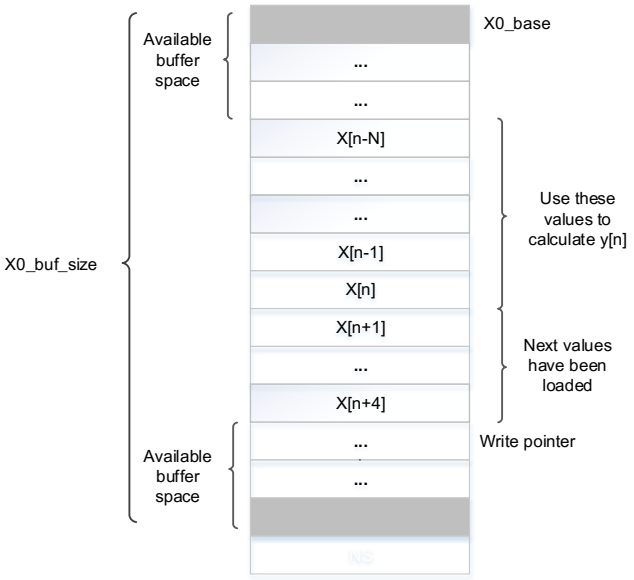
X0 buffer and X1 buffer store the input data of MAC, every multiplication operation multiplies two data together, one data from X0 buffer and the other from X1 buffer. The read address offset is generated by the pointer of the control unit, which is relative to the buffer base address.

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.

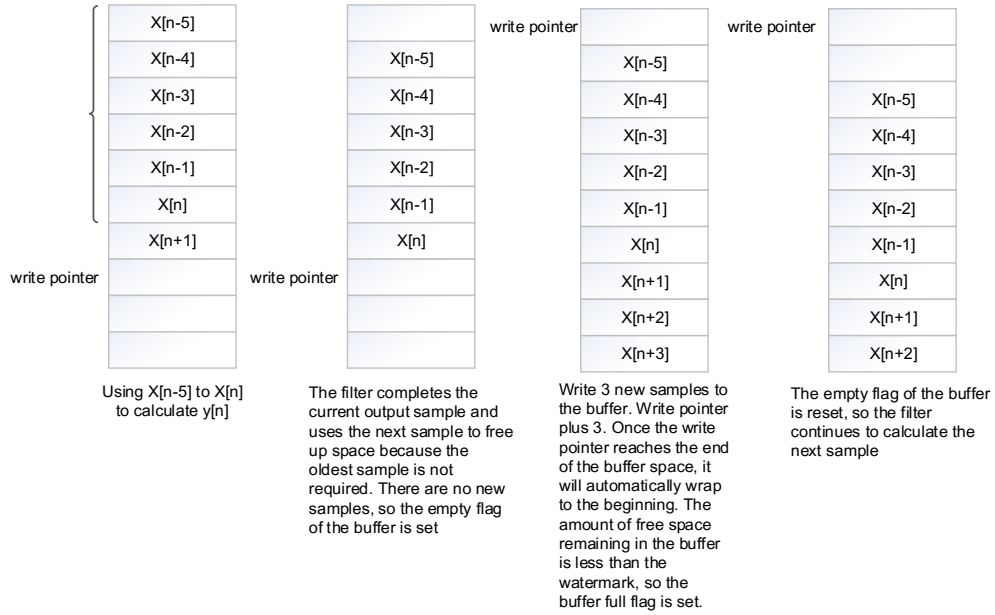
For a circular buffer, the allocated space of the buffer (x0_buf_size) should be greater than element numbers in use, therefore, new values are always available in the buffer. ***Figure 35-3 Circular input buffer area*** show the buffer layout for a filter operation. For calculating an output sample 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 bit field 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

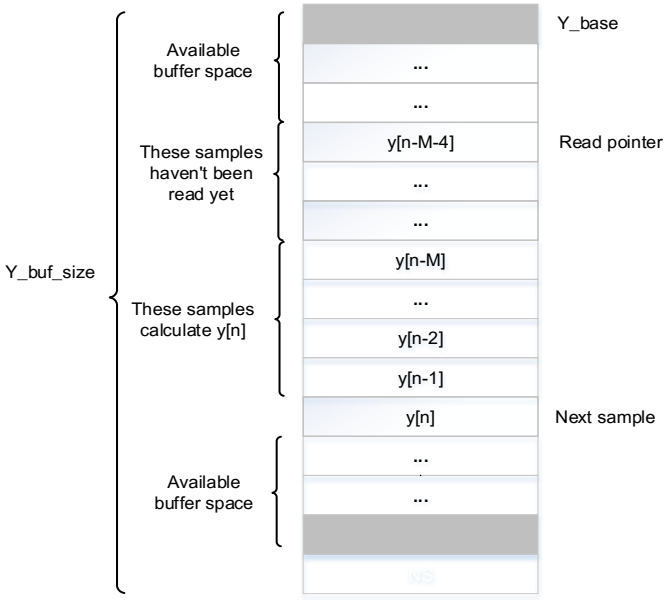


35.3.4. Output buffer

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



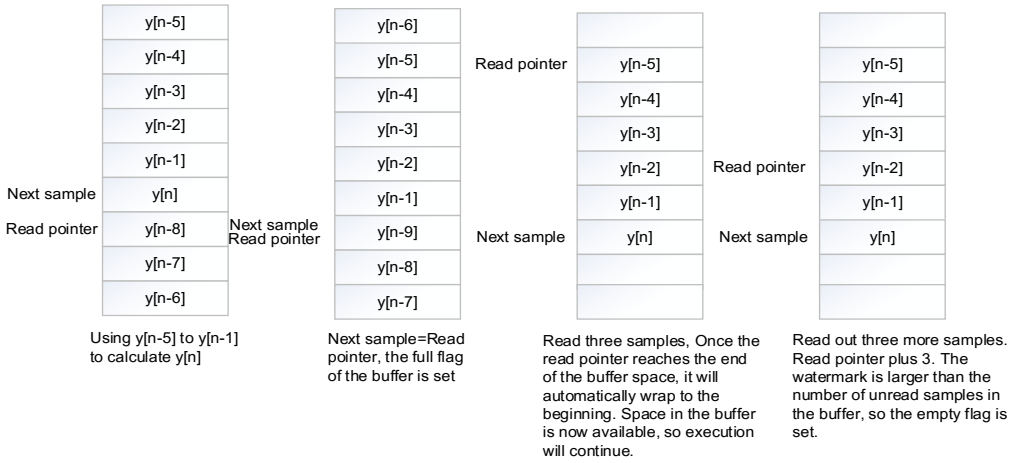
The Y buffer could work in circular buffer mode. The buffer full flag is set if the address for next output data is equal to the address which is indicated by the read pointer, and operation is stalled until the output data is read.

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 bit field 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 bit field 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 bit field 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.

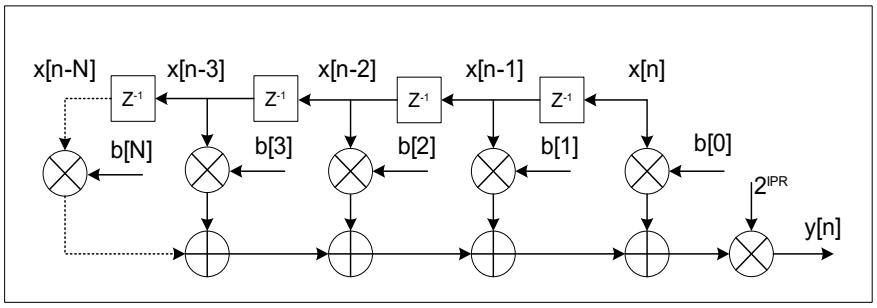
FIR filter: $\underline{Y} = \underline{B} * \underline{X}$

$$y_n = 2^{IPR} \sum_{k=0}^{N} (b_k \times x_{n-k}) \quad (35\text{-}1)$$

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 filter function***.

Figure 35-7 The structure of FIR filter 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.

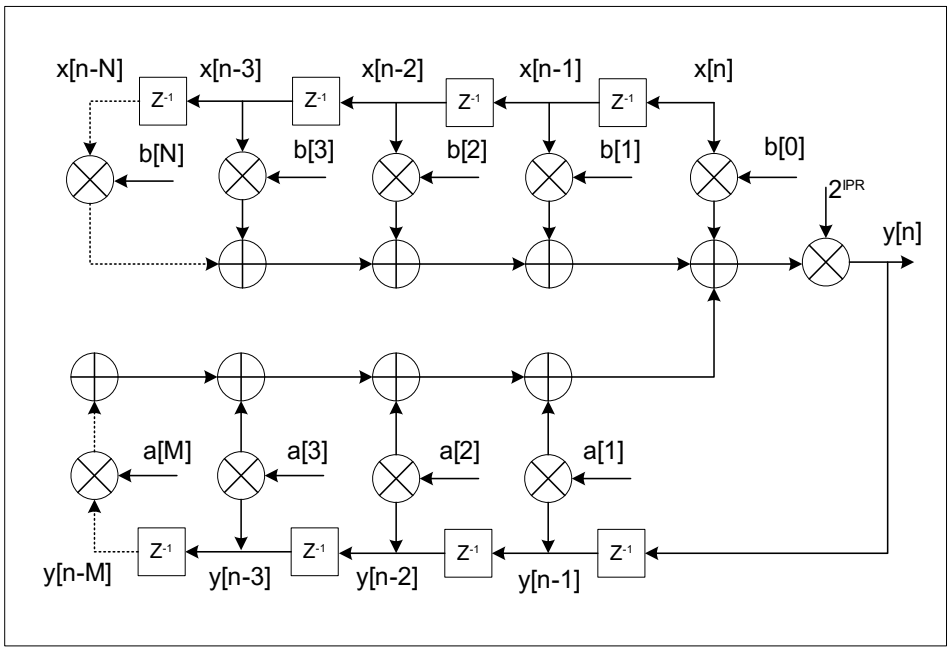
IIR filter: Y = B*X+A*Y

$$y_n = 2^{IPR}\left(\sum_{k=0}^{N}(x_{n-k} \times b_k) + \sum_{k=1}^{M}\left(y_{n-k} \times a_k\right)\right) \quad (35\text{-}2)$$

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 filter.***

Figure 35-8 The structure of IIR filter



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 bit 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 negative 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 32 bit 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.

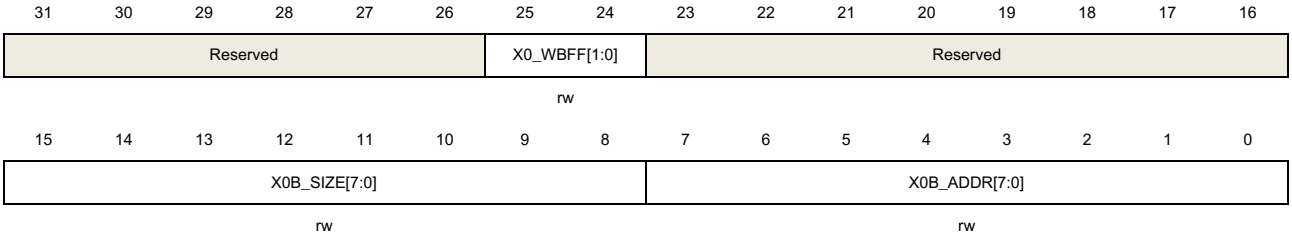
35.4. Register definition

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.

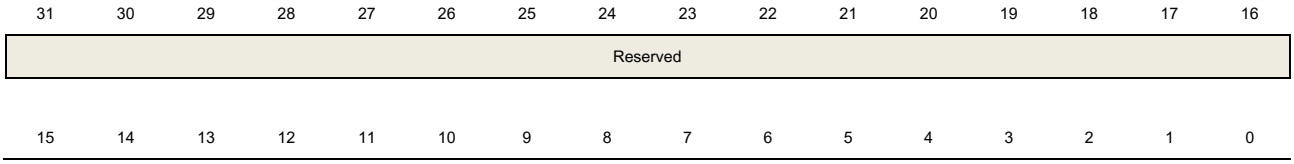


Bits	Fields	Descriptions
31:26	Reserved	Must be kept at reset value.
25:24	X0_WBFF[1:0]	Buffer full flag of watermark. If free spaces number in the buffer is less than 2^{X0_WBFF}, the flag is set. 00: Threshold = 1(if DMA write requests are enabled) 01: Threshold = 2 10: Threshold = 4 11: Threshold = 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)



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 bit field can not be changed.
7:0	X1B_ADDR[7:0]	X1 buffer base address When FAC is running (EXE = 1), this bit field 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

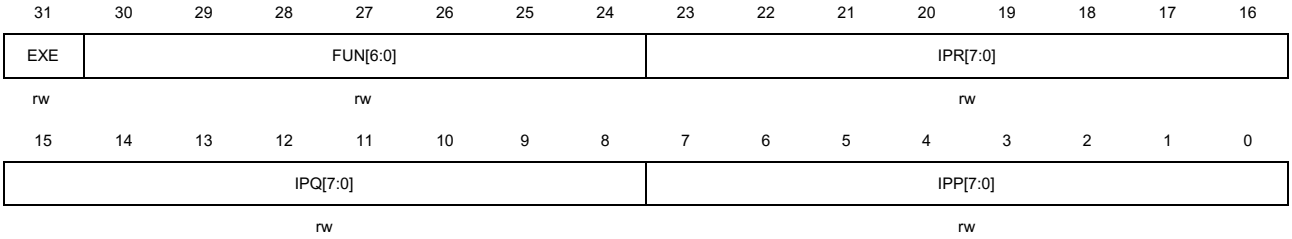
		taps for IIR filters.
7:0	YB_ADDR[7:0]	Y buffer base address

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 bit field, 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, FAC_STAT register and FAC_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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDATA[15:0]															
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. Image processing accelerator (IPA)

36.1. Overview

The IPA provides a configurable and flexible image format conversion from one or two source image to the destination image, with the following four conversion modes:

- Copy one source image to the destination image
- Convert one source image to the destination image with specific pixel format
- Convert and blend two source images to the destination image with specific pixel format
- Fill up the destination image with a specific color

18 pixel formats for foreground from 4-bit up to 32-bit per pixel, 11 pixel formats for background from 4-bit up to 32-bit per pixel, and 7 pixel formats from 16-bit up to 32-bit per pixel for the destination image are supported. Two 256*32 bits LUTs (Look-Up Table) separately for the two source images are implemented for the indirect pixel formats.

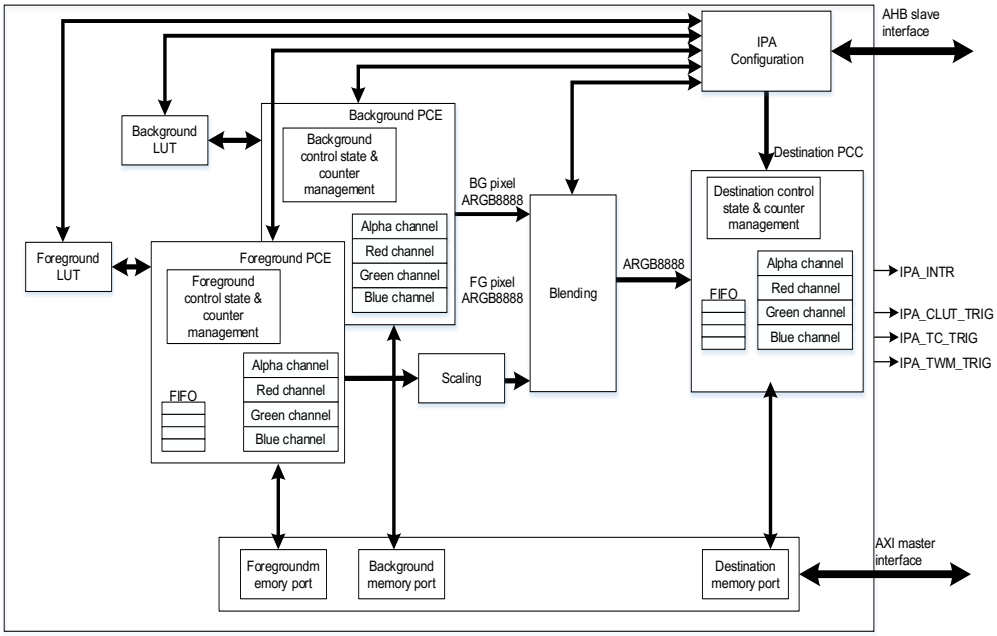
36.2. Characteristics

- One AXI master interface for memory access
- One AHB slave interface for IPA configuration with 8-bit, 16-bit, 32-bit
- Three four-doubleword depth 64-bit FIFOs independently for the source and destination images
- Support four pixel-format-convert modes
  - Copy one source image to the destination image
  - Convert one source image to the destination image with specific pixel format
  - Convert and blend two source images to the destination image with specific pixel format
  - Fill up the destination image with a specific color
- Support configurable LUT size independently for two source images
- Support two LUT pixel formats separately for two source images
- Support LUT automatically loading for two source images
- Support transfer hang up and stop
- Support pixel offset per line independently for the source and destination images
- Support pre-defined pixel channel value independently for the source and destination images
- Support three alpha channel value calculation algorithms separately for two source images
- Support 18 pixel formats for foreground image
- Support 11 pixel formats for background image
- Support 7 pixel formats for the destination image
- Support configurable image size

- Support automatically AXI bandwidth adjustment with an internal timer
- Support one interrupt with six types of event flags
- Support interrupt enable and clear
- Support decimal scaling and bilinear scaling
- Support image rotation (0, 90, 180, 270 degrees)
- Support interlace input for the foreground image

36.3. Block diagram

Figure 36-1. IPA block diagram



As showed in ***Figure 36-1. IPA block diagram***, the IPA consists of 7 main parts:

- IPA configuration through AHB slave interface
- Image data access through AXI master interface
- Foreground and background LUT
- Foreground and background pixel channel extension (PCE)
- Foreground scaling block
- Foreground and background pixel blending
- Destination pixel channel compression (PCC)

36.4. Signal description

The signals of IPA are listed in ***Table 36-1. Signals description of IPA***.

Table 36-1. Signals description of IPA

I/O port	Type	Description
IPA_INTR	O	IPA global interrupt signal
IPA_CLUT_TRIG	O	IPA CLUT transfer complete trigger signal
IPA_TC_TRIG	O	IPA transfer complete trigger signal
IPA_TWM_TRIG	O	IPA transfer watermark trigger signal

36.5. Function overview

The IPA is a pixel format converter, supporting multiple conversion modes, foreground pixel formats and line offset, background pixel formats and line offset, destination pixel formats and line offset to allow for flexible application by configuring the corresponding bits in the IPA registers. All the IPA registers (expect for LUT accesses only 32-bit supported) can be 8-bit, 16-bit and 32-bit configured through AHB slave interface.

Four conversion modes are supported, which is determined by the PFCM bits in the IPA_CTL register, as listed in the *Table 36-2. IPA conversion mode*.

- Copy foreground image to the destination image

  In this mode, the pixel data in the foreground memory are copied to the destination memory without pixel conversion. So the configured pixel format of the foreground and destination images have no specific meaning. The foreground pixel format only defines the bit number per pixel.

- Convert foreground image to the destination image

  In this mode, the pixel data in the foreground memory are converted from the foreground pixel format to the destination pixel format, and then written into the destination memory. If the foreground pixel format is indirect (L8, AL44, AL88, L4), the data read from the foreground memory is used as an index to retrieve the pixel data from the foreground LUT.

- Convert and blend the foreground and background images to the destination image

  In this mode, the pixel data in the foreground and background memory are firstly converted from the foreground and background pixel format to 'ARGB8888'. Pairs of foreground and background pixel value are blended and converted from 'ARGB8888' to the destination pixel format, and then written into the destination memory.

  If the foreground pixel format is indirect, the data read from the foreground memory is used as an index to retrieve the pixel data from the foreground LUT.

  If the background pixel format is indirect, the data read from the background memory is used as an index to retrieve the pixel data from the background LUT.

- Fill up the destination image with a specific color

In this mode, the destination image is filled up with the pre-defined pixel channel value, corresponding with the destination pixel format.

Table 36-2. IPA conversion mode

PFCM[1:0]	Conversion mode		Pixel conversion	Blending
	Source	Destination		
00	Foreground image	Destination image	No	No
01	Foreground image	Destination image	Yes	No
10	Foreground and background image	Destination image	Yes	Yes
11	Pixel value pre-defined in the register	Destination image	No	No

36.5.1. Conversion operation

An IPA transaction consists of seven operations:

1) Read pixel data from the foreground memory addressed through the IPA_FMADDR. Retrieve the pixel data from the foreground LUT if the foreground pixel format is indirect.
2) Extend the foreground pixel value to a 32-bit value, and calculate the alpha channel value according to the FAVCA bits in the IPA_FPCTL register.
3) Read pixel data from the background memory addressed through the IPA_BMADDR. Retrieve the pixel data from the background LUT if the background pixel format is indirect.
4) Extend the background pixel value to a 32-bit value, and calculate the alpha channel value according to the BAVCA bits in the IPA_BPCTL register.
5) Blend the processed foreground and background pixel data.
6) Compress the pixel data into the value with the destination pixel format determined by the DPF bits in the IPA_DPCTL register.
7) Write the converted pixel data into the destination memory addressed through the IPA_DMADDR.

Three eight-doubleword depth 64-bit FIFOs are implemented for the foreground, background and four-doubleword depth 64-bit FIFO is for the destination pixel data processing. The foreground and background FIFO are buffers to store the data reading from the corresponding source memory and the destination FIFO is pushed with the processed pixel data which is ready to write into the destination memory when the AXI bus is idle.

If the PFCM bits in the IPA_CTL register is configured to '00' or '01' to copy or convert foreground image to the destination image, only the foreground FIFO and destination FIFO are activated. If the IPA operates to fill up the destination image with the specific color, none of these three FIFOs is activated.

36.5.2. Foreground and background LUT

Two LUTs are implemented in the IPA to store the pixel value for the usage of the indirect pixel format. The pixel value must be written into the LUT before the IPA transfer is enabled when the pixel format is indirect. The pixel value in the LUT can be updated in two ways:

- Automatically loading:

  Enable the FLLEN/BLLEN bit in the IPA_FPCTL/IPA_BPCTL register. The FCNP or BCNP bits in the IPA_FPCTL or IPA_BPCTL register define the number of pixels to be loaded, which is equal to FCNP+ 1 or BCNP + 1.

- Software program:

  The pixel data is written into the corresponding memory address through the IPA AHB slave interface. The base address offset of foreground LUT is 0x0400, and the base address offset of background LUT is 0x0800.

Two pixel formats are supported for the LUTs, including 'ARGB8888' and 'RGB888', which is determined by the FLPF or BLPF bit in the IPA_FPCTL or IPA_BPCTL register, as listed in the *Table 36-3. Foreground and background CLUT pixel format*.

Table 36-3. Foreground and background CLUT pixel format

BLPF/FLPF	LUT pixel format	Memory address			
		base + 0x3	base + 0x2	base + 0x1	base + 0x0
0	ARGB8888	A_0[7:0]	R_0[7:0]	G_0[7:0]	B_0[7:0]
1	RGB888	R_3[7:0]	G_3[7:0]	B_3[7:0]	R_2[7:0]
		G_2[7:0]	B_2[7:0]	R_1[7:0]	G_1[7:0]
		B_1[7:0]	R_0[7:0]	G_0[7:0]	B_0[7:0]

Note: If the pixel format is 'RGB888', the alpha value is fixed to 0xFF when updating the pixel data in the LUT.

36.5.3. Foreground and background pixel channel extension (PCE)

In the IPA pixel-format-convert mode with pixel conversion, the foreground (and background) pixel values are extended from the foreground or background pixel format to the 'ARGB8888' format.

The FPF and BPF bits in the IPA_FPCTL and IPA_BPCTL register determine the pixel format of the foreground and background image, as listed in the *Table 36-4. Foreground and background pixel format*.

A pixel consists of five channels:

- Alpha channel: opacity, 0x00: transparent; 0xFF: opaque.
- Red channel: redness, 0x00 No red, 0xFF: fully red.
- Green channel: greenness, 0x00 No green, 0xFF: fully green.

- Blue channel: blueness, 0x00 No blue, 0xFF: fully blue.
- Luminance channel: In the IPA, the value of the luminance channel is used as an index to retrieve the pixel data from the foreground or background LUT.

Table 36-4. Foreground and background pixel format

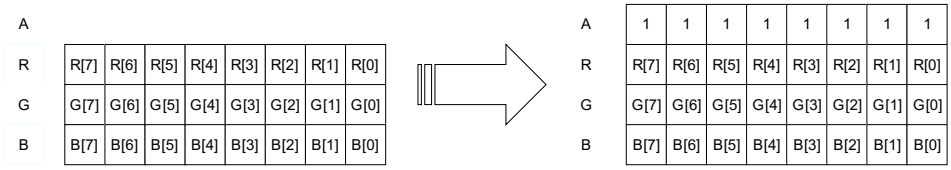
BPF[3:0]/ FPF[3:0]	YUV_SWAP	Pixel format	Memory address			
			base + 0x3	base + 0x2	base + 0x1	base + 0x0
0000	-	ARGB8888	A_0[7:0]	R_0[7:0]	G_0[7:0]	B_0[7:0]
0001	-	RGB888	R_3[7:0]	G_3[7:0]	B_3[7:0]	R_2[7:0]
			G_2[7:0]	B_2[7:0]	R_1[7:0]	G_1[7:0]
			B_1[7:0]	R_0[7:0]	G_0[7:0]	B_0[7:0]
0010	-	RGB565	R_1[4:0]G_1[5:3]	G_1[2:0]B_1[4:0]	R_0[4:0]G_0[5:3]	G_0[2:0]B_0[4:0]
0011	-	ARGB1555	A_1[0]R_1[4:0]G_1[4:3]	G_1[2:0]B_1[4:0]	A_0[0]R_0[4:0]G_0[4:3]	G_0[2:0]B_0[4:0]
0100	-	ARGB4444	A_1[3:0]R_1[3:0]	G_1[3:0]B_1[3:0]	A_0[3:0]R_0[3:0]	G_0[3:0]B_0[3:0]
0101	-	L8	L_3[7:0]	L_2[7:0]	L_1[7:0]	L_0[7:0]
0110	-	AL44	A_3[3:0]L_3[3:0]	A_2[3:0]L_2[3:0]	A_1[3:0]L_1[3:0]	A_0[3:0]L_0[3:0]
0111	-	AL88	A_1[7:0]	L_1[7:0]	A_0[7:0]	L_0[7:0]
1000	-	L4	L_7[3:0]L_6[3:0]	L_5[3:0]L_4[3:0]	L_3[3:0]L_2[3:0]	L_1[3:0]L_0[3:0]
1001	-	A8	A_3[7:0]	A_2[7:0]	A_1[7:0]	A_0[7:0]
1010	-	A4	A_7[3:0]A_6[3:0]	A_5[3:0]A_4[3:0]	A_3[3:0]A_2[3:0]	A_1[3:0]A_0[3:0]
1011	-	YUV444*	Y_3[7:0]	U_3[7:0]	V_3[7:0]	Y_2[7:0]
			U_2[7:0]	V_2[7:0]	Y_1[7:0]	U_1[7:0]
			V_1[7:0]	Y_0[7:0]	U_0[7:0]	V_0[7:0]
1101	1	YUYV422*	V_{01}[7:0]	Y_1[7:0]	U_{01}[7:0]	Y_0[7:0]
1100	1	YVYU422*	U_{01}[7:0]	Y_1[7:0]	V_{01}[7:0]	Y_0[7:0]
1100	0	UYVY422*	Y_1[7:0]	V_{01}[7:0]	Y_0[7:0]	U_{01}[7:0]
1101	0	VYUY422*	Y_1[7:0]	U_{01}[7:0]	Y_0[7:0]	V_{01}[7:0]
1110	-	YUV420*	…	…	…	…
			V_{2367}[7:0]	U_{2367}[7:0]	V_{0145}[7:0]	U_{0145}[7:0]
			…	…	…	…
			Y_7[7:0]	Y_6[7:0]	Y_5[7:0]	Y_4[7:0]@line2
			Y_3[7:0]	Y_2[7:0]	Y_1[7:0]	Y_0[7:0]@line1
1111	-	YVU420*	…	…	…	…
			U_{2367}[7:0]	V_{2367}[7:0]	U_{0145}[7:0]	V_{0145}[7:0]
			…	…	…	…
			Y_7[7:0]	Y_6[7:0]	Y_5[7:0]	Y_4[7:0]@line2
			Y_3[7:0]	Y_2[7:0]	Y_1[7:0]	Y_0[7:0]@line1

Note*: YUV format are only supported in foreground image.

If the pixel format is 'RGB888', the alpha channel value is equal to 0xFF when extending the pixel data, as shown in *Figure 36-2. Pixel extension from 'RGB888' to 'ARGB8888'*.

Figure 36-2. Pixel extension from 'RGB888' to 'ARGB8888'

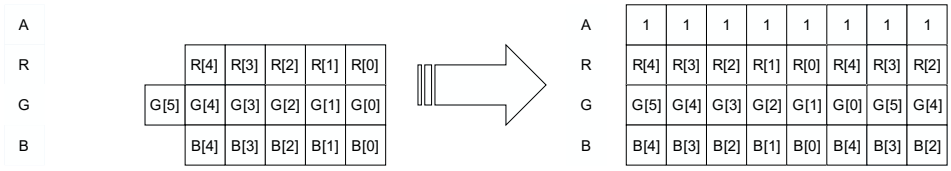
§ RGB888 à ARGB8888



If the pixel format is 'RGB565', the alpha channel value is equal to 0xFF when extending the pixel data. The red, green and blue channel value is extended to 8-bit by setting the MSBs to the original bits and copying the MSBs to the LSBs, as shown in ***Figure 36-3. Pixel extension from 'RGB565' to 'ARGB8888'***.

Figure 36-3. Pixel extension from 'RGB565' to 'ARGB8888'

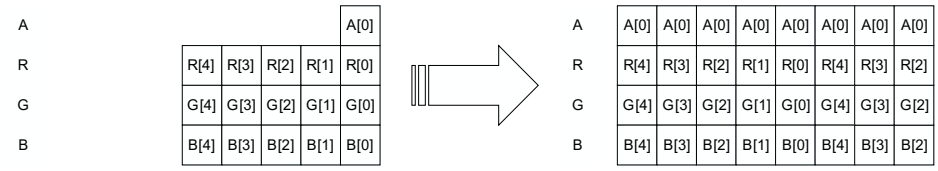
§ RGB565 à ARGB8888



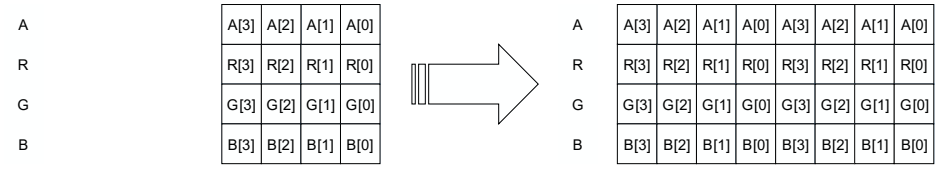
If the pixel format is 'ARGB1555' or 'ARGB4444', The value of every channel is extended to 8-bit by setting the MSBs to the original bits and copying the MSBs to the LSBs, as shown in the ***Figure 36-4. Pixel extension from 'ARGB1555' or 'ARGB4444' to 'ARGB8888'***.

Figure 36-4. Pixel extension from 'ARGB1555' or 'ARGB4444' to 'ARGB8888'

§ ARGB1555 à ARGB8888



§ ARGB4444 à ARGB8888



If the pixel format is 'L8' or 'L4', the pixel data is retrieved from the LUT with the 8-bit luminance channel value (MSBs filled with '0' when 'L4').

If the pixel format is 'AL44', only the red, green and blue channel values are retrieved from the LUT with the 8-bit luminance channel value (MSBs filled with '0'). And the alpha channel value is extended to 8-bit by setting the MSBs to the original bits and copying the MSBs to the LSBs.

If the pixel format is 'AL88', only the red, green and blue channel values are retrieved from the LUT with the 8-bit luminance channel value.

If the pixel format is 'A8', the red, green and blue channel values are separately equal to the FPDRV or BPDRV bits, FPDGV or BPDGV bits and FPDBV or BPDBV bits in the IPA_FPV or IPA_BPV register.

If the pixel format is 'A4', the alpha channel value is extended to 8-bit by setting the MSBs to the original bits and copying the MSBs to the LSBs. The red, green and blue channel values are separately equal to the FPDRV or BPDRV bits, FPDGV or BPDGV bits and FPDBV or BPDBV bits in the IPA_FPV or IPA_BPV register.

The PCE also support color conversion from YUV/YcbCr to ARGB8888 format. The following equations are used to perform this process. The constants will be stored in the IPA_CSCC_CFGx (x=0…2) control registers as two's compliment values, which allow flexibility in the implementation and differences in the video encode and decode operations. In addition, it also provides a software mechanism to manipulate brightness or contrast.

$$R = C0(Y+Yoffset) + C1(V+Uvoffset) \quad (36\text{-}1)$$

$$G = C0(Y+Yoffset) + C3(U+Uvoffset) + C2(V+Uvoffset) \quad (36\text{-}2)$$

$$B = C0(Y+Yoffset) + C4(U+Uvoffset) \quad (36\text{-}3)$$

Table 36-5. Expected coefficients for YUV and YcbCr modes indicates the expected coefficients for YUV and YcbCr modes.

Table 36-5. Expected coefficients for YUV and YcbCr modes

Coff	YUV	YcbCr
YOFF	0x000	0x1F0 (-16)
UVOFF	0x000	0x180 (-128)
C0	0x100 (1.00)	0x129 (1.164)
C1	0x123 (1.140)	0x198 (1.596)
C2	0x76B (-0.581)	0x730 (-0.813)
C3	0x79C (-0.394)	0x79C (-0.392)
C4	0x208 (2.032)	0x204 (2.017)

Three algorithms are supported to modulate the alpha channel value, which is determined by the FAVCA or BAVCA bits in the IPA_FPCTL or IPA_BPCTL register, as described in *Table 36-6. Alpha channel value modulation*.

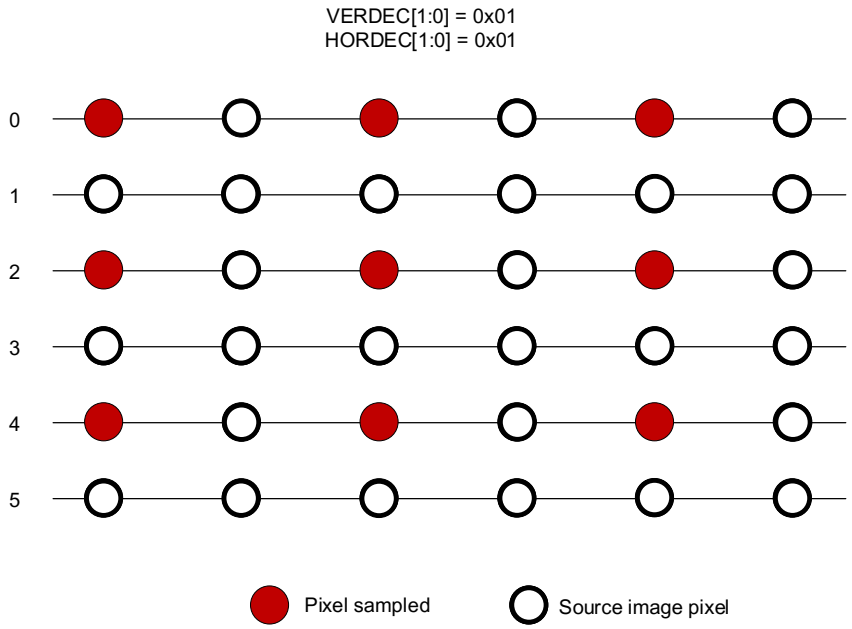
Table 36-6. Alpha channel value modulation

FAVCA[1:0]/BAVCA[1:0]	Alpha calculation algorithm
00/11	No effect, equal to the original value
01	Equal to the FPDAV or BPDAV bits in the IPA_FPCTL or IPA_BPCTL register
10	Equal to the FPDAV or BPDAV bits multiplied by the original alpha value and divided by 255

36.5.4. Foreground channel scaling

IPA implements a bilinear scaling filter to adjust the input image to different resolutions in foreground channel. The HORDEC and VERDEC in IPA_DPCTL register define the horizontal and vertical pre decimation filter control. The XSCALE and YSCALE in IPA_BSCTL register provide the X and Y scaling factor for the Foreground. The maximum down scaling factors of decimation filter and bilinear filter are 8 and 2 respectively, so that the maximum scaling factor can be up to 16. The detailed sampling method of the decimation filter is shown in ***Figure 36-5. Sampling method of decimation filter***.

Figure 36-5. Sampling method of decimation filter

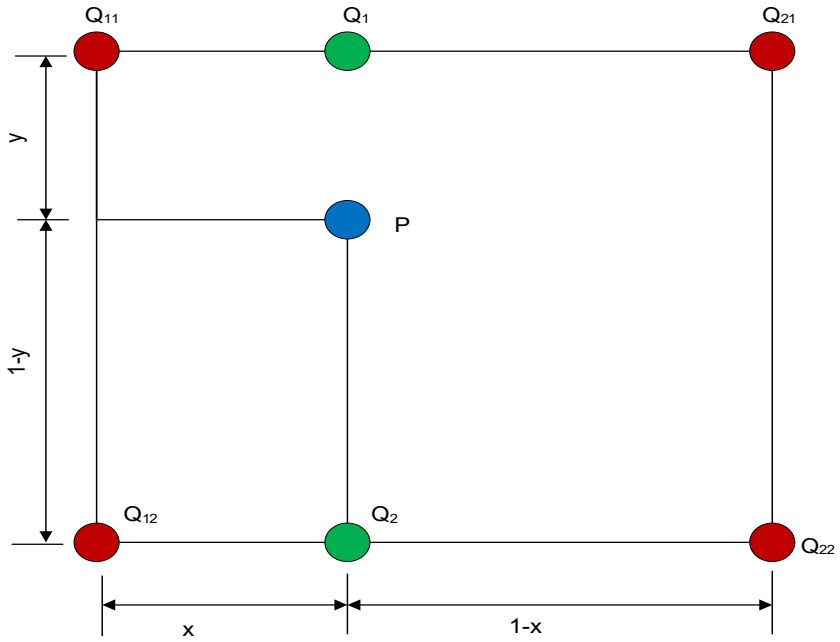


The maximum down scaling factor is 1/2 and the maximum up scaling factor is 2^12 for each axis. In order to realize the scaling function, the reciprocal of the scale factor should be written to IPA_BSCTL register. For example:

```
REG32(IPA_BSCTL) = 0x20002000;      /* 1/2x scaling (0x2.0) */
REG32(IPA_BSCTL) = 0x18001800;      /* 2/3x scaling (0x1.8) */
REG32(IPA_BSCTL) = 0x08000800;      /* 2x scaling (0x0.8) */
REG32(IPA_BSCTL) = 0x04000400;      /* 4x scaling (0x0.4) */
```

The bilinear filter is the weighted average of the four nearest pixels. Suppose P is the output pixel and Q_{11}, Q_{12}, Q_{21}, Q_{22} are the surrounding four source image pixels. As is shown in ***Figure 36-6. Bilinear Scaling diagram***.

Figure 36-6. Bilinear Scaling diagram



The value of output pixel P is computed by the equation below.

$$P = Q_1(1-y) + Q_2{}^*y \tag{36-4}$$

Where the equations for Q_1 and Q_2 are as follows:

$$Q_1 = Q_{11}{}^*(1-x) + Q_{21}{}^*x \tag{36-5}$$

$$Q_2 = Q_{12}{}^*(1-x) + Q_{22}{}^*x \tag{36-6}$$

The Scaling are only effective when PFCM eques to 2'b01 or 2'b10.

36.5.5. Blending

When the IPA operates to convert and blend the foreground and background images to the destination image, the foreground and background pixel data after extending are blended by pair to get a 32-bit pixel value.

The alpha channel value is blended on the base of the following equations (A_F is the foreground alpha value, A_B is the background alpha value):

$$A_{mix} = \frac{A_F \times A_B}{255} \tag{36-7}$$

$$A_{blend} = A_F + A_B - A_{mix} \tag{36-8}$$

The red, green and blue channel value are blended on the base of the following equations (R_F, G_F, B_F is the foreground red, green and blue value; R_B, G_B, B_B is the background red, green and blue value):

$$R_{blend} = \frac{R_F \times A_F + R_B \times A_B - R_B \times A_{mix}}{A_{blend}} \tag{36-9}$$

$$G_{blend} = \frac{G_F \times A_F + G_B \times A_B - G_B \times A_{mix}}{A_{blend}} \quad (36\text{-}10)$$

$$B_{blend} = \frac{B_F \times A_F + B_B \times A_B - B_B \times A_{mix}}{A_{blend}} \quad (36\text{-}11)$$

Note: 1) The quotient of the division is rounded down to the nearest integer.
2) If the A_{blend} is equal to zero, the R_{blend}, G_{blend} and B_{blend} is equal to '0xFF'.
3) The width and height of the background and the destination shall be consistent.

36.5.6. Destination pixel channel compression (PCC)

In the IPA pixel-format-convert mode with pixel conversion, the pixel data need to be compressed from the 'ARGB8888' format into the destination pixel format before they are written into the destination memory.

The DPF bits in the IPA_DPCTL register determine the pixel format of the destination image, as listed in ***Table 36-7. Destination pixel format***.

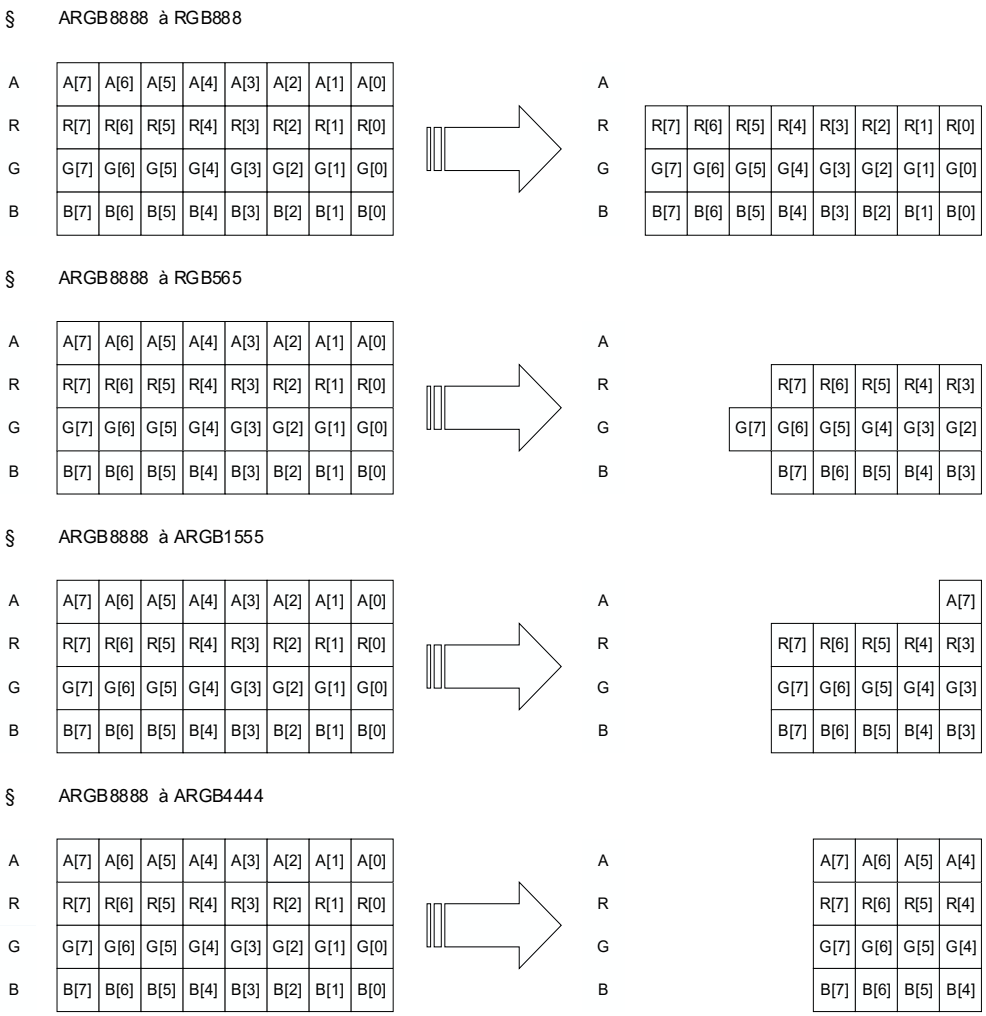
Table 36-7. Destination pixel format

DPF[2:0]	Pixel format	Memory address			
		base + 0x3	base + 0x2	base + 0x1	base + 0x0
000	ARGB8888	A_0[7:0]	R_0[7:0]	G_0[7:0]	B_0[7:0]
001	RGB888	R_3[7:0]	G_3[7:0]	B_3[7:0]	R_2[7:0]
		G_2[7:0]	B_2[7:0]	R_1[7:0]	G_1[7:0]
		B_1[7:0]	R_0[7:0]	G_0[7:0]	B_0[7:0]
010	RGB565	R_1[4:0]G_1[5:3]	G_1[2:0]B_1[4:0]	R_0[4:0]G_0[5:3]	G_0[2:0]B_0[4:0]
011	ARGB1555	A_1[0]R_1[4:0]G_1[4:3]	G_1[2:0]B_1[4:0]	A_0[0]R_0[4:0]G_0[4:3]	G_0[2:0]B_0[4:0]
100	ARGB4444	A_1[3:0]R_1[3:0]	G_1[3:0]B_1[3:0]	A_0[3:0]R_0[3:0]	G_0[3:0]B_0[3:0]
101	BGR888	B_3[7:0]	G_3[7:0]	R_3[7:0]	B_2[7:0]
		G_2[7:0]	R_2[7:0]	B_1[7:0]	G_1[7:0]
		R_1[7:0]	B_0[7:0]	G_0[7:0]	R_0[7:0]
110	BGR565	B_1[4:0]G_1[5:3]	G_1[2:0]R_1[4:0]	B_0[4:0]G_0[5:3]	G_0[2:0]R_0[4:0]

Note: If the PFCM bits in the IPA_CTL register are equal to '00' (copy the foreground image to the destination image), the DPF bits have no meaning, and the FPF bits in the IPA_FPCTL register determine the bit number per pixel for both the source and destination.

As shown in ***Figure 36-7. Pixel compression***, the destination compression is performed by discarding the LSBs.

Figure 36-7. Pixel compression



36.5.7. Rotation

The IPA support rotations of 0, 90, 180, and 270 degrees. When ROT bits in IPA_DPCTL are configured, the IPA read input image line by line and put each converted output pixel into the generated output pixel address after rotation.

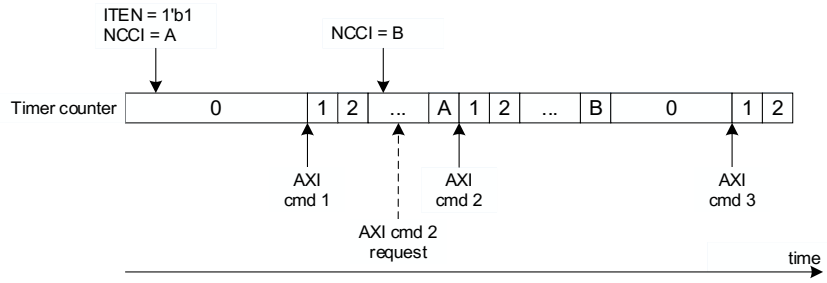
36.5.8. Inter-timer

To reduce the AXI bandwidth usage of IPA AXI master interface, a timer is implemented to insert a number of clocks between two consecutive AXI commands during IPA transmission and LUT automatic loading.

The internal timer is enabled by setting the ITEN bit in the IPA_ITCTL register. The NCCI bits in the IPA_ITCTL register define the minimum number of clock inserted between two consecutive AXI write or read address channel commands, and these bits have no meaning when the timer is disabled.

Updating the NCCI bits when the ITEN bit is enabled have no effect until the current counting is completed, as shown in *Figure 36-8. Inter timer operation*.

Figure 36-8. Inter timer operation



36.5.9. Line mark

The marked line number can be set by configuring the LM bits in the IPA_LM register. As soon as the last pixel data of the line mark has been written into the destination memory, the TLMIF bit in the IPA_INTF register is asserted to detailing the progression of the IPA transfer.

Note: If the LM bits are equal to zero, no line mark flag is asserted during the transmission.

36.5.10. Transfer flow

The foreground/background LUT automatically loading is enabled by setting the FLLEN/BLLEN bit in the IPA_FPCTL/IPA_BPCTL register. Once the loading transfer is launched, the FLLEN/BLLEN bit is used as a transmission flag and writing '0' to the FLLEN/BLLEN bit has no meaning. The FLLEN/BLLEN can be automatically cleared when the loading is finished.

The IPA transfer is enabled by setting the TEN bit in the IPA_CTL register. Once the IPA transfer is launched, the TEN bit is used as a transmission flag and writing '0' to it has no meaning. The TEN bit can be automatically cleared when the IPA transfer is finished.

At any time, the foreground/background LUT automatic loading and IPA transfer can be hanged up by setting the THU bit in the IPA_CTL register. The LUT loading and IPA transfer is paused until the THU bit is cleared by software. When none of the foreground/background LUT automatically loading and IPA transfer is enabled, setting the THU bit has no effect and the THU bit is read as 0.

The foreground/background LUT automatic loading and IPA transfer can be stopped by setting the TST bit in the IPA_CTL register. The LUT loading or IPA transfer is stopped immediately by resetting the FLLEN/BLLEN bit in the IPA_FPCTL/IPA_BPCTL register or the TEN bit in the IPA_CTL register even though the LUT loading or IPA transfer is being hanged up. The TST bit is automatically reset when the current transfer is disabled. When none of the foreground/background LUT automatic loading and IPA transfer is enabled, setting the TST bit has no effect and the TST bit is read as 0.

Only one of the foreground LUT loading, background LUT loading and IPA transfer can be working at a time. For example, when the IPA transfer is ongoing, setting the FLLEN or BLLEN bit has no effect and the FLLEN and BLLEN bit is automatically reset.

36.5.11. Configuration

Before launching any transfers, it is necessary to read the TEN, FLLEN and BLLEN bit to check whether the IPA transfer or the LUT loading is active. If one of them is ongoing, set the TST bit to stop it or wait it finished. When all of the TEN, FLLEN and BLLEN bit are read as 0, starting a new transfer is allowed.

Foreground LUT loading

When starting a new foreground LUT loading, it is recommended to respect the following steps:

1. Configure the IPA_FLMADDR register to set the foreground LUT memory base address.
2. Configure the FLPF bit in the IPA_FPCTL register to set the foreground LUT pixel format.
3. Configure the FCNP bits in the IPA_FPCTL register to set the number of pixel in the foreground LUT to be loaded.
4. Configure the needed enable bit for wrong configuration interrupt, LUT loading finish interrupt, LUT access conflict interrupt and transfer access error interrupt in the IPA_CTL register.
5. Configure the FLLEN bit with '1' in the IPA_FPCTL register to enable the foreground LUT automatically loading.

Background LUT loading

When starting a new background LUT loading, it is recommended to respect the following steps:

1. Configure the IPA_BLMADDR register to set the background LUT memory base address.
2. Configure the BLPF bit in the IPA_BPCTL register to set the background LUT pixel format.
3. Configure the BCNP bits in the IPA_BPCTL register to set the number of pixel in the background LUT to be loaded.
4. Configure the needed enable bit for wrong configuration interrupt, LUT loading finish interrupt, LUT access conflict interrupt and transfer access error interrupt in the IPA_CTL register.
5. Configure the BLLEN bit with '1' in the IPA_BPCTL register to enable the background LUT automatically loading.

IPA transfer

When starting a new IPA transfer, the configuration steps corresponding with the pixel format convert mode are as follows:

Copy the foreground image to the destination image

1. Configure the IPA_FMADDR and IPA_DMADDR register to set the foreground and destination memory base address.
2. Configure the FPF bits in the IPA_FPCTL register to set the foreground pixel format.
3. Configure the FLOFF and DLOFF bits in the IPA_FLOFF and IPA_DLOFF register to set the foreground and destination line offset.
4. Configure the LM bits in the IPA_LM register to set the line mark if needed.
5. Configure the WIDTH and HEIGHT bits in the IPA_IMS register to set the image size.
6. Configure the needed enable bit for wrong configuration interrupt, LUT access conflict interrupt, transfer line mark interrupt, full transfer finish interrupt and transfer access error interrupt in the IPA_CTL register.
7. Configure the TEN bit with '1' in the IPA_CTL register to enable the IPA transfer.

Convert foreground image to the destination image

If the foreground pixel format is indirect, the pixel data must be loaded into the foreground LUT before starting the IPA transfer. The LUT automatic loading procedure is described in the ***Foreground LUT loading***.

1. Configure the IPA_FMADDR and IPA_DMADDR register to set the foreground and destination memory base address.
2. Configure the FAVCA and FPF bits in the IPA_FPCTL register to set the foreground alpha value calculation algorithm and the foreground pixel format. Configure the IPA_CSCC_CFGx (x=0…2) when foreground pixel format is YUV/YcbCr.
3. Configure the pre-defined pixel value, including alpha, red, green and blue value in the IPA_FPCTL and IPA_FPV register if the foreground format is not ARGBxxxx type.
4. Configure the DPF bits in the IPA_DPCTL register to set the destination pixel format.
5. Configure the FLOFF and DLOFF bits in the IPA_FLOFF and IPA_DLOFF register to set the foreground and destination line offset.
6. Configure the VERDEC/HORDEC bits in the IPA_DPCTL, XSCAL/YSCAL bits in IPA_BSCTL and DWIDTH/DHEIGHT bits in the IPA_DIMS if scaling is needed.
7. Configure the ROT bits in the IPA_DPCTL when rotation is needed.
8. Configure the LM bits in the IPA_LM register to set the line mark if needed.
9. Configure the WIDTH and HEIGHT bits in the IPA_IMS register to set the image size.
10. Configure the needed enable bit for wrong configuration interrupt, LUT access conflict interrupt, transfer line mark interrupt, full transfer finish interrupt and transfer access error interrupt in the IPA_CTL register.
11. Configure the TEN bit with '1' in the IPA_CTL register to enable the IPA transfer.

Convert and blend the foreground and background images to the destination image

If the foreground or background pixel format is indirect, the pixel data must be loaded into the corresponding LUT before starting the IPA transfer. The foreground and background LUT automatically loading procedure is described in the ***Foreground LUT loading*** and ***Background LUT loading***.

1. Configure the IPA_FMADDR, IPA_BMADDR and IPA_DMADDR register to set the foreground, background and destination memory base address.
2. Configure the FAVCA and FPF bits in the IPA_FPCTL register to set the foreground alpha value calculation algorithm and the foreground pixel format. Configure the IPA_CSCC_CFGx (x=0…2) when foreground pixel format is YUV/YcbCr.
3. Configure the foreground pre-defined pixel value, including alpha, red, green and blue value in the IPA_FPCTL and IPA_FPV register if the foreground format is not ARGBxxxx type.
4. Configure the BAVCA and BPF bits in the IPA_BPCTL register to set the background alpha value calculation algorithm and the background pixel format. Configure the IPA_CSCC_CFGx (x=0…2) when foreground pixel format is YUV/YcbCr.
5. Configure the background pre-defined pixel value, including alpha, red, green and blue value in the IPA_BPCTL and IPA_BPV register if the background format is not ARGBxxxx type.
6. Configure the DPF bits in the IPA_DPCTL register to set the destination pixel format.
7. Configure the FLOFF, BLOFF and DLOFF bits in the IPA_FLOFF, IPA_BLOFF and IPA_DLOFF register to set the foreground, background and destination line offset.
8. Configure the LM bits in the IPA_LM register to set the line mark if needed.
9. Configure the WIDTH and HEIGHT bits in the IPA_IMS register to set the image size.
10. Configure the VERDEC/HORDEC bits in the IPA_DPCTL, XSCAL/YSCAL bits in IPA_BSCTL and DWIDTH/DHEIGHT bits in the IPA_DIMS if scaling is needed.
11. Configure the ROT bits in the IPA_DPCTL when rotation is needed.
12. Configure the needed enable bit for wrong configuration interrupt, LUT access conflict interrupt, transfer line mark interrupt, full transfer finish interrupt and transfer access error interrupt in the IPA_CTL register.
13. Configure the TEN bit with '1' in the IPA_CTL register to enable the IPA transfer.

Fill up the destination image with a specific color

1. Configure the IPA_DMADDR register to set the destination memory base address.
2. Configure the DPF bits in the IPA_DPCTL register to set the destination pixel format.
3. Configure the destination pre-defined pixel value, including alpha, red, green and blue value in the IPA_DPV register.
4. Configure the DLOFF bits in the IPA_DLOFF register to set the destination line offset.
5. Configure the LM bits in the IPA_LM register to set the line mark if needed.
6. Configure the WIDTH and HEIGHT bits in the IPA_IMS register to set the image size.
7. Configure the needed enable bit for wrong configuration interrupt, LUT access conflict interrupt, transfer line mark interrupt, full transfer finish interrupt and transfer access error interrupt in the IPA_CTL register.
8. Configure the TEN bit with '1' in the IPA_CTL register to enable the IPA transfer.

Configuration rules

The IPA configuration must respect a number of rules, otherwise the transfer or loading is automatically reset and the WCFIF bit in the IPA_INTF register is asserted immediately after

it is enabled. The rules are described as follows:

When the foreground LUT automatically loading is enabled:

- The FLMADDR bits in the IPA_FLMADDR register must be 32-bit alignment when the FLPF bit in the IPA_FPCTL register is equal to '0'.

When the background LUT automatically loading is enabled:

- The BLMADDR bits in the IPA_BLMADDR register must be 32-bit alignment when the BLPF bit in the IPA_BPCTL register is equal to '0'.

When the IPA transfer is enabled:

1) The FMADDR bits in the IPA_FMADDR register must be 32-bit alignment when the FPF bits in the IPA_FPCTL register are 'ARGB8888', 'UYVY422', 'VYUY422', 'YUV420' or 'YVU420' and be 16-bit alignment when the FPF bits are 'RGB565', 'ARGB1555', 'ARGB4444' or 'AL88' .
2) The FLOFF bits in the IPA_FLOFF register must be even when the FPF bits in the IPA_FPCTL register are 'A4', 'L4', 'UYVY422' or 'VYUY422', and be 4-bit alignment when the FPF bits in the IPA_FPCTL register are 'YUV420' or 'YVU420'.
3) The BMADDR bits in the IPA_BMADDR register must be 32-bit alignment when the BPF bits in the IPA_BPCTL register are 'ARGB8888', and be 16-bit alignment when the BPF bits are 'RGB565', 'ARGB1555', 'ARGB4444' or 'AL88' .
4) The BLOFF bits in the IPA_BLOFF register must be even when the BPF bits in the IPA_BPCTL register are 'A4' or 'L4'.
5) The FPF bits in the IPA_FPCTL register must be valid and less than or equal to '0b1111' when PFCM bits are not '0b00' or '0b11', and be valid and less than or equal to '0b1101' when PFCM bits are '0b00'.
6) The BPF bits in the IPA_BPCTL register must be valid and less than or equal to '0b1010'.
7) The DPF bits in the IPA_DPCTL register must be valid and less than or equal to '0b110'.
8) The DMADDR bits in the IPA_DMADDR register must be 32-bit alignment when the DPF bits in the IPA_DPCTL register are 'ARGB8888' and be 16-bit alignment when the DPF bits are 'RGB565', 'ARGB1555', 'ARGB4444'.
9) The DLOFF bits in the IPA_DLOFF register must be even when the FPF bits in the IPA_FPCTL register are 'A4' or 'L4'.
10) The WIDTH bits in the IPA_IMS register must be even when the FPF bits in the IPA_FPCTL register are 'A4', 'L4', 'UYVY422' or 'VYUY422', and be 4-bit-alignment when the FPF bits in the IPA_FPCTL register are 'YUV420' or 'YVU420'.
11) The WIDTH bits in the IPA_IMS register must be even when the BPF bits in the IPA_BPCTL register are 'A4' or 'L4'.
12) The WIDTH bits in the IPA_IMS register must be greater than zero.
13) The HEIGHT bits in the IPA_IMS register must be greater than zero.
14) The XSCALE and YSCALE bits in the IPA_BSCTL register must be greater than zero and less than '0x2001'.
15) When the FIIMEN bit is set, the FPF bits in the IPA_FPCTL register must not be 'YUV420' and 'YVU420', and the EFUVMADDR bits in the IPA_EF_UV_MADDR register must be 32-bit alignment when the FPF bits in the IPA_FPCTL register are 'ARGB8888',

'UYVY422', 'VYUY422', and be 16-bit alignment when the FPF bits are 'RGB565', 'ARGB1555', 'ARGB4444' or 'AL88'.

16) The EFUVMADDR bits in the IPA_EF_UV_MADDR register must be 16-bit alignment when the FPF bits in the IPA_FPCTL register are 'YUV420' or 'YVU420'.
17) When the forechannel scaling is enabled, that is, the HORDEC, VERDEC, XSCALE, YSCALE are not default value. The DWIDTH bits in the IPA_DIMS must be even when the BPF bits in the IPA_BPCTL register are 'A4', 'L4'.
18) When the forechannel scaling and rotation function are enabled, that is, the HORDEC , VERDEC, XSCALE, YSCALE, ROT are not default value. The DHEIGHT bits in the IPA_DIMS must be even when the BPF bits in the IPA_BPCTL register are 'A4' , 'L4'.

When the PFCM bits are equal to '00', only 1), 2), 5), 9), 10), 12), 13) are considerable.
When the PFCM bits are equal to '01', only 1), 2), 5), 7), 8), 10), 12), 13) ,14), 15), 16)are considerable.
When the PFCM bits are equal to '10', all the configuration rules except 9) are considerable.
When the PFCM bits are equal to '11', only 7), 8), 12), 13) are considerable.

36.6. Interrupts

There are six interrupt events connected to the IPA interrupt, including wrong configuration interrupt, LUT loading finish interrupt, LUT access conflict interrupt, transfer line mark interrupt, full transfer finish interrupt and transfer access error interrupt. An IPA interrupt can be produced when any interrupt events occurs.

Each interrupt event has a dedicated flag bit in the IPA_INTF register, a dedicated clear bit in the IPA_INTC register, and a dedicated enable bit in the IPA_CTL register. The relationship is described in the ***Table 36-8. IPA interrupt events***.

Table 36-8. IPA interrupt events

Interrupt event	Flag bit	Enable bit	Clear bit
	IPA_INTF	IPA_CTL	IPA_INTC
wrong configuration interrupt	WCFIF	WCFIE	WCFIFC
LUT loading finish interrupt	LLFIF	LLFIE	LLFIFC
LUT access conflict interrupt	LACIF	LACIE	LACIFC
transfer line mark interrupt	TLMIF	TLMIE	TLMIFC
full transfer finish interrupt	FTFIF	FTFIE	FTFIFC
transfer access error interrupt	TAEIF	TAEIE	TAEIFC

Wrong configuration interrupt

The wrong configuration interrupt flag is asserted immediately after the LUT loading or IPA transfer is enabled, when any of the configuration rules listed in the ***Configuration rules*** is broken. The LUT loading or IPA transfer is automatically disabled without launching any access.

When the wrong configuration interrupt flag is asserted and the enabled bit for wrong

configuration interrupt is set, an IPA interrupt is generated.

LUT loading finish interrupt

The LUT loading finish interrupt flag is asserted immediately after the last pixel data has been loaded into the foreground or background LUT. A stop operation during the loading cannot assert the LUT loading finish interrupt flag.

When the LUT loading finish interrupt flag is asserted and the enabled bit for LUT loading finish interrupt is set, an IPA interrupt is generated.

LUT access conflict interrupt

A number of rules must be respected when accessing the foreground and background LUT by software:

- During the foreground LUT automatic loading, the foreground LUT is forbidden to be accessed by software.
- During the background LUT automatic loading, the background LUT is forbidden to be accessed by software.
- During the IPA transfer with the PFCM bits equal to '0b01' or '0b10', if the foreground pixel format is indirect, the foreground LUT is forbidden to be accessed by software.
- During the IPA transfer with the PFCM bits equal to '0b10', if the background pixel format is indirect, the background LUT is forbidden to be accessed by software.

When one of the above rules is broken, the LUT access conflict interrupt flag is asserted and the software access has no effect (writing access is not be executed, reading access is returned with an invalid value).

When the LUT access conflict interrupt flag is asserted and the enabled bit for the LUT access conflict interrupt is set, an IPA interrupt is generated.

Transfer line mark interrupt

The transfer line mark interrupt flag is asserted immediately after the last pixel data of the line mark is written into the destination memory. If the LM bits in the IPA_LM register are equal to 0, the transfer line mark interrupt flag will never be asserted during the IPA transmission.

When the transfer line mark interrupt flag is asserted and the enabled bit for the transfer line mark interrupt is set, an IPA interrupt is generated.

Full transfer finish interrupt

The full transfer finish interrupt flag is asserted immediately after the last pixel data has been written into the destination memory. A stop operation during the IPA transmission cannot assert the full transfer finish interrupt flag.

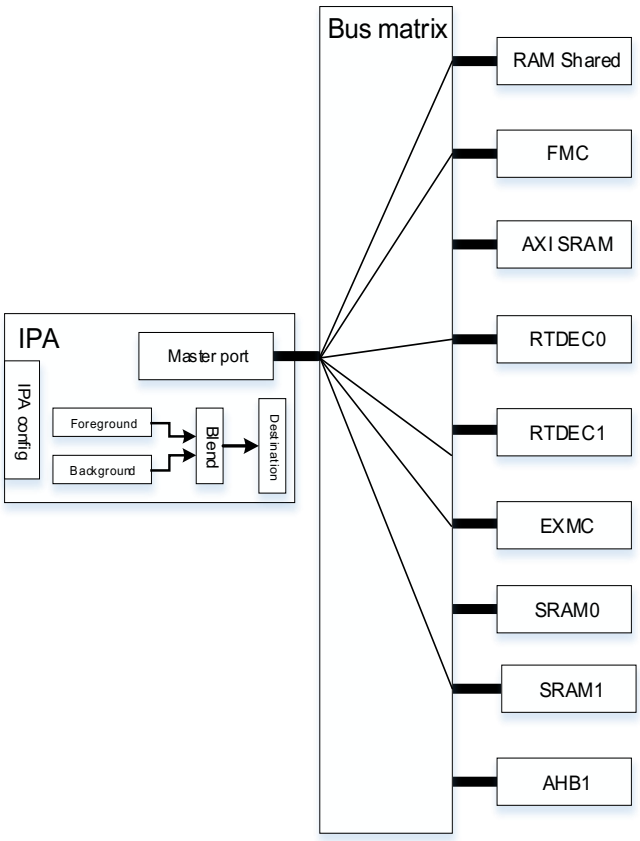
When the full transfer finish interrupt flag is asserted and the enabled bit for the full transfer finish interrupt is set, an IPA interrupt is generated.

Transfer access error interrupt

When the address accessed by the IPA is beyond the allowed area, a response error will be received and the transfer (LUT loading or IPA transfer) is disabled immediately without asserting the LUT loading finish interrupt flag or the full transfer finish interrupt flag. The allowed and forbidden access region for IPA is shown in ***Figure 36-9. System connection of IPA***.

When the transfer access error interrupt flag is asserted and the enabled bit for the transfer access error interrupt is set, an IPA interrupt is generated.

Figure 36-9. System connection of IPA



36.7. Register definition

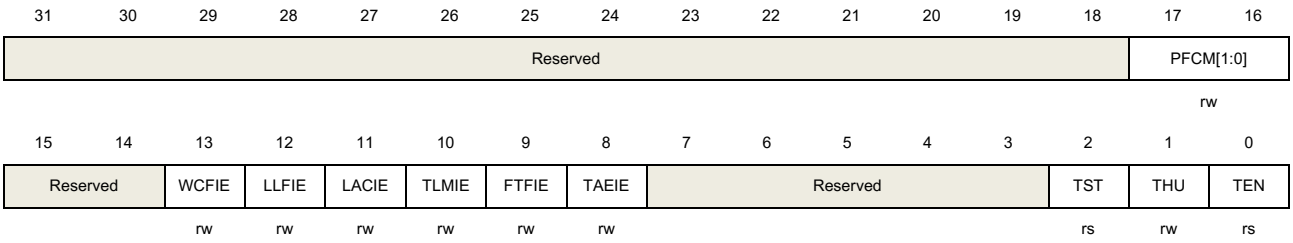
IPA base address: 0x5200 1000

36.7.1. Control register (IPA_CTL)

Address offset: 0x00

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:18	Reserved	Must be kept at reset value.
17:16	PFCM[1:0]	Pixel format convert mode Software set and clear. 00: Foreground memory to destination memory without pixel format convert 01: Foreground memory to destination memory with pixel format convert 10: Blending foreground and background memory to destination memory 11: Fill up destination memory with specific color These bits can NOT be written when TEN is '1'.
15:14	Reserved	Must be kept at reset value.
13	WCFIE	Enable bit for wrong configuration interrupt Software set and clear 0: Disable configuration error interrupt 1: Enable configuration error interrupt
12	LLFIE	Enable bit for LUT loading finish interrupt Software set and clear 0: Disable LUT loading finish interrupt 1: Enable LUT loading finish interrupt
11	LACIE	Enable bit for LUT access conflict interrupt Software set and clear 0: Disable LUT access conflict interrupt 1: Enable LUT access conflict interrupt
10	TLMIE	Enable bit for transfer line mark interrupt

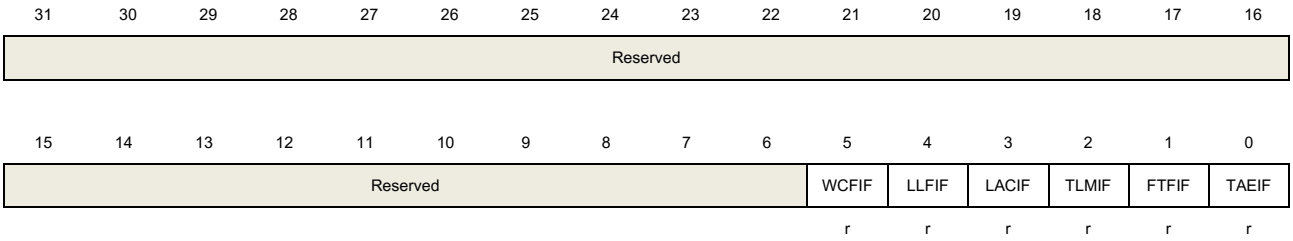
		Software set and clear 0: Disable transfer line mark interrupt 1: Enable transfer line mark interrupt
9	FTFIE	Enable bit for full transfer finish interrupt Software set and clear 0: Disable full transfer finish interrupt 1: Enable full transfer finish interrupt
8	TAEIE	Enable bit for transfer access error interrupt Software set and clear 0: Disable transfer access error interrupt 1: Enable transfer access error interrupt
7:3	Reserved	Must be kept at reset value.
2	TST	Transfer stop Software set, software and hardware clear. 0: No effect 1: Stop the current transfer When this bit is enabled, the current transfer (including LUT automatic loading and IPA transfer) is stopped. This bit can be cleared by hardware immediately when the current transfer is disabled.
1	THU	Transfer hang up Software set, software and hardware clear. 0: No effect 1: Hang up the current transfer When this bit is enabled, the current transfer (including LUT automatic loading and IPA transfer) is hanged up. When this bit is cleared, the current transfer continues. This bit can be cleared by hardware immediately when the current transfer is disabled.
0	TEN	Transfer enable Software set, hardware clear. 0: Transfer disable 1: Transfer enable When this bit is enabled, the IPA transfer is started. This bit is automatically cleared when one of the following situations occurs: – When the TST bit is enabled to stop the current transfer. – When the transfer is fully finished. – When a wrong configuration or a transfer access error is detected. – When the foreground LUT or background LUT is being loaded (FLLEN bit in the IPA_FPCTL register or BLLEN bit in the IPA_BPCTL register is '1').

36.7.2. Interrupt flag register (IPA_INTF)

Address offset: 0x04

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:6	Reserved	Must be kept at reset value.
5	WCFIF	Wrong configuration interrupt flag Hardware set, Software cleared by enable 'WCFIFC' bit in the IPA_INTC register. 0: No wrong configuration is detected when IPA transfer or LUT loading is enable. 1: A wrong configuration is detected when IPA transfer or LUT loading is enable.
4	LLFIF	LUT loading finish interrupt flag Hardware set, software cleared by enable 'LLFIFC' bit in the IPA_INTC register. 0: No LUT loading finish is detected 1: A LUT loading finish is detected
3	LACIF	LUT access conflict interrupt flag Hardware set, software cleared by enable 'LACIFC' bit in the IPA_INTC register. 0: No LUT access conflict is detected. 1: A LUT access conflict is detected.
2	TLMIF	Transfer line mark interrupt flag Hardware set, software cleared by enable 'TLMIFC' bit in the IPA_INTC register. 0: The number of pixel transferred has not exactly reached the line mark 1: The number of pixel transferred has exactly reached the line mark
1	FTFIF	Full transfer finish interrupt flag Hardware set, software cleared by enable 'FTFIFC' bit in the IPA_INTC register. 0: No full transfer finish is detected. 1: A full transfer finish is detected.
0	TAEIF	Transfer access error interrupt flag Hardware set, software cleared by enable 'TAEIFC' bit in the IPA_INTC register. 0: No transfer access error is detected. 1: A transfer access error is detected.

36.7.3. Interrupt flag clear register (IPA_INTC)

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															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										WCFIFC	LLFIFC	LACIFC	TLMIFC	FTFIFC	TAEIFC
										rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

Bits	Fields	Descriptions
31:6	Reserved	Must be kept at reset value.
5	WCFIFC	Clear bit for wrong configuration interrupt flag Software set, hardware clear 0: No effect 1: Clear wrong configuration interrupt flag
4	LLFIFC	Clear bit for LUT loading finish interrupt flag Software set, hardware clear 0: No effect 1: Clear LUT loading finish interrupt flag
3	LACIFC	Clear bit for LUT access conflict interrupt flag Software set, hardware clear 0: No effect 1: Clear LUT access conflict interrupt flag
2	TLMIFC	Clear bit for transfer line mark interrupt flag Software set, hardware clear 0: No effect 1: Clear transfer line mark interrupt flag
1	FTFIFC	Clear bit for full transfer finish interrupt flag Software set, hardware clear 0: No effect 1: Clear full transfer finish interrupt flag
0	TAEIFC	Clear bit for transfer access error interrupt flag Software set, hardware clear 0: No effect 1: Clear transfer access error interrupt flag

36.7.4. Foreground memory base address register (IPA_FMADDR)

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
FMADDR[31:16]															
rw															

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

Bits	Fields	Descriptions
31:0	FMADDR[31:0]	Foreground memory base address These bits must be aligned to 8-bit, 16-bit or 32-bit corresponding with the foreground pixel format. If foreground pixel format is ARGB8888, UYVY422, VYUY422 ,YUV420 or YVU420, these bits must be 32-bit aligned; If the foreground pixel format is RGB565, ARGB1555, ARGB4444 or AL88, these bits must be 16-bit aligned. If the above alignment rules are broken, a wrong configuration will be detected when the transfer is enable. These bits can NOT be written when TEN in the IPA_CTL register is '1'.

36.7.5. Foreground line offset register (IPA_FLOFF)

Address offset: 0x10

Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value.
13:0	FLOFF[13:0]	Foreground line offset These bits indicate the number of pixel between the last pixel of the current line and the first pixel of the next line. If the foreground pixel format is A4, L4, UYVY422 or VYUY422, the FLOFF must be configured to be an even number, and if the foreground pixel format is YUV420 or YVU420, the FLOFF must be configured 4-bit alignment otherwise a wrong configuration will be detected when the transfer is

enable.

These bits can NOT be written when TEN in the IPA_CTL register is '1'.

36.7.6. Background memory base address register (IPA_BMADDR)

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
BMADDR[31:16]															
rw															

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

Bits	Fields	Descriptions
31:0	BMADDR[31:0]	Background memory base address These bits must be aligned to 8-bit, 16-bit or 32-bit corresponding with the background pixel format. If background pixel format is ARGB8888, UYVY422 and VYUY422, these bits must be 32-bit aligned; If the background pixel format is RGB565, ARGB1555, ARGB4444 or AL88, these bits must be 16-bit aligned. If the above alignment rules are broken, a wrong configuration will be detected when the transfer is enable. These bits can NOT be written when TEN in the IPA_CTL register is '1'.

36.7.7. Background line offset register (IPA_BLOFF)

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		BLOFF[13:0]													
		rw													

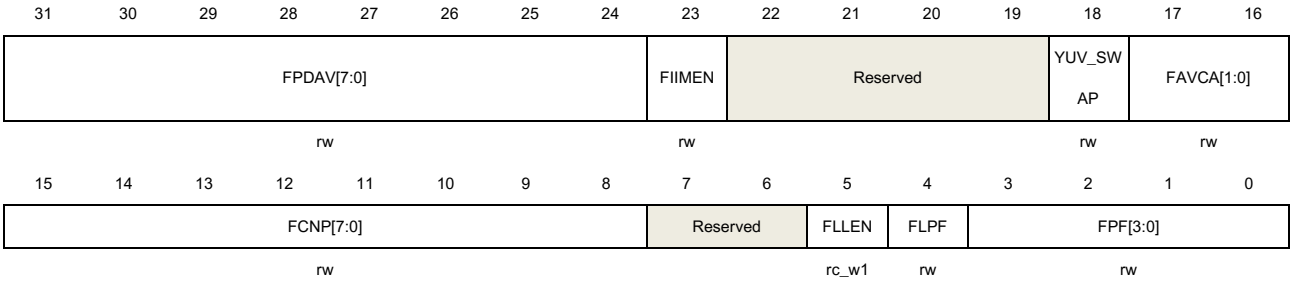
Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value.
13:0	BLOFF[13:0]	Background line offset These bits indicate the number of pixel between the last pixel of the current line and

the first pixel of the next line. If the background pixel format is A4 or L4, the BLOFF must be configured to be an even number, otherwise a configuration error will be detected when the transfer is enable.
These bits can NOT be written when TEN in the IPA_CTL register is '1'.

36.7.8. Foreground pixel control register (IPA_FPCTL)

Address offset: 0x1C
Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:24	FPDAV[7:0]	Foreground pre-defined alpha value Software set and clear These bits define an alpha value. These bits are used to calculate the foreground alpha channel value with the alpha value read from foreground memory or foreground LUT according to the foreground alpha calculation algorithm. These bits can NOT be written when TEN in the IPA_CTL register is '1'.
23	FIIMEN	Foreground input interlace mode enable 0: Foreground input interlace mode disable 1: Foreground input interlace mode enable
22:19	Reserved	Must be kept at reset value.
18	YUV_SWAP	Swap the YUV422 format of foreground pixel format 0:the foreground pixel format is FPF 1:when FPF is UYVY422, the foreground pixel format is actually YVYU422; when FPF is VYUY422, the foreground pixel format is actually YUYV422
17:16	FAVCA[1:0]	Foreground alpha value calculation algorithm Software set and clear 00: No effect 01: FPDAV[7:0] is selected as the foreground alpha value 10: FPDAV[7:0] multiplied by the alpha data read from foreground memory or foreground LUT divided by 255 is selected as the foreground alpha value 11: Reserved

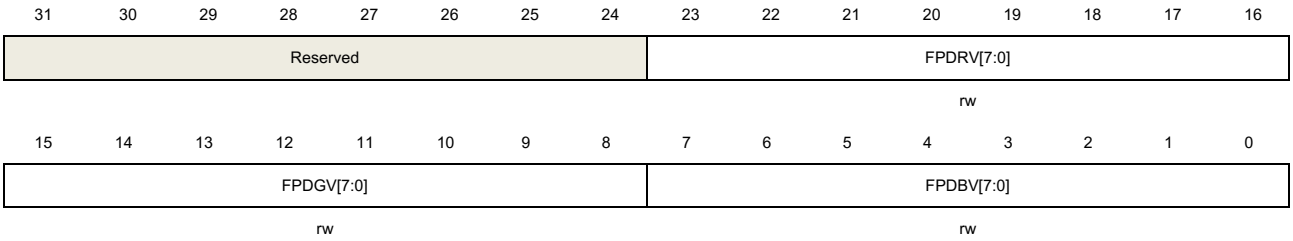
		These bits can NOT be written when TEN in the IPA_CTL register is '1'.
15:8	FCNP[7:0]	Foreground LUT number of pixel Software set and clear The pixel number of foreground LUT is equal to FCNP + 1. These bits can NOT be written when FLLEN is '1'.
7:6	Reserved	Must be kept at reset value.
5	FLLEN	Foreground LUT loading enable Software set, hardware clear. 0: Disable foreground LUT loading 1: Enable foreground LUT loading When this bit is enabled, the foreground LUT loading is started. This bit is automatically cleared when one of the following situations occurs: – When the TST bit is enabled – When the foreground LUT loading is finished – When a wrong configuration or a transfer error is detected – When the IPA transfer is ongoing or the background LUT is being loaded (TEN bit in the IPA_CTL register or BLLEN bit in the IPA_BPCTL register is '1').
4	FLPF	Foreground LUT pixel format Software set and clear 0: ARGB8888 1: RGB888 This bit can NOT be written when FLLEN is '1'.
3:0	FPF[3:0]	Foreground pixel format software set and clear 0000: ARGB8888 0001: RGB888 0010: RGB565 0011: ARGB1555 0100: ARGB4444 0101: L8 0110: AL44 0111: AL88 1000: L4 1001: A8 1010: A4 1011: YUV444 (1 Plane) 1100: UYVY422 (1 Plane) 1101: VYUY422 (1 Plane) 1110: YUV420 (2 Plane) 1111: YVU420 (2 Plane) These bits can NOT be written when TEN in the IPA_CTL register is '1'.

36.7.9. Foreground pixel value register (IPA_FPV)

Address offset: 0x20

Reset value: 0x0000 0000

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



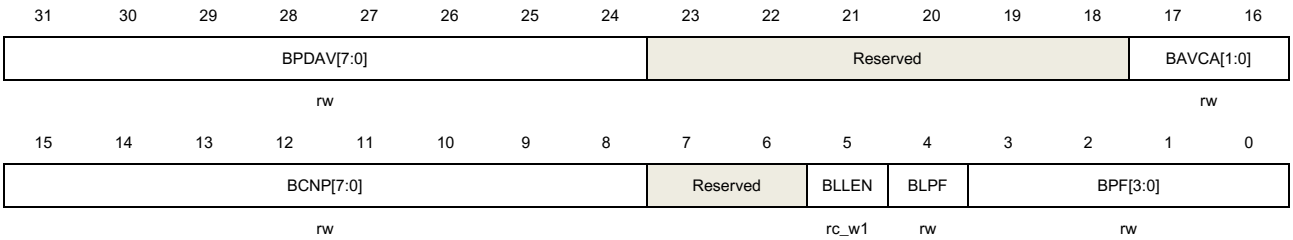
Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:16	FPDRV[7:0]	Foreground pre-defined red value Software set and clear When the foreground pixel format is A4 or A8, these bits are used as the foreground red value. These bits can NOT be written when TEN in the IPA_CTL register is '1'.
15:8	FPDGV[7:0]	Foreground pre-defined green value Software set and clear When the foreground pixel format is A4 or A8, these bits are used as the foreground green value. These bits can NOT be written when TEN in the IPA_CTL register is '1'.
7:0	FPDBV[7:0]	Foreground pre-defined blue value Software set and clear When the foreground pixel format is A4 or A8, these bits are used as the foreground blue value. These bits can NOT be written when TEN in the IPA_CTL register is '1'.

36.7.10. Background pixel control register (IPA_BPCTL)

Address offset: 0x24

Reset value: 0x0000 0000

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



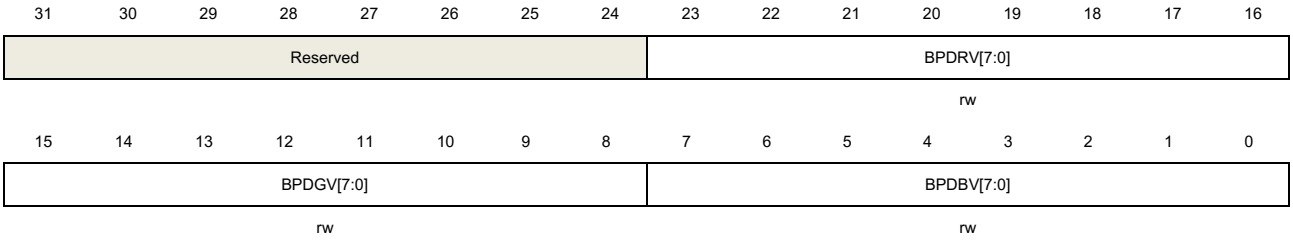
Bits	Fields	Descriptions
31:24	BPDAV[7:0]	Background pre- defined alpha value Software set and clear These bits define an alpha value. These bits are used to calculate the background alpha channel value with the alpha value read from background memory or background LUT according to the background alpha calculation algorithm. These bits can NOT be written when TEN in the IPA_CTL register is '1'.
23:18	Reserved	Must be kept at reset value.
17:16	BAVCA[1:0]	Background alpha value calculation algorithm Software set and clear 00: No effect 01: BPDAV[7:0] is selected as the background alpha value 10: BPDAV[7:0] multiplied by the alpha data read from background memory or background LUT divided by 255 is selected as the background alpha value 11: Reserved These bits can NOT be written when TEN in the IPA_CTL register is '1'.
15:8	BCNP[7:0]	Background LUT number of pixel Software set and clear The number of pixel of background LUT is equal to BCNP + 1. These bits can NOT be written when BLLEN is '1'.
7:6	Reserved	Must be kept at reset value.
5	BLLEN	Background LUT loading enable Software set, hardware clear. 0: Background LUT loading disable 1: Background LUT loading enable This bit is automatically cleared when one of the following situations occurs: – When the TST bit is enabled – When the background LUT loading is finished – When a wrong configuration or a transfer error is detected – When the IPA transfer is ongoing or the foreground LUT is being loaded (TEN bit in the IPA_CTL register or FLLEN bit in the IPA_FPCTL register is '1').
4	BLPF	Background LUT pixel format Software set and clear 0: ARGB8888 1: RGB888 This bit can NOT be written when BLLEN is '1'.
3:0	BPF[3:0]	Background pixel format software set and clear 0000: ARGB8888

0001: RGB888
0010: RGB565
0011: ARGB1555
0100: ARGB4444
0101: L8
0110: AL44
0111: AL88
1000: L4
1001: A8
1010: A4
1011 ~ 1111: Reserved
These bits can NOT be written when TEN in the IPA_CTL register is '1'.

36.7.11. Background pixel value register (IPA_BPV)

Address offset: 0x28
Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:16	BPDRV[7:0]	Background pre-defined red value Software set and clear When the background pixel format is A4 or A8, these bits are used as the background red value. These bits can NOT be written when TEN in the IPA_CTL register is '1'.
15:8	BPDGV[7:0]	Background pre-defined green value Software set and clear When the background pixel format is A4 or A8, these bits are used as the background green value. These bits can NOT be written when TEN in the IPA_CTL register is '1'.
7:0	BPDBV[7:0]	Background pre-defined blue value Software set and clear When the background pixel format is A4 or A8, these bits are used as the background blue value.

These bits can NOT be written when TEN in the IPA_CTL register is '1'.

36.7.12. Foreground LUT memory base address register (IPA_FLMADDR)

Address offset: 0x2C

Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:0	FLMADDR[31:0]	Foreground LUT memory base address Software set and clear The address must be aligned to 8-bit, 16-bit or 32-bit corresponding with the foreground LUT pixel format. If foreground LUT pixel format is ARGB8888, these bits must be 32-bit aligned. If the above alignment rule is broken, a wrong configuration will be detected when the foreground LUT loading is enable. These bit can NOT be written when FLLEN in the IPA_FPCTL register is '1'.

36.7.13. Background LUT memory base address register (IPA_BLMADDR)

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
BLMADDR[31:16]															
rw															

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

Bits	Fields	Descriptions
31:0	BLMADDR[31:0]	Background LUT memory base address Software set and clear The address must be aligned to 8-bit, 16-bit or 32-bit corresponding with the background LUT pixel format. If background LUT pixel format is ARGB8888, these bits must be 32-bit aligned. If the above alignment rule is broken, a wrong configuration will be detected when the background LUT loading is enable.

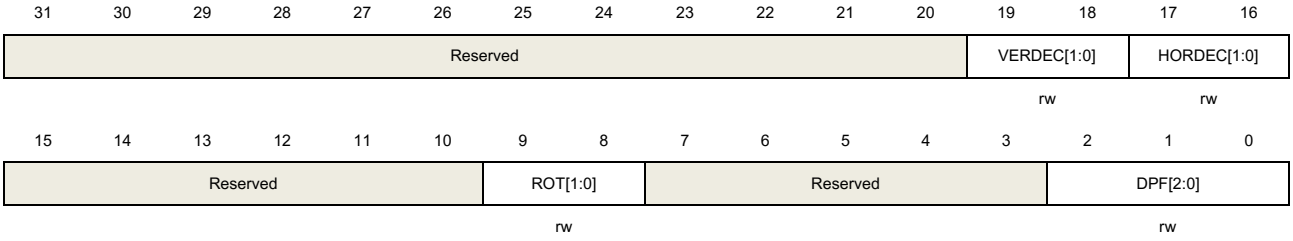
These bit can NOT be written when BLLEN in the IPA_BPCTL register is '1'.

36.7.14. Destination pixel control register (IPA_DPCTL)

Address offset: 0x34

Reset value: 0x0000 0000

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



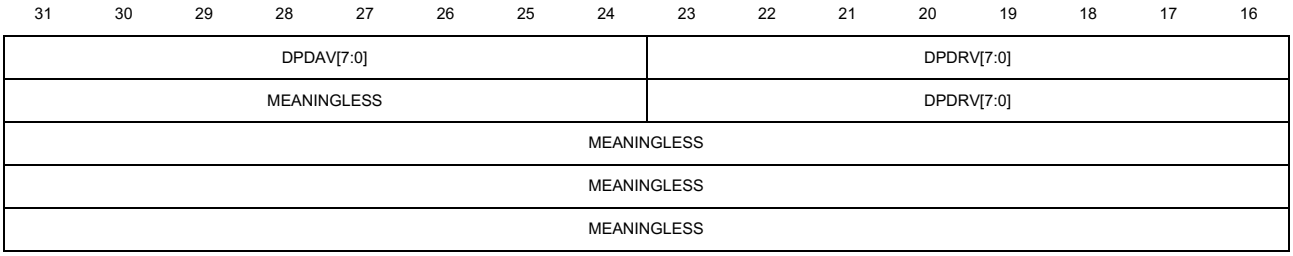
Bits	Fields	Descriptions
31:20	Reserved	Must be kept at reset value.
19:18	VERDEC[1:0]	Vertical pre decimation filter control 00: Disable the pre-decimation filter 01: Down scaling factors of decimation filter is 2 10: Down scaling factors of decimation filter is 4 11: Down scaling factors of decimation filter is 8
17:16	HORDEC[1:0]	Horizontal pre decimation filter control 00: Disable the pre-decimation filter 01: Down scaling factors of decimation filter is 2 10: Down scaling factors of decimation filter is 4 11: Down scaling factors of decimation filter is 8
15:10	Reserved	Must be kept at reset value.
9:8	ROT[1:0]	Indicates the clockwise rotation to be applied at the output. 00: No Rotation 01: Rotate 90 degree 10: Rotate 180 degree 11: Rotate 270 degree
7:3	Reserved	Must be kept at reset value.
2:0	DPF[2:0]	Destination pixel format Software set and clear 000: ARGB8888 001: RGB888 010: RGB565 011: ARGB1555 100: ARGB4444

101:BGR888
110:BGR565
111: Reserved
These bits can NOT be written when TEN in the IPA_CTL register is '1'.

36.7.15. Destination pixel value register (IPA_DPV)

Address offset: 0x38
Reset value: 0x0000 0000

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



When the destination pixel format is ARGB8888, the FIRST row is valid.

Bits	Fields	Descriptions
31:24	DPDAV[7:0]	Destination pre-defined alpha value Software set and clear When IPA is configured to fill up destination memory with specific color, these bits are used as the destination alpha value. These bits can NOT be written when TEN in the IPA_CTL register is '1'.
23:16	DPDRV[7:0]	Destination pre-defined red value Software set and clear When IPA is configured to fill up destination memory with specific color, these bits are used as the destination red value. These bits can NOT be written when TEN in the IPA_CTL register is '1'.
15:8	DPDGV[7:0]	Destination pre-defined green value Software set and clear When IPA is configured to fill up destination memory with specific color, these bits are used as the destination green value.

		These bits can NOT be written when TEN in the IPA_CTL register is '1'.
7:0	DPDBV[7:0]	Destination pre-defined blue value Software set and clear When IPA is configured to fill up destination memory with specific color, these bits are used as the destination blue value. These bits can NOT be written when TEN in the IPA_CTL register is '1'.

When the destination pixel format is RGB888, the SECOND row is valid.

Bits	Fields	Descriptions
31:24	Meaningless	These bit can be set and cleared by software, but these bits have no meaning when the destination pixel format is RGB888.
23:16	DPDRV[7:0]	Destination pre-defined red value Software set and clear When IPA is configured to fill up destination memory with specific color, these bits are used as the destination red value. These bits can NOT be written when TEN in the IPA_CTL register is '1'.
15:8	DPDGV[7:0]	Destination pre-defined green value Software set and clear When IPA is configured to fill up destination memory with specific color, these bits are used as the destination green value. These bits can NOT be written when TEN in the IPA_CTL register is '1'.
7:0	DPDBV[7:0]	Destination pre-defined blue value Software set and clear When IPA is configured to fill up destination memory with specific color, these bits are used as the destination blue value. These bits can NOT be written when TEN in the IPA_CTL register is '1'.

When the destination pixel format is RGB565, the THIRD row is valid.

Bits	Fields	Descriptions
31:16	Meaningless	These bit can be set and cleared by software, but these bits have no meaning when the destination pixel format is RGB565.
15:11	DPDRV[4:0]	Destination pre-defined red value Software set and clear When IPA is configured to fill up destination memory with specific color, these bits are used as the destination red value. These bits can NOT be written when TEN in the IPA_CTL register is '1'.
10:5	DPDGV[5:0]	Destination pre-defined green value Software set and clear When IPA is configured to fill up destination memory with specific color, these bits

		are used as the destination green value. These bits can NOT be written when TEN in the IPA_CTL register is '1'.
4:0	DPDBV[4:0]	Destination pre-defined blue value Software set and clear When IPA is configured to fill up destination memory with specific color, these bits are used as the destination blue value. These bits can NOT be written when TEN in the IPA_CTL register is '1'.

When the destination pixel format is ARGB1555, the FOURTH row is valid.

Bits	Fields	Descriptions
31:16	Meaningless	These bit can be set and cleared by software, but these bits have no meaning when the destination pixel format is ARGB1555.
15	DPDAV	Destination pre-defined alpha value Software set and clear When IPA is configured to fill up destination memory with specific color, these bits are used as the destination alpha value. These bits can NOT be written when TEN in the IPA_CTL register is '1'.
14:10	DPDRV[4:0]	Destination pre-defined red value Software set and clear When IPA is configured to fill up destination memory with specific color, these bits are used as the destination red value. These bits can NOT be written when TEN in the IPA_CTL register is '1'.
9:5	DPDGV[4:0]	Destination pre-defined green value Software set and clear When IPA is configured to fill up destination memory with specific color, these bits are used as the destination green value. These bits can NOT be written when TEN in the IPA_CTL register is '1'.
4:0	DPDBV[4:0]	Destination pre-defined blue value Software set and clear When IPA is configured to fill up destination memory with specific color, these bits are used as the destination blue value. These bits can NOT be written when TEN in the IPA_CTL register is '1'.

When the destination pixel format is ARGB4444, the FIFTH row is valid.

Bits	Fields	Descriptions
31:16	Meaningless	These bit can be set and cleared by software, but these bits have no meaning when the destination pixel format is ARGB4444.
15:12	DPDAV[3:0]	Destination pre-defined alpha value Software set and clear

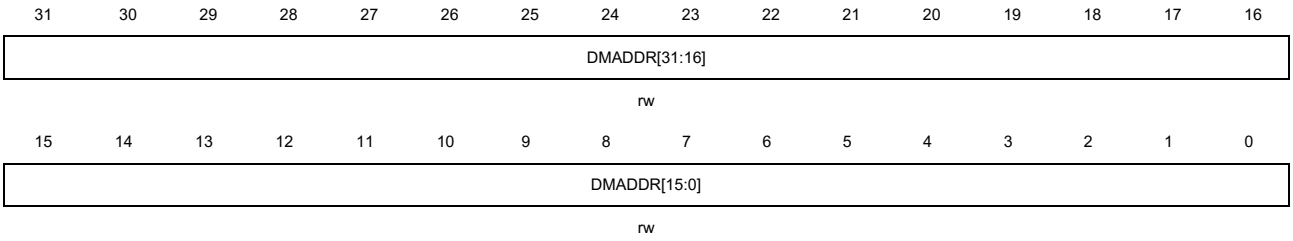
		When IPA is configured to fill up destination memory with specific color, these bits are used as the destination alpha value. These bits can NOT be written when TEN in the IPA_CTL register is '1'.
11:8	DPDRV[3:0]	Destination pre-defined red value Software set and clear When IPA is configured to fill up destination memory with specific color, these bits are used as the destination red value. These bits can NOT be written when TEN in the IPA_CTL register is '1'.
7:4	DPDGV[3:0]	Destination pre-defined green value Software set and clear When IPA is configured to fill up destination memory with specific color, these bits are used as the destination green value. These bits can NOT be written when TEN in the IPA_CTL register is '1'.
3:0	DPDBV[3:0]	Destination pre-defined blue value Software set and clear When IPA is configured to fill up destination memory with specific color, these bits are used as the destination blue value. These bits can NOT be written when TEN in the IPA_CTL register is '1'.

36.7.16. Destination memory base address register (IPA_DMADDR)

Address offset: 0x3C

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:0	DMADDR[31:0]	Destination memory base address software set and clear The address must be aligned to 8-bit, 16-bit or 32-bit corresponding with the destination pixel format. If the destination pixel format is ARGB8888, these bits must be 32-bit aligned; If the destination pixel format is RGB565, ARGB1555 or ARGB4444, these bits must be 16-bit aligned. If the above alignment rules are broken, a wrong configuration will be detected when the transfer is enable. These bit can NOT be written when TEN in the IPA_CTL register is '1'.

36.7.17. Destination line offset register (IPA_DLOFF)

Address offset: 0x40

Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value.
13:0	DLOFF[13:0]	Destination line offset software set and clear These bits indicate the number of pixel between the last pixel of the current line and the first pixel of the next line. When PFCM in the IPA_CTL register is configured to '00', if the foreground pixel format is A4 or L4, these bits must be configured to be an even number, or a wrong configuration will be detected when the transfer is enable. These bits can NOT be written when TEN in the IPA_CTL register is '1'.

36.7.18. Image size register (IPA_IMS)

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
Reserved		WIDTH[13:0]													
		rw													

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

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29:16	WIDTH[13:0]	Width of the image to be processed Software set and clear These bits signify the number of pixels per line. If the foreground or background pixel format is A4, L4, UYVY422 or VYUY422, these bits must be configured to be

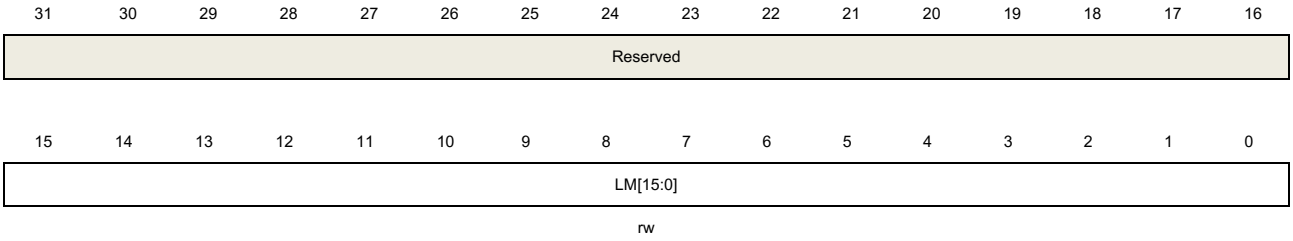
		an even number, otherwise a wrong configuration will be detected when the transfer is enable. These bits can NOT be written when TEN in the IPA_CTL register is '1'.
15:0	HEIGHT[15:0]	Height of the image to be processed Software set and clear These bits specify the number of lines of image to be processed. These bits can NOT be written when TEN in the IPA_CTL register is '1'.

36.7.19. Line mark register (IPA_LM)

Address offset: 0x48

Reset value: 0x0000 0000

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



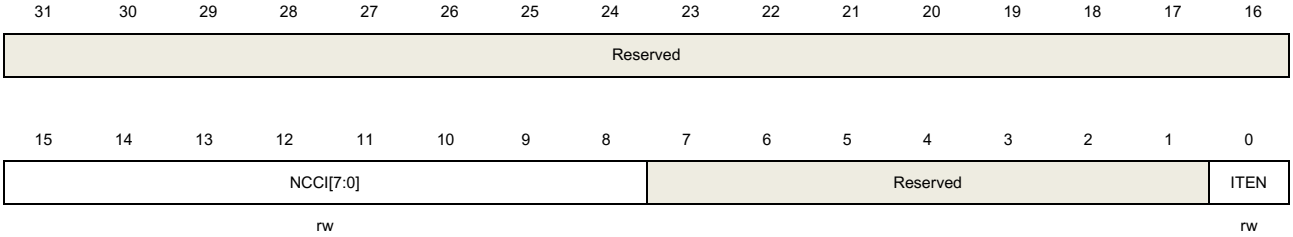
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	LM[15:0]	line mark Software set and clear These bits define a line number to signify the transfer level. An interrupt flag is asserted as soon as the last pixel of the marked line has been written into the destination memory. These bits can NOT be written when TEN in the IPA_CTL register is '1'.

36.7.20. Inter-timer control register (IPA_ITCTL)

Address offset: 0x4C

Reset value: 0x0000 0000

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



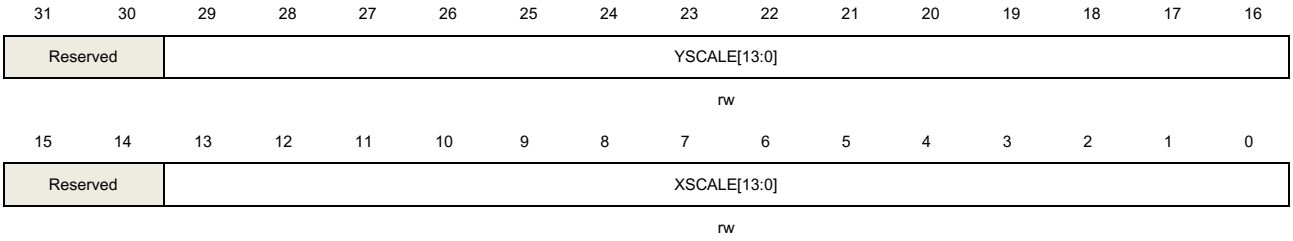
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:8	NCCI[7:0]	Number of clock cycles interval Software set and clear These bits have no meaning if ITEN is '0'. If the ITEN is '1', two consecutive AHB commands are issued with an interval equal to or greater than these bits.
7:1	Reserved	Must be kept at reset value.
0	ITEN	Inter-timer enable An inter-timer is implemented to reduce the AHB bus bandwidth usage of IPA. 0: Disable inter-timer 1: Enable inter-timer

36.7.21. Bilinear scaling control register (IPA_BSCTL)

Address offset: 0x50

Reset value: 0x1000 1000

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



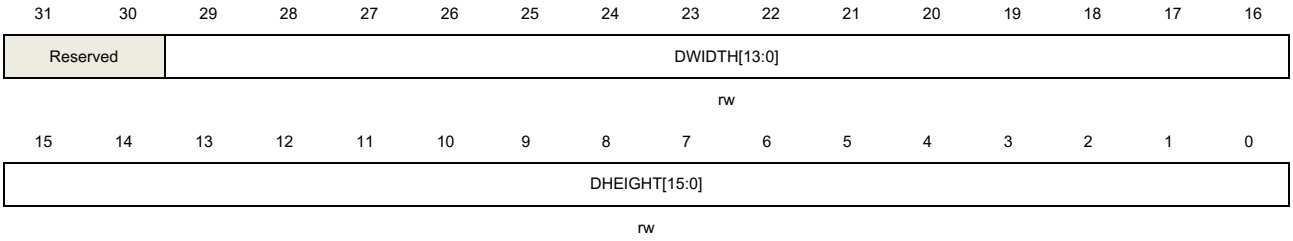
Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29:16	YSCALE[13:0]	It consists of 2-bit integer and 12-bit fractional, representing the Y scaling factor for the Foreground. **Note:** The maximum down scaling factor is 1/2 and the maximum up scaling factor is 2^12.
15:14	Reserved	Must be kept at reset value.
13:0	XSCALE[13:0]	It consists of 2-bit integer and 12-bit fractional, representing of the X scaling factor for the Foreground. **Note:** The maximum down scaling factor is 1/2 and the maximum up scaling factor is 2^12.

36.7.22. Scaling destination image size register (IPA_DIMS)

Address offset: 0x54

Reset value: 0x0000 0000

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

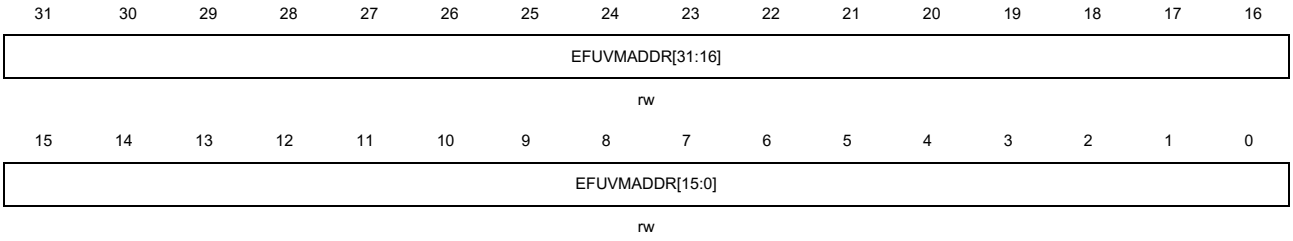


Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29:16	DWIDTH[13:0]	Width of the image to be processed Software set and clear. These bits signify the number of pixels per line. If the foreground or background pixel format is A4 or L4, these bits must be configured to be an even number, otherwise a wrong configuration will be detected when the transfer is enable. These bits can NOT be written when TEN in the IPA_CTL register is '1'. Set these bits when scaling.
15:0	DHEIGHT[15:0]	Height of the image to be processed Software set and clear These bits specify the number of lines of image to be processed. These bits can NOT be written when TEN in the IPA_CTL register is '1'. Set these bits when scaling.

36.7.23. Foreground even frame/UV memory base address register (IPA_EF_UV_MADDR)

Address offset: 0x5C
Reset value: 0x0000 0000

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



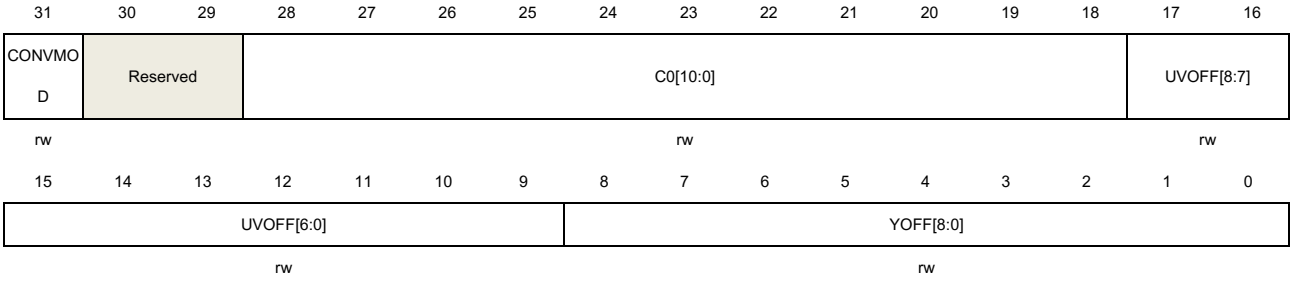
Bits	Fields	Descriptions
31:0	EFUVMADDR[31:0]	Foreground Even Frame/UV memory base address These bits must be aligned to 8-bit, 16-bit or 32-bit corresponding with the

foreground pixel format. If foreground pixel format is ARGB8888,UYVY422 or VYUY422, these bits must be 32-bit aligned; If the foreground pixel format is YUV420, YVU420, RGB565, ARGB1555, ARGB4444 or AL88, these bits must be 16-bit aligned. If the above alignment rules are broken, a wrong configuration will be detected when the transfer is enable.
These bits can NOT be written when TEN in the IPA_CTL register is '1'.

36.7.24. Color space conversion coefficient configure register 0 (IPA_CSCC_CFG0)

Address offset: 0x60
Reset value: 0x0400 0000

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



Bits	Fields	Descriptions
31	CONVMOD	Color space convert mode 0: Convert YUV to RGB, the UV value are 8-bit two's compliment 1: Convert YcbCr to RGB, the CbCr value are 8-bit unsigned number
30:29	Reserved	Must be kept at reset value.
28:18	C0[10:0]	Y multiplier coefficient. For YUV, it is typically 0x100 (1.000). For YcbCr, it is typically 0x12A (1.164).
17:9	UVOFF[8:0]	Phase offset implicated for UV/CbCr data, which is used for YUV/YcbCr to RGB conversion. For YUV, it is typically 0x000. For YcbCr, it is typically 0x180 (-128).
8:0	YOFF[8:0]	Amplitude offset implicated for Y data. For YUV, it is typically 0. For YcbCr, it is typically 0x1F0 (-16).

36.7.25. Color space conversion coefficient configure register 1 (IPA_CSCC_CFG1)

Address offset: 0x64

Reset value: 0x0123 0208

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					C1[10:0]										
					rw										

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					C4[10:0]										
					rw										

Bits	Fields	Descriptions
31:27	Reserved	Must be kept at reset value.
26:16	C1[10:0]	V/Cr Red multiplier coefficient. For YUV, it is typically 0x123 (1.140). For YcbCr it is typically 0x198 (1.596).
15:11	Reserved	Must be kept at reset value.
10:0	C4[10:0]	U/Cb Blue multiplier coefficient. For YUV, it is typically 0x208 (2.032). For YcbCr it is typically 0x204 (2.017).

36.7.26. Color space conversion coefficient configure register 2 (IPA_CSCC_CFG2)

Address offset: 0x68

Reset value: 0x076B 079C

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					C2[10:0]										
					rw										

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					C3[10:0]										
					rw										

Bits	Fields	Descriptions
31:27	Reserved	Must be kept at reset value.
26:16	C2[10:0]	V/Cr Green multiplier coefficient. For YUV, it is typically 0x76B (-0.581). For YcbCr, it is typically 0x730 (-0.813).
15:11	Reserved	Must be kept at reset value.
10:0	C3[10:0]	U/Cb Green multiplier coefficient. For YUV, it is typically 0x79C (-0.394). For YcbCr, it is typically 0x79C (-0.392).

37. Encoder Divided-Output controller (EDOUT)

37.1. Overview

The encoder divided-output controller (EDOUT) is used to output location information obtained from the encoder in the form of A-phase, B-phase, and Z-phase pulses.

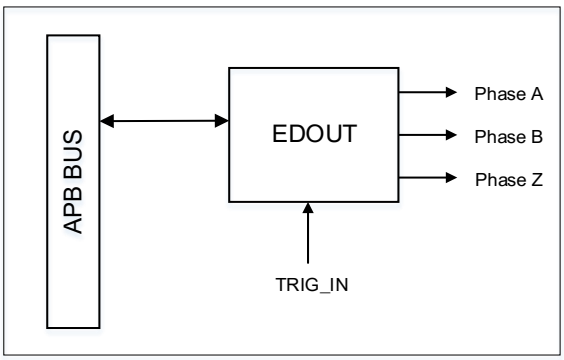
37.2. Characteristics

- Support for changing the active polarity of the B-phase output.
- Support for configuring the Z-phase output start location and pulse width.
- Number of edges per rotation: 16 to 65536, in multiples of four.
- Support for the input of update period event signals from the TRIGSEL.

37.3. Function overview

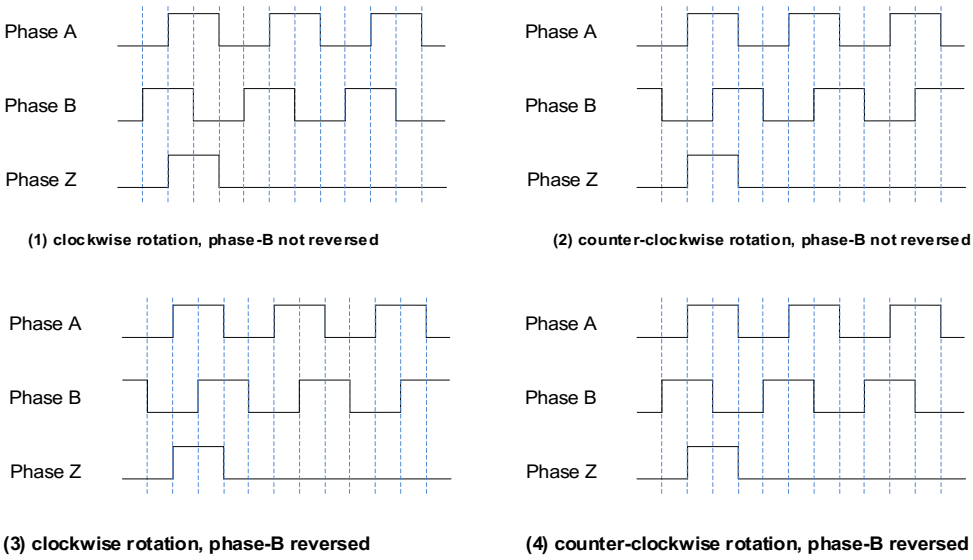
The EDOUT block diagram is shown in ***Figure 37-1. Block diagram of EDOUT***. The signals labeled Phase A, Phase B and Phase Z are the A-phase, B-phase and Z-phase output signals, respectively. TRIG_IN is the input signal from TRIGSEL.

Figure 37-1. Block diagram of EDOUT



The EDOUT output waveforms are similar to those of an incremental encoder. The output waveforms reflect the current location information according to the register settings, as shown in ***Figure 37-2. ABZ-phase output waveforms***.

Figure 37-2. ABZ-phase output waveforms



By configuring its registers, EDOUT converts location information from an incremental or absolute encoder into A-phase, B-phase and Z-phase output signals.

In addition to the initial setup of EDOUT, the CPU gets the location change from the encoder, updates the EDOUT register in each update period, and obtains the AB-phase and Z-phase output signals.

At this point, the event signal from the TIMER module must be input to EDOUT via TRIGSEL.

37.4. Z-phase output mode

Z-phase supports two operation modes:

■ Operation mode 0

In operation mode 0, the Z-phase output is determined by the current location. Configure the ZOSP and ZOWH fields in the EDOUT_ZCR register. When the current location, reflected by the LOCCNT field in EDOUT_LCNT, matches ZOSP, the Z-phase output pulse starts. Its width is ZOWH edges.

■ Operation mode 1

In operation mode 1, the Z-phase output is determined by the output edge number within an update period. Configure the ZOSP and ZOWH fields in EDOUT_ZCR. In each update period, when the output edge number matches ZOSP, the Z-phase output pulse starts. Its width is ZOWH edges.

In the above two operation modes, if the ZOWH bit field is set to 0, it means that the Z-phase is not output. Note that the ZCR register must be configured during the EDOUT run (EDOUT_ENABLE.EDOUTEN = 1); otherwise, the Z-phase has no valid output.

37.5. Operation guidance

EDOUT outputs AB-phase and Z-phase signals based on register settings. The update period of the AB-phase and Z-phase output is determined by the input of the TRIGSEL. Generate an update period using a timer, etc., and output a signal from TRIGSEL to EDOUT. The configuration procedure is described below.

37.5.1. EDOUT initialization

The EDOUT initialization steps are as follows:

1. Configure the EDOUT relevant output pins
2. Initialization settings for the EDOUT

With EDOUTEN set to 0, configure EDOUT_CTL and EDOUT_LOC. For example, for negative B-phase active polarity and 88 edges per rotation (4 × 22), set POL to 1 and LOCMAX to 87 (4 × 22 - 1).

3. Set the initial value in EDOUT_LCNT register

The initial value is set to the initial location of the absolute or incremental encoder for conversion to the AB-phase and Z-phase signals represented by the value from 0 to LOCMAX in EDOUT_LCNT register. The formula for expressing location value with value from 0 to LOCMAX is [encoder location value]×[number of edges per rotation]/[resolution of encoder location value](rounded down).

For 88 edges per rotation and an encoder resolution of 20 bits (1048576 location values), an initial encoder location value of 931802 gives LOCCNT = 78 after rounding down (931802 × 88 / 1048576).

4. Set the TRIGSEL

When a timer is used as the trigger source, configure it to generate the update-period trigger signals and select its trigger output as the EDOUT input through TRIGSEL. EDOUT also supports other trigger sources. The high-level pulse width of the selected trigger signal must be greater than T_{PCLK}.

5. Enable EDOUT and configure the Z-phase output

Set the EDOUTEN bit in the EDOUT_ENABLE register to 1. Then configure the ZOMD, ZOSP and ZOWH fields in the EDOUT_ZCR register while EDOUTEN remains 1. Set ZOWH to 0 if no Z-phase output is required.

Configure EDOUT_ZCR before starting the timer used to generate the update-period trigger signals. Writing EDOUT_ZCR only while EDOUTEN is 0 does not satisfy the configuration requirement for valid Z-phase output.

6. Start the TIMER operation.

Start the timer configured in step 4 to generate the update-period trigger signals.

37.5.2. EDOUT update processing

The EDOUT update processing steps are as follows:

1. Get location information.

Get location information from an absolute or incremental encoder.

2. Set the value in EDOUT_OCNT register

Firstly, calculate the value to be set in the PDC and EDGC bit fields of EDOUT_OCNT register. The current location represented by the value from 0 to LOCMAX is calculated as follows:

(m) = [location value obtained in step 1]×[number of edges per rotation]/[resolution of encoder location value] (rounded down)

At this point, the value of EDGC bit field is:

(n) = (m) - [the previous calculated value of (m)]

When the absolute value of (n) is greater than half of the number of rotated edges, then:

A. If (n) is positive, the value of EDGC is (n) - [number of edges per rotation].

B. If (n) is negative, the value of EDGC is (n) + [number of edges per rotation].

When EDGC bit field is not 0, PDC bit field is [update period]/[T_{PCLK}]/[absolute value of the EDGC bit field] (rounded down). When the EDGC bit field is 0, set the PDC bit field to 0xFFFF.

Set the calculated PDC and EDGC bit fields value in the EDOUT_OCNT register.

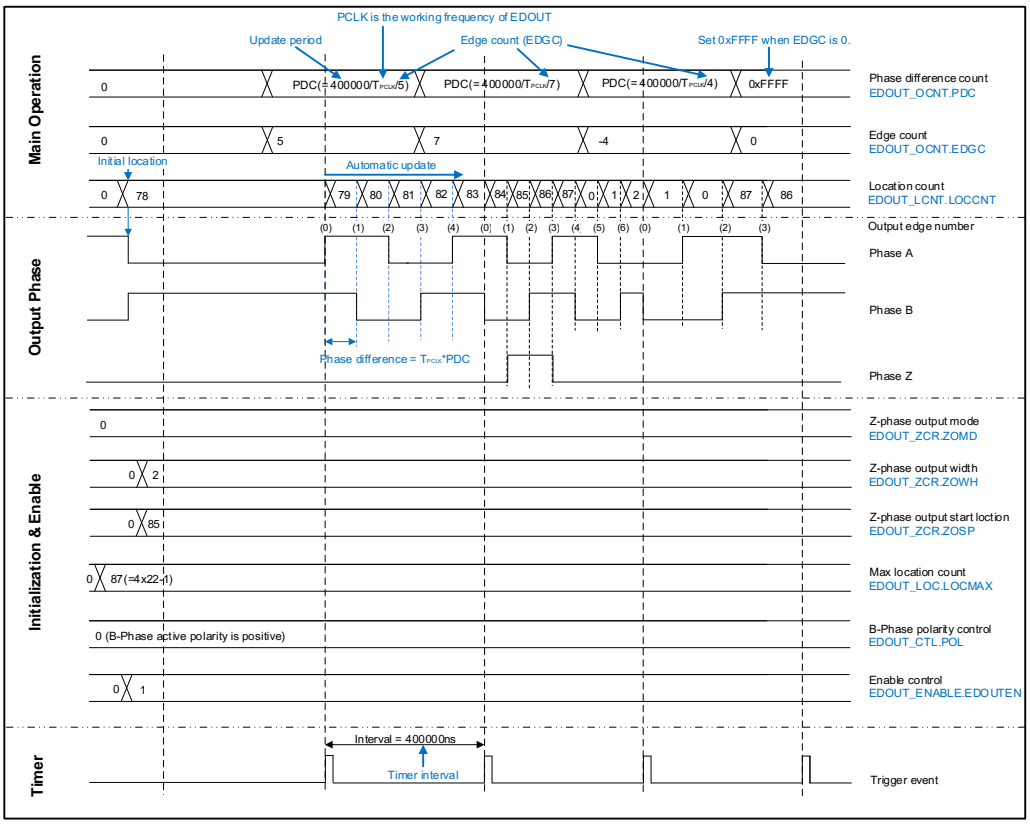
37.5.3. EDOUT working example

Figure 37-3. Example of the settings of EDOUT and the AB-Phase and Z-Phase output waveforms shows the EDOUT settings and the A-phase, B-phase and Z-phase output waveforms. In this example, a timer generates the update-period trigger signals, and its trigger output is selected as the EDOUT input. The update period is 400000 ns, and the number of edges per rotation is 88. The B-phase active polarity is positive, and the Z-phase output is determined by the current location. The encoder location value, expressed in the range 0 to LOCMAX, changes from the initial value of 78 to 83, then to 2, and finally to 86.

In this chapter, PCLK denotes the EDOUT working-clock frequency, and T_{PCLK} denotes its period (T_{PCLK} = 1/PCLK). Express the update period and T_{PCLK} in the same time unit when calculating PDC.

The update period is determined by the trigger signals supplied to EDOUT through TRIGSEL. In the example shown in the figure below, successive trigger events are 400000 ns apart, and all time values are expressed in ns.

Figure 37-3. Example of the settings of EDOUT and the AB-Phase and Z-Phase output waveforms



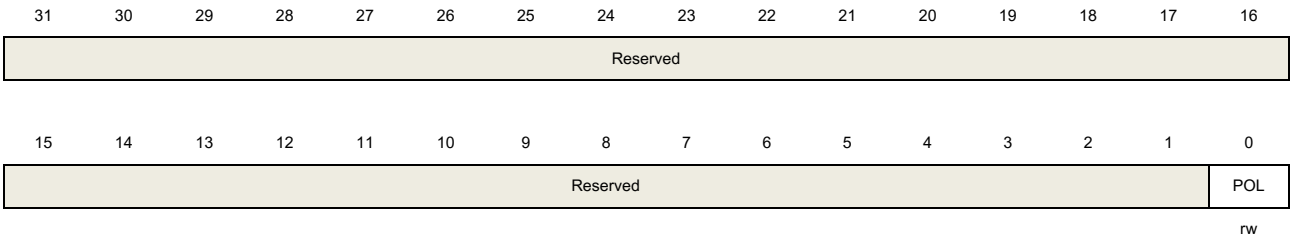
37.6. Register definition

EDOUT base address: 0x4001 8800

37.6.1. Control register (EDOUT_CTL)

Address offset: 0x00
Reset value: 0x0000 0000

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

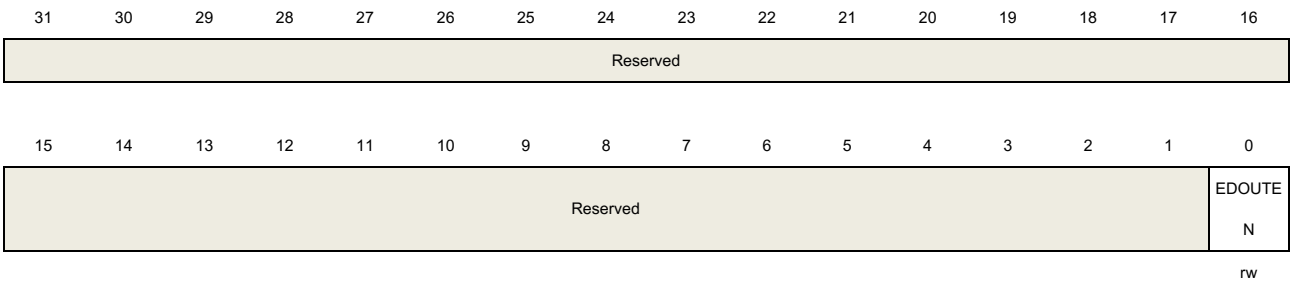


Bits	Fields	Descriptions
31:1	Reserved	Must be kept at reset value.
0	POL	B-Phase active polarity The active polarity of the B-phase output signal selection. If the EDOUTEN bit of EDOUT_ENABLE register is 0, the setting of this bit is reflected in the B phase output. Otherwise, this bit takes no effect. 0: Active polarity is positive 1: Active polarity is negative

37.6.2. Enable register (EDOUT_ENABLE)

Address offset: 0x04
Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:1	Reserved	Must be kept at reset value.
0	EDOUTEN	EDOUT enable

When this bit is 0, the A-phase and B-phase outputs immediately assume the corresponding states after EDOUT_LCNT is configured, and the Z-phase output is held at 0. When this bit is 1, EDOUT is enabled for A-phase, B-phase and Z-phase signal output. Valid Z-phase output also requires EDOUT_ZCR to be configured as described in the Z-phase output mode section.

0: EDOUT disabled

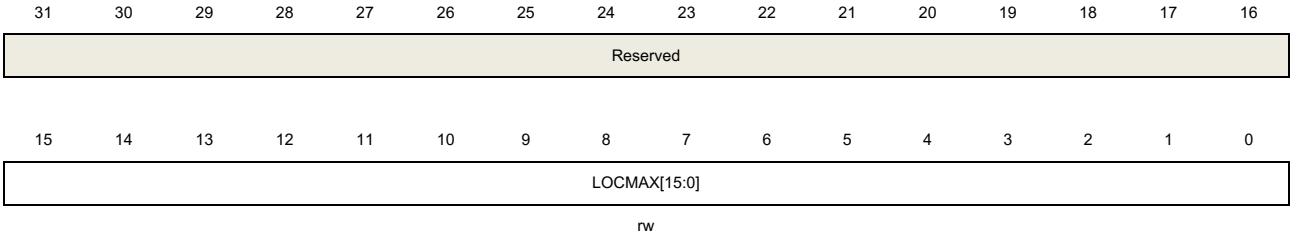
1: EDOUT enabled

37.6.3. Location register (EDOUT_LOC)

Address offset: 0x08

Reset value: 0x0000 0000

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



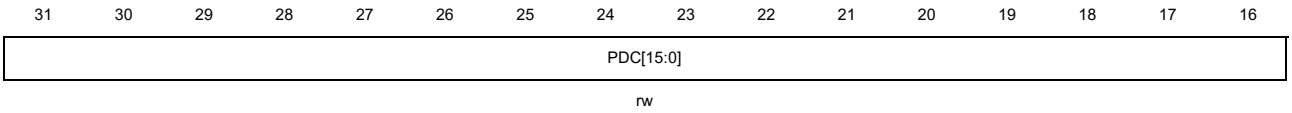
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	LOCMAX[15:0]	Maximum location value These bits set the maximum location value, which is one less than the number of edges per rotation. Let N be the number of edges per rotation. N must be a multiple of four in the range 16 to 65536. Set LOCMAX to N - 1. For example, for N = 4 × M, set LOCMAX to 4 × M - 1. The setting takes effect when the EDOUTEN bit in the EDOUT_ENABLE register changes from 0 to 1. 0x0000~0x000E: Reserved 0x000F: 16 edges per rotation; location values range from 0 to 15 ... 0xFFFF: 65536 edges per rotation; location values range from 0 to 65535

37.6.4. Output counter register (EDOUT_OCNT)

Address offset: 0x0C

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
EDGC[15:0]															
rw															

Bits	Fields	Descriptions
31:16	PDC[15:0]	Phase difference count These bits set the phase difference between the A-phase signal and the B-phase signal for the next update period. The allowable setting range is 2 to 65535, in units of T_{PCLK}. When the EDGC bit field is set to 0, set these bits to 65535; when it is not 0, set them to "Update period / T_{PCLK} / Absolute value of EDGC" (rounded down). When EDOUT is running (EDOUT_ENABLE.EDOUTEN = 1), please ensure that these bits are set before the next update period event signal is generated.
15:0	EDGC[15:0]	Edge count These bits set the number of edges of the A-phase and the B-phase signal for the next update period. For reverse rotation, set a negative value encoded in 16-bit two's complement form. The allowable setting range is -32767(0x8001) to 32767(0x7FFF). The absolute value of these bits must not be greater than "Update period / (2*T_{PCLK})". When EDOUT is running (EDOUT_ENABLE.EDOUTEN = 1), these fields must be configured before the next update-period trigger event.

37.6.5. Location counter register (EDOUT_LCNT)

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
LOCCNT[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	LOCCNT[15:0]	Current location value When EDOUT is stopped (EDOUT_ENABLE.EDOUTEN = 0), this field sets the current location. The allowable setting range is 0 to LOCMAX. After the current location is set, the A-phase and B-phase outputs immediately assume the corresponding states. When EDOUT is running (EDOUTEN = 1), this field reflects the current location changes associated with the A-phase and B-phase outputs.

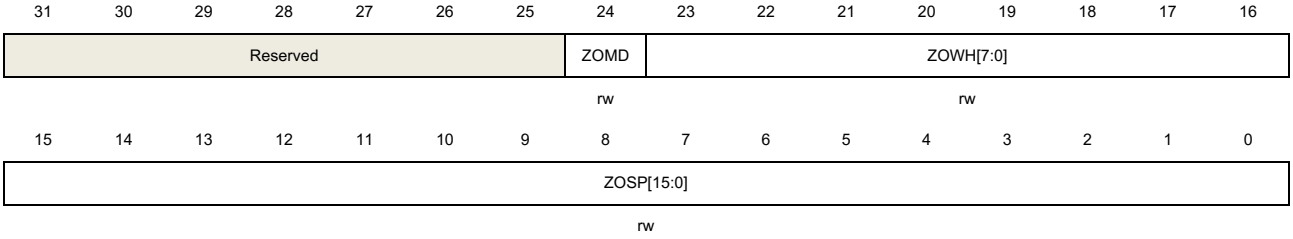
37.6.6. Z-phase configuration register (EDOUT_ZCR)

Address offset: 0x14

Reset value: 0x0000 0000

For valid Z-phase output, configure this register while EDOUT_ENABLE.EDOUTEN is 1.

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



Bits	Fields	Descriptions
31:25	Reserved	Must be kept at reset value.
24	ZOMD	Z-phase output mode 0: Output according to the current location 1: Output according to the output edge number within an update period
23:16	ZOWH[7:0]	Z-phase output pulse width This field sets the Z-phase output pulse width in units of output edges. A value of 0 disables Z-phase output in both operation modes.
15:0	ZOSP[15:0]	Z-phase output start location When ZOMD is 0, this field specifies the current location value that starts the Z-phase output pulse when matched by LOCCNT. When ZOMD is 1, this field specifies the output edge number within an update period that starts the Z-phase output pulse when matched.

38. Encoder Interface Module (EDIM)

The EDIM module includes 5 protocols: A-format encoder protocol, T-format encoder protocol, EnDat 2.2 encoder protocol, BiSS-C encoder protocol, and HIPERFACE DSL encoder protocol. These five protocols share a set of GPIO pins and interrupts. See ***Table 38-1. The five protocols of EDIM*** for details.

Table 38-1. The five protocols of EDIM

Encoder module	AFMT	TFMT	ENDAT	BISS	HDSL
Protocol	A-format	T-format	EnDat 2.2	BiSS-C	HIPERFACE DSL
Number of units	1	1	1	1	1
Maximum slave number	8	1	1	8	1
External trigger	●	●	●	●	●
Access mode	APB	APB	APB	APB	APB/SPI
DMA	●	●	●	●	x
Interrupt	●	●	●	●	●

38.1. A-format encoder interface module (AFMT)

38.1.1. Overview

The A-format encoder interface module (AFMT) serves as a master to communicate with encoders that support the A-format.

38.1.2. Characteristics

- Support the A-format version 2.0 protocol.
- Up to 8 encoders.
- Software trigger and external trigger.
- Monitor the status of the communication.
  8. Communication completion
  9. Timeout detection
  10. Transmission busy
- Interrupt generation.
- Support DMA mode.
- Multi configurable baud rates of 2.5 Mbits/s, 4 Mbits/s, 6.67 Mbits/s, 8 Mbits/s, and 16 Mbits/s

38.1.3. Block diagram

Figure 38-1. AFMT module block diagram

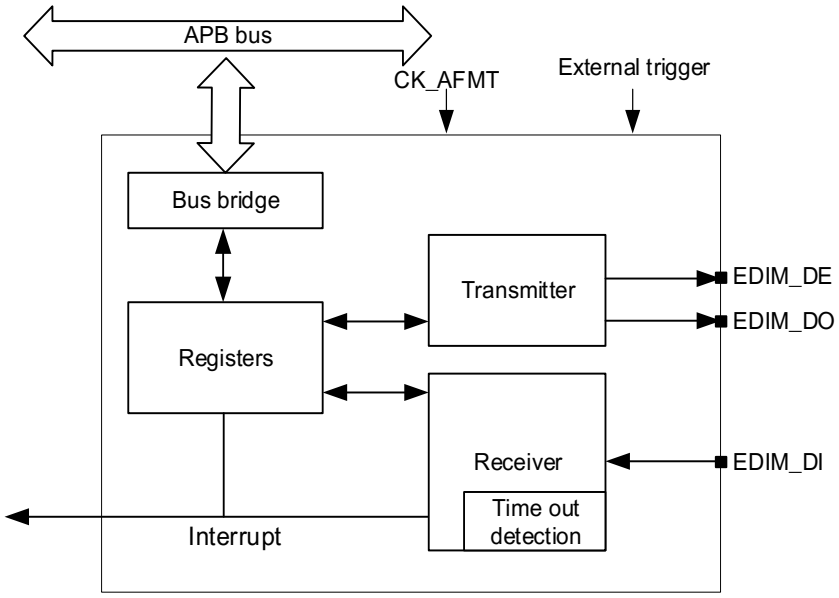


Figure 38-1. AFMT module block diagram provides details of the internal configuration of the AFMT module. *Table 38-2. AFMT pins definition* gives the AFMT pin description.

Table 38-2. AFMT pins definition

Name	Description
EDIM_DE	AFMT transceiver control signal
EDIM_DO	AFMT data output
EDIM_DI	AFMT data input

38.1.4. Function overview

AFMT module is a communication interface used to transmit control frames to encoder, receive data and status frames from encoder. Encoder status and communication alarm can read out from EDIM_AFMT_ENCnRDATAm registers.

AFMT clock

The function clock for AFMT module is CK_AFMT from RCU, with a frequency of 80MHz. The CPU accesses the AFMT module registers through the APB bus clock, which has a maximum frequency of 150MHz. A bus bridge is designed to synchronize the APB clock PCLK1 and CK_AFMT.

Note: Frequency of CK_AFMT affects the transmission baud rate. When the MCU samples data returned by the encoder, it adjusts the sampling point at the data transition edge, which increases the tolerance to the baud rate error of the slave encoder. However, when the MCU transmits data, errors in the clock frequency can affect the baud rate, which may cause errors

when the slave encoders receive data. The clock frequency and baud rate can have a 2% error margin.

10.1.9 format command

10.1.10 format command consists of 5 bits, stored in the CC[4:0] of EDIM_AFMT_CMD register. *Table 38-3. A-format command* shows command used in A-format protocol.

Table 38-3. A-format command

Frame name	CC[4:0]	Classification	Function
CDF0	5'b00000	Data request (Individual transmission)	ABS full 40-bit data request
CDF1	5'b00001		ABS lower 24-bit data request
CDF2	5'b00010		ABS upper 24-bit data request
CDF3	5'b00011		Encoder status request
CDF4	5'b00100	Data request (Broadcast transmission)	ABS full 40-bit data request
CDF5	5'b00101		ABS lower 24-bit data request
CDF6	5'b00110		ABS upper 24-bit data request
CDF7	5'b00111		Encoder status request
CDF8	5'b01000	Operating request (Individual transmission)	Status flag clear request (8 times in a row)
CDF9	5'b01001		Multi-rotation data clear request (8 times in a row)
CDF10	5'b01010		Status and multi-rotation data clear requests (8 times in a row)
CDF11	5'b01011		Encoder address setting I (8 times in a row)
CDF12	5'b01100		One-rotation data zero preset (8 times in a row)
CDF13	5'b01101	Memory access request (Individual transmission)	Memory read request (2 frames)
CDF14	5'b01110		Memory write request (4 frames)
CDF15	5'b01111	Data request (Individual transmission)	10-bit temperature data request
CDF16	5'b10000	ID code read	ID code read I (Individual)
CDF17	5'b10001		ID code read II (peer-to-peer connection)
CDF18	5'b10010	ID code write	ID code write I (Individual, 4 frames)

Frame name	CC[4:0]	Classification	Function
CDF19	5'b10011		ID code write II (peer-to-peer connection, 4 frames)
CDF20	5'b10100	Operating request (Individual transmission)	Encoder address setting II
CDF21	5'b10101	Data request (Individual transmission)	ABS lower 17-bit data request
CDF22	5'b10110	Data request (Broadcast transmission)	ABS lower 17-bit data request
CDF27	5'b11011	Data request (Individual transmission)	ABS lower 24-bit data and status requests
CDF28	5'b11100	Data request (Broadcast transmission)	ABS lower 24-bit data and status requests
CDF29	5'b11101	Data request (Individual transmission)	ABS lower 24-bit data and temperature requests
CDF30	5'b11110	Data request (Broadcast transmission)	ABS lower 24-bit data and temperature requests

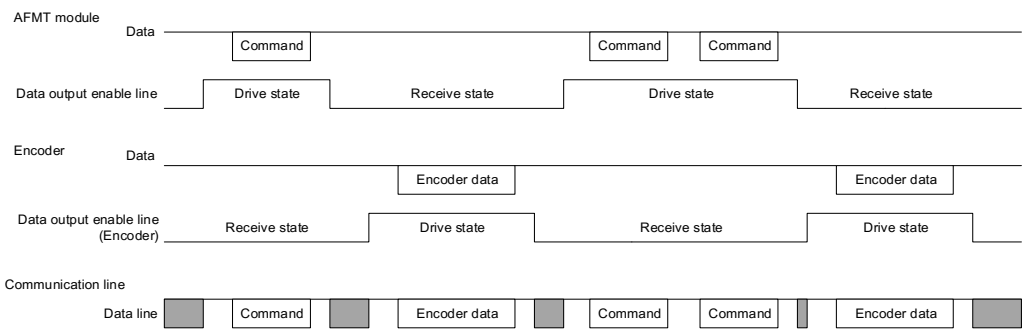
Note:

1. Other values of CC[4:0] are reserved, except for those listed in the table. CC[4:0] not in table could not be written into EDIM_AFMT_CMD register.
2. It is recommended that users employ a 1-to-1 connection when setting the slave encoder addressee with the CDF20 command to prevent encoders with the same identification code from responding simultaneously.

Individual transmission mode

In individual transmission mode, only slave encoders with matching same slave address will respond to the master command. The matching slave address is stored in the MEA[2:0] bits of the EDIM_AFMT_CMD register. The timing for serial data transmission and reception in individual transmission mode is shown in *Figure 38-2. Timing of individual transmission mode*.

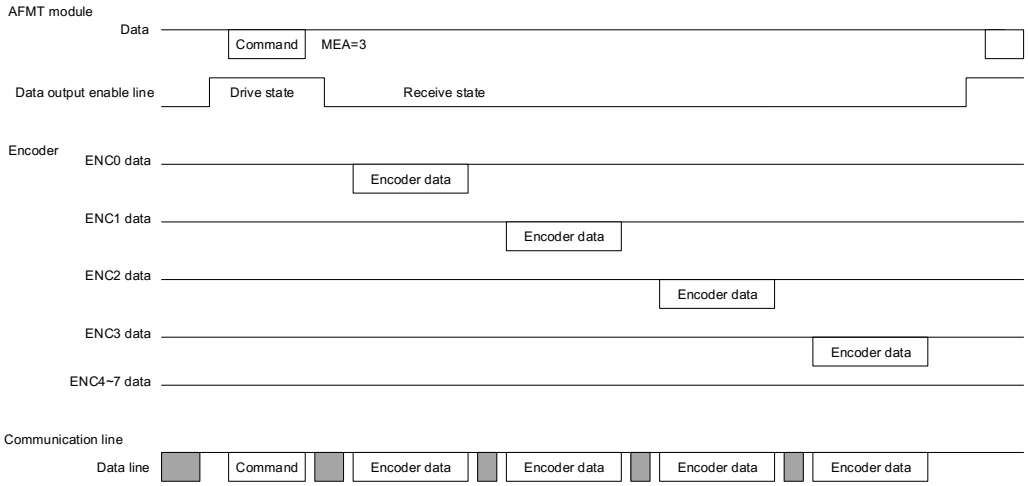
Figure 38-2. Timing of individual transmission mode



Broadcast transmission mode

In broadcast transmission mode, all slave encoders with addresses less than matching address will respond to the master command. The matching slave address is stored in the MEA[2:0] bits of the EDIM_AFMT_CMD register. The timing for serial data transmission and reception in broadcast transmission mode is shown in *Figure 38-3. Timing of broadcast transmission mode*.

Figure 38-3. Timing of broadcast transmission mode



Received data

All slave encoder data is stored in EDIM_AFMT_ENCnRDATAm registers. Each encoder has 3 data registers (EDIM_AFMT_ENCnRDATA0/1/2). For different A-format commands, the data in these registers has different meanings, as shown in *Table 38-4. Received data in EDIM_AFMT_ENCnRDATAm register*.

Table 38-4. Received data in EDIM_AFMT_ENCnRDATAm register

Frame name	Register	[31:24]	[23:16]	[15:8]	[7:0]
CDF0 CDF4	ENCnRDATA0	-	-	{CS[3:0], ES[3:0]}	{CC[4:0], EA[2:0]}
	ENCnRDATA1	ABS 40-bit full data (ABS[31:0])			
	ENCnRDATA2	-	-	-	ABS 40-bit full data (ABS[39:32])
CDF1 CDF5	ENCnRDATA0	-	-	{CS[3:0], ES[3:0]}	{CC[4:0], EA[2:0]}
	ENCnRDATA1	-	ABS lower 24-bit data (ABS[23:0])		
CDF2 CDF6	ENCnRDATA0	-	-	{CS[3:0], ES[3:0]}	{CC[4:0], EA[2:0]}
	ENCnRDATA1	-	ABS upper 24-bit data (ABS[39:16])		
CDF3	ENCnRDATA0	-	-	{CS[3:0],	{CC[4:0],

Frame name	Register	[31:24]	[23:16]	[15:8]	[7:0]
CDF7-12				ES[3:0]}	EA[2:0]}
	ENCnRDATA1	-	-	Status code (ALM[15:0])	
CDF13 CDF14	ENCnRDATA0	-	-	{CS[3:0], ES[3:0]}	{CC[4:0], EA[2:0]}
	ENCnRDATA1	-	8-bit encoder EEPROM address	16-bit encoder EEPROM data	
CDF15	ENCnRDATA0	-	-	{CS[3:0], ES[3:0]}	{CC[4:0], EA[2:0]}
	ENCnRDATA1	-	-	10-bit temperature data (DB[9:0])	
CDF16-20	ENCnRDATA0	-	-	{CS[3:0], ES[3:0]}	{CC[4:0], EA[2:0]}
	ENCnRDATA1	-	24-bit ID code (ABS[23:0])		
CDF21 CDF22	ENCnRDATA0	{CS[3:1], 0, ES, EA[2:0]}	ABS lower 17-bit data (ABS[16:0])		
CDF27 CDF28	ENCnRDATA0	-	-	{CS[3:0], ES[3:0]}	{CC[4:0], EA[2:0]}
	ENCnRDATA1	-	ABS lower 24-bit data (ABS[23:0])		
	ENCnRDATA2	-	-	Status code (ALM[15:0])	
CDF29 CDF30	ENCnRDATA0	-	-	{CS[3:0], ES[3:0]}	{CC[4:0], EA[2:0]}
	ENCnRDATA1	-	ABS lower 24-bit data (ABS[23:0])		
	ENCnRDATA2	-	-	10-bit temperature data (DB[9:0])	

Note:

1. '-' is 0.
2. CS[3:0] indicates four types of communication errors, as shown in *Table 38-5. Communication state flag*.

Table 38-5. Communication state flag

Bit	Function
CS[3]	When CS[3] = 1, it indicates transmitted CRC is not matched with CRC calculated from received data.
CS[2]	When CS[2] = 1, it indicates encoder data does not return within the predefined time after command code transmission. If a communication timeout error interrupt is generated, please check CS[2] for all EDIM_AFMT_ENCnRDATA0 registers to identify which slave encoder occurred timeout error.
CS[1]	When CS[1] = 1, It indicates a mismatch in the start bit and stop bit for the field, a lack of receive data length, or a sync code mismatch in the IF field.

Bit	Function
CS[0]	When CS[0] = 1, it indicates transmitted command is not matched with received command.

3. ES[3:0], EA[3:0], and ALM[15:0] are the status bits, encoder address, and status code values received from the slave encoder, as shown in *Table 38-6. Encoder state flag*.

Table 38-6. Encoder state flag

Type	Bit	Function
ES[3:0]/ES	ES[3]	STERR+PSERR+MTERR+INCERR
	ES[2]	OVSPD+MEMERR+OVTEMP+OVFLOW
	ES[1]	BATT
	ES[0]	BUSY+MEMBUSY
	ES	ES[0]+ES[1]+ES[2]+ES[3] (only for CDF21-22)
ALM[15:0]	ALM[15]	0 fixed
	ALM[14]	0 fixed
	ALM[13]	0 fixed
	ALM[12]	0 fixed
	ALM[11]	0 fixed
	ALM[10]	INCERR
	ALM[9]	OVTEMP
	ALM[8]	MEMBUSY
	ALM[7]	BUSY
	ALM[6]	PSERR
	ALM[5]	STERR
	ALM[4]	MEMERR
	ALM[3]	OVSPD
	ALM[2]	OVFLOW
	ALM[1]	MTERR (IBERR)
	ALM[0]	BATT (IBALARM)

Note: Detail information please refer to A-format protocol specification.

Command transmission process

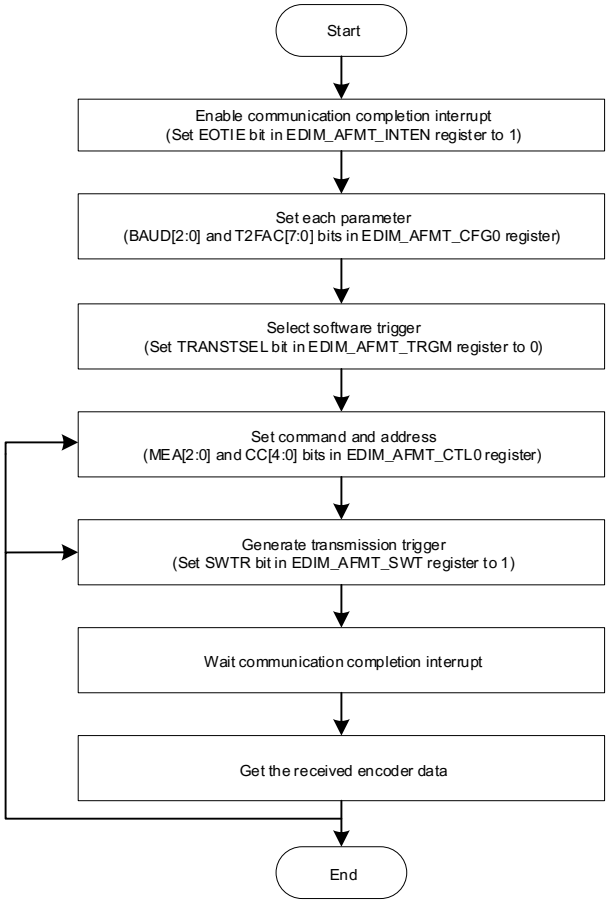
After writing commands and memory data, the command transmission starts with a software or external hardware trigger, based on the TRANSTSEL bit in EDIM_AFMT_TRGM register.

Startup by software trigger

When TRANSTSEL bit in the EDIM_AFMT_TRGM register is set to 0, the software trigger is selected.

In this mode, when SWTR bit in the EDIM_AFMT_SWT register is set to 1, command data transmission starts. The command transmission process is shown in *Figure 38-4. Command transmission process (software trigger)*.

Figure 38-4. Command transmission process (software trigger)

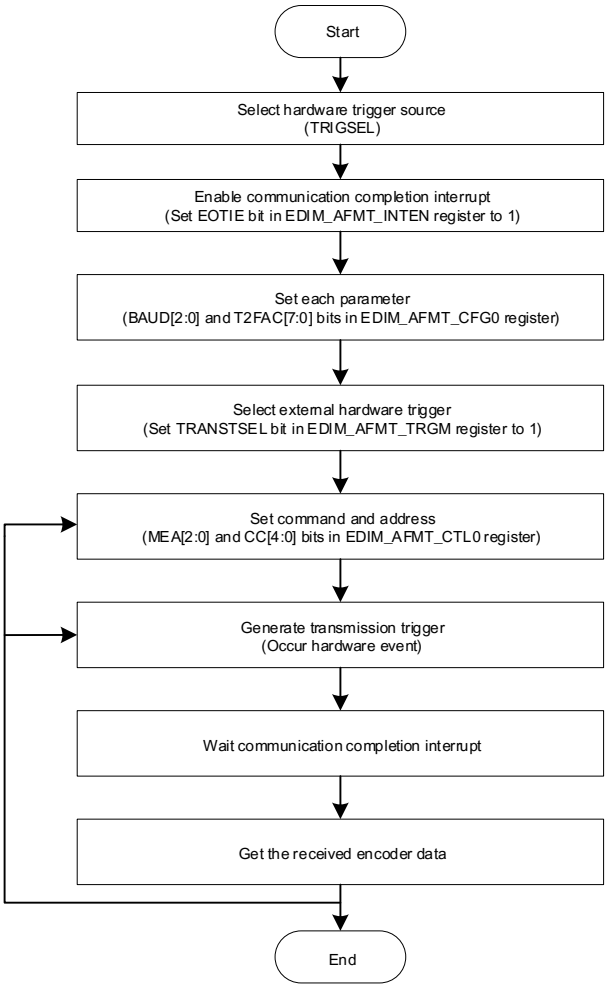


Startup by external hardware trigger

When TRANSTSEL bit in the EDIM_AFMT_TRGM register is set to 1, the external hardware trigger is selected.

In this mode, the command transmission can be triggered by TRIGSEL. The command transmission process is shown in ***Figure 38-5. Command transmission process (external hardware trigger)***.

Figure 38-5. Command transmission process (external hardware trigger)



DMA mode

In some situations, position value requests from the encoder may be frequent, and the CPU may not be able to read out the received data in time. In such case, users can configure DFEN bit and DFENx (x = 0…3) bits in EDIM_AFMT_CFG register to enable DMA request. A DMA request will be triggered after active encoder data is stored in FIFO. The total DMA request number depends on the number of active encoder and the configuration of DFENx (x = 0…3) bit. The DMA request is reset by hardware when the circuit receives an acknowledgment from DMA.

The DMA request is controlled by a DMA interface that includes a synchronized FIFO. The FIFO level is 3*4 bytes. When DFENx (x = 0…2) bit in EDIM_AFMT_CFG register is set to 1, the received data is not only stored in the corresponding data registers but also in a FIFO structure, which can be accessed via the EDIM_AFMT_FIFODATA register. User can readout data of EDIM_AFMT_FIFODATA register by DMA.

For example, if users wish to read EDIM_AFMT_ENC0RDATA0 and EDIM_AFMT_ENC1RDATA0 of the active encoders 0 and 1 by DMA. DFEN bit and DFEN0

bit in EDIM_AFMT_CFG register should be set to 1. The data of EDIM_AFMT_ENC0RDATA0 and EDIM_AFMT_ENC1RDATA0 are also transferred to FIFO. When data in the EDIM_AFMT_ENC0RDATA0 and EDIM_AFMT_ENC1RDATA0 registers is stored into the FIFO, it triggers a DMA request for each. If EDIM_AFMT_FIFODATA register address is set as DMA source address, DMA will transfer the data of EDIM_ AFMT_FIFODATA register.

If FIFO is overrun or underrun, FIFOOVRIF and FIFOUNRIF bits are set to 1 in EDIM_AFMT_STAT register. Overrun and underrun don't affect data reading. But the data read out is not reliable. FIFOFH bit in EDIM_AFMT_CFG register is set to 1, which can execute to flush FIFO.

If DMA acknowledge doesn't come back to AFMT module within 1024 PCLK1 cycles, DMATOIF bit in EDIM_AFMT_STAT register will be set to 1 and DMA request will be cleared by hardware.

AFMT interrupts

The interrupt can be produced on one of the events:

- End of communication
- Communication timeout detection error
- FIFO overrun, underrun
- DMA timeout error

When new encoder data is received and stored in EDIM_AFMT_ENCnRDATAm registers (for n = 0…7 and m = 0…2), and the whole receiving and sending process is completed, the end of communication bit (EOC) in EDIM_AFMT_STAT register is set to 0. The EOC bit is automatically set to 1 as soon as the first output data is read.

When encoder data is not returned within the predefined time (10 us) for a request, the communication timeout error occurs and TOIF bit in EDIM_AFMT_STAT register is set to 1.

38.1.5. Registers definition

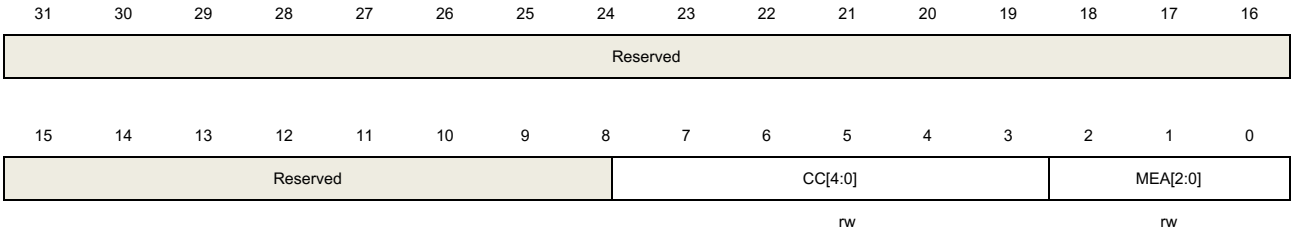
AFMT base address: 0x4000 C400

AFMT command register (EDIM_AFMT_CMD)

Address offset: 0x80

Reset value: 0x0000 0000

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



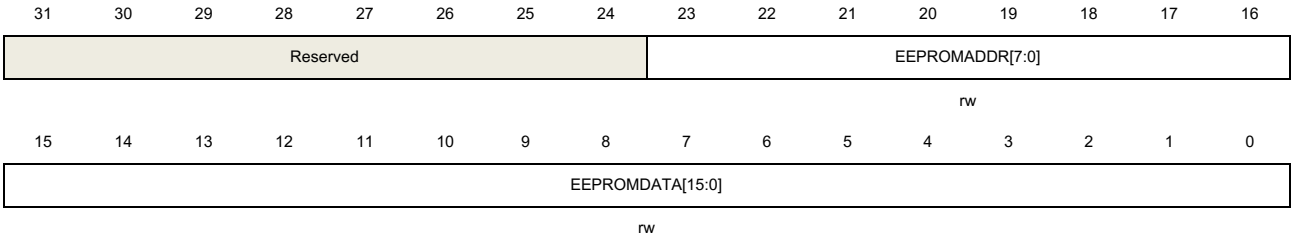
Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value
7:3	CC[4:0]	Encoder receive communication command code of A-format The values of these bits are as shown in ***Table 38-3. A-format command***.
2:0	MEA[2:0]	Matching address of encoder The values of these bits must match the address of the encoder in communication.

AFMT EEPROM register (EDIM_AFMT_EEPROM)

Address offset: 0x84

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	EEPROMADDR[7:0]	EEPROM accessing address of encoder When CC[4:0] = 5'b01101 or CC[4:0] = 5'b01110 in the EDIM_AFMT_CMD register, these bits must be written to the EEPROM address of encoder by software.

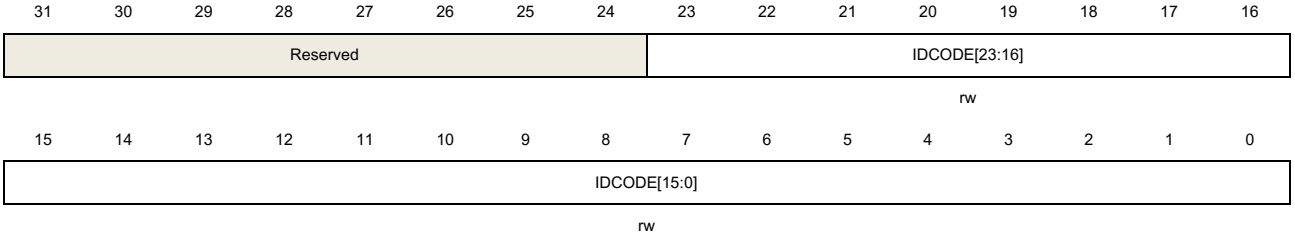
15:0	EEPROMDATA[15:0]	Data to be writtern into EEPROM of encoder When CC[4:0] = 5'b01110 in the EDIM_AFMT_CMD register, these bits must be written to the EEPROM data of encoder by software.

AFMT ID code register (EDIM_AFMT_IDCODE)

Address offset: 0x88

Reset value: 0x0000 0000

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



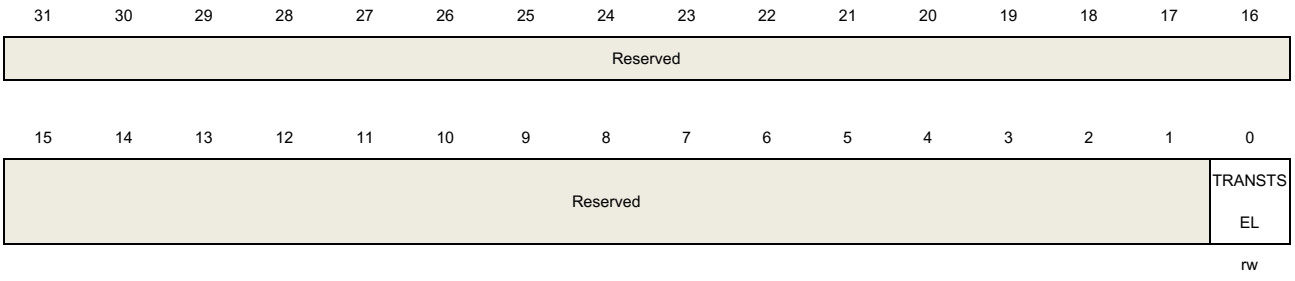
Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value
23:0	IDCODE[24:0]	ID code for communication When CC[4:0] = 5'b10010 or CC[4:0] = 5'b10011 in the EDIM_AFMT_CMD register, ID code must be written in IDCODE[23:0] bits by software.

AFMT trigger mode register (EDIM_AFMT_TRGM)

Address offset: 0x8C

Reset value: 0x0000 0000

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



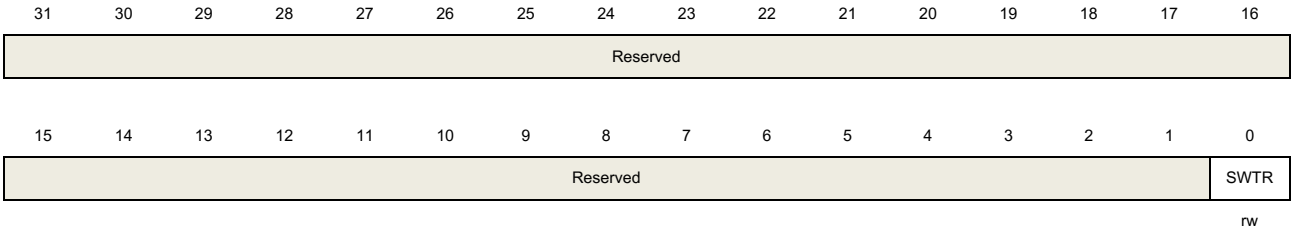
Bits	Fields	Descriptions
31:1	Reserved	Must be kept at reset value
0	TRANSTSEL	Transmission start trigger selection 0: Transmittion trigger is input from EDIM_AFMT_SWT 1: Transmittion trigger is input from TRIGSEL

AFMT software trigger register (EDIM_AFMT_SWT)

Address offset: 0x90

Reset value: 0x0000 0000

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



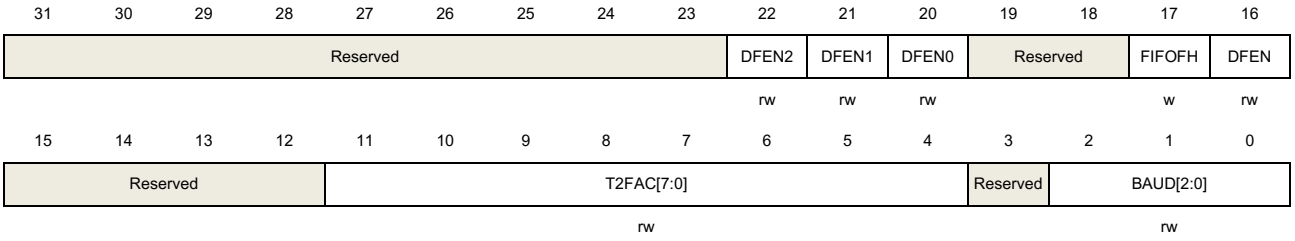
Bits	Fields	Descriptions
31:1	Reserved	Must be kept at reset value
0	SWTR	Data transmission software trigger, cleared by hardware 0: Software trigger disabled 1: Software trigger enabled This bit is set by software to generate a transmission trigger.

AFMT configuration register (EDIM_AFMT_CFG)

Address offset: 0x94

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:23	Reserved	Must be kept at reset value
22	DFEN2	DMA request and FIFO enable for EDIM_AFMT_ENCnRDATA2 register of active encoder 0: DMA request and FIFO disabled 1: DMA request and FIFO enabled If DFEN2 = 1, the data of EDIM_AFMT_ENCnRDATA2 of active encoder n is also transferred to FIFO. A DMA request will be generated after the data is stored in FIFO. The total DMA request number depends on the number of active encoder.

N21	DFEN1	DMA request and FIFO enable for EDIM_AFMT_ENCnRDATA1 register of active encoder 0: DMA request and FIFO disabled 1: DMA request and FIFO enabled If DFEN1 = 1, the data of EDIM_AFMT_ENCnRDATA1 of active encoder n is also transferred to FIFO. A DMA request will be generated after the data is stored in FIFO. The total DMA request number depends on the number of active encoder.
N20	DFEN0	DMA request and FIFO enable for EDIM_AFMT_ENCnRDATA0 register of active encoder n 0: DMA request and FIFO disabled 1: DMA request and FIFO enabled If DFEN0 = 1, the data of EDIM_AFMT_ENCnRDATA0 of active encoder n is also transferred to FIFO. A DMA request will be generated after the data is stored in FIFO. The total DMA request number depends on the number of active encoder.
19:18	Reserved	Must be kept at reset value
17	FIFOFH	FIFO flush 0: No action 1: Clear the FIFO data
16	DFEN	DMA request and FIFO enable 0: DMA request disabled 1: DMA request enabled
15:12	Reserved	Must be kept at reset value
11:4	T2FAC[7:0]	Time step factor of t_2 t_2 = 1 μs + N × 0.25 μs 00000000: N = 0 00000001: N = 1 … 11111111: N = 255
3	Reserved	Must be kept at reset value
2:0	BAUD[2:0]	Communication baud rate configuration 000: 2.5 Mbits/s 001: 4 Mbits/s 010: 6.67 Mbits/s 011: 8 Mbits/s 100: 16 Mbits/s Others: Reserved

AFMT status register (EDIM_AFMT_STAT)

Address offset: 0x98

Reset value: 0x0000 0003

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved													DMATOIF	FIFOUNRIF	FIFOOVRIF
													rc_w1	rc_w1	rc_w1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													TOIF	EOC	CBUSYF
													rc_w1	r	r

Bits	Fields	Descriptions
31:19	Reserved	Must be kept at reset value
18	DMATOIF	DMA request timeout error interrupt flag 0: No DMA request timeout error occurred 1: DMA request timeout error occurred
17	FIFOUNRIF	FIFO underrun error interrupt flag 0: No underrun error occurred 1: Underrun error occurred
16	FIFOOVRIF	FIFO overrun error interrupt flag 0: No overrun error occurred 1: Overrun error occurred
15:3	Reserved	Must be kept at reset value
2	TOIF	Timeout error interrupt flag 0: No timeout error occurred 1: Timeout error occurred
1	EOC	End of communitcation 0: End of communitcation 1: Communitcation status is idle or busy. This bit is set to 0 by hardware and is automatically set to 1 after reading the received data in EDIM_AFMT_ENCnRDATAm register.
0	CBUSYF	Communitcation busy flag 0: Busy 1: Idle

AFMT interrupt enable register (EDIM_AFMT_INTEN)

Address offset: 0x9C

Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:19	Reserved	Must be kept at reset value
18	DMATOIE	Interrupt enable of DMATOIF 0: Interrupt disabled 1: Interrupt enabled
17	FIFOUNRIE	Interrupt enable of FIFOUNRIF 0: Interrupt disabled 1: Interrupt enabled
16	FIFOOVRIE	Interrupt enable of FIFOOVRIF 0: Interrupt disabled 1: Interrupt enabled
15:3	Reserved	Must be kept at reset value
2	TOIE	Interrupt enable of TOIF 0: Interrupt disabled 1: Interrupt enabled
1	EOCIE	Interrupt enable of EOC 0: Interrupt disabled 1: Interrupt enabled
0	Reserved	Must be kept at reset value

AFMT FIFO data register (EDIM_AFMT_FIFODATA)

Address offset: 0x140

Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:0	FIFODATA[31:0]	FIFO data from EDIM_AFMT_ENCnRDATAm (n = 0…7, m = 0…2) registers.

AFMT encoder n received data m register (EDIM_AFMT_ENCnRDATAm) (n=0...7, m=0...2)

Address offset: 0x04 * m + 0x10 * n

Reset value: 0x0000 0000

This register 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[31:16]															
r															

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

Bits	Fields	Descriptions
31:0	RDATA[31:0]	Received data from encoder n In different command code, RDATA contains different information

38.2. T-format encoder interface module (TFMT)

38.2.1. Overview

The T-format encoder interface module (TFMT) serves as a master to communicate with encoders that support the Tamagawa format.

38.2.2. Characteristics

- Support communication with Tamagawa smart-abs encoder.
- Monitor the status of the communication.
  11. Encoder error
  12. Timeout detection
  13. Transmission/receive busy
  14. Communication completion
  15. Received frame error
- Interrupt generation.
- Support DMA mode.
- Baud rate: 2.5 Mbits/s.

38.2.3. Block diagram

Figure 38-6. TFMT module block diagram

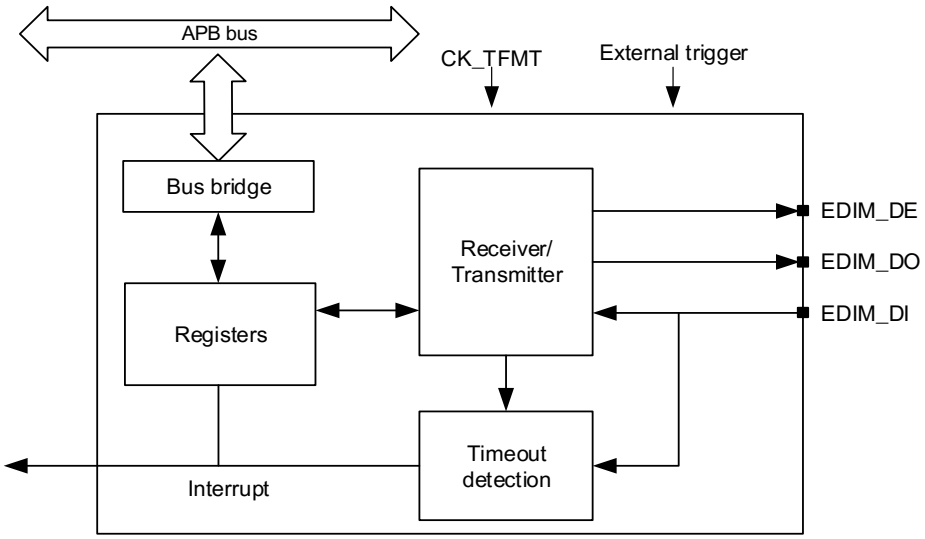


Figure 38-6. TFMT module block diagram provides details of the internal configuration of the TFMT module. *Table 38-7. TFMT pins definition* gives the TFMT pin description.

Table 38-7. TFMT pins definition

Name	Description
EDIM_DE	TFMT transceiver control signal
EDIM_DO	TFMT data output
EDIM_DI	TFMT data input

38.2.4. Function overview

TFMT module is a communication interface used to transmit control frames to encoder, receive data and status frames from encoder.

TFMT clock

The function clock for TFMT module is CK_TFMT, with a frequency of 80MHz. The CPU accesses the TFMT module registers through the APB bus, which has a maximum frequency of 150MHz. A bus bridge is designed to synchronize the APB clock PCLK1 and CK_TFMT.

Note: Frequency of CK_TFMT affects the transmission baud rate. When the MCU samples data returned by the encoder, it adjusts the sampling point at the data transition edge, which increases the tolerance to the baud rate error of the slave encoder. However, when the MCU transmits data, errors in the clock frequency can affect the baud rate, which may cause errors when the slave encoders receive data. The clock frequency and baud rate can have a 2% error margin.

Tamagawa format data ID code

Tamagawa format protocol supports different data ID codes, which consist of 4 command bits and one parity bit, stored in the CC[4:0] of EDIM_TFMT_DATAID register. *Table 38-8. Tamagawa format Data ID code* shows data ID code transmitted to encoder in control field (CF).

Table 38-8. Tamagawa format Data ID code

Function	Data ID	Code				Parity
		CC[3]	CC[2]	CC[1]	CC[0]	CC[4]
Data readout	Data ID 0	0	0	0	0	0
	Data ID 1	0	0	0	1	1
	Data ID 2	0	0	1	0	1
	Data ID 3	0	0	1	1	0
Writing to EEPROM	Data ID 6	0	1	1	0	0
Readout from EEPROM	Data ID D	1	1	0	1	1
Reset	Data ID 7	0	1	1	1	1
	Data ID 8	1	0	0	0	1
	Data ID C	1	1	0	0	0

Received data

Frames received from encoder is stored in EDIM_TFMT_RDATAx registers, ***Table 38-9. Received data in EDIM_TFMT_RDATAx register*** shows the data stored in EDIM_TFMT_RDATAx registers.

Table 38-9. Received data in EDIM_TFMT_RDATAx register

Data ID code	Register	[31:24]	[23:16]	[15:8]	[7:0]
Data ID 0 Data ID 7 Data ID 8 Data ID C	EDIM_TFMT_RDATA0	-	-	-	CC[3:0]
	EDIM_TFMT_RDATA1	ABS2	ABS1	ABS0	CD[7:0]
	EDIM_TFMT_RDATA2	-	-	-	SF
	EDIM_TFMT_RDATA3	-	-	-	-
Data ID 1	EDIM_TFMT_RDATA0	-	-	-	CC[3:0]
	EDIM_TFMT_RDATA1	ABM2	ABM1	ABM0	CD[7:0]
	EDIM_TFMT_RDATA2	-	-	-	SF
	EDIM_TFMT_RDATA3	-	-	-	-
Data ID 2	EDIM_TFMT_RDATA0	-	-	-	CC[3:0]
	EDIM_TFMT_RDATA1	-	SF	ENID	CD[7:0]
	EDIM_TFMT_RDATA2	-	-	-	-
	EDIM_TFMT_RDATA3	-	-	-	-
Data ID 3	EDIM_TFMT_RDATA0	-	-	-	CC[3:0]
	EDIM_TFMT_RDATA1	ABS2	ABS1	ABS0	CD[7:0]
	EDIM_TFMT_RDATA2	ABM2	ABM1	ABM0	ENID
	EDIM_TFMT_RDATA3	-	-	SF	ALMC
Data ID 6 Data ID D	EDIM_TFMT_RDATA0	-	-	-	CC[3:0]
	EDIM_TFMT_RDATA1	-	DATA[7:0]	ADDR[6:0]	CD[7:0]
	EDIM_TFMT_RDATA2	-	-	-	-
	EDIM_TFMT_RDATA3	-	-	-	-

Note:

1. '-'is 0.
2. ABS0-2 are absolute data in one revolution, with a length of 24 bits, where ABS0 is the lower byte and ABS2 is the higher byte. ABM0-2 are multi-turn, with a length of 24 bits, where ABM0 is the lower byte and ABM2 is the higher byte. SF is the status field received from the encoder. It is stored in bit-reversed order. ENID is the encoder ID. ALMC is the encoder error. Detail information please refer to Tamagawa format protocol specification.
3. DATA[7:0] and ADDR[6:0] are the received EEPROM data from the encoder.
4. CD[7:0] contains data ID code and error information, as shown in ***Table 38-10. Data ID code and error information***.

Table 38-10. Data ID code and error information

Bit	Function
CD[7:3]	CC[4:0]
CD[2]	CD[2] is always 0
CD[1]	When CD[1] = 1, It indicates Data ID code received is not equal to transmitted.
CD[0]	When CD[0] = 1, it indicates error occur when receiving.

Command transmission trigger

Data ID code, EEPROM address and data should be written into EDIM_TFMT_DATAID and EDIM_TFMT_EEPROM registers. After configuring those registers, command transmission will start when TFMT module get a start trigger.

Transmission start trigger can be selected by TRANSTSEL bit in EDIM_TFMT_CFG. When TRANSTSEL bit is set to 1, transmittion trigger is selected external hardware signal from TRIGSEL.

When TRANSTSEL bit is set to 0, transmittion trigger is selected software signal. In this mode, when SWTR bit in the EDIM_TFMT_SWT register is set to 1, command data transmission starts.

DMA mode

In some situations, position value requests from the encoder may be frequent, and the CPU may not be able to read out the received data in time. In such case, users can configure DFEN bit and DFENx (x = 0…3) bit in EDIM_TFMT_CFG register to enable DMA request. If DMA is enabled, DMA requests will be triggered k times by hardware when the EOC bit in the EDIM_TFMT_STAT register is set to 1. K depends on the configuration of DFENx (x = 0…3) bit. The DMA request is reset by hardware when the circuit receives an acknowledgment from DMA.

The DMA request is controlled by a DMA interface that includes a synchronized FIFO. The FIFO level is 4*4 bytes. When DFENx (x = 0…3) bit in EDIM_TFMT_CFG register is set to 1, the received data is not only stored in the corresponding data registers but also in a FIFO structure, which can be accessed via the EDIM_TFMT_FIFODATA register. User can readout data of EDIM_TFMT_FIFODATA register by DMA.

For example, if users wish to read EDIM_TFMT_RDATA0 and EDIM_TFMT_RDATA3 registers by DMA, the DFEN bit, DFEN0 bit and DFEN3 bit in EDIM_TFMT_CFG register should be set to 1. The data of EDIM_TFMT_RDATA0 and EDIM_TFMT_RDATA3 are also transferred to FIFO. The DMA request will be triggered 2 times by hardware when EOC is set to 1. If EDIM_TFMT_FIFODATA register address is set as DMA source address, DMA will transfer the data of EDIM_TFMT_FIFODATA register.

If FIFO is overrun or underrun, FIFOOVRIF and FIFOUNRIF bits are set to 1 in EDIM_TFMT_STAT register. Overrun and underrun don't affect data reading. But the data read out is not reliable. FIFOFH bit in EDIM_TFMT_CFG register is set to 1, which can

execute to flush FIFO.

If DMA acknowledge doesn't come back to TFMT module within 1024 PCLK1 cycles, DMATOIF bit in EDIM_TFMT_STAT register will be set to 1 and DMA request will be cleared by hardware.

TFMT interrupts

The interrupt can be produced on one of the events:

- End of communication
- Communication timeout error
- CRC failure
- Mismatched Data ID codes
- Parity error interrupt flag in received control field of frame
- Start bit error or delimiter error in received frame
- EEPROM busy
- Delimiter error interrupt flag in request frame
- Parity error interrupt flag in request frame
- Encoder counting error
- Encoder alarm
- FIFO overrun, underrun
- DMA timeout error

Before beginning another frame transfer, the EDIM_TFMT_RDATAx (x = 0…3) registers must be read to clear the communication completion interrupt.

When encoder data is not returned within the predefined time (10 us) for a request, the communication timeout error occurs and TOIF bit in EDIM_TFMT_STAT register is set to 1. When next communication start, the timeout error interrupt is cleared.

When a CRC failure, parity check failure, or start bit/delimiter mismatch occurs, received data is not stored in EDIM_TRDATAx registers. However, if the received Data ID code doesn't match the transmitted one, the data is still stored in EDIM_TFMT_RDATAx (x = 0…3) registers, and CCIF bit in EDIM_TFMT_STAT register is set to 1.

38.2.5. Registers definition

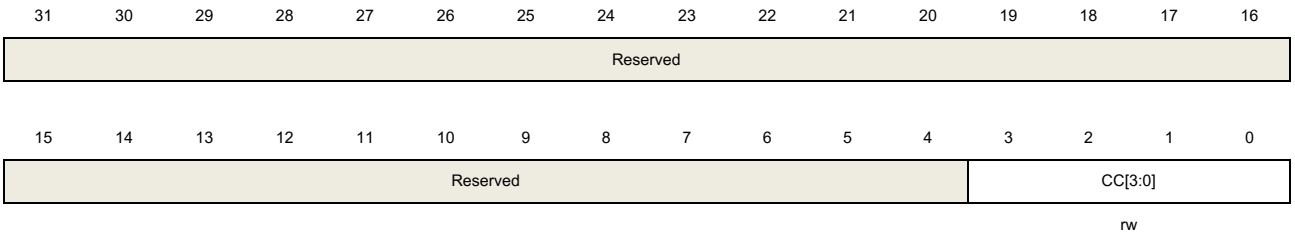
TFMT base address: 0x4000 C800

TFMT ID code register (EDIM_TFMT_DATAID)

Address offset: 0x00

Reset value: 0x0000 0000

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



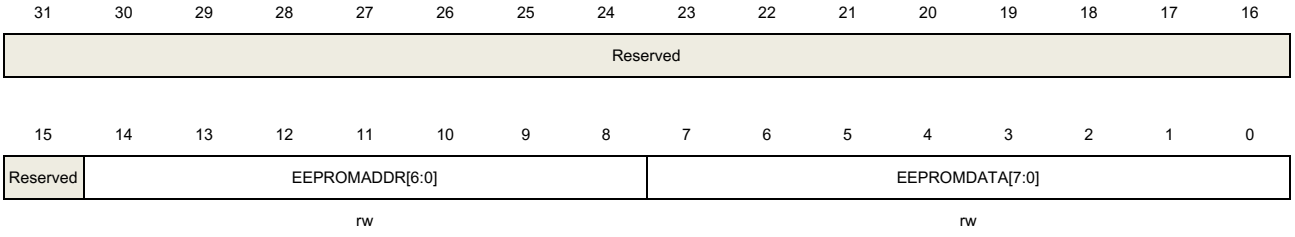
Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value.
3:0	CC[3:0]	Data ID code 0000 - 0011: Data readout 0110: Writing to EEPROM 0111: Reset of all error 1000: Reset of one revolution data 1100: Reset of multi-turn data and all error 1101: Readout from EEPROM Others: Reserved Tamagawa format mast uses Data ID code to control Tamagawa smart-abs encoder. The Data field send back to Tamagawa master is up to Data ID code.

TFMT EEPROM register (EDIM_TFMT_EEPROM)

Address offset: 0x04

Reset value: 0x0000 0000

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



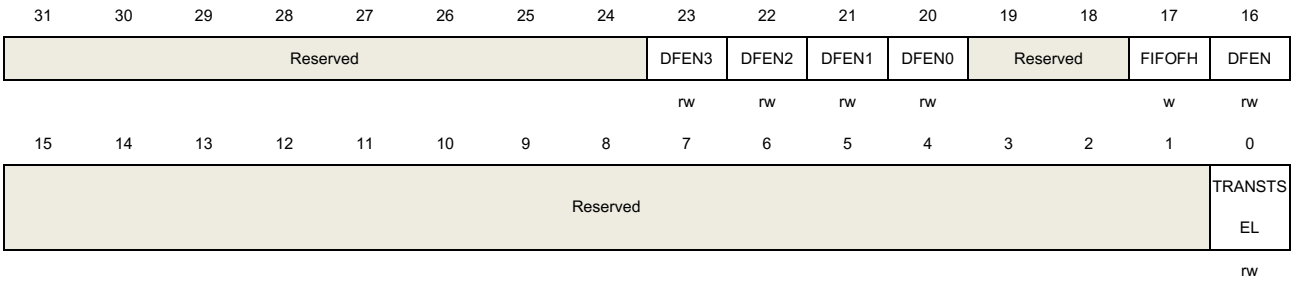
Bits	Fields	Descriptions
31:15	Reserved	Must be kept at reset value
14:8	EEPROMADDR[6:0]	EEPROM accessing address of encoder
7:0	EEPROMDATA[7:0]	Data to be written into EEPROM of encoder

TFMT configuration register (EDIM_TFMT_CFG)

Address offset: 0x08

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	DFEN3	DMA request and FIFO enable for EDIM_TFMT_RDATA3 register 0: DMA request and FIFO disabled 1: DMA request and FIFO enabled If DFEN3 = 1, the data of EDIM_TFMT_RDATA3 is transferred to FIFO. A DMA requests will be generated when EOC is set to 1.
22	DFEN2	DMA request and FIFO enable for EDIM_TFMT_RDATA2 register 0: DMA request and FIFO disabled 1: DMA request and FIFO enabled If DFEN2 = 1, the data of EDIM_TFMT_RDATA2 is transferred to FIFO. A DMA requests will be generated when EOC is set to 1.
21	DFEN1	DMA request and FIFO enable for EDIM_TFMT_RDATA1 register 0: DMA request and FIFO disabled 1: DMA request and FIFO enabled If DFEN1 = 1, the data of EDIM_TFMT_RDATA1 is transferred to FIFO. A DMA requests will be generated when EOC is set to 1.
20	DFEN0	DMA request and FIFO enable for EDIM_TFMT_RDATA0 register 0: DMA request and FIFO disabled 1: DMA request and FIFO enabled If DFEN0 = 1, the data of EDIM_TFMT_RDATA0 is transferred to FIFO. A DMA

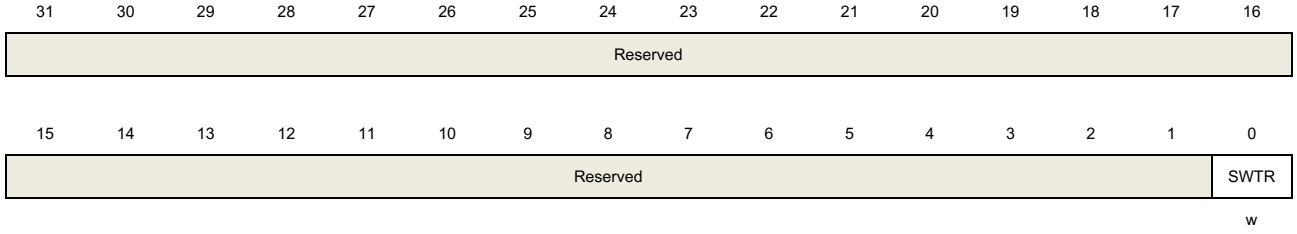
		requests will be generated when EOC is set to 1.
19:18	Reserved	Must be kept at reset value
17	FIFOFH	FIFO flush 0: No action 1: Clear the FIFO data
16	DFEN	DMA request and FIFO enable 0: DMA request disabled 1: DMA request enabled
15:1	Reserved	Must be kept at reset value
0	TRANSTSEL	Transmission start trigger selection 0: Transmittion trigger is input from EDIM_TFMT_SWT 1: Transmittion trigger is input from TRIGSEL

TFMT software trigger register (EDIM_TFMT_SWT)

Address offset: 0x0C

Reset value: 0x0000 0000

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



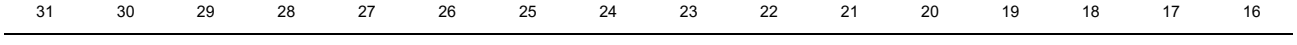
Bits	Fields	Descriptions
31:1	Reserved	Must be kept at reset value.
0	SWTR	Data transmission software trigger, cleared by hardware 0: Software trigger disabled 1: Software trigger enabled This bit is set by software to generate a transmission trigger.

TFMT status register (EDIM_TFMT_STAT)

Address offset: 0x10

Reset value: 0x0000 0000

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



Reserved													DMATOIF	FIFOUNRIF	FIFOOVRIF
													rc_w1	rc_w1	rc_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			CNTIF	ALRMIF	PREQIF	DELIF	EEPROMBUSYIF	CCIF	PCTLIF	SDELIF	CRCIF	TOIF	RBUSYF	TBUSYF	EOC
			rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	r	r	r

Bits	Fields	Descriptions
31:19	Reserved	Must be kept at reset value
18	DMATOIF	DMA request timeout error interrupt flag 0: No DMA request timeout error occurred 1: Timeout error occurred
17	FIFOUNRIF	FIFO underrun error interrupt flag 0: No FIFO underrun error occurred 1: Underrun error occurred
16	FIFOOVRIF	FIFO overrun error interrupt flag 0: No FIFO overrun error occurred 1: Overrun error occurred
15:13	Reserved	Must be kept at reset value
12	CNTIF	Counting error interrupt flag of encoder 0: No counting error occurred 1: Counting error occurred
11	ALRMIF	Alarm interrupt flag of encoder 0: No alarm 1: Over-heat, Multi-turn error, battery error and battery alarm occur in encoder When this bit is set, configure CC[3:0] = 0011 to confirm the contents of alarm in the data frame.
10	PREQIF	Parity error interrupt flag in request frame 0: No parity error n request frame occurred 1: Parity error in request frame When this bit is set, the received data is considered invalid, and the same request frame should be transmitted again.
9	DELIF	Delimiter error interrupt flag in request frame 0: No delimiter error occurred 1: Delimiter error occurred When this bit is set, the received data is considered invalid, and the same request frame should be transmitted again.
8	EEPROMBUSYIF	EEPROM busy interrupt flag

		0: EEPROM access is completed 1: EEPROM is busy, the process of writing to or readout from the EEPROM has not been completed.
7	CCIF	Data ID code error interrupt flag 0: No Data ID code error occurred 1: The Data ID code received does not match the Data ID code transmitted
6	PCTLIF	Parity error interrupt flag in received control field of frame 0: No parity error in received control field of frame occurred 1: Parity error in received control field
5	SDELIF	Start bit error interrupt flag or delimiter error flag in received frame 0: No Start bit error or delimiter error in received frame occured 1: Start bit error or delimiter error in received frame
4	CRCIF	CRC error interrupt flag in received frame 0: No CRC error in received frame occurred 1: CRC error in received frame
3	TOIF	Encoder transmission timeout error interrupt flag 0: No encoder transmission timeout error occurred 1: Encoder failed to start transmitting a frame within 200 us of receiving the request frame.
2	RBUSYF	Receive busy flag 0: Idle 1: Receiving from encoder
1	TBUSYF	Transmit busy flag 0: Idle 1: Transmitting to encoder
0	EOC	End of communication 0: TFMT module is idle or communicating with encoder 1: End of communication Set by hardware at the end of communication and cleared by hardware after readout received data.

TFMT interrupt enable register (EDIM_TFMT_INTEN)

Address offset: 0x14

Reset value: 0x0000 0000

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

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

															rw	rw	rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Reserved			CNTIE	ALRMIE	PREQIE	DELIE	EEPROM BUSYIE	CCIE	PCTLIE	SDELIE	CRCIE	TOIE	Reserved		EOCIE				
			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw			rw				

Bits	Fields	Descriptions
31:19	Reserved	Must be kept at reset value
18	DMATOIE	Interrupt enable of DMATOIF 0: Interrupt disabled 1: Interrupt enabled
17	FIFOUNRIE	Interrupt enable of FIFOUNRIF 0: Interrupt disabled 1: Interrupt enabled
16	FIFOOVRIE	Interrupt enable of FIFOOVRIF 0: Interrupt disabled 1: Interrupt enabled
15:13	Reserved	Must be kept at reset value
12	CNTIE	Interrupt enable for CNTIF 0: Interrupt disabled 1: Interrupt enabled
11	ALRMIE	Interrupt enable for ALRMIF 0: Interrupt disabled 1: Interrupt enabled
10	PREQIE	Interrupt enable for PREQIF 0: Interrupt disabled 1: Interrupt enabled
9	DELIE	Interrupt enable for DELIF 0: Interrupt disabled 1: Interrupt enabled
8	EEPROMBUSYIE	Interrupt enable for EEPROMBUSYIF 0: Interrupt disabled 1: Interrupt enabled
7	CCIE	Interrupt enable for CCIF 0: Interrupt disabled 1: Interrupt enabled
6	PCTLIE	Interrupt enable for PCTLIF 0: Interrupt disabled

		1: Interrupt enabled
5	SDELIE	Interrupt enable for SDELIF 0: Interrupt disabled 1: Interrupt enabled
4	CRCIE	Interrupt enable for CRCIF 0: Interrupt disabled 1: Interrupt enabled
3	TOIE	Interrupt enable for TOIF 0: Interrupt disabled 1: Interrupt enabled
2:1	Reserved	Must be kept at reset value.
0	EOCIE	Interrupt enable for EOC 0: Interrupt disabled 1: Interrupt enabled

TFMT received data register x (EDIM_TFMT_RDATAx) (x = 0...3)

Address offset: 0x18 + 0x04 * x (x = 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
RDATA[31:16]															
r															

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

Bits	Fields	Descriptions
31:0	RDATA[31:0]	Received data from encoder In different Data ID code, RDATA contains different information

TFMT FIFO data register (EDIM_TFMT_FIFODATA)

Address offset: 0x128

Reset value: 0x0000 0000

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

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

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

FIFODATA[15:0]
r

Bits	Fields	Descriptions
31:0	FIFODATA[31:0]	FIFO data from EDIM_TFMT_RDATAx (x = 0…3) registers.

TFMT status update mode register (EDIM_TFMT_STATUPMODE)

Address offset: 0x200

Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:1	Reserved	Must be kept at reset value.
0	STATUPMODE	Encoder status update mode 0: Dynamic update, encoder status update at the end of frame transmission 1: Static update, encoder status set by hardware and cleared by software **Note:** encoder status includes: CNTIF/ALRMIF/PREQIF/DELIF/EEPROMBUSYIF/CCIF/PCTLIF/SDELIF/CRCIF/TOIF

38.3. EnDat 2.2 encoder interface module (ENDAT)

38.3.1. Overview

The EnDat 2.2 encoder interface module (ENDAT) serves as a master to communicate with encoders that support the EnDat 2.2 interface.

38.3.2. Characteristics

- Support communication with encoder supporting EnDat 2.2 and EnDat 2.1.
- Transmission clock generator.
- Continuous clock mode.
- Configurable received data length.
- Cable propagation delay time compensation.
- Recovery time Ⅱ / Ⅲ .
- Support software trigger, internal timer trigger and external trigger.
- Reset selected additional information of encoder.
- Configurable digital filter for data input to avoid spike.
- Monitor the status of the communication.
  16. Encoder error
  17. TYPE Ⅰ / Ⅱ / Ⅲ error
  18. CRC
  19. Timeout detection
- Interrupt generation.
- Support DMA mode

38.3.3. Block diagram

Figure 38-7. ENDAT module block diagram

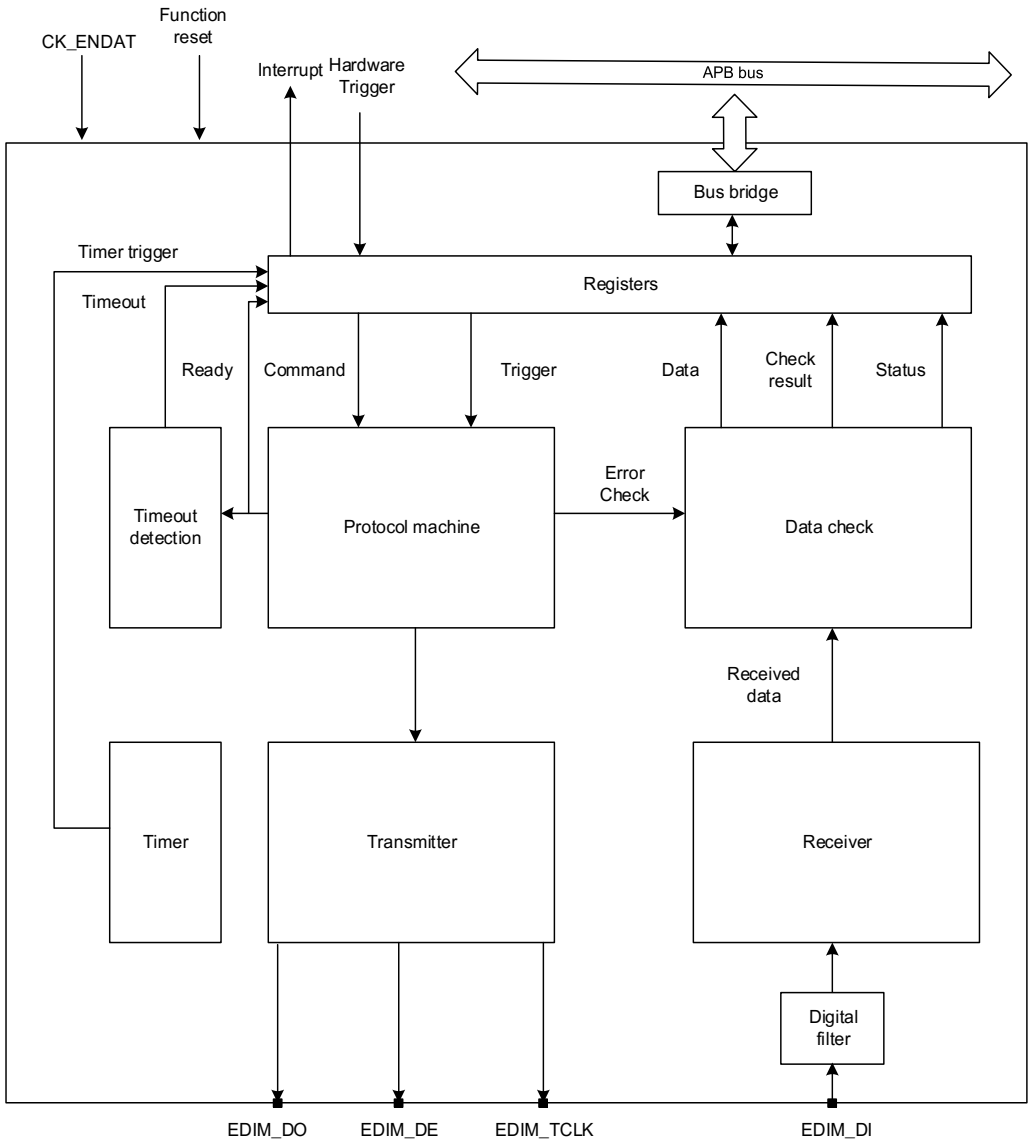


Figure 38-7. ENDAT module block diagram provides details of the internal configuration of the ENDAT module. *Table 38-11. ENDAT pins definition* gives the ENDAT pin description.

Table 38-11. ENDAT pins definition

Name	Description
EDIM_TCLK	ENDAT clock signal output
EDIM_DE	ENDAT transceiver control signal
EDIM_DO	ENDAT data output
EDIM_DI	ENDAT data input

38.3.4. Function overview

ENDAT module is a bidirectional serial interface. It starts a transmission by sending a transmission clock signal of a defined length. Along with the clock, a mode command and any other necessary data are sent to the encoder. The encoder then sends back the requested data to MCU. The direction of data transfer can switch several times during this process.

ENDAT clock

The function clock for ENDAT module is CK_ENDAT, with a frequency of 100MHz. The CPU accesses the ENDAT module registers through the APB bus, which has a maximum frequency of 150MHz. A bus bridge is designed to synchronize the APB clock PCLK1 and CK_ENDAT.

EnDat protocol command

EnDat command consists of CMD[5:0], MEMADDR[7:0] and CDATA[15:0], stored in EDIM_ENDAT_CMD register. *Table 38-12. EnDat command* shows command used in EnDat 2.1 and 2.2 protocol.

Table 38-12. EnDat command

Protocol	Function	CDM[5:0]	MEMADDR[7:0]	CDATA[15:0]
EnDat 2.1/2.2	Encoder to send position values	6'b000111	Reserved	Reserved
EnDat 2.1/2.2	Selection of memory area	6'b001110	MRS code	Reserved
EnDat 2.1/2.2	Encoder to receive parameters	6'b011100	Address	Parameters
EnDat 2.1/2.2	Encoder to send parameters	6'b100011	Address	Reserved
EnDat 2.1/2.2	Encoder to receive reset	6'b101010	Reserved	Reserved
EnDat 2.1/2.2	Encoder to send test values	6'b010101	Reserved	Reserved
EnDat 2.1/2.2	Encoder to receive test command	6'b110001	Port address	Reserved
EnDat 2.2	Encoder to send position values with additional information	6'b111000	Reserved	Reserved
EnDat 2.2	Encoder to send position values with additional information and selection of the	6'b001001	MRS code	Reserved

Protocol	Function	CDM[5:0]	MEMADDR[7:0]	CDATA[15:0]
	memory area			
EnDat 2.2	Encoder to send position values with additional information and receive parameters	6'b011011	Address	Parameters
EnDat 2.2	Encoder to send position values with additional information and parameters	6'b100100	Address	Reserved
EnDat 2.2	Encoder to send position values with additional information and receive test command	6'b110110	Port address	Reserved
EnDat 2.2	Encoder to send position values with additional information and receive error reset	6'b101101	Reserved	Reserved
EnDat 2.2	Encoder to receive communication command	6'b010010	Address	Instructions

Note:

1. EnDat 2.2 command sets are compatible with EnDat 2.1 command sets.
2. Details of MRS code, address, port address, instructions and more details of command please refer to EnDat protocol specification.

EnDat 2.2 command sets can select additional information. The additional information will be transferred back to MCU in next frame. When CDM[5:0] bits in EDIM_ENDAT_CMD register are set to 6'b001001, MRS code for selection (and deselection) of additional information determines the further progress of the protocol, i.e. whether and how many clocks will be sent to receive additional information.

Only values for the mode commands defined above are set into CDM[5:0] bits.

Received data

When EnDat 2.1 command is transmitted, RDATAL[31:0] and RDATAH[15:0] in EDIM_ENDAT_RDATA0 and EDIM_ENDAT_RDATA1 registers contain different data depend on the commands. When EnDat 2.2 command is transmitted, the position value is always stored into RDATAL[31:0] and RDATAH[15:0]. *Table 38-13. Received data in EDIM_ENDAT_RDATA0 and EDIM_ENDAT_RDATA1 registers* shows the received data in RDATAL[31:0] and RDATAH[15:0].

Table 38-13. Received data in EDIM_ENDAT_RDATA0 and EDIM_ENDAT_RDATA1 registers

Data type	RDATAH [15:8]	RDATAH [7:0]	RDATAL [31:24]	RDATAL [23:16]	RDATAL [15:8]	RDATAL [7:0]
Position value	MSB	…	…	…	…	DATA[7:0]
MRS code	-	-	-	MRS code	DATA[15:0]	
Address and parameter	-	-	-	Address	DATA[15:0]	
Port address and data	-	-	-	Port address	DATA[15:0]	
Test values	-	TESTVAL[39:0]				
Address and instruction	-	-	-	Address	DATA[15:0]	

Note:

1. '-' is invalid and is 0.
2. The position of the MSB shifts based on the length of the position value. The bits to the left of the MSB are not included in the position value.
3. The types, including MRS code, Address and parameter, Port address and data, Test values, are only applicable for EnDat 2.1 mode command.
4. The type of Address and instruction is exclusively for the case when CDM[5:0] equals 6'b010010.
5. DATA in MRS code, Address and parameter, Address and instruction represent receive parameter. TESTVAL in Test values type represents 40-bit test value received.

Command transmission trigger

Software trigger

The command transmission can be triggered by software. When SWTR bit in the EDIM_ENDAT_SWT register is set to 1, command transmission starts.

External Hardware trigger

When HWTRGEN bit in the EDIM_ENDAT_CFG0 register is set to 1, the external hardware trigger is enabled.

In this mode, the command transmission can be triggered by TRIGSEL.

APB bus has different clock frequency from CK_ENDAT. All bus write access has a synchronization time. External hardware trigger come in this time may be ignored or start a transmission which is not expected. For safety reasons, external hardware signal delay time, stored in HWTRGDELAY[7:0] in EDIM_ENDAT_CFG1 register, should be set to avoid this situation. The external hardware signal from TRIGSEL will trigger a new transmission after a

configurable hardware delay time.

Note: If both the internal timer and hardware trigger delay time are configured, then both the internal timer trigger and hardware trigger delay are disabled, and TRIGSEL signal will directly trigger transmission without delay. If only the hardware trigger delay time is configured, TRIGSEL signal will trigger transmission after a configured hardware delay time. If only the internal timer is configured, both the TRIGSEL signal and the internal timer signal can trigger data transmission, with the specific trigger depending on which one arrives first.

Internal timer trigger

ENDAT has an internal timer. The timer can trigger ENDAT start a frame transmission periodically. Timer values are set in PERIOD[7:0] of EDIM_ENDAT_CFG1 register for the trigger frequency. Two ranges are available for selection (μs and ms). ***Table 38-14. Timer values for the trigger period*** shows the programming values of trigger period.

Table 38-14. Timer values for the trigger period

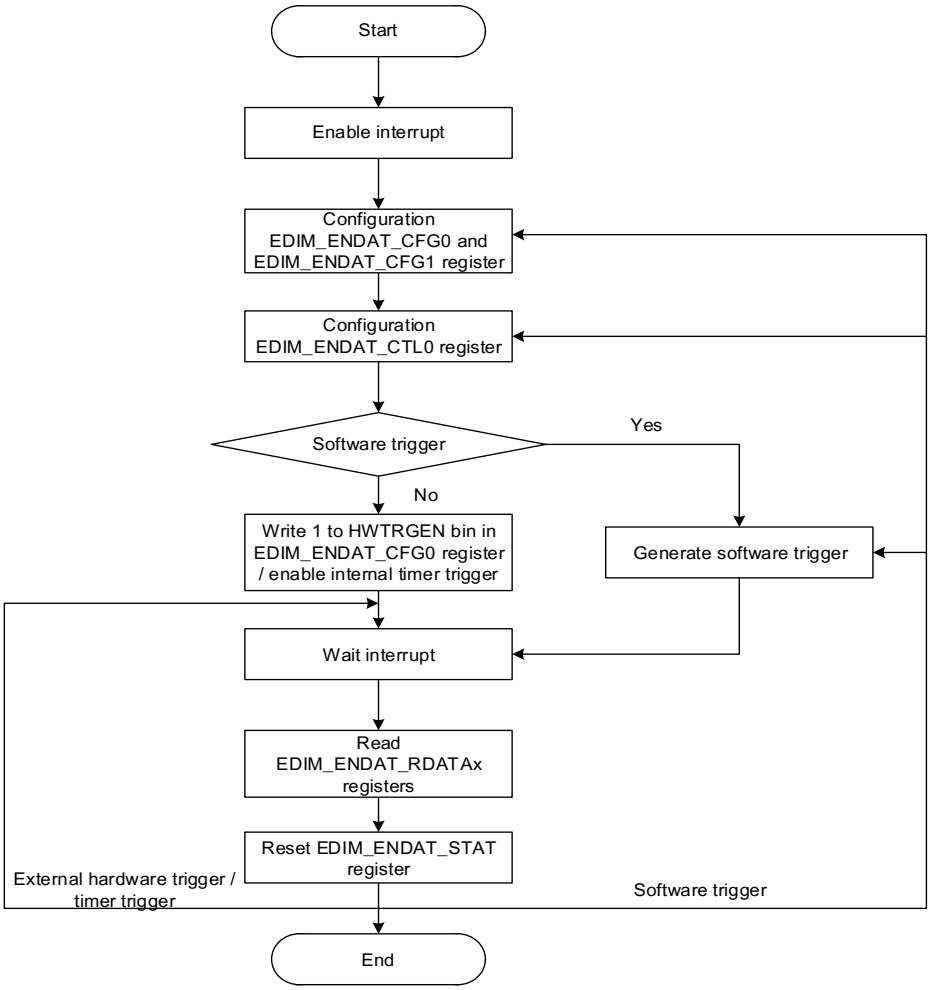
μs: PERIOD[7] = 0		ms: PERIOD[7] = 1	
PERIOD[6:0]	**Trigger period (μs)**	**PERIOD[6:0]**	**Trigger period (ms)**
7'b0000000	Disable	7'b0000000	Disable
7'b0000001	2	7'b0000001	0.2
7'b0000010	4	7'b0000010	0.4
…	…	…	…
7'b1111111	254	7'b1111111	25.4

Note: The internal timer is not affected by HWTRGEN bit in the EDIM_ENDAT_CFG0 register.

Communication process

Figure 38-8. ENDAT module communication process shows the ENDAT module communication process. CPU access registers in ENDAT module by APB bus. The clock frequency of APB is different from CK_ENDAT. Synchronization time is need to synchronize CPU write and read access. In that case, if the hardware trigger is enabled and a hardware trigger is generated immediately after writing 1 to HWTRGEN bit in EDIM_ENDAT_CFG0 register, the transmission that starts may not function correctly. Therefore, HWTRGEN bit is written to 1 after EDIM_ENDAT_CFG0 and EDIM_ENDAT_CFG1 registers have been written and EDIM_ENDAT_CMD register has been loaded.

Figure 38-8. ENDAT module communication process



Transmission clock

The clock generator creates the transmission clock required for the synchronous serial interface. The transmission clock is set in TCLK[3:0] of EDIM_ENDAT_CFG0 register.

The clock generator also makes sure the right number of clock pulses are sent for each mode command, whether or not there's delay compensation. The number of pulses depends on the data length and the mode command as per EnDat protocol specification.

When the CMD[5:0] in the EDIM_ENDAT_CMD register is set to 6'b000111 and the CTNCLK bit in the EDIM_ENDAT_CFG0 register is set to 1, EnDat allows for position value transfers with a continuous clock.

In continuous clock mode, the update rate can be extremely high. If the application doesn't reset the RDATAUPIF bit in the EDIM_ENDAT_STAT register, the EDIM_ENDAT_RDATA0 and EDIM_ENDAT_RDATA1 registers won't update. For safety applications, it's recommended to set the DCSEL bit to 1 in the EDIM_ENDAT_CFG0 register.

Data length configuration

The number of bits to receive for the position value from the encoder should be set in DATALEN[5:0] of EDIM_ENDAT_CFG0 register.

Note: When CMD[5:0] in EDIM_ENDAT_CMD register is set to 6'b010101, the DATALEN[5:0] must be set to 6'b101000.

Delay time compensation

When operating at high transmission rates and with long cable lengths, the cable propagation time must be compensated.

When PDTCEN bit in EDIM_ENDAT_CFG0 register is set to 1, CMD[5:0] in EDIM_ENDAT_CMD register needs to be set to 6'b000111, and the cable propagation time measurement is performed with the next EnDat transmission. The cable propagation time measurement is performed at a fixed 100kHz transmission clock. After propagation time measurement is completed, transmission clock is depended on TCLK[3:0] in EDIM_ENDAT_CFG0 register.

The measured value of cable propagation time is stored in PTIME[7:0] of EDIM_ENDAT_CFG0 register and is taken into account for subsequent transmissions.

After completion of the propagation time measurement, propagation time measurement complete flag and delay time compensation flag, stored in PTMCF bit and DTCF bit of EDIM_ENDAT_STAT register, are set to 1.

If the cable propagation time needs to be measured repeatedly, PDTCEN bit must be reset and then set again.

Additionally, PTIME[7:0] can be overwritten by software to correct the measured cable propagation time. In this case, PTMCF bit will be set to 0.

Note: The cable propagation time measurement function is supported with certified cables only.

Timeout detection

If a transmission timeout occurs, the transmission will be aborted, and the ENDAT module will be redefined to its initial state. However, all registers will not be reset. If propagation time measurement or delay compensation was performed beforehand, these settings will be retained and remain active.

The timeout detection threshold can be set in TOTH[7:0] of EDIM_ENDAT_CFG1 register. If timeout detection is enabled (TOTH[7:0] != 0), it starts when the EnDat transmission begins and stops when the EnDat transmission ends (NEXTRDYF = 1). The timeout detection flag is stored in TOIF bit of EDIM_ENDAT_STAT register. If TOIE bit in EDIM_ENDAT_INTEN register is set to 1, an interrupt will be generated when TOIF bit is set to 1.

The timeout detection is ineffective with the following functions:

1. Continuous clock mode (CTNCLK = 1).
2. Mode command: Encoder to receive reset (CMD[5:0] = 6'b101010).
3. Mode command: Encoder to receive parameters and Encoder to send parameters (CMD[5:0] = 6'b011100 and CMD[5:0] = 6'b100011).

Two ranges are available for selection (μs and ms). *Table 38-15. Timeout detection threshold* shows the configuration of the timeout detection threshold.

Table 38-15. Timeout detection threshold

μs: TOTH[7] = 0		ms: TOTH[7] = 1	
TOTH[6:0]	Time (μs)	TOTH[6:0]	Time (ms)
7'b0000000	Disable	7'b0000000	Disable
7'b0000001	2	7'b0000001	0.2
7'b0000010	4	7'b0000010	0.4
…	…	…	…
7'b1111111	254	7'b1111111	25.4

Recovery time Ⅱ / Ⅲ

During recovery time Ⅱ , the ENDAT module can't start a new transmission. If PDTCEN bit in EDIM_ENDAT_CFG0 register is set to 1 and PTIME[7:0] in EDIM_ENDAT_CFG0 register is higher than or equals 500 ns, the ENDAT module does not implement recovery time Ⅱ additionally. In this case, the ENDAT module is ready for a new transmission after falling edge on the data input port. NEXTRDYF bit in EDIM_ENDAT_STAT register indicates that the ENDAT module is ready for the next transmission.

When operating at high transmission rates and with long cable lengths, recovery time Ⅲ must be set in RCVTIME3[2:0] of EDIM_ENDAT_CFG1 register, and its default value is 2 μs.

RCVTIME3[2:0] can be overwritten by software if required. However, the configured recovery time Ⅲ t_{ST} in RCVTIME3[2:0] should be greater than 1.2 times the measured cable propagation delay time. (t_{ST} > 1.2 × cable propagation delay time)

Reset additional information

If users reset the ENDAT module by writing a 1 to SWRST bit in EDIM_ENDAT_CFG0 register, all register will be reset. Any additional information selected before the reset will be lost. However, the encoder retains this information and continues to expect clock pulses for the additional information. Therefore, after resetting the ENDAT module, the EnDat2.1/2.2 reset command must be transmitted to deselect the additional information in the encoder (CMD[5:0] = 6'b101010).

Digital filter configuration

A digital filter is designed to filter out spikes on the data input port. The filter length is set in FLT[2:0] of EDIM_ENDAT_CFG1 register. FLT[2:0] indicates that the data input line must have n consecutive CK_ENDAT cycles of consistency. Any changes occurring in fewer than n cycles will be disregarded. If SPKIE bit in EDIM_ENDAT_INTEN register is set to 1, an interrupt will be generated when digital filter finds spikes on the data input port.

DMA mode

In some situations, position value requests from the encoder may be frequent, and the CPU may not be able to read out the received data in time. In such case, users can configure DFEN bit and DFENx (x = 0…3) bit in EDIM_ENDAT_CFG0 register to enable DMA request. If DMA is enabled, DMA requests will be triggered k times by hardware when the DATACHKIF bit in the EDIM_ENDAT_STAT register is set to 1. K depends on the configuration of DFENx (x = 0…3) bit. The DMA request is reset by hardware when the circuit receives an acknowledgment from DMA.

The DMA request is controlled by a DMA interface that includes a synchronized FIFO. The FIFO level is 4*4 bytes. When DFENx (x = 0…3) bit in EDIM_ENDAT_CFG0 register is set to 1, the received data is not only stored in the corresponding data registers but also in a FIFO structure, which can be accessed via the EDIM_ENDAT_FIFODATA register. User can readout data of EDIM_ENDAT_FIFODATA register by DMA.

For example, if users wish to read EDIM_ENDAT_RDATA0 and EDIM_ENDAT_RDATA3 registers by DMA, the DFEN bit, DFEN0 bit and DFEN3 bit in EDIM_ENDAT_CFG0 register should be set to 1. The data of EDIM_ENDAT_RDATA0 and EDIM_ENDAT_RDATA3 are also transferred to FIFO. The DMA request will be triggered 2 times by hardware when DATACHKIF bit is set to 1. If EDIM_ENDAT_FIFODATA register address is set as DMA source address, DMA will transfer the data of EDIM_ENDAT_FIFODATA register.

If FIFO is overrun or underrun, FIFOOVRIF and FIFOUNRIF bits are set to 1 in EDIM_ENDAT_STAT register. Overrun and underrun don't affect data reading. But the data read out is not reliable. FIFOFH bit in EDIM_ENDAT_CFG0 register is set to 1, which can execute to flush FIFO.

If DMA acknowledge doesn't come back to ENDAT module within 256 PCLK1 cycles, DMATOIF bit in EDIM_ENDAT_STAT register will be set to 1 and DMA request will be cleared by hardware.

ENDAT interrupts

The interrupt can be produced on one of the events:

- Received data check completion.
- Type Ⅰ error
- Type Ⅱ error

- Type Ⅲ error
- Frame transmission timeout
- Spike detection
- "WRN" bit transmitted in the EnDat protocol
- "RM" bit transmitted in the EnDat protocol
- "BUSY" bit transmitted in the EnDat protocol
- CRC error of additional information 2
- CRC error of additional information 1
- CRC error of the received data
- "Error message2" bit transmitted in the EnDat protocol
- EDIM_ENDAT_RDATA3 register update
- EDIM_ENDAT_RDATA2 register update
- EDIM_ENDAT_RDATA0/1 registers update
- MRS or address error
- "Error message1" bit transmitted in the EnDat protocol
- FIFO overrun, underrun
- DMA timeout error

38.3.5. Registers definition

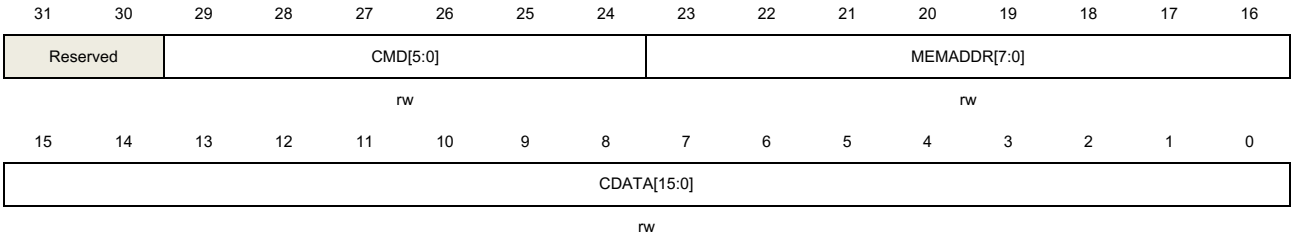
ENDAT base address: 0x4000 CC00

ENDAT command register (EDIM_ENDAT_CMD)

Address offset: 0x00

Reset value: 0x0700 0000

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



Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29:24	CMD[5:0]	EnDat communication mode code 000111: Encoder to send position values 001001: Encoder to send positon values with additional information and selection of the memory area 001110: Selection of memory area 010010: Encoder to receive communication command 010101: Encoder to send test values 011011: Encoder to send positon values with additional information and receive parameters 011100: Encoder to receive parameters 100011: Encoder to send parameters 100100: Encoder to send position values with additional information and parameters 101010: Encoder to receive reset 101101: Encoder to send position values with additional information and receive error reset 110001: Encoder to receive test command 110110: Encoder to send position values with additional information and receive test command 111000: Encoder to send position values with additional information
23:16	MEMADDR[7:0]	Memory range selection, address and port address of encoder The values of these bits depend on the CMD[5:0], as shown in *Table 38-12. EnDat command*.
15:0	CDATA[15:0]	Encoder receive communication data of instruction or parameters

The values of these bits depend on the CMD[5:0], as shown in *Table 38-12. EnDat command*.

ENDAT received data register 0 (EDIM_ENDAT_RDATA0)

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

Bits	Fields	Descriptions
31:0	RDATAL[31:0]	Received data from encoder These bits indicate received data bit 31 to 0. The components of the received data depend on the CMD[5:0], as shown in *Table 38-13. Received data in EDIM_ENDAT_RDATA0 and EDIM_ENDAT_RDATA1 registers*.

ENDAT received data register 1 (EDIM_ENDAT_RDATA1)

Address offset: 0x08

Reset value: 0x0040 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									INVF2	F1	CRC[4:0]				
									r	r			r		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDATAH[15:0]															
r															

Bits	Fields	Descriptions
31:23	Reserved	Must be kept at reset value
22	INVF2	The inverse of the safety-relevant error bit (F2) in the received data 0: A safety-relevant error has occurred, or forced dynamic sampling is active. 1: No error
21	F1	The safety-relevant error bit (F1) in the received data 0: No error

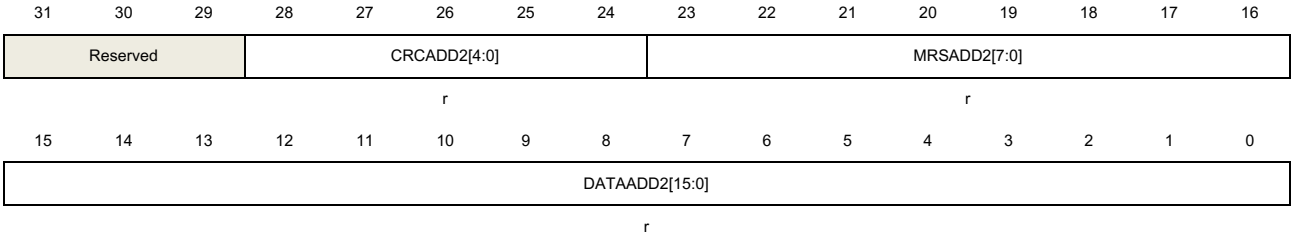
		1: A safety-relevant error has occurred, or forced dynamic sampling is active.
20:16	CRC[4:0]	CRC in received data from encoder
15:0	RDATAH[15:0]	Received data from encoder These bits indicate received data bit 47 to 32. The components of the received data depend on the CMD[5:0], as shown in ***Table 38-13. Received data in EDIM_ENDAT_RDATA0 and EDIM_ENDAT_RDATA1 registers***.

ENDAT received data register 2 (EDIM_ENDAT_RDATA2)

Address offset: 0x0C

Reset value: 0x0000 0000

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



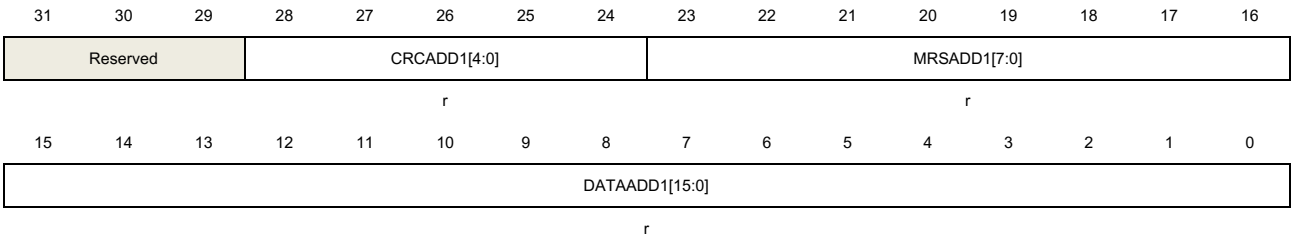
Bits	Fields	Descriptions
31:29	Reserved	Must be kept at reset value
28:24	CRCADD2[4:0]	CRC in additional information 2
23:16	MRSADD2[7:0]	MRS in additional information 2
15:0	DATAADD2[15:0]	Data in additional information 2

ENDAT received data register 3 (EDIM_ENDAT_RDATA3)

Address offset: 0x10

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	CRCADD1[4:0]	CRC in additional information 1
23:16	MRSADD1[7:0]	MRS in additional information 1
15:0	DATAADD1[15:0]	Data in additional information 1

ENDAT configuration register 0 (EDIM_ENDAT_CFG0)

Address offset: 0x14

Reset value: 0x0000 0DF0

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	FIFOFH	SWRST	DFEN3	DFEN2	DFEN1	DFEN0	PDTCEN	PTIME[7:0]							
	w	w	rw	rw	rw	rw	rw	rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AUTORST	RSTMSEL	DATALEN[5:0]						TCLK[3:0]				DFEN	CTNCLK	DCSEL	HWTRGEN
rw	rw	rw						rw				rw	rw	rw	rw

Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30	FIFOFH	FIFO flush 0: No action 1: Clear the FIFO data
29	SWRST	Software reset all registers of ENDAT module 0: Software reset inactive 1: Software reset active.
28	DFEN3	DMA request and FIFO enable for EDIM_ENDAT_RDATA3 register 0: DMA request and FIFO disabled 1: DMA request and FIFO enabled
27	DFEN2	DMA request and FIFO enable for EDIM_ENDAT_RDATA2 register 0: DMA request and FIFO disabled 1: DMA request and FIFO enabled
26	DFEN1	DMA request and FIFO enable for EDIM_ENDAT_RDATA1 register 0: DMA request and FIFO disabled 1: DMA request and FIFO enabled
25	DFEN0	DMA request and FIFO enable for EDIM_ENDAT_RDATA0 register 0: DMA request and FIFO disabled

		1: DMA request and FIFO enabled
24	PDTCEN	Propagation delay time compensation enable 0: Propagation delay compensation disabled 1: Propagation delay compensation enabled
23:16	PTIME[7:0]	Cable propagation time These bits are the measured cable propagation time. If these bits are written by software, the PTMCF bit in the EDIM_ENDAT_STAT register is set to 0 by hardware. The time step is equal to the system clock cycle CK_ENDAT and the single transit time through the digital filter is also included in the cable propagation time.
15	AUTORST	Automatic reset of the status and error registers For safety mode, this bit is set to 0. 0: The EDIM_ENDAT_STAT register is reset by software. 1: The EDIM_ENDAT_STAT register is automatically reset at the start of data acquisition in the next EnDat transmission cycle.
14	RSTMSEL	Reset moment selection For safety mode, this bit is set to 1. 0: The EDIM_ENDAT_STAT register can be reset by software at any time. 1: The EDIM_ENDAT_STAT register must be reset by software after EDIM_ENDAT_STAT register is fully updated (DATACHKIF bit in EDIM_ENDAT_STAT register is set). Resetting will not take effect after a transmission is started.
13:8	DATALEN[5:0]	EnDat data length The EnDat data length is the sum of multi-turn data bits and single-turn data bits. 00000-000111: Reserved 001000: Data length = 8 bits 001001: Data length = 9 bits … 110000: Data length = 48 bits 110001-111111: Reserved
7:4	TCLK[3:0]	Transmission clock for EnDat 0000: 16.67 MHz 0001: 16.67 MHz 0010: 16.67 MHz 0011: 16.67 MHz 0110: 8.33 MHz 1011: 4.16 MHz 1100: 2 MHz 1101: 1 MHz 1110: 200 kHz

		1111: 100 kHz
3	DFEN	DMA request and FIFO enable 0: DMA request and FIFO disabled 1: DMA request and FIFO enabled
2	CTNCLK	Continuous clock mode 0: Continuous clock disabled 1: Continuous clock enabled
1	DCSEL	Data condition selection This bit defines whether the data transfer to the EDIM_ENDAT_RDATAx registers (x = 0…3) after the completion of data transmission depends on the flag bits being set in the EDIM_ENDAT_STAT register. These flag bits include TYPE1IF, TYPE2IF, TYPE3IF, CRCADD2IF,CRCADD1IF, CRCIF, MRSADDIF, ERR2MIF, ERR1MIF, RDATA3UPIF, RDATA2UPIF and RDATAUPIF. For safety mode, this bit must be set to 1. 0: Data transfer depend on the flag bits being set to 1 in the EDIM_ENDAT_STAT register 1: Data transfer despite the flag bits in the EDIM_ENDAT_STAT register
0	HWTRGEN	Hardware trigger enable 0: Hardware trigger disabled 1: Hardware trigger enabled

ENDAT configuration register 1 (EDIM_ENDAT_CFG1)

Address offset: 0x18

Reset value: 0x0004 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HWTRGDELAY[7:0]								Reserved		FLT[2:0]			RCVTIME3[2:0]		
rw										rw			rw		

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

Bits	Fields	Descriptions
31:24	HWTRGDELAY[7:0]	Hardware trigger delay 00000000: Disable 00000001: Reserved 00000010: Reserved 00000011: 3 CK_ENDAT cycles 00000100: 4 CK_ENDAT cycles …

		11111111: 255 CK_ENDAT cycles
23:22	Reserved	Must be kept at reset value.
21:19	FLT[2:0]	Digital filter for data input interface from encoder These bits indicates that the data input line must have n consecutive CK_ENDAT cycles of consistency. Any changes occurring in fewer than n cycles will be disregarded. 000: Disable 001: n = 3 010: n = 4 011: n = 5 100: n = 6 101: n =10 110: n = 20 111: n = 40
18:16	RCVTIME3[2:0]	Recovery time Ⅲ t_{ST} 000: 1/2 TCLK 001: 0.5 μs 010: 1 μs 011: 1.5 μs 100: 2 μs 101: 4 μs 110: 8 μs 111: 10 μs
15:8	TOTH[7:0]	Frame transmission timeout threshold These bits have 256 different threshold values, as detailed in ***Table 38-15. Timeout detection threshold***.
7:0	PERIOD[7:0]	Internal timer trigger period These bits have 256 different values for trigger period, as detailed in ***Table 38-14. Timer values for the trigger period***.

ENDAT status register (EDIM_ENDAT_STAT)

Address offset: 0x1C

Reset value: 0x4000 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
DATACHKIF	NEXTRDYF	Reserved			DMATOIF	FIFOUNRIF	FIFOOVRIF	PTMCF	DTCF	Reserved			TYPE3IF	TOIF	SPKIF
rc_w1	r				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

WRNMIF	RMMIF	BUSYMIF	CRCADD2IF	CRCADD1IF	ERR2MIF	RDATA3UPIF	RDATA2UPIF	Reserved	MRSADDRIF	TYPE2IF	TYPE1IF	CRCIF	ERR1MIF	RDATAUPIF
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	DATACHKIF	Received data check interrupt flag 0: The EDIM_ENDAT_STAT is not yet completely updated and all checks have not been performed. 1: The EDIM_ENDAT_STAT is completely updated and all checks have been performed. Data transmission is not yet completed, however, meaning that the EnDat protocol automation machine is not yet ready again.
30	NEXTRDYF	Next transmisson ready flag 0: The data transmission is not yet completed. 1: Data transmission has ended and the EnDat automation machine is ready for the next transmisson. Recovery timer Ⅰ and Ⅱ are completed.
29:27	Reserved	Must be kept at reset value.
26	DMATOIF	DMA request timeout interrupt flag 0: No DMA request timeout error occurred 1: DMA timeout error occurred
25	FIFOUNRIF	FIFO underrun error interrupt flag 0: No FIFO underrun error occurred 1: Underrun error occurred
24	FIFOOVRIF	FIFO overrun error interrupt flag 0: No FIFO overrun error occurred 1: Overrun error occurred
23	PTMCF	Propagation time measurement completion flag This bit reports that propagation time measurement was successfully completed. When PDTCEN bit in EDIM_ENDAT_CFG0 register is set to 1, this bit is valid. 0: PTMCF incomplete 1: PTMCF complete
22	DTCF	Delay time compensation flag If PDTCEN bit in EDIM_ENDAT_CFG0 register is set to 0, this bit and PTMCF bit will be reset by hardware. 0: Delay time compensation inactive 1: Delay time compensation active
21:19	Reserved	Must be kept at reset value.
18	TYPE3IF	Type Ⅲ error interrupt flag 0: No Type Ⅲ error occurred 1: A transmission layer error occurred in the EnDat master.

Note: Error handling for Type Ⅲ is only executed in EnDat 2.1 commands.

17	TOIF	Frame transmission timeout interrupt flag 0: No frame transmission timeout occurred 1: Transmission timeout
16	SPKIF	Spike detection interrupt flag A spike is detected at data input port of ENDAT module When the FLT[2:0] bit in the EDIM_ENDAT_CFG0 register is enabled, this bit is valid. 0: No spike 1: Spike occurred
15	WRNMIF	Mirror interrupt flag for "WRN" bit transmitted in the EnDat protocol. 0: WRN = 0 1: WRN = 1
14	RMMIF	Mirror interrupt flag for "RM" bit transmitted in the EnDat protocol. 0: RM = 0 1: RM = 1
13	BUSYMIF	Mirror interrupt flag for "BUSY" bit transmitted in the EnDat protocol. 0: BUSY = 0 1: BUSY = 1
12	CRCADD2IF	CRC result interrupt flag of additional information 2 0: No CRC result of additional information 2 error occurred 1: CRC error occurred
11	CRCADD1IF	CRC result interrupt flag of additional information 1 0: No CRC result of additional information 1 error occurred 1: CRC error occurred
10	ERR2MIF	Mirror interrupt flag for "Error message2" bit transmitted in the EnDat protocol (only with EnDat2.2 commands). 0: Error message2 occurred 1: No Error message2 occurred
9	RDATA3UPIF	EDIM_ENDAT_RDATA3 register update interrupt flag This bit indicates that EDIM_ENDAT_RDATA3 register has been updated. To rewrite data, when AUTORST bit in EDIM_ENDAT_CFG0 register is set to 0, it must be cleared by software after the EDIM_ENDAT_RDATA3 register is read. 0: EDIM_ENDAT_RDATA3 not updated 1: EDIM_ENDAT_RDATA3 updated
8	RDATA2UPIF	EDIM_ENDAT_RDATA2 register update interrupt flag This bit indicates that EDIM_ENDAT_RDATA2 register has been updated. To rewrite data, when AUTORST bit in EDIM_ENDAT_CFG0 register is set to 0, it must be cleared by software after the EDIM_ENDAT_RDATA2 register is read.

0: EDIM_ENDAT_RDATA2 not updated
1: EDIM_ENDAT_RDATA2 updated updated

7:6	Reserved	Must be kept at reset value.
5	MRSADDRIF	MRS or address error interrupt flag When Type Ⅰ / Ⅱ error occurs, this bit is set. If the MRS and address bits are incorrectly recognized due to interference, only the MRSADDRF bit will be set, not the TYPE1IF and TYPE2IF bits. 0: No MRS or address error occurred 1: A MRS or address error occurred
4	TYPE2IF	Type Ⅱ error interrupt flag 0: No Type Ⅱ error occurred 1: A Type Ⅱ error occurred
3	TYPE1IF	Type Ⅰ error interrupt flag 0: No Type Ⅰ error occurred 1: A Type Ⅰ error occurred
2	CRCIF	CRC error interrupt flag of the received data (position value, parameter or test value). 0: No CRC error interrupt flag of the received data occurred 1: CRC error occurred
1	ERR1MIF	Mirror interrupt flag for "Error message1" bit transmitted in the EnDat protocol 0: No Error message1 occurred 1: Error message1 occurred
0	RDATAUPIF	EDIM_ENDAT_RDATA0 and EDIM_ENDAT_RADTA1 registers update interrupt flag This bit indicates that EDIM_ENDAT_RDATA0 and EDIM_ENDAT_RDATA1 registers has been updated. To rewrite data, when AUTORST bit in EDIM_ENDAT_CFG0 register is set to 0, it must be cleared by software after the data registers are read. 0: EDIM_ENDAT_RDATA0 and EDIM_ENDAT_RDATA1 not updated 1: EDIM_ENDAT_RDATA0 and EDIM_ENDAT_RDATA1 updated

ENDAT interrupt enable register (EDIM_ENDAT_INTEN)

Address offset: 0x20
Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATACHKIE	Reserved				DMATOIE	FIFOUNRIE	FIFOOVRIE	Reserved					TYPE3IE	TOIE	SPKIE

rw					rw	rw	rw							rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
WRNIE	RMIE	BUSYIE	CRCADD2IE	CRCADD1IE	ERR2IE	RDATA3UPIE	RDATA2UPIE	Reserved		MRSADDRIE	TYPE2IE	TYPE1IE	CRCIE	ERR1IE	RDATAUPIE	
rw	rw	rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw	

Bits	Fields	Descriptions
31	DATACHKIE	Interrupt enable for DATACHKIF 0: Interrupt disabled 1: Interrupt enabled
30:27	Reserved	Must be kept at reset value.
26	DMATOIE	Interrupt enable of DMATOIF 0: Interrupt disabled 1: Interrupt enabled
25	FIFOUNRIE	Interrupt enable of FIFOUNRIF 0: Interrupt disabled 1: Interrupt enabled
24	FIFOOVRIE	Interrupt enable of FIFOOVRIF 0: Interrupt disabled 1: Interrupt enabled
23:19	Reserved	Must be kept at reset value.
18	TYPE3IE	Interrupt enable of TYPE3IF 0: Interrupt disabled 1: Interrupt enabled
17	TOIE	Interrupt enable of TOIF 0: Interrupt disabled 1: Interrupt enabled
16	SPKIE	Interrupt enable of SPKIF 0: Interrupt disabled 1: Interrupt enabled
15	WRNIE	Interrupt enable of WRNMIF 0: Interrupt disabled 1: Interrupt enabled
14	RMIE	Interrupt enable of RMMIF 0: Interrupt disabled 1: Interrupt enabled
13	BUSYIE	Interrupt enable of BUSYMIF 0: Interrupt disabled

		1: Interrupt enabled
12	CRCADD2IE	Interrupt enable of CRCADD2IF 0: Interrupt disabled 1: Interrupt enabled
11	CRCADD1IE	Interrupt enable of CRCADD1IF 0: Interrupt disabled 1: Interrupt enabled
10	ERR2IE	Interrupt enable of ERR2MIF 0: Interrupt disabled 1: Interrupt enabled
9	RDATA3UPIE	Interrupt enable of RDATA3UPIF 0: Interrupt disabled 1: Interrupt enabled
8	RDATA2UPIE	Interrupt enable of RDATA2UPIF 0: Interrupt disabled 1: Interrupt enabled
7:6	Reserved	Must kept at reset value
5	MRSADDRIE	Interrupt enable of MRSADDRIF 0: Interrupt disabled 1: Interrupt enabled
4	TYPE2IE	Interrupt enable of TYPE2IF 0: Interrupt disabled 1: Interrupt enabled
3	TYPE1IE	Interrupt enable of TYPE1IF 0: Interrupt disabled 1: Interrupt enabled
2	CRCIE	Interrupt enable of CRCIF 0: Interrupt disabled 1: Interrupt enabled
1	ERR1IE	Interrupt enable of ERR1MIF 0: Interrupt disabled 1: Interrupt enabled
0	RDATAUPIE	Interrupt enable of RDATAUPIF 0: Interrupt disabled 1: Interrupt enabled

ENDAT software trigger register (EDIM_ENDAT_SWT)

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															SWTR
															w

Bits	Fields	Descriptions
31:1	Reserved	Must kept at reset value
0	SWTR	Data transmission software trigger, cleared by hardware 0: Software trigger disabled 1: Software trigger enabled This bit is set by software to generate a transmission trigger.

ENDAT FIFO data register (EDIM_FIFODATA)

Address offset: 0x128

Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:0	FIFODATA[31:0]	FIFO data from EDIM_ENDAT_RDATAx (x = 0…3) registers.

ENDAT status update mode register (EDIM_ENDAT_STATUPMODE)

Address offset: 0x200

Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:3	Reserved	Must kept *at* reset value
2	WRNMIFMODE	WRNMIF status update mode 0: Dynamic Update, WRNMIF update at the end of frams transmission 1: Static Update, WRNMIF set by hardware and cleared by software
1	RMMIFMODE	RMMIF status update mode 0: Dynamic Update, RMMIF update at the end of frams transmission 1: Static Update, RMMIF set by hardware and cleared by software
0	BUSYMIFMODE	BUSYMIF status update mode 0: Dynamic Update, BUSYMIF update at the end of frams transmission 1: Static Update, BUSYMIF set by hardware and cleared by software

38.4. BiSS-C encoder interface module (BISS)

38.4.1. Overview

The BiSS-C encoder interface module serves as a master to communicate with encoders that support the BiSS-C interface.

38.4.2. Characteristics

- Support BiSS-C communication with up to 8 slaves.
  20. Each slave process data up to 64 bits.
- CRC checks with up to 16 bits for data verification.
- Process data transfer and register communication simultaneously.
- Compensating for line delays and conversion times automatically.
- The independent memory banks enable the controller to access data freely while BiSS sensor data is being transferred.
- The memory for bidirectional register communication is 64 bytes.
- A parallel interface with a 32-bit data/address bus.
- Support point-to-point connection.
- Interrupt generation.
- Support DMA mode.

38.4.3. Block diagram

Figure 38-9. BISS module block diagram

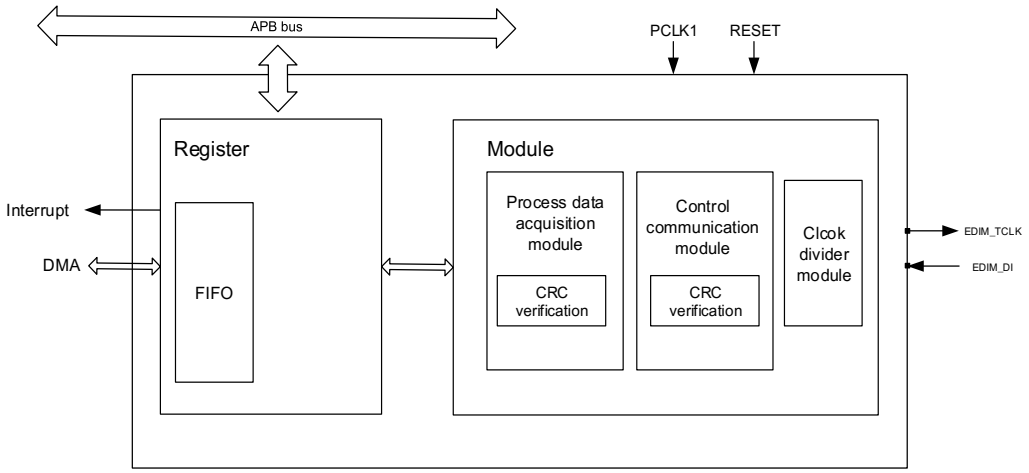


Figure 38-9. BISS module block diagram provides details of the internal configuration of the BISS module. *Table 38-16. BISS pins definition* gives the BISS pin description.

Table 38-16. BISS pins definition

Name	Description
EDIM_TCLK	BISS clock signal output

Name	Description
EDIM_DI	BISS data input

38.4.4. Function overview

BISS module is a communication interface used to transmit frames to encoder, receive frames from encoder. More details of BiSS-C protocol please refer to BiSS C Protocol Description.

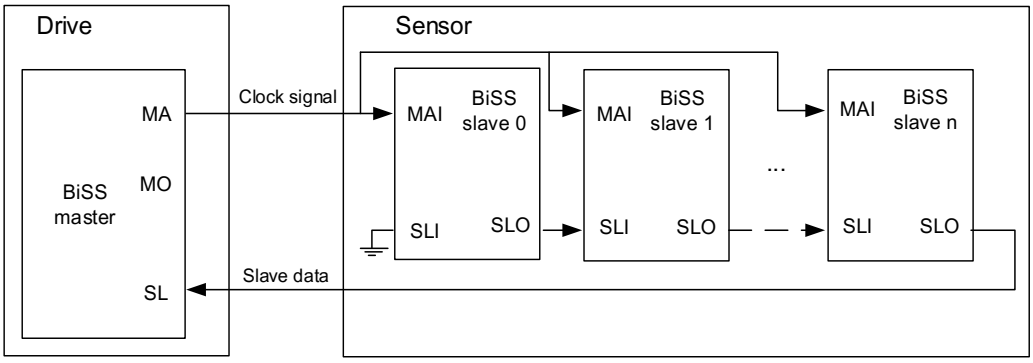
BISS clock

The system clock for BISS module is PCLK1. And the CPU accesses the BISS module registers through the APB bus.

Point-to-point connection mode

BISS module supports point-to-point connection mode, as shown in ***Figure 38-10. Point-to-point connection***.

Figure 38-10. Point-to-point connection



The BiSS master connects to the clock line via MA pin and to the data line via its SL pin.

The clock signal at the MA pin manages the communication frame, providing the clock source for the BiSS slave and marking the start and end of the transmission cycle. Additionally, this clock signal transmits one control bit (CDM) from the BiSS master to the BiSS slave for control communication.

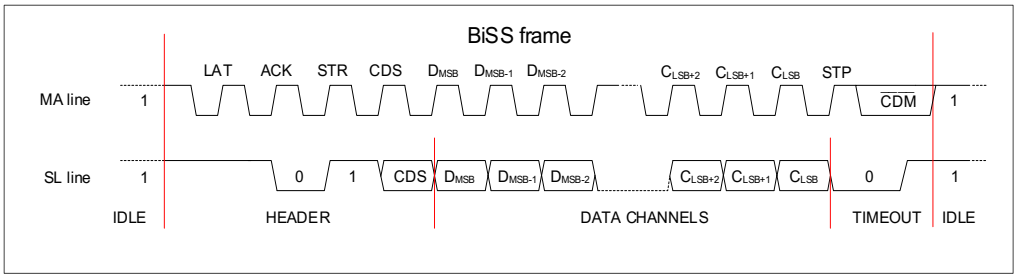
Inside the sensor device, the BiSS slave connects to the clock line through its MAI pin and to the data output line through its SLO pin.

Generally, BiSS operates as a digital protocol with both the BiSS master and slave generating data signals on the rising edge of the clock signal MA. The BiSS slave samples data at the falling edge of MA. The BiSS master sampling rate relative to the clock signal MA is flexible and can be adjusted to compensate for line delays.

BiSS-C frame

BiSS-C frames are controlled by the clock signal at the MA pin of the BISS module, and can be divided into four stages: idle, data header, data channels, and timeout, as shown in ***Figure 38-11. Simple BiSS frame***

Figure 38-11. Simple BiSS frame



In the idle phase, no data is transmitted, and both the MA and SL pins are typically held high.

In the data header phase, a new BiSS frame is initiated with a falling edge followed by a rising edge on the MA line. The first rising edge, known as LAT (latch), triggers the BiSS Slave to latch or generate process data. The BiSS slave responds on the SL line at each rising edge of the MA clock signal, with each edge representing one bit of the BiSS frame. The first bit, ACK, is used to acknowledgment the start of a new cycle. Since SL line is high during idle, the ACK signal is always 0.

In the data channels phase, this phase is used to transmit process data of each slave. Each slave transmits up to 64 bits of process data ($D_{MSB}..D_{LSB}$). Each slave data channel is typically protected by a CRC of up to 16 bits ($C_{MSB}...C_{LSB}$).

In the timeout phase, this phase is used to synchronize the end of a BiSS cycle between the BiSS master and the BiSS slaves. The BiSS master stops the clock signal on the MA line, and each BiSS slave individually measures the timeout. After the timeout, a rising edge on the SL line signals the BiSS master. Consequently, the BiSS slave generates a stop bit at the last rising edge on the MA line (STP). To transmit the control data master bit (CDM) for control communication, the BiSS slave samples the clock line MA at the end of the timeout. BiSS systems support two types of timeout modes: static timeout mode and adaptive timeout mode.

Note:

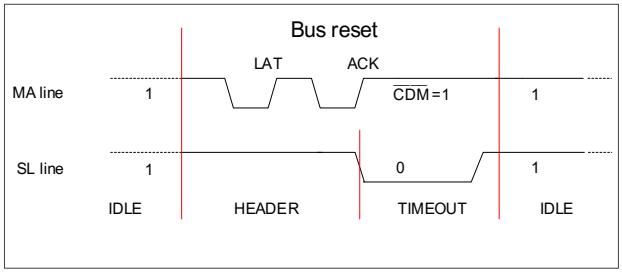
1. The BiSS master adjusts its sampling points for the remaining data bits based on the delay observed in the last acknowledgment clock cycle.
2. The BiSS slave must respond with the acknowledge bit within a specific time interval. Any response that comes after this interval should be flagged as an error by the drive.

BiSS-C protocol include process data communication and control communication. Process data are completely transmitted in each frame used as sensor data, and control data are transmitted one bit (CDM/CDS) per frame used for commands and register access.

BiSS-C process data communication

Bus reset

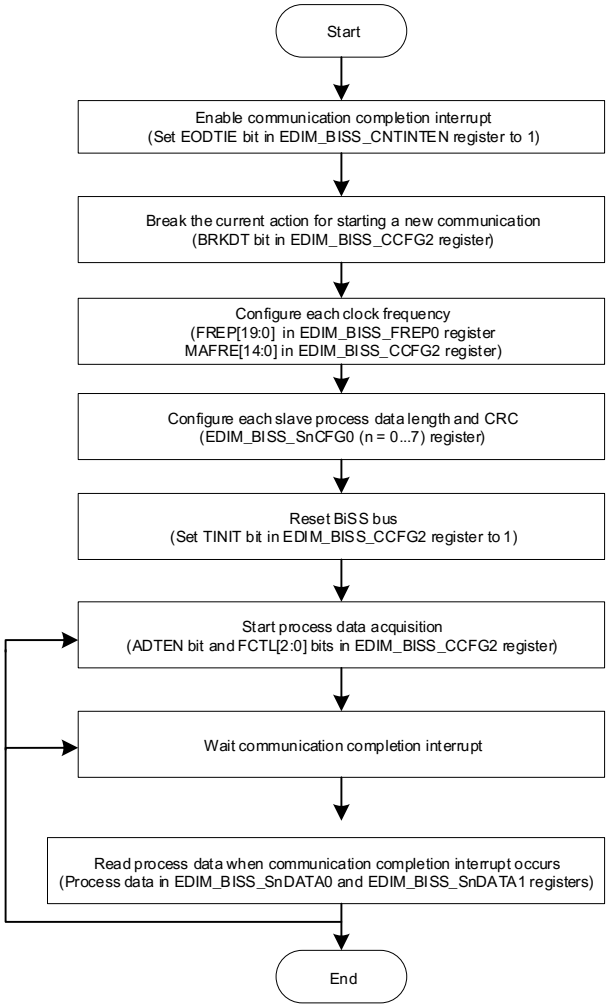
Figure 38-12. Typical reset sequence



In the point-to-point connection mode, slave 0 must generate the start and stop bits for the BiSS frame. To inform slave 0 of its active role, its data input line SLI needs to be connected to ground, and the system should be initialized after power-on. When TINIT bit in EDIM_BISS_CCFG2 register is set to 1, a bus reset frame is transmitted, as shown in ***Figure 38-12. Typical reset sequence***. The BiSS master sends at least two low pulses on the MA line. After the timeout, the system is ready for the BiSS communication, and the idle level of SL should be 1.

Received data

Figure 38-13. Basic operation for process data



The basic operation to receive process data is shown in ***Figure 38-13. Basic operation for process data***. Received data is stored in the EDIM_BISS_SnDATAm (n = 1…8, m = 0..1) registers. These data registers can hold data from eight slaves, with each slave containing up to 64 bits of process data. The distribution of the received process data in the registers is as shown in the ***Table 38-17. Received data in EDIM_BISS_SnDATAm register***.

Table 38-17. Received data in EDIM_BISS_SnDATAm register

Slave ID	Register	Process data
Slave 0	EDIM_BISS_S0DATA0	DATA[31:0]
	EDIM_BISS_S0DATA1	DATA[63:32]
Slave 1	EDIM_BISS_S1DATA0	DATA[31:0]
	EDIM_BISS_S1DATA1	DATA[63:32]
Slave 2	EDIM_BISS_S2DATA0	DATA[31:0]
	EDIM_BISS_S2DATA1	DATA[63:32]
Slave 3	EDIM_BISS_S3DATA0	DATA[31:0]

Slave ID	Register	Process data
	EDIM_BISS_S3DATA1	DATA[63:32]
Slave 4	EDIM_BISS_S4DATA0	DATA[31:0]
	EDIM_BISS_S4DATA1	DATA[63:32]
Slave 5	EDIM_BISS_S5DATA0	DATA[31:0]
	EDIM_BISS_S5DATA1	DATA[63:32]
Slave 6	EDIM_BISS_S6DATA0	DATA[31:0]
	EDIM_BISS_S6DATA1	DATA[63:32]
Slave 7	EDIM_BISS_S7DATA0	DATA[31:0]
	EDIM_BISS_S7DATA1	DATA[63:32]

Note: DATA represents the received process data and CRC. The memory is written byte by byte with unused bits set to 0. Unused bytes are not changed.

In BiSS, the CRC bits are inverted before transmission from slaves. The received CRC bits are re-inverted and stored in the most significant byte(s) of EDIM_BISS_SnRDATAm (n = 0…7, m = 0…1) register. The CRC storage function is set in CRCSEN bit of EDIM_BISS_CCFG1 register. Additionally, if the process data exceeds 32 bits, the CRC bits will not be stored.

Example:

1. Slave 0: The total length of received sensor data is 28 bits, including 21 bits of process data, 7 bits of CRC.
2. Slave 1: The total length of received sensor data is 22 bits, including 13 bits of process data, 9 bits of CRC.
3. Slave 2: The total length of received sensor data is 54 bits, including 47 bits of process data, 7 bits of CRC.

In the example, the distribution of the received sensor data in the registers is as shown in the *Table 38-18. Received data in EDIM_BISS_SnDATAm register for example*.

Table 38-18. Received data in EDIM_BISS_SnDATAm register for example

Slave ID	Register	[31:24]	[23:16]	[15:8]	[7:0]
Slave 0	EDIM_BISS_S0DATA0	-	{000, process data 1 [20:0]}		
	EDIM_BISS_S0DATA1	{0, CRC 1 [6:0]}	-		
Slave 1	EDIM_BISS_S1DATA0	-		{000, process data 2 [12:0]}	
	EDIM_BISS_S1DATA1	{0000000, CRC 2 [8:0]}		-	
Slave 2	EDIM_BISS_S2DATA0	process data 3 [31:0]			
	EDIM_BISS_S2DATA1	-		{0, process data 3 [46:32]}	

Note: '-' represents the byte is unchanged.

The BiSS module utilizes two memory banks to store process data, allowing for the import of new data during system access. During a BiSS frame's processing, while the first RAM bank is being updated with new process data, the second bank can be accessed to retrieve data from the last frame. At the end of the BiSS-C frame, a swap of the process data memory banks takes place. Simultaneously, the validity flag SLVF[15:0] in EDIM_BISS_STAT0 register, which corresponds to the RAM bank, is also swapped. The bank swap function can be configured in RBSWDIS bit of EDIM_BISS_CCFG2 register.

Data transmission trigger

The BISS module has three data transmission trigger modes: automatic trigger, external trigger, and software trigger.

When ADTEN bit in EDIM_BISS_CCFG2 register is set to 1 and FREP[19:0] in EDIM_BISS_FREP register is not set to 0x00001, the trigger mode is automatic trigger. If FREP[19:0] is set to 0, data transmission is triggered automatically as soon as the slave's timeout is detected. If FREP[19:0] is set to 0x00002 - 0xFFFFF, the data transmission is triggered automatically in a configured constant frame repetition rate.

When ADTEN bit is set to 1 and FREP[19:0] is set to 0x00001, the trigger mode is external trigger. The trigger signal is from TRIGSEL. The process data master clock frequency f_{MA}, as set in MAFRE[14:0] of EDIM_BISS_CCFG2 register, determines the delay from the external trigger event to the falling edge at MA. The delay can be calculated using the formula:

$$t_{delay} = 2 / f_{PCLK} + 1 / (2 * f_{MA}) \quad (38\text{-}1)$$

When ADTEN bit is set to 0, FCTL[2:0] in EDIM_BISS_CCFG2 register is used to trigger data transmission. The master begins data transmission after FCTL[2:0] is written and resets the FCTL[2:0] after the frame is transmitted. If the SL line is low at frame initiation, an automatic data transmission error will be generated, ADTERRIF bit in EDIM_BISS_STAT0 register is set to 0. The error can be avoided by resetting FREP[19:0], effectively waiting for a high level of the SL line. BiSS-C control communication can also be initiated with FCTL[2:0]. Additionally, reduced BiSS-C frames can be enabled for control communication using BiSS commands. During active control communication, the CDM bit is generated automatically. Otherwise, the generation of the CDM bit is controlled by FCTL[2:0].

To obtain better trigger response characteristics, the following two guidelines are recommended when configuring the BISS data transfer trigger:

Use a higher process data master clock frequency f_{MA} where possible. The delay from the trigger event to the first falling edge of MA contains a component determined by fMA (see $t_{delay} = 2/f_{PCLK} + 1/(2 \cdot f_{MA})$). Increasing f_{MA}, which is configured by MAFRE[14:0] in the EDIM_BISS_CCFG2 register, shortens this delay and therefore yields a faster trigger response.

In constant frame repetition rate auto-trigger mode, keep the frame repetition rate in an

integer-multiple relationship with f_{MA}. When FREP[19:0] in the EDIM_BISS_FREP register is set to a value between 0x00002 and 0xFFFFF to enable auto-triggering at a constant frame repetition rate, it is recommended to configure the frame repetition period as an integer multiple of the MA clock period, so that the start of each frame is aligned with an MA clock edge, which provides better frame transfer timing characteristics.

Note: Both recommendations can be applied together. For applications with high real-time requirements or multi-axis synchronization, it is advisable to select a higher f_{MA} first, and then plan the frame repetition rate as an integer multiple of the MA clock period.

BiSS-C control communication

BiSS-C control communication includes BiSS commands and register communication.

The CTS bit in EDIM_BCCFG1 register defines what type of control communication is being selected by the BiSS master.

The CDM and CDS bits are used for BiSS-C control data communication, being transmitted by the BiSS master and slave in every BiSS-C frame. Multiple CDM and CDS bits form a control communication.

BiSS commands

When CTS bit is set to 0, the BiSS commands is used for control communication. A BiSS command can address either a single or multiple slaves. The number of addressed slaves is configured in IDS[7:0] bits of EDIM_BISS_CCDATAn (n = 0…15) register. To address all slaves using broadcast mode, IDS[7:0] must be set to 0. When IDS[N] is set, it represents the slave(7-N) is addressed. CMD[2:0] in EDIM_BISS_CCFG1 register is used to determine which BiSS command is sent. The broadcast BiSS command communication and addressed BiSS command communication are shown in ***Figure 38-14. Broadcast BiSS command communication*** and ***Figure 38-15. Addressed BiSS command communication***.

Figure 38-14. Broadcast BiSS command communication

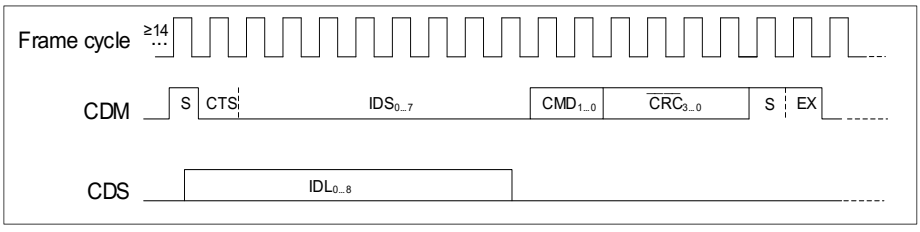
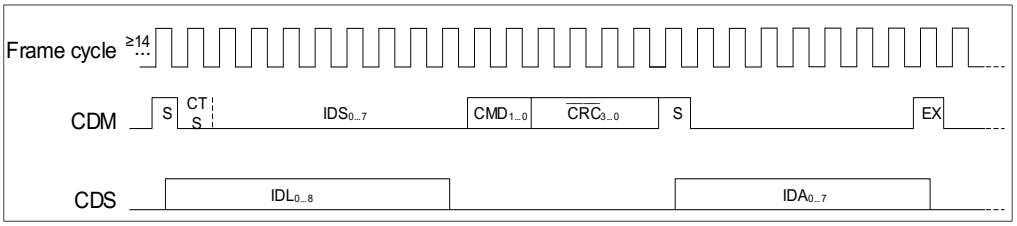


Figure 38-15. Addressed BiSS command communication



Register communication

When CTS bit is set to 1, the register communication is used for control communication. The register communication is used to write and read memory-mapped resources of the specific BiSS slave. The addressed slave is selected by CMD[2:0] in EDIM_BISS_CCFG1 register.

The register communication start address, register communication direction selection and number of consecutive communication register are configured in REGADDR[6:0], REGWRSEL and REGNUM[5:0] of EDIM_BISS_CCFG0 register. The register communication for reading and writing are shown in *Figure 38-16. Register communication for reading* and *Figure 38-17. Register communication for writing*.

Figure 38-16. Register communication for reading

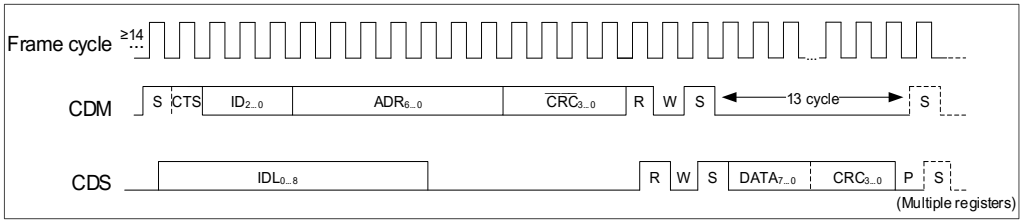
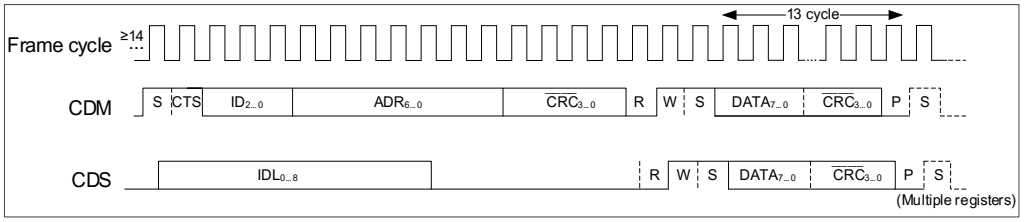


Figure 38-17. Register communication for writing



DMA mode

In some situations, position value requests from the encoder may be frequent, and the CPU may not be able to read out the received data in time. In such case, users can configure DFEN bit and DFENx (x = 0…15) bit in EDIM_BISS_DFCTL register to enable DMA request. If DMA is enabled, DMA requests will be triggered k times by hardware when the EODT bit in the EDIM_BISS_STAT0 register is set to 1. K depends on the configuration of DFENx (x = 0…15) bit. The DMA request is reset by hardware when the circuit receives an acknowledgment from DMA.

The DMA request is controlled by a DMA interface that includes a synchronized FIFO. The FIFO level is 16*4 bytes. When DFENx (x = 0…15) bit is set to 1, the received data is not only stored in the corresponding data registers but also in a FIFO structure, which can be accessed via the EDIM_BISS_FIFODATA register. User can readout data of EDIM_BISS_FIFODATA register by DMA.

For example, if users wish to read EDIM_BISS_S0RDATA0 and EDIM_BISS_S0DATA1 registers by DMA, the DFEN bit, DFEN0 bit and DFEN1 bit in EDIM_BISS_DFCTL register should be set to 1. The data of EDIM_BISS_S0RDATA0 and EDIM_ BISS_S0DATA1 are also

transferred to FIFO. The DMA request will be triggered 2 times by hardware when EODT bit is set to 1. If EDIM_BISS_FIFODATA register address is set as DMA source address, DMA will transfer the data of EDIM_BISS_FIFODATA register.

If FIFO is overrun or underrun, FIFOOVRIF and FIFOUNRIF bits are set to 1 in EDIM_BISS_STAT1 register. Overrun and underrun don't affect data reading. But the data read out is not reliable. FIFOFH bit in EDIM_BISS_DFCTL register is set to 1, which can execute to flush FIFO.

If DMA acknowledge doesn't come back to BISS module within 1024 PCLK1 cycles, DMATOIF bit in EDIM_BISS_STAT1 register will be set to 1 and DMA request will be cleared by hardware.

BISS Interrupts

The interrupt can be produced on one of the events:

- End of data transmission
- Automatic data transmission error
- Start bit error in register communication
- Process data transmission error
- Control communication error
- FIFO overrun, underrun
- DMA timeout error

EODT bit in EDIM_BISS_STAT0 register represents the data transmission status of BISS module. EODT bit is set when BISS module is idle and cleared by hardware when BiSS-C frame transmission begin. The rising edge of the EODT bit triggers the setting of EODTIF bit in EDIM_BISS_STAT1 register. If EODTIE bit in EDIM_BISS_CNTINTEN register is set, an interrupt will be generated when transmission finish.

38.4.5. Registers definition

BISS base address: 0x4000 D000

BISS slave n data register 0 (EDIM_BISS_SnDATA0) (n = 0...7)

Address offset: 0x00 + 0x08 * n
Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:0	DATAL[31:0]	Low double word process data of slave n. These bits indicate received data bit 31 to 0.

BISS slave n data register 1 (EDIM_BISS_SnDATA1) (n=0...7)

Address offset: 0x04 + 0x08 * n
Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:0	DATAH[31:0]	High double word process data of slave n. These bits indicate received data bit 63 to 32.

BISS control communication data x register (EDIM_BISS_CCDATAx) (x = 0...15)

Address offset: 0x80 + 0x04 * x
Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:0	CCDATA[31:0]	**As storing the data of automatic register communication:** When CTS = 1 in EDIM_BISS_CCFG1 register, these bits are used to store the data of register communication. In case of writing to the slave register, these bits must be written before starting the BiSS communication. In case of reading the slave register, the register data is stored in these bits after a successful BiSS control communication. Besides, the register can only be accessed if no control communication is currently active (EOCC = 1 in the EDIM_BISS_STAT0 register). **As slave n selection for BiSS commands** When CTS = 0 in EDIM_BISS_CCFG1 register, these bits are used for BiSS commands. A BiSS command can address one or serval slaves by different IDS[7:0] values. The IDS[7:0] bits correspond to the lower 8 bits of EDIM_BISS_CCDATA0 register. When IDS[7:0] = 0x00, all slaves are addressed. IDS[N] = 0: slave (7-N) is not addressed. IDS[N] = 1: slave (7-N) is addressed.

BISS slave n process data transmission configuration register (EDIM_BISS_SnPDTCFG) (n = 0...7)

Address offset: 0xC0+ 0x04 * n

Reset value: 0x0000 0000

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

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

Bits	Fields	Descriptions
31:16	CRCIVAL[15:0]	CRC calculation initial value Different CRC calculation start values can be utilized to distinguish process data from various slaves.
15	CRCMSEL	CRC polynomial mode selection When CRCPOLY[6:0] = 0, the CRC verification is disabled.

		0: Select prefined CRC polynomials. 1: Select individual CRC polynomials.
14:8	CRCPOLY[6:0]	**As prefined CRC polynomial:** When CRCMSEL = 0, these bits are use to select prefined CRC polynomials 0000000: CRC verification is disabled. 0000011: CRC polynomial is 0x0B 0000100: CRC polynomial is 0x13 0000101: CRC polynomial is 0x25. 0000110: CRC polynomial is 0x43 0000111: CRC polynomial is 0x89 0001000: CRC polynomial is 0x12F 0010000: CRC polynomial is 0x190D9 Others: Reserved. **As individual CRC polynomial:** When CRCMSEL = 1, these bits are used to select a maximum of 8-bit CRC polynomial. The high 7 bits of these bits are determined by CRCPOLY[6:0]. If the CRC polynomial length does not require 8 bits, the higher bits are filled with 0. For example: If the CRC polynomial is 8'b01011101, then CRCPOLY[6:0] is 7'b0101110. When CRCPOLY[6:0] = 0, the CRC verification is disabled.
7	Reserved	Must be kept at reset value
6	PDTEN	Process data transmisson enable 0: Process data is disabled. 1: Process data is enabled.
5:0	PDLEN[5:0]	Process data length 000000: Process data bit length is 1. 000001: Process data bit length is 2. … 111111: Process data bit length is 64.

BISS communication configuration register 0 (EDIM_BISS_CCFG0)

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		REGNUM[5:0]						REGWRSEL	REGADDR[6:0]						
		rw						rw	rw						

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

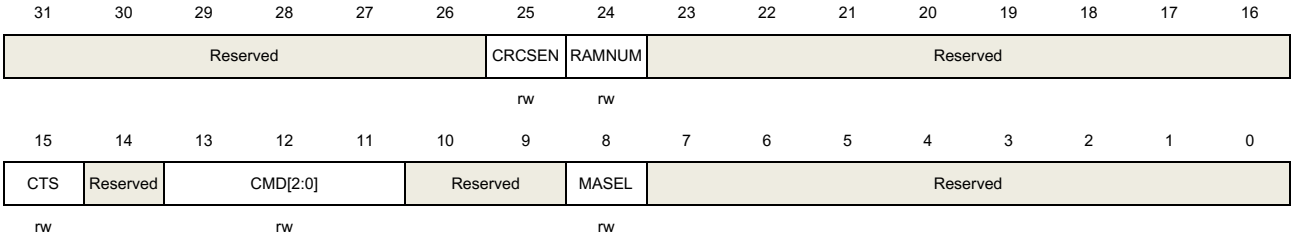
Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value
29:24	REGNUM[5:0]	The number of consecutive communication registers in register communication mode 000000: Register number is 1. 000001: Register number is 2. ... 111111: Register number is 64.
23	REGWRSEL	Register communication write/read mode selection 0: Read register data. 1: Write register data.
22:16	REGADDR[6:0]	Register communication start address
15:0	Reserved	Must be kept at reset value

BISS communication configuration register 1 (EDIM_BISS_CCFG1)

Address offset: 0xE4

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	CRCSEN	CRC storage enable 0: CRC of process data can be stored in EDIM_BISS_SnDATA0/1 (n = 0…7) registers. 1: CRC of process data cannot be stored in EDIM_BISS_SnDATA0/1 (n = 0…7) registers. **Note:** When CRCSEN bit is set to 1 from 0, the last CRC will be leaved in EDIM_BISS_SnDATA0/1 register
24	RAMNUM	Number of RAM banks used for process data 0: Two RAM banks are used for process data. 1: One RAM bank is used for process data.

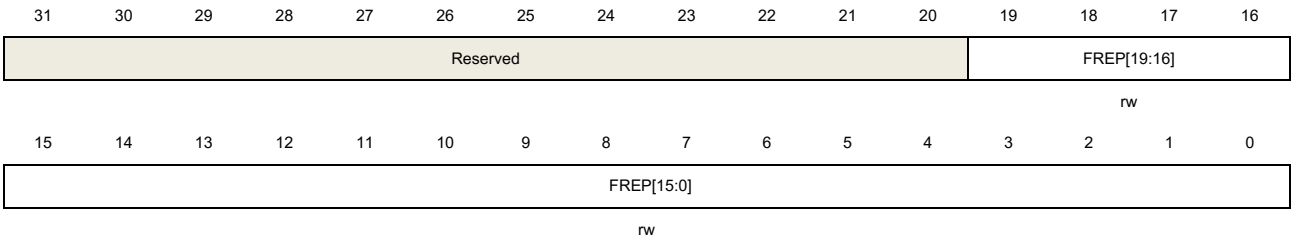
23:16	Reserved	Must be kept at reset value
15	CTS	Control communication mode selection 0: BiSS commands 1: Register communication
14	Reserved	Must be kept at reset value
13:11	CMD[2:0]	**As slave addressing for register communication:** When CTS = 1, these bits are defined as slave addressing for register communication. **As command selection for BiSS commands:** When CTS = 0, CMD[2:1] bits determine which BiSS command is sent. Some BiSS commands are predefined according to the protocol, while others can be customized based on the specific functions of the addressed slave. **As command execution moment selection for BiSS commands:** When CTS = 0, CMD[0] bit determines the execution moment. 0: Execute immediately 1: After successful checking of the ID Acknowledge bit, the master triggers the execution of the BiSS command; if not, the command will not be executed.
10:9	Reserved	Must be kept at reset value
8	MASEL	MA signal level selection at the end of frame. 0: After the slave timeout is detected on SL, MA changes to the digital high level. 1: MA maintains level until the next frame begins.
7:0	Reserved	Must be kept at reset value

BISS frame repetition register (EDIM_BISS_FREP)

Address offset: 0xE8

Reset value: 0x000F FFFF

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



Bits	Fields	Descriptions
31:20	Reserved	Must be kept at reset value
19:0	FREP[19:0]	Frame automatic repetition rate 0x00000: The highest frame rate for automatically restarting the next frame.

0x00001: The master requires a trigger signal to restart the next frame.
0x00002: f_{PCLK1} / (2 * (N + 1)), N = 2
0x00003: f_{PCLK1} / (2 * (N + 1)), N = 3
0xFFFFF: f_{PCLK1} / (2 * (N + 1)), N = 1048575

BISS counter and interrupt enable register (EDIM_BISS_CNTINTEN)

Address offset: 0xEC
Reset value: 0x1FFF 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			FCNT[12:0]												
			rw												

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						FIFOOVRIE	FIFOUNRIE	DMATOIE	Reserved	ADTERRIE	SERRIE	PDTERRIE	CCERRIE	EODTIE	Reserved
						rw	rw	rw		rw	rw	rw	rw	rw	

Bits	Fields	Descriptions
31:29	Reserved	Must be kept at reset value
28:16	FCNT[12:0]	The count value from the BiSS frame's slave response ACK to the CDM timeout confirmation bit. Counting is performed at the clock PCLK1. These bits need to be calculated by the user according to the 40us specified by the BiSS-C protocol. For example, If the frequency of PCLK1 is 100MHz, these bits need to be set to 40us * 100MHz = 4000.
15:10	Reserved	Must be kept at reset value
9	FIFOOVRIE	Interrupt enable of FIFOOVRIF 0: Disable 1: Enable
8	FIFOUNRIE	Interrupt enable of FIFOUNRIF 0: Disable 1: Enable
7	DMATOIE	Interrupt enable of DMATOIF 0: Disable 1: Enable
6	Reserved	Must be kept at reset value
5	ADTERRIE	Interrupt enable of ADTERRIF 0: Disable

		1: Enable
4	SERRIE	Interrupt enable of SERRIF 0: Disable 1: Enable
3	DTERRIE	Interrupt enable of PDTERRIF 0: Disable 1: Enable
2	CCERRIE	Interrupt enable of CCERRIF 0: Disable 1: Enable
1	EODTIE	Interrupt enable of EODTIF 0: Disable 1: Enable
0	Reserved	Must be kept at reset value

BISS status register 0 (EDIM_BISS_STAT0)

Address offset: 0xF0

Reset value: 0x0000 00FB

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CDMTOF	CDS	VREGNUM[5:0]						SLVF[15:8]							
r	r	r						rc_w1							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SLVF[7:0]								INERRF	ADTERRIF	SERRIF	PDTERRIF	CCERRIF	EOCC	EOFT	EODT
rc_w1								r	rs	rs	rs	rs	r	r	r

Bits	Fields	Descriptions
31	CDMTOF	CDM timeout completion flag in control communication In BiSS-C control communication, a minimum number of BiSS-C frame cycles with CDM set to 0 must be sent before initiating new control communication. For manual control communication by software, the CDMTOF bit indicates that at least 14 BiSS-C frame cycles with CDM = 0 have been transmitted. 0: CDM timeout of control communication not completed 1: CDM timeout of control communication completed
30	CDS	CDS bit sent by slave 0: CDS = 0 1: CDS = 1

Note: There are two RAM banks for CDS bit that switch simultaneously to EDIM_BISS_SnDATA0/1 (n = 0…7) RAM banks.

29:24	VREGNUM[5:0]	Number of valid register from addressed slave in register communication These bits define the number of valid register in register communication. They are reset when initializing register communication or starting a new control communication. These bits are written only in the event of a register communication error.
23:8	SLVF[15:0]	Slave n data validity flag Each slave is allocated two adjacent bits, the higher of which indicates data validity, while the lower is fixed at 0. Data validity is determined based on the CRC check result of the process data. If CRC checking is disabled, the corresponding bit for the slave n is set to 1 after the process data has been sampled. 0: Process data is invalid. 1: Process data is valid. **Note:** After reading process data, the corresponding bit should be reset. These bits are cleared by writing a 1 to them. There are two RAM banks for SLVF[15:0] that switch simultaneously to EDIM_BISS_SnDATA0/1 (n = 0…7) RAM banks.
7	INERRF	Internal error flag The internal errors include automatic data transmission error, start bit error in register communication, process data transmission error and control communication error. 0: Any internal error occurred 1: No internal error occurred
6	ADTERRIF	Automatic data transmission error interrupt flag When automatic communication is enabled (ADTEN = 1 in the EDIM_BISS_CCFG2 register), this bit is valid. This bit is set after power on and executing the initialization sequence (TINIT = 1 in the EDIM_BISS_CCFG2 register). The automatic data transmission error can be divided into five type: - When SL line level is low and current state is idle phase of BiSS-C frame, FCTL[2:0] in EDIM_BISS_CCFG2 register is written. - When ADTEN = 1, a new frame transmission cycle could not be initiated. - When ADTEN = 1, if the last BiSS frame is not completed in time, the next BiSS frame will be skipped. - When the initialization frame is sent (TINIT = 1), the slave does not respond with an ACK within 40 us. - If a new BiSS frame is scheduled before RBSWDIS bit in EDIM_BISS_CCFG2 register is reset, no RAM bank will be switched and ADTERRIF bit will be set. 0: Automatic data transmission error occurred. 1: No automatic data transmission error occurred.
5	SERRIF	Start bit error interrupt flag in register communication

This bit is set after power on and executing the initialization sequence (TINIT = 1 in the EDIM_BISS_CCFG2 register).
0: Start bit was missed in the last register communication. The addressed slave might not be executed or the time to access the slave register is too long (>20ms).
1: No start bit error occurred..
Note: The master can terminate register communication after t_{busy_r} as defined in the BiSS-C Protocol Description.

4	PDTERRIF	Process data transmission error interrupt flag This bit indicates the error in process data, verified using CRC. This bit is set after power on and executing the initialization sequence (TINIT = 1 in the EDIM_BISS_CCFG2 register). 0: Process data transmission error occurred in the last transmission. The faulty slave data can be ensured in SLAVF[15:0] 1: No process data transmission error occurred.
3	CCERRIF	Control communication error interrupt flag This bit is set after power on and executing the initialization sequence (TINIT = 1 in the EDIM_BISS_CCFG2 register). If an error occurs, data transmission is halted. The control communication error can be divided into three type: - Read register access issues: CRC errors or an inverted R-bit - Errors in writing register access within repeated CDS data. - BiSS Commands: Errors in the IDA field. When a register data error in register communication is detected, the VREGNUM[5:0] in EDIM_BISS_STAT0 register represents the correct register number before the error occurred. This bit is set after power on and executing the initialization sequence (TINIT = 1 in the EDIM_BISS_CCFG2 register). 0: Last control communication error occurred. 1: No last control communication error occurred.
2	EOCC	End of control communication flag 0: Control communication has not been finished or has not been started after initialization sequence (TINIT = 1 in the EDIM_BISS_CCFG2 register). 1: Control communication has been finished or has been terminated due to an error.
1	EOFT	End of frame transmission flag This bit indicates the completion of the entire BiSS-C frame, including the timeout phase. It is cleared by hardware when a BISS-C frame starts. 0: Frame transmission active. 1: Frame transmission not active.
0	EODT	End of data transmission flag This bit only indicates the end of process data and CRC transmission and does not include the timeout phase of BiSS-C frame. It is cleared by hardware when a BISS-

C frame starts.

0: Data transmission active.

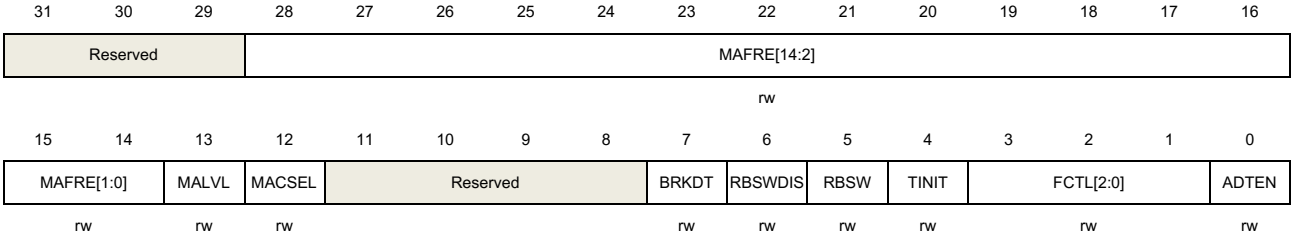
1: Data transmission not active.

BISS communication configuration register 2 (EDIM_BISS_CCFG2)

Address offset: 0xF4

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
28:14	MAFRE[14:0]	Process data master clock frequency f_{MA} The f_{MA} is generated for BiSS from the system clock (PCLK1) and depends on the frequency division by MAFRE[14:0]. 0x0000 - 0x0007: Reserved 0x0008: f_{PCLK1} / (N + 1), N = 8 … 0x7FFF: f_{PCLK1} / (N + 1), N = 32767
13	MALVL	MA line force level 0: Low 1: High
12	MACSEL	MA line level control source selection 0: MA line level is not controlled by MALVL bit 1: MA line level is controlled by MALVL bit
11:8	Reserved	Must be kept at reset value
7	BRKDT	Break data transmission This bit will be cleared by hardware. 0: Not break 1: Break **Note:** TINIT bit should be set after breaking data transmission to ensure resetting INERRF, ADTERRIF, SERRIF, PDTERRIF and CCERRIF bits in EDIM_BISS_STAT0 register.
6	RBSWDIS	RAM bank automatic switching disable

0: Automatic RAM bank switching is enabled

1: Automatic RAM bank switching is disabled

Note: If RBSWDIS = 1, RAM bank will be switched after RBSWDIS bit is set to 0. If a new BiSS frame is scheduled before RBSWDIS is set to 0, no RAM bank will be switched and ADTERRIF bit in the EDIM_BISS_STAT0 register will be set.

5 RBSW RAM bank force switching

RAM bank is switched immediately by toggling this bit..

4 TINIT Transmit BiSS initialization sequence

This bit can only be configured when ADTEN is 0 and during the idle phase of the BiSS-C frame.

0: Not transmit

1: Transmit

Note: A line delay will be measured after transmitting BiSS initialization sequence, and the measured data will be stored in DATAL[12:0] of the EDIM_BISS_S1DATA0 register. The line delay time is DATAL[12:0] / f_{PCLK}. When the line delay time exceeds 40us, the initialization sequence will be break and ADTERRIF bit in the EDIM_BISS_STAT0 register will be reset. When the BiSS initialization sequence is sent, the state of the SL line is not checked. As a result, a constant SL = 0 is ignored.

3:1 FCTL[2:0] Frame control

These bits can only be configured when ADTEN is 0 and during the idle phase of the BiSS-C frame.

As trigger BiSS frame transmission:

When ADTEN = 0, these bits are used to trigger BiSS frame transmission.

000: No frame transmission

001: Trigger one frame with CDM = 1, and these bits will be automatically reset.

010: Trigger one frame with CDM = 0, and these bits will be automatically reset.

011: Trigger one frame with CDM being the inverse of CDS, and these bits will be automatically reset.

100: Trigger one frame with control communication, and these bits will be automatically reset.

101 - 110: Reserved.

111: Trigger one reduced frame with control communication, and these bits will be automatically reset.

As control next BiSS frame:

When ADTEN = 1, these bits are used to control next BiSS frame transmission.

000: Reserved

001: Next frame with CDM = 1.

010: Next frame with CDM = 0.

011: Next frame with CDM being the inverse of CDS.

100: Next frame with control communication.

101 - 110: Reserved.

111: Next reduced frame with control communication.

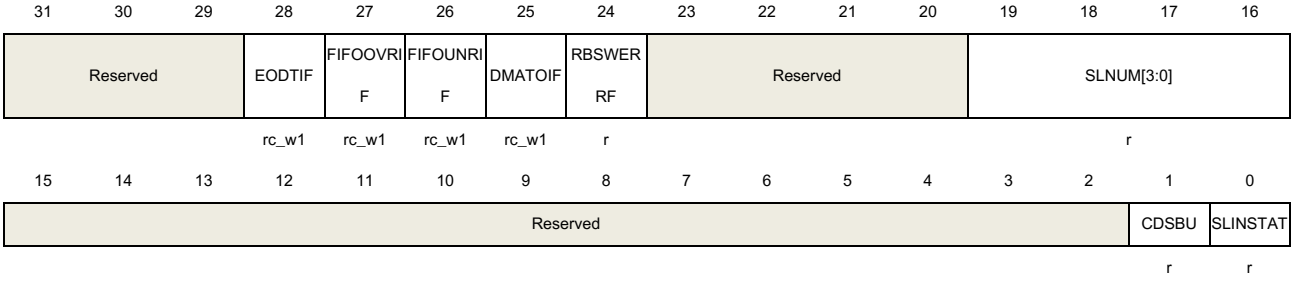
0	ADTEN	Automatic data transmission enable 0: Automatic data transmission is disabled. 1: Automatic data transmission is enabled.

BISS status register 1 (EDIM_BISS_STAT1)

Address offset: 0xF8

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	EODTIF	End of data transmission interrupt flag 0: Data transmission is not completed 1: Data transmission is completed
27	FIFOOVRIF	FIFO overrun error interrupt flag 0: No FIFO overrun error occurred 1: Overrun error occurred
26	FIFOUNRIF	FIFO underrun error interrupt flag 0: No FIFO underrun error occurred 1: Underrun error occurred
25	DMATOIF	DMA request timeout error interrupt flag 0: No DMA requeset timeout error occurred 1: DMA request timeout error occurred
24	RBSWERRF	RAM bank switching error flag The RAM bank switching error type: - When a RAM bank switch is executed at the end of a BiSS-C frame but RBSWDIS bit is set to 1 then RBSWERRF will be set to 1. 0: No RAM bank switching error occurred. 1: RAM bank switching error occurred.
23:20	Reserved	Must be kept at reset value
19:16	SLNUM[3:0]	Number of slaves in control communication These bits are valid after a control communication are performed.

0000: No slaves
0001: 1 slave
0010: 2 slaves
0011: 3 slaves
0100: 4 slaves
0101: 5 slaves
0110: 6 slaves
0111: 7 slaves
1000: 8 slaves
1001: More than 8 slaves.
Others: Reserved

15:2	Reserved	Must be kept at reset value
1	CDSBU	CDS bit backup This bit is the same as the CDS bit in the EDIM_BISS_STAT0 register. **Note:** There are two RAM banks for CDSBU bit that switch simultaneously to EDIM_BISS_SnDATA0/1 (n = 0…7) RAM banks.
0	SLINSTAT	SL line input level state 0: Low 1: High

BISS DMA and FIFO control register (EDIM_BISS_DFCTL)

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
DFEN15	DFEN14	DFEN13	DFEN12	DFEN11	DFEN10	DFEN9	DFEN8	DFEN7	DFEN6	DFEN5	DFEN4	DFEN3	DFEN2	DFEN1	DFEN0
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														FIFOFH	DFEN
														w	rw

Bits	Fields	Descriptions
31	DFEN15	DMA request and FIFO enable for EDIM_BISS_S7RDATA1 register 0: DMA request and FIFO disabled 1: DMA request and FIFO enabled
30	DFEN14	DMA request and FIFO enable for EDIM_BISS_S7RDATA0 register 0: DMA request and FIFO disabled 1: DMA request and FIFO enabled
29	DFEN13	DMA request and FIFO enable for EDIM_BISS_S6RDATA1 register

0: DMA request and FIFO disabled
1: DMA request and FIFO enabled

28	DFEN12	DMA request and FIFO enable for EDIM_BISS_S6RDATA0 register 0: DMA request and FIFO disabled 1: DMA request and FIFO enabled
27	DFEN11	DMA request and FIFO enable for EDIM_BISS_S5RDATA1 register 0: DMA request and FIFO disabled 1: DMA request and FIFO enabled
26	DFEN10	DMA request and FIFO enable for EDIM_BISS_S5RDATA0 register 0: DMA request and FIFO disabled 1: DMA request and FIFO enabled
25	DFEN9	DMA request and FIFO enable for EDIM_BISS_S4RDATA1 register 0: DMA request and FIFO disabled 1: DMA request and FIFO enabled
24	DFEN8	DMA request and FIFO enable for EDIM_BISS_S4RDATA0 register 0: DMA request and FIFO disabled 1: DMA request and FIFO enabled
23	DFEN7	DMA request and FIFO enable for EDIM_BISS_S3RDATA1 register 0: DMA request and FIFO disabled 1: DMA request and FIFO enabled
22	DFEN6	DMA request and FIFO enable for EDIM_BISS_S3RDATA0 register 0: DMA request and FIFO disabled 1: DMA request and FIFO enabled
21	DFEN5	DMA request and FIFO enable for EDIM_BISS_S2RDATA1 register 0: DMA request and FIFO disabled 1: DMA request and FIFO enabled
20	DFEN4	DMA request and FIFO enable for EDIM_BISS_S2RDATA0 register 0: DMA request and FIFO disabled 1: DMA request and FIFO enabled
19	DFEN3	DMA request and FIFO enable for EDIM_BISS_S1RDATA1 register 0: DMA request and FIFO disabled 1: DMA request and FIFO enabled
18	DFEN2	DMA request and FIFO enable for EDIM_BISS_S1RDATA0 register 0: DMA request and FIFO disabled 1: DMA request and FIFO enabled
17	DFEN1	DMA request and FIFO enable for EDIM_BISS_S0RDATA1 register 0: DMA request and FIFO disabled

		1: DMA request and FIFO enabled
16	DFEN0	DMA request and FIFO enable for EDIM_BISS_S0RDATA0 register 0: DMA request and FIFO disabled 1: DMA request and FIFO enabled
15:2	Reserved	Must be kept at reset value
1	FIFOFH	FIFO flush 0: No action 1: Clear the FIFO data
0	DFEN	DMA request and FIFO enable 0: DMA request and FIFO disabled 1: DMA request and FIFO enabled

BISS FIFO data register (EDIM_BISS_FIFODATA)

Address offset: 0x200

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FIFODATA[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIFODATA[15:0]															
r															

Bits	Fields	Descriptions
31:0	FIFODATA[31:0]	FIFO data from EDIM_BISS_SnDATA0/1 (n = 0…7) registers.

38.5. HIPERFACE DSL encoder interface module (HDSL)

38.5.1. Overview

The HIPERFACE DSL encoder interface module serves as a master to communicate with encoders that support the HIPERFACE DSL

38.5.2. Characteristics

- Support communication with encoder supporting HIPERFACE DSL.
- Support HIPERFACE DSL safety function.
- Drive interface and safe channel 1 interface can be configured for access either from the internal bus or through SPI interface 1.
- Safe channel 2 interface can be configured for access either from the internal bus or through SPI interface 2.
- System diagnostics function
- Fast position function
  - Estimator
  - Free running mode
  - SYNC mode
- Monitor the status and error messages
  - Event register
  - Online status
  - Status summary of the motor feedback system
  - Motor feedback system error messages
- Communication status error interrupt.
- Supports baud rate up to 9.375 MHz

38.5.3. Block diagram

Figure 38-18. HDSL master block diagram

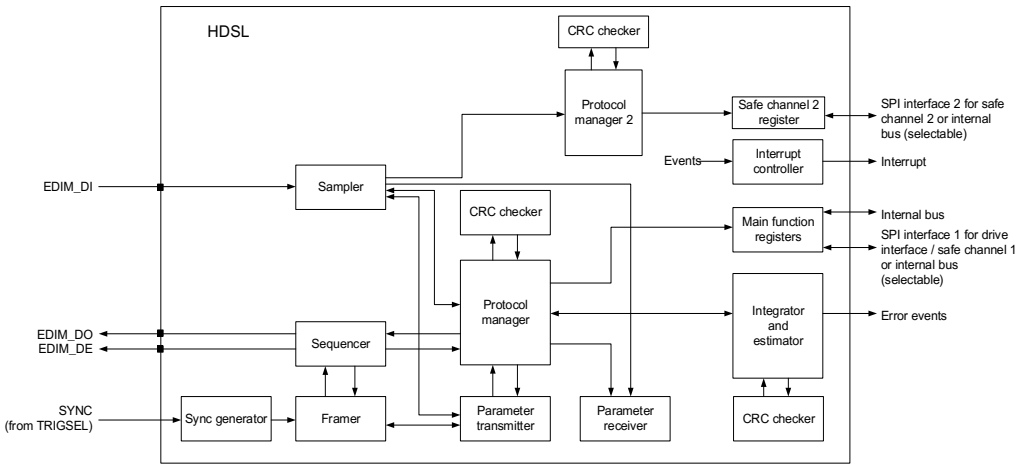


Figure 38-18. HDSL master block diagram provides details of the internal configuration of the DSL master. *Table 38-19. HDSL pins definition* gives the HDSL pin description.

Table 38-19. HDSL pins definition

Name	Description
EDIM_DE	HDSL transceiver control signal
EDIM_DO	HDSL data output
EDIM_DI	HDSL data input
HDSL_SEL1	HDSL SPI selection safe channel 1
HDSL_CLK1	HDSL SPI clock safe channel 1
HDSL_MISO1	HDSL data output safe channel 1
HDSL_MOSI1	HDSL data input safe channel 1
HDSL_SEL2	HDSL SPI selection safe channel 2
HDSL_CLK2	HDSL SPI clock safe channel 2
HDSL_MISO2	HDSL data output safe channel 2
HDSL_MOSI2	HDSL data input safe channel 2
HDSL_LINK	HDSL link
HDSL_SMPL	HDSL test signal line sampler

38.5.4. Function overview

This HIPERFACE DSL® MASTER is licensed from SICK AG. For details, see the SICK document: HIPERFACE DSL® MASTER Safety Integration Manual.

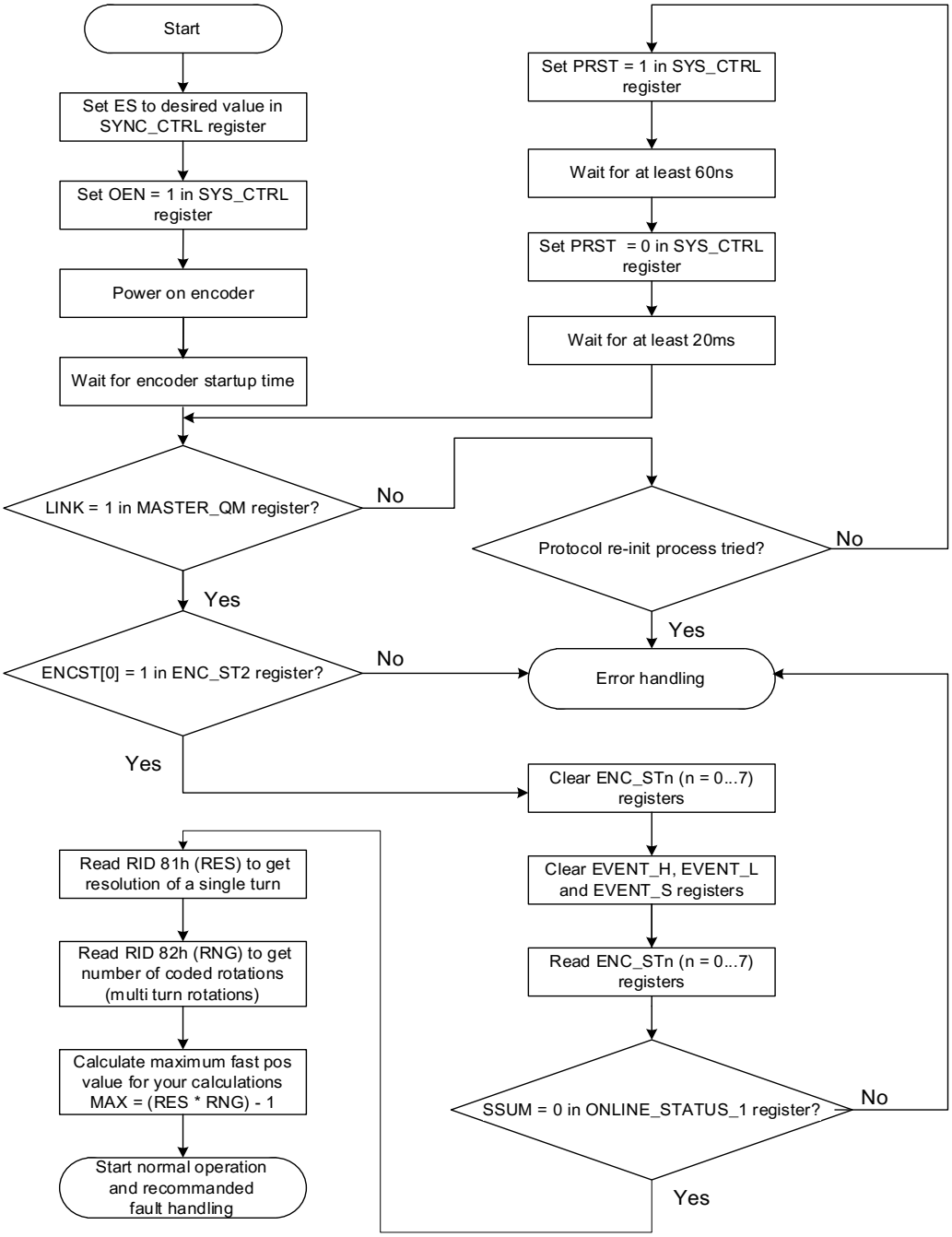
HDSL clock

The function clock of DSL Master is CK_HDSL, with a frequency of 75MHz. The CK_HDSL is provided by an PLL or system clock, configured in corresponding RCU register.

System start

The recommended procedure for system start is shown in *Figure 38-19. HDSL master system start procedure*.

Figure 38-19. HDSL master system start procedure



Quality monitoring

The MASTER_QM register contains the quality monitoring value for the data connection.

As soon as the DSL Master detects events that indicate an improvement or degradation of the quality of the data connection, these events are indicated as values higher or lower than the quality monitoring value, as shown in ***Table 38-20. Quality monitoring events***.

Table 38-20. Quality monitoring events

Protocol event	Value change in quality monitoring
Wrong synchronization in Safe Channel (last byte)	-4
Wrong synchronization in Safe Channel (1st …7th bytes)	-6
RSSI <2	-4
Wrong encoding in parameter or SensorHub channels	-1
Wrong encoding in process data channel	-2
Unknown special characters in the protocol package	-2
Any identified error in Safe Channel 1	-6
Any identified error in Safe Channel 2	-8
Correct synchronization in Safe Channel	+1
Correct CRC value in Safe Channel	+1

Quality monitoring is initiated with the value "8".

The maximum quality monitoring value is "15". This is the standard value during operation.

Note: Particular attention must be paid to the quality monitoring value during the development of a DSL drive controller. If a value lower than "15" is indicated, the cause may be a problem with the connection circuit, particularly if the value is continuously displayed.

If the quality monitoring value falls below "14", QMLW information is indicated in Online Status and in the EVENT_L register.

If the quality monitoring value falls to "0", a forced reset of the protocol is carried out.

This is indicated by the PRST bit in Online Status and in the EVENT_H register.

HDSL interrupts

The interrupt will be generated if an interrupt condition has been fulfilled in the DSL Master. The interrupt conditions are configured in MASK_H, MASK_L and MASK_SUM registers. The interrupt conditions for safety are configured in MASK_S register.

Figure 38-20. Interrupt condition and *Figure 38-21. Interrupt condition for safety* show an interrupt condition as a diagram.

Figure 38-20. Interrupt condition

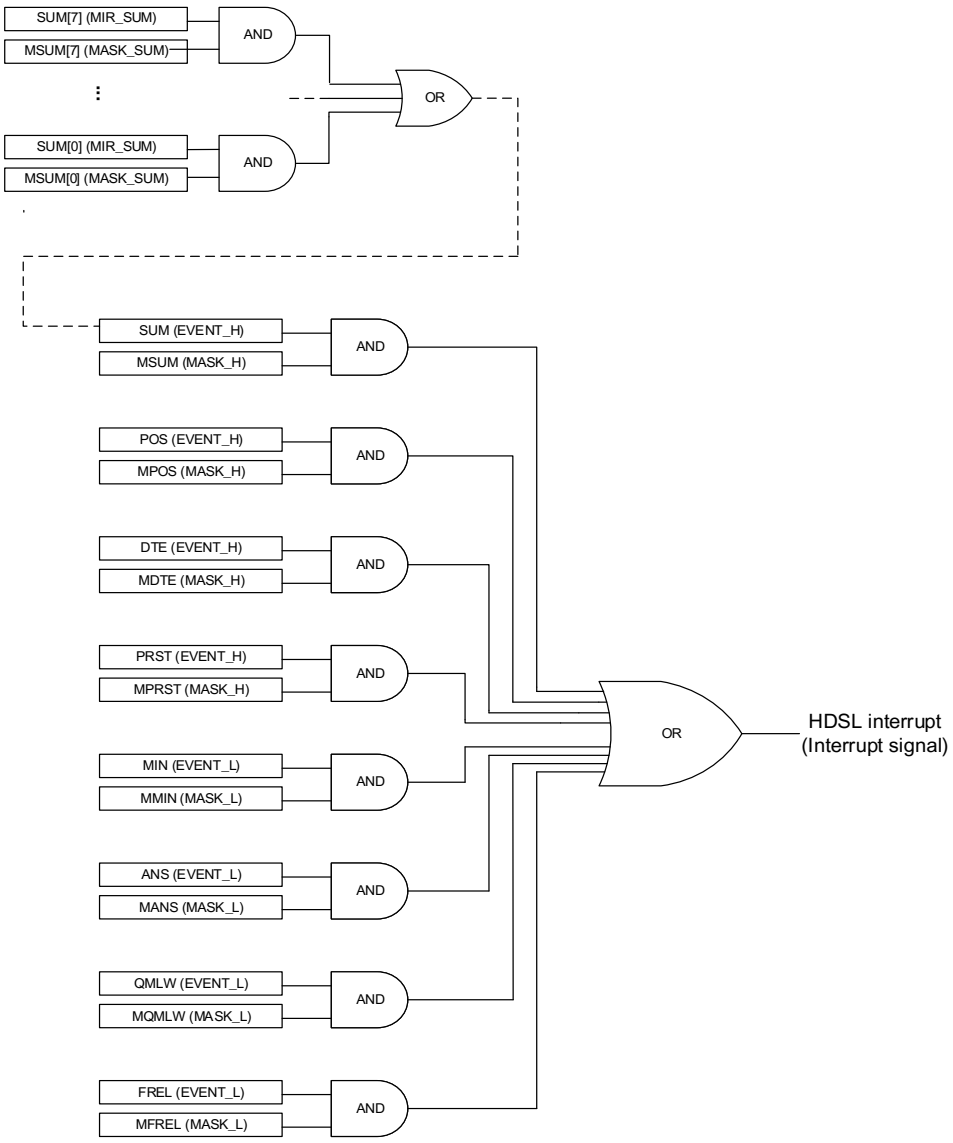
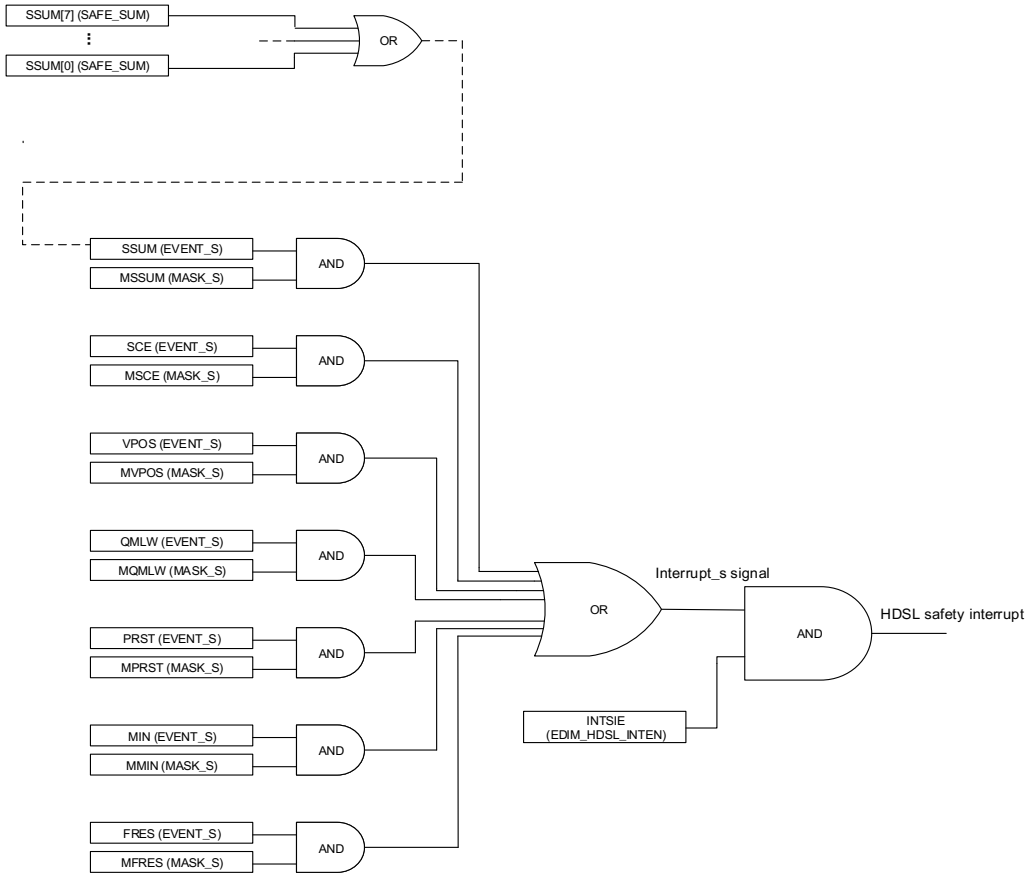


Figure 38-21. Interrupt condition for safety



Fast position

The fast position and the rotation speed of the encoder shaft are transmitted on the DSL motor feedback system process data channel. These values are the main process values for the drive application control circuit.

HIPERFACE DSL stores the fast position in the POSn (n = 0…4) registers and the rotation speed in the VELn (n = 0…2) registers.

The position is given as a 40 bit value that includes the angle setting ("single-turn" value) and the number of rotations ("multi-turn" value). Only the position bits actually measured by the motor feedback system are accessible and are stored in the registers as a right-justified value. The other (higher value) bits are constantly set at "0", as shown in ***Table 38-21. Position data in POSn (n = 0…4) registers***.

Table 38-21. Position data in POSn (n = 0…4) registers

Type	POS4	POS3	POS2	POS1	POS0
Single turn 17 bit	-	-	ST[16]	ST[15:8]	ST[7:0]
Single turn 17 bit Multi turn 12 bit	-	MT[11:7]	{MT[6:0], ST[16]}	ST[15:8]	ST[7:0]

Type	POS4	POS3	POS2	POS1	POS0
Single turn 23 bit Multi turn 12 bit	MT[11:9]	MT[8:1]	{MT[0], ST[22:16]}	ST[15:8]	ST[7:0]

Note:

1. '-' default value is 0.
2. ST: Single turn bit (angular position)
3. MT: Multi turn bit (number of revolutions)

The motor feedback system fast position is sampled and transmitted if the DSL Master receives a SYNC signal.

This SYNC signal can be created in two different ways: free running mode and SYNC mode.

Free running mode

In free-running mode, the SYNC signal is automatically created by the DSL Master, for which the maximum frame transmission frequency is used. The free-running mode is the standard DSL master operating mode at start-up. This operating mode can also be selected manually, by setting the ES[7:0] in the SYNC_CTRL register to 0.

SYNC mode

In SYNC mode, the DSL Master depends on a prepared cyclic control signal. This control signal triggers position measurements and enables polling of position and rotation speed values synchronously with the control signal. The control signal must be applied to the SYNC input and have the characteristics prescribed for the DSL Master.

The position is available after a set delay in relation to the leading edge of the control signal.

When SYNC mode is used the following points must be noted:

1. A correct control signal must be applied at the sync input. The signal must correspond with the specifications for pulse width and cycle time.
2. Setting or deleting the SPOL bit in the SYS_CTRL register determines whether the position measurements are to be triggered by the leading or the trailing edge of the control signal. The set latency of the DSL system is measured from this edge.
3. The correct ES[7:0] divider must be set in the SYNC_CTRL register. This divider determines how many position samplings and transmissions will be undertaken for each control signal. Do not change setting of ES[7:0] divider during start-up or operation.

38.5.5. Registers definition

HDSL has main function registers for the drive interface, safe channel 1 interface, safe channel 2 interface and general control interface.

The drive interface registers, safe channel 1 interface registers, safe channel 2 interface register and general control interface can be accessed through the internal bus.

The drive interface registers and safe channel 1 interface registers can also be accessed through SPI interface 1. The safe channel 2 interface registers can also be accessed through SPI interface 2. In this case, register address should be referred to offset address only.

For drive interface, safe channel 1 interface and safe channel 2 interface registers access operation can be executed with 8-bit/16-bit/32-bit access. However, when 16-bit/32-bit access with unappropriate R/W attribution for the registers is executed, HDSL operation cannot be guaranteed.

HDSL drive interface registers

HDSL drive interface base address: 0x4000 D800

HDSL system control register (SYS_CTRL)

Address offset: 0x00
Reset value: 0x80

The SYS_CTRL register contains the main control bits of the DSL master.

This register has to be accessed by byte (8-bit).

7	6	5	4	3	2	1	0
PRST	MRST	FRST	LOOP	Reserved		SPOL	OEN
rw	rw	rw	rw			rw	rw

Bits	Fields	Descriptions
7	PRST	Protocol reset 0: Normal protocol action 1: A forced reset of the protocol status will be initiated. If the bit is cleared, a restart of the connection is triggered.
6	MRST	Message reset 0: Normal parameters channel action 1: The parameters channel is reset. Current short and long messages are discarded.
5	FRST	Pipeline FIFO reset 0: Normal FIFO access 1: The FIFO is reset. Data is not stored and cannot be read.

4	LOOP	Test drive interface Value for the read-back test for drive interface. This value has no other purpose.
3:2	Reserved	Must be kept at reset value
1	SPOL	Polarity of the synchronization pulse 0: The SYNC tralling edge is used. 1: The SYNC leading edge is used.
0	OEN	Activation of the output 0: The impedance of the DSL cable is high. 1: The DSL cables are activated for output to the DSL Slave.

HDSL synchronization control register (SYNC_CTRL)

Address offset: 0x01
Reset value: 0x00

The synchronization control register for control of the synchronization contains the bit with which the synchronization source for position sampling is controlled.

This register has to be accessed by byte (8-bit).

7	6	5	4	3	2	1	0
ES[7:0]							
w							

Bits	Fields	Descriptions
7:0	ES[7:0]	External synchronization 0x00: Position sampling during free running at the shortest cycle time. Others: Position sampling with the SYNC input synchronized. The value from ES determines the number of position samplings carried out in one SYNC cycle. The user must match the number of samplings per cycle to the shortest frame length.

HDSL quality monitoring register (MASTER_QM)

Address offset: 0x03
Reset value: 0x08

The MASTER_QM register contains the quality monitoring value for the data connection.

This register has to be accessed by byte (8-bit).

7	6	5	4	3	2	1	0
LINK	Reserved			QM[3:0]			
r				r			

Bits	Fields	Descriptions
7	LINK	DSL protocol connection status

		It should be noted that LINK is also represented at the HDSLn_LINK output. (n = 0, 1). 0: No connection present or connection error due to a communications error. 1: Protocol connection between DSL master and Slave was established
6:4	Reserved	Must be kept at reset value
3:0	QM[3:0]	Quality monitoring bits Quality monitoring value. Higher values indicate a better connection. If the quality monitoring reaches the value 0x0, a forced reset of the protocol is carried out.

HDSL high byte event register (EVENT_H)

Address offset: 0x04
Reset value: 0x00

This register has to be accessed by byte (8-bit).

7	6	5	4	3	2	1	0
INT	SUM	Reserved		POS	Reserved	DTE	PRST
r	rc_w0			rc_w0		rc_w0	rc_w0

Bits	Fields	Descriptions
7	INT	Interrupt status 0: No interrupt in DSL master 1: DSL master interrupt occurred
6	SUM	Remote event monitoring When the SUM bit is set, an error or a warning has been transmitted from the DSL Slave. The frequency inverter application must check the MIR_SUM register to obtain a detailed description. This bit is level sensitive. 0: All DSL Slave events are cleared. 1: The DSL Slave has signaled an event and the summary mask is set accordingly.
5:4	Reserved	Must be kept at reset value
3	POS	Estimator turned on This error usually indicates a transmission error on the DSL connection. If this error occurs frequently, the wiring of the DSL connection should be checked. If this error occurs continuously, there is probably an error in the motor feedback system. 0: The data for the fast position was correctly transmitted. 1: Fast position data consistency error. The fast position read through drive interface is supplied by the estimator.
2	Reserved	Must be kept at reset value
1	DTE	Estimator deviation threshold error 0: Current value of deviation smaller than the specified maximum.

		1: Current value of deviation greater than the specified maximum.
0	PRST	Protocol reset warning 0: Normal protocol action 1: The forced protocol reset was triggered.

HDSL low byte event register (EVENT_L)

Address offset: 0x05
Reset value: 0x00

The EVENT_H / EVENT_L register contains the messaging bits for all warning and error modes of the DSL system.

All messaging bits are set by the DSL master if a corresponding status is determined.

An event bit that has been set is not reset by the DSL master. The frequency inverter application must clear bits that have been set.

This register has to be accessed by byte (8-bit).

7	6	5	4	3	2	1	0
Reserved		MIN	ANS	Reserved	QMLW	FREL	Reserved
		rc_w0	rc_w0		rc_w0	rc_w0	

Bits	Fields	Descriptions
7:6	Reserved	Must be kept at reset value
5	MIN	Message initialization When this warning is displayed, the Parameters Channel is still in the initialization status and no "short message" or "long message" can be triggered. 0: No acknowledgment for the initialization received. 1: An acknowledgment was received from the Slave for the initialization of a message.
4	ANS	Erroneous answer to "long message" This error indicates that the transmission of an answer from the DSL Slave to the last "long message" failed. The frequency inverter application must send the "long message" again. 0: The last answers to "long messages" were error-free. 1: An error occurred during the answer to a long message. The effectiveness of the previous transaction is not known.
3	Reserved	Must be kept at reset value
2	QMLW	Quality monitoring low-value warning This warning indicates that a transmission error occurred at bit level for one of the CRC values. If this error occurs frequently, the wiring of the DSL connection should be checked.

		0: Quality monitoring value greater than or equal to "14" 1: Quality monitoring value below "14"
1	FREL	Channel free for "long message" If the bit is set, the frequency inverter application can trigger a "long message". Provided no answer has been received from the DSL Slave, this bit remains deleted. As the processing duration of a "long message" in the motor feedback system is not specified, a user time limit condition should be installed via the frequency inverter application. When a time limit is exceeded, the MRST bit in the SYS_CTRL register is set, which causes the Parameters Channel to be reset. 0: No "long message" can be sent. 1: A "long message" can be sent on the Parameters Channel.
0	Reserved	Must be kept at reset value

HDSL high byte event mask register (MASK_H)

Address offset: 0x06
Reset value: 0x00

This register has to be accessed by byte (8-bit).

7	6	5	4	3	2	1	0
Reserved	MSUM	Reserved		MPOS	Reserved	MDTE	MPRST
	w			w		w	w

Bits	Fields	Descriptions
7	Reserved	Must be kept at reset value
6	MSUM	Mask for remote event monitoring 0: DSL Slave events that are masked in the SUMMARY register do not set the interrupt signal. 1: DSL Slave events that are masked in the SUMMARY register set the interrupt signal.
5:4	Reserved	Must be kept at reset value
3	MPOS	Mask for fast position error 0: An error in the fast position does not set the interrupt signal. 1: An error in the fast position sets the interrupt signal.
2	Reserved	Must be kept at reset value
1	MDTE	Mask for estimator deviation threshold error warning 0: A high estimator deviation threshold error value does not set the interrupt signal. 1: A high estimator deviation threshold error value sets the interrupt signal.

0	MPRST	Mask for protocol reset warning 0: A protocol reset does not set the interrupt signal. 1: A protocol reset sets the interrupt signal.

HDSL low byte event mask register (MASK_L)

Address offset: 0x07
Reset value: 0x00

In the MASK_H/MASK_L registers, the events are set with which the interrupt signal is set.

This register has to be accessed by byte (8-bit).

7	6	5	4	3	2	1	0
Reserved		MMIN	MANS	Reserved	MQMLW	MFREL	Reserved
		w	w		w	w	

Bits	Fields	Descriptions
7:6	Reserved	Must be kept at reset value
5	MMIN	Mask for message initialization confirmation 0: The acknowledgment for the initialization of a DSL Slave message does not set the interrupt signal. 1: The acknowledgment for the initialization of a DSL Slave message sets the interrupt signal.
4	MANS	Mask for erroneous answer to "long message" 0: A transmission error during the answer to a long message does not set the interrupt signal. 1: A transmission error during the answer to a long message sets the interrupt signal.
3	Reserved	Must be kept at reset value
2	MQMLW	Mask for low-quality monitoring value warning 0: A low-quality monitoring value does not set the interrupt signal. 1: A low-quality monitoring value sets the interrupt signal.
1	MFREL	Mask for "channel free for "long message"" 0: If a "long message" can be sent on the Parameters Channel, the interrupt signal is not set. 1: If a "long message" can be sent on the Parameters Channel, the interrupt signal is set.
0	Reserved	Must be kept at reset value

HDSL summary mask register (MASK_SUM)

Address offset: 0x08
Reset value: 0x00

In the MASK_SUM register, the DSL Slave collective events are determined with which the SUM event monitoring in the EVENT_H register, as well as the interrupt signal, are set.

Several events can be masked to trigger an interrupt. In addition, events from the DSL master can be combined with these events.

This register has to be accessed by byte (8-bit).

7	6	5	4	3	2	1	0
MSUN[7:0]							
w							

Bits	Fields	Descriptions
7:0	MSUM[7:0]	Mask for status summary bits 0: In the set status, the corresponding status summary bit does not set the SUM event monitoring and the interrupt signal. 1: In the set status, the corresponding status summary bit sets the SUM event monitoring and the interrupt signal.

HDSL edge register (EDGES)

Address offset: 0x09
Reset value: 0x00

The EDGES register contains the time control for the DSL cable bit sampling and can be used to monitor the connection quality.

Each individual EDGES register bit is set if, at system start-up, an edge of the test signal is detected during the time period of the corresponding bit. An edge is defined as a change in cable value between successive detections. The sampling is carried out eight times as fast as the cable bit rate.

The register is write-protected. The contents of this register do not change after the start-up phase. A new bit sampling pattern is only generated after a forced reset of the protocol.

This register has to be accessed by byte (8-bit).

7	6	5	4	3	2	1	0
EDGES[7:0]							
r							

Bits	Fields	Descriptions
7:0	EDGES[7:0]	Identification of edges in the cable signal 0: No edge was detected in the time period of the corresponding bit.

1: An edge was detected in the time period of the corresponding bit.

HDSL run time delay register (DELAY)

Address offset: 0x0A
Reset value: 0x00

The DELAY register stores information about the run-time delay of the system cable and the signal strength. The register can be used to monitor the connection quality.

The register is write-protected.

The value for Line Delay does not change after the start-up phase. A fresh value for Line Delay is only measured after a forced reset of the protocol.

This register has to be accessed by byte (8-bit).

7	6	5	4	3	2	1	0
RSSI[3:0]				CBLDLY[3:0]			
r				r			

Bits	Fields	Descriptions
7:4	RSSI[3:0]	Indication of the received signal strength 4-bit value for the cable signal strength, from "0" to "12". Higher values indicate better connection quality. Low connection (< 2) quality affects QM. RSSI is continuously updated during operation and used for signal monitoring during run time.
3:0	CBLDLY	4-bit value for cable delay This value gives the cable signal round trip delay of cable and transceivers in bits. This value enables a rough estimate of cable length to be made. 0000：DSL connection cable delay: 100 to 200 ns. Cable length DSL connection 10 to 20 m. 0001：DSL connection cable delay: 100 to 200 ns. Cable length DSL connection 10 to 20 m. 0010：DSL connection cable delay: 200 to 300 ns. Cable length DSL connection 20 to 30 m. 0011：DSL connection cable delay: 300 to 400 ns. Cable length DSL connection 30 to 40 m. 0100：DSL connection cable delay: 400 to 500 ns. Cable length DSL connection 40 to 50 m. 0101：DSL connection cable delay: 500 to 600 ns. Cable length DSL connection 50 to 60 m. 0110：DSL connection cable delay: 600 to 700 ns. Cable length DSL connection 60 to 70 m. 0111：DSL connection cable delay: 700 to 800 ns. Cable length DSL connection 70 to 80 m.

1000：DSL connection cable delay: 800 to 900 ns. Cable length DSL connection 80 to 90 m.
1001：DSL connection cable delay: 900 to 1000 ns. Cable length DSL connection 90 to 100 m.

HDSL version register (VERSION)

Address offset: 0x0B
Reset value: 0x57

The VERSION register contains the release version of the DSL master module.

The register is write-protected.

This register has to be accessed by byte (8-bit).

7	6	5	4	3	2	1	0
CODE[1:0]		MAJOR[1:0]		MINOR[3:0]			
r		r		r			

Bits	Fields	Descriptions
7:6	CODE[1:0]	Type of module 01: DSL master Others: Reserved
5:4	MAJOR[1:0]	Major release number The current version is 1.
3:0	MINOR[3:0]	Minor release number The current version is 7.

HDSL encoder ID 2 register (ENC_ID2)

Address offset: 0x0D
Reset value: 0x00

The ENC_ID2/1/0 registers contain the designation code of the motor feedback system connected to the DSL master. In the current protocol specification, the designation code is 20 bits long. With later enhancements, the free bits in the encoder ID registers are used to indicate special characters.

These registers are write-protected.

This register has to be accessed by byte (8-bit).

7	6	5	4	3	2	1	0
Reserved	SCI[2:0]			ENCID[3:0]			
	r			r			

Bits	Fields	Descriptions
7	Reserved	Must be kept at reset value
6:4	SCI[2:0]	Indication of special characters 3-bit special character for later enhancements of the encoder designation code. Not allocated.
3:0	ENCID[3:0]	Encoder designation code bit 19 to bit 16

HDSL encoder ID 1 register (ENC_ID1)

Address offset: 0x0E
Reset value: 0x00

The ENC_ID2/1/0 registers contain the designation code of the motor feedback system connected to the DSL master. In the current protocol specification, the designation code is 20 bits long. With later enhancements, the free bits in the encoder ID registers are used to indicate special characters.

These registers are write-protected.

This register has to be accessed by byte (8-bit).

7	6	5	4	3	2	1	0
ENCID[7:0]							
r							

Bits	Fields	Descriptions
7:0	ENCID[7:0]	Encoder designation code bit 15 to bit 8

HDSL encoder ID 0 register (ENC_ID0)

Address offset: 0x0F
Reset value: 0x00

The ENC_ID2/1/0 registers contain the designation code of the motor feedback system connected to the DSL master. In the current protocol specification, the designation code is 20 bits long. With later enhancements, the free bits in the encoder ID registers are used to indicate special characters.

These registers are write-protected.

This register has to be accessed by byte (8-bit).

7	6	5	4	3	2	1	0
ENCID[7:0]							
r							

Bits	Fields	Descriptions

7:0	ENCID[7:0]	Encoder designation code bit 7 to bit 0

HDSL fast position byte n register (POSn) (n = 0…4)

Address offset: 0x10 + (0x04 - n)
Reset value: 0x00

The POSn registers for the fast position contain the value of the motor feedback system connected. This position is generated incrementally from the safe position at start-up and is updated with every protocol frame.

After every eight protocol frames, the fast position is checked against the safe position (VPOSn register).

The position sampling point is determined by the ES bits in SYNC_CTRL register.

Only those POS bits are activated that lie within the range that the motor feedback system has actually measured. All other higher value bits are read as "0". The number of measurable bits can be taken from ENCID bits 9 to 0 in the ENC_ID0 to 2 registers.

If sign is set in the ENC_Idn register, the value of the fast position is given signed in the two's complement.

The units of the position value are (steps). These registers are write-protected.

This register has to be accessed by byte (8-bit).

7	6	5	4	3	2	1	0
POS[7:0]							
r							

Bits	Fields	Descriptions
7:0	POS[7:0]	Fast position, byte n Position value of the motor feedback system (length: 40 bits), incrementally generated.

HDSL speed byte n register (VELn) (n = 0…2)

Address offset: 0x15 + (0x02 - n)
Reset value: 0x00

The VELn registers contain the speed values of the connected motor feedback system. This value is calculated as a Δ position from the acceleration value (Δ δ position) transmitted on the process data channel and the currently updated protocol frame.

The speed sampling point is determined by the ES bits of the SYNC_CTRL register. The units of the speed value are steps/ frame cycle time.

These registers are write-protected.

This register has to be accessed by byte (8-bit).

7	6	5	4	3	2	1	0
VEL[7:0]							
r							

Bits	Fields	Descriptions
7:0	VEL[7:0]	Speed, byte n Speed of the motor feedback system (length: 24 bits)

HDSL mirror status summary register (MIR_SUM)

Address offset: 0x18
Reset value: 0x00

The MIR_SUM register contains the same information for the drive application. The MIR_SUM register is based on the encoder mirror registers MIR_ST7 to MIR_ST0 in the DSL Slave and therefore delayed when compared to the SAFE_SUM register. The bits in the MIR_SUM register can be read by the drive application only.

This register has to be accessed by byte (8-bit).

7	6	5	4	3	2	1	0
SUM[7:0]							
r							

Bits	Fields	Descriptions
7:0	SUM[7:0]	Status summary bit bit [7:1] (external resource): 0: The corresponding error, warning, or event is not active. 1: An error, a warning, or an event associated with the DSL Slave external resources was triggered. Bit [0] (interface): 0: The DSL Slave protocol has not triggered an error, warning, or event. 1: An error, a warning, or an event associated with the DSL Slave protocol interface was triggered.

HDSL parameters channel buffer byte n register (PC_BUFFERn) (n = 0…7)

Address offset: 0x20 + n
Reset value: 0x00

The eight PC_BUFFERn registers of the Parameters Channel buffer contain the answer to the last "long message" request or the data for a "long message" write operation.

These registers are also for the reporting of error conditions arising from a "long message" operation. If, when accessing a resource, an error due to a "long message" arises (e.g. invalid

data, error in the A/D conversion), after the answering message has been received the LOFF bit in the PC_ADD_H register is set. In this case the Parameters Channel buffer bytes 0 and 1 contain an error code.

The meaning of the error code depends on the particular HIPERFACE DSL® encoder and is described in detail in the data sheet.

This register has to be accessed by byte (8-bit).

7	6	5	4	3	2	1	0
PCBUF[7:0]							
rw							

Bits	Fields	Descriptions
7:0	PCBUF[7:0]	Parameters Channel buffer, byte 0-7: 8 bytes for the answer to a long message (read operation) or for a "long message" write operation. Long message error code, byte 0-1: 2 bytes for reports about errors in encoder resources arising from the previous "long message" operation.

HDSL high byte long message address register (PC_ADD_H)

Address offset: 0x28
Reset value: 0x00

The addresses and the addressing mode for "long messages" sent over the Parameters Channel are determined in the PC_ADD_H/PC_ADD_L long message address registers.

In addition, the PC_ADD_H contains indications of errors arising from "long message" operations. For this sort of error, the Parameters Channel buffer contains the error code in bytes 0 and 1 associated with this status.

This register has to be accessed by byte (8-bit).

7	6	5	4	3	2	1	0
Reserved	LRW	LOFF	LIND	LLEN[1:0]		LADDH[1:0]	
	w	rw	w	w		w	

Bits	Fields	Descriptions
7	Reserved	Must be kept at reset value
6	LRW	Long message, read/write mode 0: "long message" write operation 1: "long message" read operation
5	LOFF	Long message addressing mode/long message error 0: **Write access**: Addressing of "long message" without offset. The offset value from the PCOFFn registers is not used.

		Read access: The last "long message" was correctly processed. 1: **Write access:** Offset addressing of "long message". The offset value from the PCOFFn registers is used in the resource of the selected database entry as a subaddress. **Read access:** The last "long message" caused an error.
4	LIND	Indirect addressing of long messages 0: Direct addressing of "long messages". The operation affects the database entry given in the current address. 1: Indirect addressing of "long messages". During this operation, the stored address content in the given database entry is evaluated.
3:2	LLEN[1:0]	Data length of the "long message" 00: No data bytes 01: 2 data bytes 10: 4 data bytes 11: 8 data bytes
1:0	LADDH[1:0]	Long message address, bit 9 to bit 8 Database entry with 10-bit address for a "long message" operation

HDSL low byte long message address register (PC_ADD_L)

Address offset: 0x29
Reset value: 0x00

The addresses and the addressing mode for "long messages" sent over the Parameters Channel are determined in the PC_ADD_H/PC_ADD_L long message address registers.

This register has to be accessed by byte (8-bit).

7	6	5	4	3	2	1	0
LADDL[7:0]							
w							

Bits	Fields	Descriptions
7:0	LADDL[7:0]	Long message address, bit 7 to bit 0 Database entry with 10-bit address for a "long message" operation

HDSL high byte long message address offset register (PC_OFF_H)

Address offset: 0x2A
Reset value: 0x00

The PC_OFF_H/PC_OFF_L address offset registers for long messages are used in "long message" operations, if LOFF is set in the PC_ADD_H register. In this case the LOFFADD value from these registers is used to communicate with the sub-address of a multiple byte

encoder resource.

Only write access is possible for these registers.

This register has to be accessed by byte (8-bit).

7	6	5	4	3	2	1	0
LID	LOFFADDH[6:0]						
r	w						

Bits	Fields	Descriptions
7	LID	Long message identification. The value must be 1.
6:0	LOFFADDH[6:0]	Long message offset value, bit 14 to bit 8 The 15 bit offset value of the "long message" address offset is stored in these bits.

HDSL low byte long message address offset register (PC_OFF_L)

Address offset: 0x2B
Reset value: 0x00

The PC_OFF_H/PC_OFF_L address offset registers for long messages are used in "long message" operations, if LOFF is set in the PC_ADD_H register. In this case the LOFFADD value from these registers is used to communicate with the sub-address of a multiple byte encoder resource.

Only write access is possible for these registers.

This register has to be accessed by byte (8-bit).

7	6	5	4	3	2	1	0
LOFFADDL[7:0]							
w							

Bits	Fields	Descriptions
7:0	LOFFADDH[6:0]	Long message offset value, bit 7 to bit 0 The 15 bit offset value of the "long message" address offset is stored in these bits.

HDSL parameters channel control register (PC_CTRL)

Address offset: 0x2C
Reset value: 0x00

The PC_CTRL control register for the Parameters Channel handles the start of "long message" transactions. After setting all "long message" registers (registers PC_BUFFER0 to 7, PC_ADD_H/PC_ADD_L and PC_OFF_H/L), the "long message" is transmitted to the DSL Slave by setting the LSTA bit.

This register has to be accessed by byte (8-bit).

7	6	5	4	3	2	1	0
Reserved							LSTA
							w

Bits	Fields	Descriptions
7:1	Reserved	Must be kept at reset value
0	LSTA	Control of the long message start 0: No effect 1: A "long message" transaction is started with the values currently stored in the "long message" registers.

HDSL sensorhub channel status register (PIPE_S)

Address offset: 0x2D
Reset value: 0x04

The PIPE_S register provides information about the current status of the SensorHub Channel. When this register is read, the current data from the FIFO buffer is read and stored in an intermediate register so that a subsequent read process in the PIPE_D register can be considered to be completed at the same time as the PIPE_S register is read. Using this mechanism, any deviation between status and data information in this instance will prevent new data entering the SensorHub Channel during access to the FIFO buffer.

This register has to be accessed by byte (8-bit).

7	6	5	4	3	2	1	0
Reserved				POVR	PEMP	PERR	PSCI
				r	r	r	r

Bits	Fields	Descriptions
7:4	Reserved	Must be kept at reset value
3	POVR	SensorHub Channel overflow This bit is deleted after the read process. 0: The capacity of the FIFO buffer for SensorHub Channel data is not yet exhausted. 1: The capacity of the 8 byte FIFO buffer for SensorHub Channel data was exhausted and since the last read process, values have been discarded.
2	PEMP	The SensorHub channel buffer is empty This bit is updated after every access to the FIFO buffer. 0: No "buffer empty" error. 1: A read request was issued, but the FIFO buffer for SensorHub Channel data is empty. In this case, PIPE_D contains the value 0x00.

1	PERR	Coding error of the bits in the SensorHub Channel This bit is stored together with the pipeline data byte in question in the FIFO buffer. 0: No error in bit coding. 1: The bit level coding of the data currently in the SensorHub Channel is erroneous.
0	PSCI	Indication for special characters in the SensorHub Channel This bit is stored together with the pipeline data byte in question in the FIFO buffer. 0: Indication for "no special character". 1: A special character was received in the SensorHub Channel.

HDSL sensorhub channel data register (PIPE_D)

Address offset: 0x2E
Reset value: 0x00

The PIPE_D register contains the SensorHub Channel data that is stored in an 8 byte FIFO buffer. If new data arrives at the buffer when it is full, before PIPE_D is read, the oldest value is discarded and the POVR bit in PIPE_S is set. If a read request is issued when the buffer is empty, the PEMP bit in PIPE_S is set and the value 0x00 is transmitted. At the moment that the PIPE_S register is accessed, the corresponding PIPE_D value is frozen to guarantee synchronization between status and data information.

This register has to be accessed by byte (8-bit).

7	6	5	4	3	2	1	0
SCDATA[7:0]							
r							

Bits	Fields	Descriptions
7:0	SCDATA[7:0]	SensorHub Channel data 8 bit value of the FIFO buffer for SensorHub Channel data.

HDSL parameters channel short message mirror register (PC_DATA)

Address offset: 0x2F
Reset value: 0x00

When reading the mirror registers MIR_ST7 to MIR_ST0 instead of the real remote registers, the answer is available immediately in the PC_DATA register.

This register has to be accessed by byte (8-bit).

7	6	5	4	3	2	1	0
PCDATA[7:0]							
r							

Bits	Fields	Descriptions
7:0	PCDATA[7:0]	"Short message" Mirror register data 8-bit value of the requested mirror register

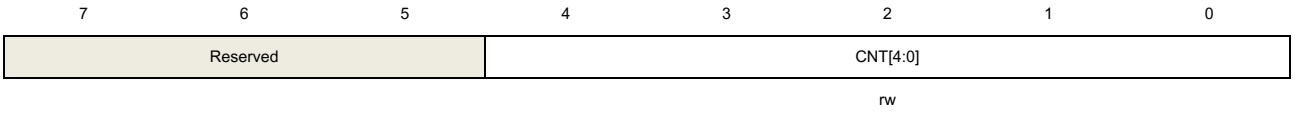
HDSL fast position error counter register (ACC_ERR_CNT)

Address offset: 0x38
Reset value: 0x00

The ACC_ERR_CNT register returns the count of transmitted fast position values with consecutive transmission errors. The value is clamped to a maximum of 31 (0x1F).

With a writing access the error threshold can be set. This value is also clamped to a maximum of 31 (0x1F). If the count of transmitted fast position values with consecutive transmission errors exceeds this threshold, the error threshold will be set to 1.

This register has to be accessed by byte (8-bit).



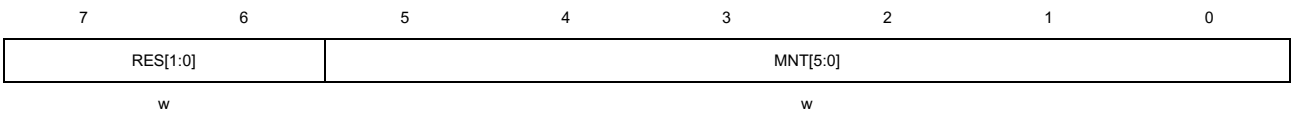
Bits	Fields	Descriptions
7:5	Reserved	Must be kept at reset value
4:0	CNT[4:0]	Position error count/threshold **Read:** 5-bit value of count of transmitted fast position values with consecutive transmission errors **Write:** 5-bit value for threshold

HDSL fast position acceleration boundary register (MAXACC)

Address offset: 0x39
Reset value: 0x00

The MAXACC register allows setting an acceleration threshold for a given application. This threshold is used by the fast position estimator to clamp the acceleration of the estimated position during communication or sensor failures of the fast position channel.

This register has to be accessed by byte (8-bit).



Bits	Fields	Descriptions
7:6	RES[1:0]	Resolution of fast position acceleration boundary

5:0	MNT[5:0]	Mantissa of fast position acceleration boundary

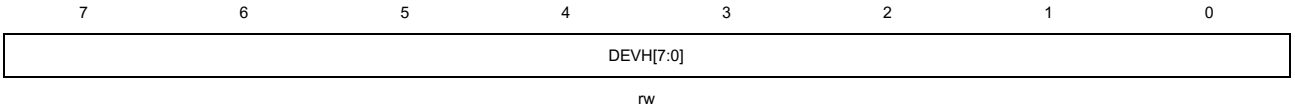
HDSL fast position estimator deviation high byte register (MAXDEV_H)

Address offset: 0x3A
Reset value: 0xFF

The MAXDEV_H/MAXDEV_L registers return the maximum absolute position deviation while the position estimator is active. The returned 16-bit value has the same format (resolution) as the fast position channel and is clamped to a maximum of 65535 steps (0xFFFF). The registers are set to the maximum value 0xFFFF at reset.

These registers also allow setting a deviation threshold value for triggering the error event. The threshold value can be written with the same format as the deviation (unsigned 16-bit, same resolution as the fast position channel).

This register has to be accessed by byte (8-bit).



Bits	Fields	Descriptions
7:0	DEVH[7:0]	Fast position estimator deviation high byte **Read:** 16-bit value of position deviation **Write:** 16-bit value for deviation threshold

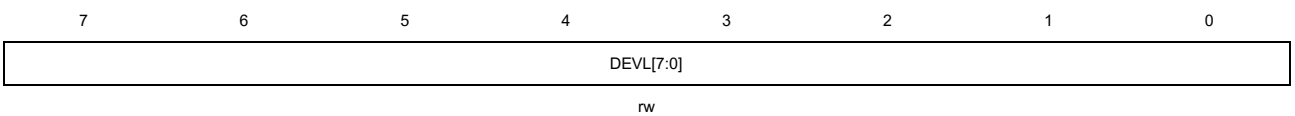
HDSL fast position estimator deviation low byte register (MAXDEV_L)

Address offset: 0x3B
Reset value: 0xFF

The MAXDEV_H/MAXDEV_L registers return the maximum absolute position deviation while the position estimator is active. The returned 16-bit value has the same format (resolution) as the fast position channel and is clamped to a maximum of 65535 steps (0xFFFF). The registers are set to the maximum value 0xFFFF at reset.

These registers also allow setting a deviation threshold value for triggering the error event. The threshold value can be written with the same format as the deviation (unsigned 16-bit, same resolution as the fast position channel).

This register has to be accessed by byte (8-bit).



Bits	Fields	Descriptions

7:0	DEVL[7:0]	Fast position estimator deviation low byte **Read:** 16-bit value of position deviation **Write:** 16-bit value for deviation threshold

HDSL dummy register (DUMMY)

Address offset: 0x3F
Reset value: 0xXX

The dummy read by the DUMMY register in Safe 1 interface is required. The dummy read by the DUMMY register in Drive interface is not required.

This register has to be accessed by byte (8-bit).

7	6	5	4	3	2	1	0
DUMMY[7:0]							
r							

Bits	Fields	Descriptions
7:0	DUMMY[7:0]	Due to the transmission of the online-status, Online Status read transactions need less time for transmission. Therefore, dummy read transactions must be inserted to avoid unwanted extra transactions. A read access to this register 0x3F has no effect and must be used for this purpose.

HDSL mirror status n register (MIR_STn) (n = 0…7)

Address offset: 0x60 + n
Reset value: 0x00

The MIR_ST encoder status registers contain all slave system errors, events, and warnings from the DSL encoder. This way, the information can be accessed via drive interface without blocking the short message channel.

The allocation between the individual bits and the slave system statuses depends on the DSL Slave specification. MIR_STn registers (n = 0 to 7) are updated only when a read Short Message is initiated on the corresponding remote status register ENC_STn and the response is read from S_PC_DATA. Trigger for updating the MIR_STn, is reading S_PC_DATA.

Background: initiating a Short Message saves the index of the register, but reading S_PC_DATA triggers the actual copy of register content to the corresponding MIR_STn.

This register has to be accessed by byte (8-bit).

7	6	5	4	3	2	1	0
MIRST[7:0]							
rc_w0							

Bits	Fields	Descriptions
7:0	MIRST[7:0]	Mirror status, byte n The individual bits indicate different errors, events and warnings. The meaning of each individual bit depends on the particular DSL Slave specification. 0: Encoder in normal status 1: Error, event, or warning status

HDSL safe channel 1 interface registers

HDSL safe channel 1 interface base address: 0x4000 DA00

HDSL safe channel 1 edge register (SAFE_EDGES)

Address offset: 0x09
Reset value: 0x00

The EDGES register contains the time control for the DSL cable bit sampling and can be used to monitor the connection quality.

Each individual EDGES register bit is set if, at system start-up, an edge of the test signal is detected during the time period of the corresponding bit. An edge is defined as a change in cable value between successive detections. The sampling is carried out eight times as fast as the cable bit rate.

The register is write-protected. The contents of this register do not change after the start-up phase. A new bit sampling pattern is only generated after a forced reset of the protocol.

This register has to be accessed by byte (8-bit).

7	6	5	4	3	2	1	0
EDGES[7:0]							
r							

Bits	Fields	Descriptions
7:0	EDGES[7:0]	Identification of edges in the cable signal 0: No edge was detected in the time period of the corresponding bit. 1: An edge was detected in the time period of the corresponding bit.

HDSL safe channel 1 run time delay register (SAFE_DELAY)

Address offset: 0x0A
Reset value: 0x00

The DELAY register stores information about the run-time delay of the system cable and the signal strength. The register can be used to monitor the connection quality.

The register is write-protected.

The value for Line Delay does not change after the start-up phase. A fresh value for Line Delay is only measured after a forced reset of the protocol.

This register has to be accessed by byte (8-bit).

7	6	5	4	3	2	1	0
RSSI[3:0]				CBLDLY[3:0]			
r				r			

Bits	Fields	Descriptions
7:4	RSSI[3:0]	Indication of the received signal strength 4-bit value for the cable signal strength, from "0" to "12". Higher values indicate better connection quality. Low connection (< 2) quality affects QM. RSSI is continuously updated during operation and used for signal monitoring during run time.
3:0	CBLDLY	4-bit value for cable delay This value gives the cable signal round trip delay of cable and transceivers in bits. This value enables a rough estimate of cable length to be made. 0000：DSL connection cable delay: 100 to 200 ns. Cable length DSL connection 10 to 20 m. 0001：DSL connection cable delay: 100 to 200 ns. Cable length DSL connection 10 to 20 m. 0010：DSL connection cable delay: 200 to 300 ns. Cable length DSL connection 20 to 30 m. 0011：DSL connection cable delay: 300 to 400 ns. Cable length DSL connection 30 to 40 m. 0100：DSL connection cable delay: 400 to 500 ns. Cable length DSL connection 40 to 50 m. 0101：DSL connection cable delay: 500 to 600 ns. Cable length DSL connection 50 to 60 m. 0110：DSL connection cable delay: 600 to 700 ns. Cable length DSL connection 60 to 70 m. 0111：DSL connection cable delay: 700 to 800 ns. Cable length DSL connection 70 to 80 m. 1000：DSL connection cable delay: 800 to 900 ns. Cable length DSL connection 80 to 90 m. 1001：DSL connection cable delay: 900 to 1000 ns. Cable length DSL connection 90 to 100 m.

HDSL safe channel 1 version register (SAFE_VERSION)

Address offset: 0x0B
Reset value: 0x57

The VERSION register contains the release version of the DSL master module.

The register is write-protected.

This register has to be accessed by byte (8-bit).

7	6	5	4	3	2	1	0
CODE[1:0]		MAJOR[1:0]		MINOR[3:0]			
r		r		r			

Bits	Fields	Descriptions
7:6	CODE[1:0]	Type of module 01: DSL master Others: Reserved
5:4	MAJOR[1:0]	Major release number The current version is 1.
3:0	MINOR[3:0]	Minor release number The current version is 7.

HDSL safe channel 1 encoder ID 2 register (SAFE_ENC_ID2)

Address offset: 0x0D
Reset value: 0x00

The ENC_ID2/1/0 registers contain the designation code of the motor feedback system connected to the DSL master. In the current protocol specification, the designation code is 20 bits long. With later enhancements, the free bits in the encoder ID registers are used to indicate special characters.

These registers are write-protected.

This register has to be accessed by byte (8-bit).

7	6	5	4	3	2	1	0
Reserved	SCI[2:0]			ENCID[3:0]			
	r			r			

Bits	Fields	Descriptions
7	Reserved	Must be kept at reset value
6:4	SCI[2:0]	Indication of special characters 3-bit special character for later enhancements of the encoder designation code. Not allocated.
3:0	ENCID[3:0]	Encoder designation code bit 19 to bit 16

HDSL safe channel 1 encoder ID 1 register (SAFE_ENC_ID1)

Address offset: 0x0E

Reset value: 0x00

The ENC_ID2/1/0 registers contain the designation code of the motor feedback system connected to the DSL master. In the current protocol specification, the designation code is 20 bits long. With later enhancements, the free bits in the encoder ID registers are used to indicate special characters.

These registers are write-protected.

This register has to be accessed by byte (8-bit).

7	6	5	4	3	2	1	0
ENCID[7:0]							
r							

Bits	Fields	Descriptions
7:0	ENCID[7:0]	Encoder designation code bit 15 to bit 8

HDSL safe channel 1 encoder ID 0 register (SAFE_ENC_ID0)

Address offset: 0x0F
Reset value: 0x00

The ENC_ID2/1/0 registers contain the designation code of the motor feedback system connected to the DSL master. In the current protocol specification, the designation code is 20 bits long. With later enhancements, the free bits in the encoder ID registers are used to indicate special characters.

These registers are write-protected.

This register has to be accessed by byte (8-bit).

7	6	5	4	3	2	1	0
ENCID[7:0]							
r							

Bits	Fields	Descriptions
7:0	ENCID[7:0]	Encoder designation code bit 7 to bit 0

HDSL safe position channel 1 byte n register (VPOSn) (n = 0…4)

Address offset: 0x19 + (0x04 - n)
Reset value: 0x00

The VPOSn registers for the safe position contain the position value from the primary channel of the motor feedback system connected. This safe position is transmitted in every eighth protocol frame.

The safe position is not synchronized with the sync signal.

Only those vpos bits are activated that lie within the range that the motor feedback system has actually measured. All other higher value bits are read as 0. The number of measurable bits can be taken from ENCID bits 9 to 0 in the ENC_ID0 to ENC_ID2 registers.

If sign is set in the ENC_ID register, the value of the safe position is given signed in the two's complement.

The units of the position value are [steps]. These registers are write-protected.

This register has to be accessed by byte (8-bit).

7	6	5	4	3	2	1	0
VPOS[7:0]							
r							

Bits	Fields	Descriptions
7:0	VPOS[7:0]	Safe position, byte n Position value transmitted through Safe Channel 1 (length: 40 bits), absolute value.

HDSL high byte position checksum channel 1 register (VPOSCRC_H)

Address offset: 0x1E
Reset value: 0x00

The CRC is checked in the DSL master. In order to guarantee, in a safety-related application, that the CRC machine in the module is functioning, these registers can be checked with an external cross-check in the diagnostics test interval.

This register has to be accessed by byte (8-bit).

7	6	5	4	3	2	1	0
VPOSCRCH[7:0]							
r							

Bits	Fields	Descriptions
7:0	VPOSCRCH[7:0]	CRC of the safe position, byte 1 16 bit CRC checksum (CRC 16) of the safe position and status summary in Safe Channel 1.

HDSL low byte position checksum channel 1 register (VPOSCRC_L)

Address offset: 0x1F
Reset value: 0x00

The CRC is checked in the DSL master. In order to guarantee, in a safety-related application, that the CRC machine in the module is functioning, these registers can be checked with an external cross-check in the diagnostics test interval.

This register has to be accessed by byte (8-bit).

7	6	5	4	3	2	1	0
VPOSCRCL[7:0]							
r							

Bits	Fields	Descriptions
7:0	VPOSCRCL[7:0]	CRC of the safe position, byte 0 16 bit CRC checksum (CRC 16) of the safe position and status summary in Safe Channel 1.

HDSL safe system control register (SAFE_CTRL)

Address offset: 0x35
Reset value: 0x80

The SAFE_CTRL register contains two control bits of the DSL master available for the safety-related application.

This register has to be accessed by byte (8-bit).

7	6	5	4	3	2	1	0
PRST	MRST	Reserved					
rw	rw						

Bits	Fields	Descriptions
7	PRST	Protocol reset 0: Normal protocol action 1: A forced reset of the protocol status will be initiated. If the bit is deleted, a restart of the connection is triggered.
6	MRST	Message reset 0: Normal Parameters Channel action 1: The Parameters Channel is reset. Current short and long messages are discarded.
5:0	Reserved	Must be kept at reset value

HDSL safe status summary register (SAFE_SUM)

Address offset: 0x36
Reset value: 0x00

The SAFE_SUM register contains the summarized DSL Slave status information for the safety-related application. It is based on the encoder status ENC_STn (n = 0 to 7). Each status summary bit contains the summarized information from 8 error, warning, and event modes of the DSL Slave. The bits in the status summary register can be read by the safety-related application only.

This register has to be accessed by byte (8-bit).

7	6	5	4	3	2	1	0
SSUM[7:0]							
r							

Bits	Fields	Descriptions
7:0	SSUM[7:0]	Status summary bit bit [7:1] (external resource): 0: The corresponding error, warning, or event is not active. 1: An error, a warning, or an event associated with DSL Slave external resources was triggered. Bit [0] (interface): 0: The DSL Slave protocol has not triggered an error, a warning, or an event. 1: An error, a warning, or an event associated with the DSL Slave protocol interface was triggered.

HDSL parameters channel short message register (S_PC_DATA)

Address offset: 0x37
Reset value: 0x00

The S_PC_DATA register for the Parameters Channel short message contains the results of "short message" transactions. "Short message" transactions are generated if operations are carried out with remote registers (DSL Slave). Generally, FRES (in the EVENT_S register) must be set after a transaction is started. Only then will SPCDATA contain valid information.

This register has to be accessed by byte (8-bit).

7	6	5	4	3	2	1	0
SPCDATA[7:0]							
r							

Bits	Fields	Descriptions
7:0	SPCDATA[7:0]	"Short message" parameter channel data 8-bit value of the requested remote register

HDSL safe event register (EVENT_S)

Address offset: 0x3D
Reset value: 0x00

The EVENT_S register contains the message Quality monitoring bits for all warning and error modes of the DSL system.

All messaging bits are set by the DSL master if a corresponding status is determined.

The following bit description lists the effects of warning and error conditions as well as the

reactions to errors that must be installed in the safety-related application.

An event bit that has been set is not reset by the DSL master. The safety-related application must delete bits that have been set.

Both edge and level-sensitive flags are present in the EVENT_S register. Edge-sensitive bits are set when the corresponding status arises. They are only set again if the corresponding status disappears and then arises once more. This is the standard action. The level-sensitive bits set a bit as long as the corresponding status exists.

This register has to be accessed by byte (8-bit).

7	6	5	4	3	2	1	0
SINT	SSUM	SCE	VPOS	QMLW	PRST	MIN	FRES
r	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

Bits	Fields	Descriptions
7	SINT	Safe interrupt status This bit reflects the status of the safety-related interrupt.
6	SSUM	Remote event monitoring When the bit is set, an error or a warning has been transmitted from the DSL Slave. The safety-related application must check the SAFE_SUM register to obtain a detailed description. This bit is level sensitive. 0: All DSL Slave events are cleared. 1: The DSL Slave has signaled an event.
5	SCE	Error on the Safe Channel This error usually indicates a transmission error on the DSL connection. If this error occurs frequently, the wiring of the DSL connection should be checked. If this error occurs continuously, there is probably an error in the motor feedback system. This error affects quality monitoring and produces the QMLW warning or a protocol reset. 0: Safe Channel data was correctly transmitted. 1: Data consistency error on the Safe Channel.
4	VPOS	Safe position error This error usually indicates an encoder sensor error. If this error occurs continuously, there is probably an error in the motor feedback system. 0: The safe position is correct. 1: Sensor error
3	QMLW	Quality monitoring low-value warning This warning indicates that a transmission error occurred. If this error occurs frequently, the wiring of the DSL connection should be checked. 0: Quality monitoring value, QM[3:0] in the MASTERQM register, greater than or equal to "14"

		1: Quality monitoring value below "14"
2	PRST	Protocol reset warning This error message indicates that the protocol connection to the DSL Slave has been reinitialized. This error message can be caused by a frequency inverter application request (PRST bit in SYS_CTRL register), a safety-related application request (PRST bit in SAFE_CTRL register), generated by the DSL master itself, or activated via the reset input. The DSL master causes a protocol reset if too many transmission errors indicate a connection problem. A protocol reset causes a re-synchronization with the DSL Slave that can improve the connection quality. 0: Normal protocol action 1: The forced protocol reset was triggered.
1	MIN	Message init When this warning is displayed, the Parameters Channel is still in the initialization status and no "short message" or "long message" can be triggered. 0: No acknowledgment for the initialization received. 1: An acknowledgment was received from the Slave for the initialization of a message.
0	FRES	Channel free for "short message" If the bit is set, the frequency inverter application can trigger a "short message". Provided no answer has been received from the DSL Slave, this bit remains 0. As the processing duration of a "long message" in the motor feedback system is not specified, a time limit condition is installed in the DSL master. If the time limit is exceeded, attempts are made again automatically. 0: No "short message" can be sent. 1: A "short message" can be sent on the Parameters Channel

HDSL safe event mask register (MASK_S)

Address offset: 0x3E
Reset value: 0x00

In the MASK_S register, events are chosen that can trigger the safety-related interrupt signal.

Several events can be masked to trigger an interrupt. In addition, events from the DSL Slave summary can be combined with these events.

This register has to be accessed by byte (8-bit).

7	6	5	4	3	2	1	0
Reserved	MSSUM	MSCE	MVPOS	MQMLW	MPRST	MMIN	MFRES
	w	w	w	w	w	w	w

Bits	Fields	Descriptions

7	Reserved	Must be kept at reset value
6	MSSUM	Mask for remote event monitoring 0: DSL Slave events that are set in the SAFE_SUM register do not set the safety-related interrupt signal. 1: DSL Slave events that are set in the SAFE_SUM register sets the safety-related interrupt signal.
5	MSCE	Mask for transmission errors on the Safe Channel 0: A transmission error on the Safe Channel does not set the safety-related interrupt signal. 1: A transmission error on the Safe Channel sets the safety-related interrupt signal.
4	MVPOS	Mask for safe position error 0: An error in the safe position does not set the safety-related interrupt signal. 1: An error in the safe position sets the safety-related interrupt signal.
3	MQMLW	Mask for low-quality monitoring value warning 0: A low-quality monitoring value does not set the safety-related interrupt signal. 1: A low-quality monitoring value sets the safety-related interrupt signal.
2	MPRST	Mask for protocol reset warning 0: A protocol reset does not set the safety-related interrupt signal. 1: A protocol reset sets the safety-related interrupt signal.
1	MMIN	Mask for message initialization confirmation 0: The acknowledgment for the initialization of a DSL Slave message does not set the safety-related interrupt signal. 1: The acknowledgment for the initialization of a DSL Slave message sets the safety related interrupt signal.
0	MFRES	Mask for "channel free for "short message"" 0: If a "short message" can be sent on the Parameters Channel, the safety-related interrupt signal is not set. 1: If a "short message" can be sent on the Parameters Channel, the safety-related interrupt signal is set.

HDSL encoder status n register (ENC_STn) (n = 0…7)

Address offset: 0x40 + n
Reset value: 0x00

The ENC_STn encoder status registers contain all slave system errors, events, and warnings from the DSL encoder.

The allocation between the individual bits and the slave system statuses depends on DSL Slave specification.

This register has to be accessed by byte (8-bit).

7	6	5	4	3	2	1	0
ENCST[7:0]							
rc_w0							

Bits	Fields	Descriptions
7:0	ENCST[7:0]	Encoder status, byte n The individual bits indicate different errors, events, and warnings. The meaning of each individual bit is determined by the particular DSL Slave installation. 0: Encoder in normal status 1: Error, event, or warning status

HDSL slave RSSI register (SRSSI)

Address offset: 0x7C
Reset value: 0x00

The SRSSI register for indicating the received signal strength at the slave (Slave Received Signal Strength Indicator, RSSI) provides an indication of the strength of the signal arriving at the slave.

The value of the register is only updated from frame to frame if the measurement result deteriorates. After a read access to this register, the register is reset to the value "7" (maximum signal strength).

The register is write-protected.

This register has to be accessed by byte (8-bit).

7	6	5	4	3	2	1	0
Reserved					SRSSI[2:0]		
					r		

Bits	Fields	Descriptions
7:3	Reserved	Must be kept at reset value
2:0	SRSSI[2:0]	Value of the Slave RSSI The values for the Slave RSSI range from "0" (poorest signal strength) to "7" (best signal strength).

HDSL slave mail register (MAIL)

Address offset: 0x7E
Reset value: 0x00

The MAIL multi-purpose register of the slave is used for fast communication with the DSL motor feedback system processor. The content of the slave mail register is transmitted to the

encoder processor by the most rapid route possible.

The slave mail register is used for the transmission of test messages to the DSL motor feedback system.

This register has to be accessed by byte (8-bit).

7	6	5	4	3	2	1	0
MAIL[7:0]							
w							

Bits	Fields	Descriptions
7:0	MAIL[7:0]	Slave-Mail 8-bit slave mail data for multiple applications

HDSL slave ping register (PING)

Address offset: 0x7F
Reset value: 0x00

The PING register of the slave is used to carry out connection tests on behalf of the DSL Slave. The register can be written to and read externally, without this affecting the DSL interface.

On start-up, the register is initialized with the DSL Slave interface.

This register has to be accessed by byte (8-bit).

7	6	5	4	3	2	1	0
PING[7:0]							
rw							

Bits	Fields	Descriptions
7:0	PING[7:0]	Slave-Ping 8-bit hardware version of the DSL encoder at start-up. On first reading, the Ping value of the slave can be used as a connection test on behalf of the slave.

HDSL safe channel 2 interface registers

HDSL safe channel 2 interface base address: 0x4000 DB00

HDSL version in safe channel 2 register (VERSION2)

Address offset: 0x0B
Reset value: 0x57

The VERSION2 register in safe channel 2 is identical to the VERSION register in the drive and safe channel 1 interfaces.

This register has to be accessed by byte (8-bit).

7	6	5	4	3	2	1	0
CODE[1:0]		MAJOR[1:0]		MINOR[3:0]			
r		r		r			

Bits	Fields	Descriptions
7:6	CODE[1:0]	Type of module 01: DSL master Others: Reserved
5:4	MAJOR[1:0]	Major release number The current version is 1.
3:0	MINOR[3:0]	Minor release number The current version is 7.

HDSL encoder ID in safe channel 2 register (ENC2_ID)

Address offset: 0x0F
Reset value: 0x00

The ENC2_ID register for the ID of the encoder in safe channel 2 contains the designation code for the safe channel 2 channel of the DSL encoder.

The contents of ENC2_ID are defined during the actual installation of the motor feedback system and can be found in the corresponding data sheet.

These registers are write-protected.

This register has to be accessed by byte (8-bit).

7	6	5	4	3	2	1	0
ENC2ID[7:0]							
r							

Bits	Fields	Descriptions
7:0	ENC2ID[7:0]	Designation of the safe channel 2 channel 0x02/0x22: SIL2 arrangement. The same position data is transmitted to the safe channel 2 channel as to the safe channel 1 channel. 0x03/0x33: SIL3 arrangement. Position data is transmitted to the safe channel 2 channel that originates from the secondary sensor in the HIPERFACE DSL® motor feedback system.

HDSL safe channel 2 status register (STATUS2)

Address offset: 0x18
Reset value: 0x00

The STATUS2 status register contains the status information for safe channel 2 of the HIPERFACE DSL® motor feedback system.

This register has to be accessed by byte (8-bit).

7	6	5	4	3	2	1	0
TOG2	TEST2	ERR2	FIX2[4:0]				
r	r	r	r				

Bits	Fields	Descriptions
7	TOG2	Safe Channel 2 Toggle For successive position transmissions on safe channel 2, TOG2 must always toggle between 0 and 1. The starting value for TOG2 is 0. If the toggle bit does not change its value, it is probable that a transmission error occurred and the transmitted absolute value for safe channel 2 is invalid. Suitable measures must be installed in the user application.
6	TEST2	Safe Channel 2 Test TEST2 bit is set to 1 if a test message is executed via the MAIL register while the safe channel 2 status and position data are valid. TEST2 can only be valid if the user application has previously requested a test. Corresponding error indications for TEST2 are either the ERR2 bit or a discrepancy between the position and the CRC of safe channel 2.
5	ERR2	Safe Channel 2 Position Error 0: The last safe position received in safe channel 2 is correct. 1: The last safe position received in safe channel 2 is invalid. Suitable measures must be installed in the user application.
4:0	FIX2[4:0]	Safe Channel 2 Fixed Bit Pattern FIX2 is a fixed bit pattern for each SPI transaction that indicates errors arising in the SPI interface due to unchanged bit values. The standard value of the fixed bit pattern is 11100b. All other values indicate an error on safe channel 2 of the DSL system. Suitable measures must be installed in the user application.

HDSL safe position channel 2 byte n register (VPOS2n) (n = 0…4)

Address offset: 0x19 + (0x04 - n)
Reset value: 0x00

The VPOS2n registers contain the position value from the secondary channel of the motor feedback system connected. This safe position is transmitted in every eighth protocol frame if the validity of the data transfer has been checked.

The safe position is not synchronized with the most recent sync signal but to the (user or internally generated) sync signal of one of the last 8 protocol frames. Only those VPOS2 bits are relevant that lie within the range that the motor feedback system has actually measured.

Also, typically channel 2 has a lower resolution than channel 1. The operating manual and datasheet of the product specify the resolution of each channel.

The units of the position value are [steps].

These registers are write-protected.

This register has to be accessed by byte (8-bit).

7	6	5	4	3	2	1	0
VPOS2[7:0]							
r							

Bits	Fields	Descriptions
7:0	VPOS2[7:0]	Safe position, Channel 2, byte n Position value at motor feedback system Safe Channel 2 (length: 40 bits), transmitted as an absolute value complement.

HDSL high byte position checksum channel 2 register (VPOSCRC2_H)

Address offset: 0x1E
Reset value: 0x00

The VPOSCRC2_H/VPOSCRC2_L registers for the position checksum in safe channel 2 contain the CRC checksum for the safe position VPOS2n.

The CRC is checked in the DSL master IP Core. In order to guarantee, in a safety-related application, that the CRC machine is functioning, these registers can be checked with an external cross-check in the diagnostics test interval.

This register has to be accessed by byte (8-bit).

7	6	5	4	3	2	1	0
VPOSCRC2H[7:0]							
r							

Bits	Fields	Descriptions
7:0	VPOSCRC2H[7:0]	CRC of the safe position 2, bit 15 to bit 8 16-bit CRC checksum (CRC 16) of the safe position and status summary in Safe Channel 2.

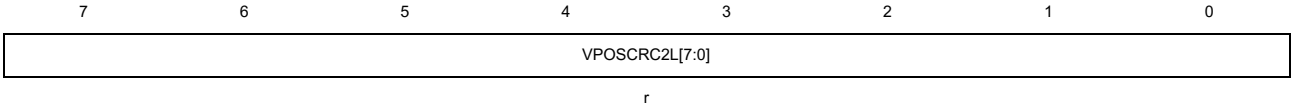
HDSL low byte position checksum channel 2 register (VPOSCRC2_L)

Address offset: 0x1F
Reset value: 0x00

The VPOSCRC2_H/VPOSCRC2_L registers for the position checksum in safe channel 2 contain the CRC checksum for the safe position VPOS2n.

The CRC is checked in the DSL master IP Core. In order to guarantee, in a safety-related application, that the CRC machine is functioning, these registers can be checked with an external cross-check in the diagnostics test interval.

This register has to be accessed by byte (8-bit).



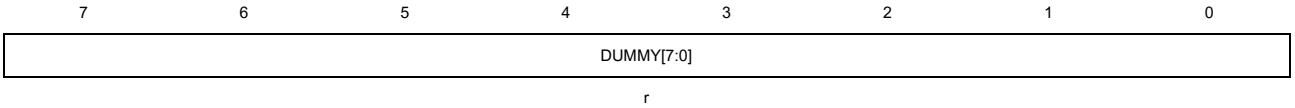
Bits	Fields	Descriptions
7:0	VPOSCRC2L[7:0]	CRC of the safe position 2, bit 7 to bit 0 16-bit CRC checksum (CRC 16) of the safe position and status summary in Safe Channel 2.

HDSL dummy in safe channel 2 register (DUMMY2)

Address offset: 0x3F
Reset value: 0xXX

The dummy read by the DUMMY2 register in the safe channel 2 is required.

This register has to be accessed by byte (8-bit).



Bits	Fields	Descriptions
7:0	DUMMY[7:0]	Due to the transmission of the online-status, Online Status read transactions need less time for transmission. Therefore, dummy read transactions must be inserted to avoid unwanted extra transactions. A read access to this register 0x3F has no effect and must be used for this purpose.

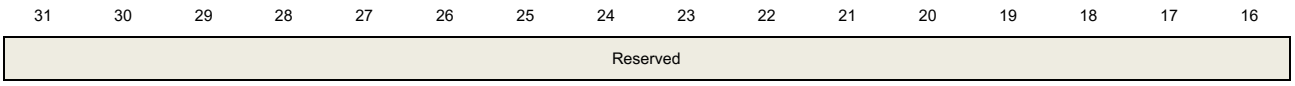
HDSL general registers

HDSL general registers base address: 0x4000 DB80

HDSL control register (EDIM_HDSL_CTL)

Address offset: 0x00
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												SSEL2	DSEL	SSEL1	ENDIAN
												rw	rw	rw	rw

Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value.
3	SSEL2	HIPERFACE DSL unit 0 safe channel 2 interface select Select the interface mode for drive interface 0: SPI interface 1: Internal bus interface
2	DSEL	HIPERFACE DSL unit 0 driver interface select Select the interface mode for drive interface 0: Internal bus interface 1: SPI interface **Note:** DSEL bit and SSEL1 bit cannot set simultaneously.
1	SSEL1	HIPERFACE DSL unit 0 safe channel 1 interface select Select the interface mode for drive interface 0: Internal bus interface 1: SPI interface **Note:** DSEL bit and SSEL1 bit cannot set simultaneously.
0	ENDIAN	HIPERFACE DSL unit 0 endian mode Byte sequence selection for register access 0: Big endian 1: Little endian **Note:** When using the accompanying firmware library, this bit needs to be set to 0.

HDSL online status (drive) register (ONLINE_STATUS_D)

Address offset: 0x04

Reset value: 0x0000 1B04

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT	SUM	Reserved		POS	Reserved	DTE	PRST	POSTX[1:0]		MIN	ANS	Reserved	QMLW	FREL	Reserved
r	r			r		r	r	r		r	r		r	r	r

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.

15	INT	Status of the interrupt signal 0: Interrupt output on the low level 1: Interrupt output on the high level
14	SUM	Summary byte 0: The last valid value from MIR_SUM was zero. 1: The last valid value from MIR_SUM was not zero. The importance of this flag depends on the particular error source that leads to a set MIR_SUM.
13:12	Reserved	Must be kept at reset value.
11	POS	Estimator turned on 0: No fast position error 1: A source of an error in the fast position was identified or an alignment procedure is currently being carried out. It is probable that the last fast position is invalid. Be aware that the fast position read through drive interface is provided by the estimator.
10	Reserved	Must be kept at reset value.
9	DTE	Deviation threshold error 0: Current value of deviation smaller than the specified maximum 1: Current value of deviation greater than the specified maximum
8	PRST	Protocol reset 0: HDSLn running 1: HDSLn has restarted the protocol
7:6	POSTX[1:0]	Position transmission status 00: Position request is transmitted to the DSL encoder 01: Safe position 1 was received 10: Fast position was received or position newly updated by estimator 11: Safe position 2 was received
5	MIN	Acknowledgment of message Initialization 0: Parameter Channel not functioning 1: The DSL encoder sends a figure by which the initialization of the Parameter Channel is acknowledged
4	ANS	Incorrect answer detected 0: No error detected in the last answer to a long message 1: The last answer to a long message was damaged
3	Reserved	Must be kept at reset value.
2	QMLW	Quality monitoring at low level 0: Current value of quality monitoring greater than or equal to 14 1: Current value of quality monitoring less than 14
1	FREL	Channel status for long message

0: The channel for the long message is in use.
1: The channel for the long message is free.

0	Reserved	Must be kept at reset value.

HDSL online status (safe channel 1) register (ONLINE_STATUS_1)

Address offset: 0x08
Reset value: 0x0000 3504

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SINT	SSUM	SCE	Reserved		VPOS	Reserved	PRST	POSTX[1:0]		MIN	Reserved		QMLW	Reserved	FRES
r	r	r			r		r	r		r			r		r

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	SINT	Status of the interrupt signal 0: Interrupt output on the low level 1: Interrupt output on the high level
14	SSUM	Safe summary bit 0: The last valid value from SAFE_SUM was zero. 1: The last valid value from SAFE_SUM was not zero. The importance of this flag depends on the particular error source that leads to a set SAFE_SUM.
13	SCE	CRC Error on the Safe Channel 0: The last Safe Channel CRC received was correct. 1: The last Safe Channel CRC received was wrong. It is expected that the last safe position 2 transmitted is invalid.
12:11	Reserved	Must be kept at reset value.
10	VPOS	Safe Position Invalid 0: The last safe position received was correct. 1: An error in the safe position was identified. It is expected that the safe position transmitted from the encoder is invalid.
9	Reserved	Must be kept at reset value.
8	PRST	Protocol reset 0: HDSL0 running 1: HDSL0 has restarted the protocol

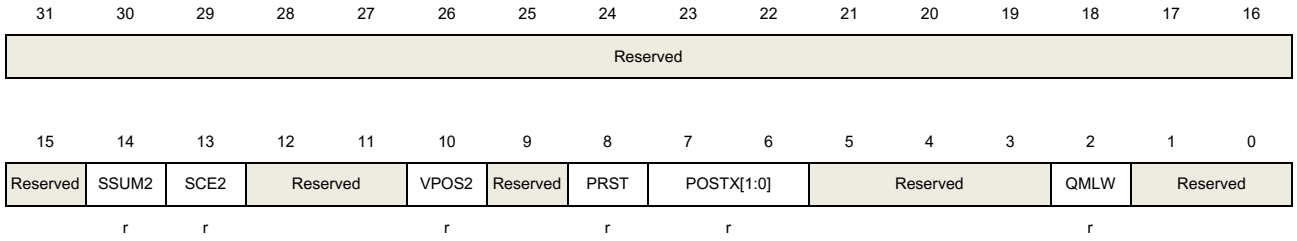
7:6	POSTX[1:0]	Position transmission status 00: Position request is transmitted to the DSL encoder 01: Safe position 1 was received 10: Fast position was received or position newly updated by estimator 11: Safe position 2 was received
5	MIN	Acknowledgment of message Initialization 0: Parameter Channel not functioning 1: The DSL encoder sends a figure by which the initialization of the Parameter Channel is acknowledged
4:3	Reserved	Must be kept at reset value.
2	QMLW	Quality monitoring at low level 0: Current value of quality monitoring greater than or equal to 14 1: Current value of quality monitoring less than 14
1	Reserved	Must be kept at reset value.
0	FRES	Channel Status for the short message 0: The channel for the short message is in use. 1: The channel for the short message is free.

HDSL online status (safe channel 2) register (ONLINE_STATUS_2)

Address offset: 0x0C

Reset value: 0x0000 3504

This register has to be accessed by word (32-bit).



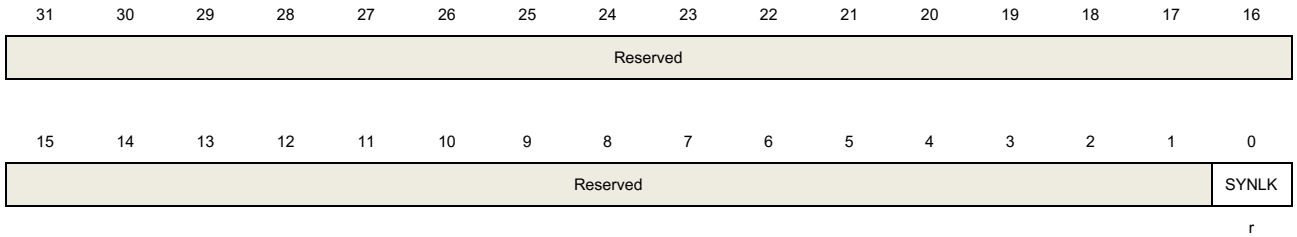
Bits	Fields	Descriptions
31:15	Reserved	Must be kept at reset value.
14	SSUM2	Summary byte Channel 2 0: Neither TEST2 nor ERR2 is set. 1: One of the indications TEST2 or ERR2 is set. The error reaction to the flag depends on the meaning of the bit.
13	SCE2	Transmission Error Channel 2 0: The last data received in channel 2 was correct. 1: The last safe channel 2 CRC received was wrong. It is expected that the last

		safe position 2 transmitted is invalid. Suitable measures must be installed in the user application.
12:11	Reserved	Must be kept at reset value.
10	VPOS2	Safe Position 2 Invalid 0: The last safe position received in channel 2 was correct. 1: A source of an error in the safe position in channel 2 was identified. It is probable that the safe position transmitted from Channel 2 is invalid. Suitable measures must be installed in the user application.
9	Reserved	Must be kept at reset value.
8	PRST	Protocol reset 0: HDSL0 running 1: HDSL0 has restarted the protocol
7:6	POSTX[1:0]	Position transmission status 00: Position request is transmitted to the DSL encoder 01: Safe position 1 was received 10: Fast position was received or position newly updated by estimator 11: Safe position 2 was received
5:3	Reserved	Must be kept at reset value.
2	QMLW	Quality monitoring at low level 0: Current value of quality monitoring greater than or equal to 14 1: Current value of quality monitoring less than 14
1:0	Reserved	Must be kept at reset value.

HDSL sync lock register (EDIM_HDSL_SYNLK)

Address offset: 0x10
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:1	Reserved	Must be kept at reset value.
0	SYNLK	Position sampling resolution locked

0: Not locked

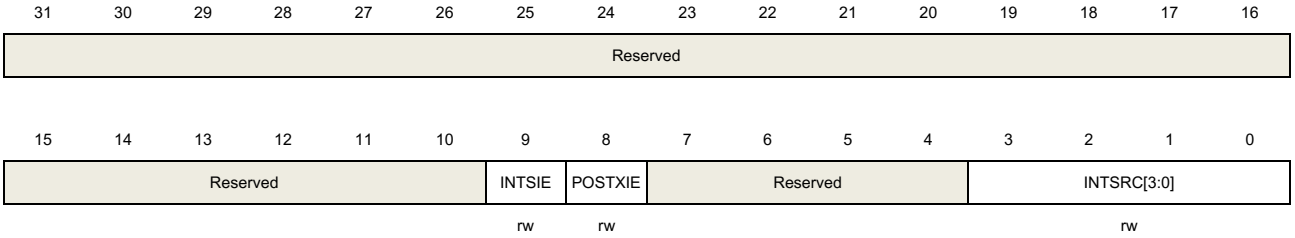
1: Locked

HDSL interrupt enable register (EDIM_HDSL_INTEN)

Address offset: 0x14

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	INTSIE	Interrupt_s enable 0: Interrupt_s disable 1: Interrupt_s enable
8	POSTXIE	POSTX interrupt enable 0: POSTX interrupt disable 1: POSTX interrupt enable
7:4	Reserved	Must be kept at reset value.
3:0	INTSRC[3:0]	POSTX interrupt source 0000: When POSTX=2'b01 0001: When POSTX=2'b10 0010: When POSTX=2'b11 0011: When POSTX = 2'b01, 2'b10, 2'b11 0100: When POSTX transfer from 2'b00 to 2'01 0101: When POSTX transfer from 2'b00 to 2'10 0111: When POSTX transfer from 2'b00 to 2'01 or 2'b10 1000: When POSTX transfer from 2'b01 to 2'10 1001: When POSTX transfer from 2'b10 to 2'b11 1010: When POSTX changes. Others: Reserved.

39. Resolver-to-digital module (RDCM)

39.1. Overview

The RDCM is a complete 14-bit tracking resolver-to-digital converter, integrating an on-chip programmable sinusoidal oscillator that provides sine wave excitor for resolvers.

The excitation signal frequency range output by the RDCM is 2 kHz to 100 kHz (DAC buffer on). The input I/Q signals utilize a Type II tracking loop to track the input signals and convert the input sine and cosine information into a digital representation of the input angle and velocity.

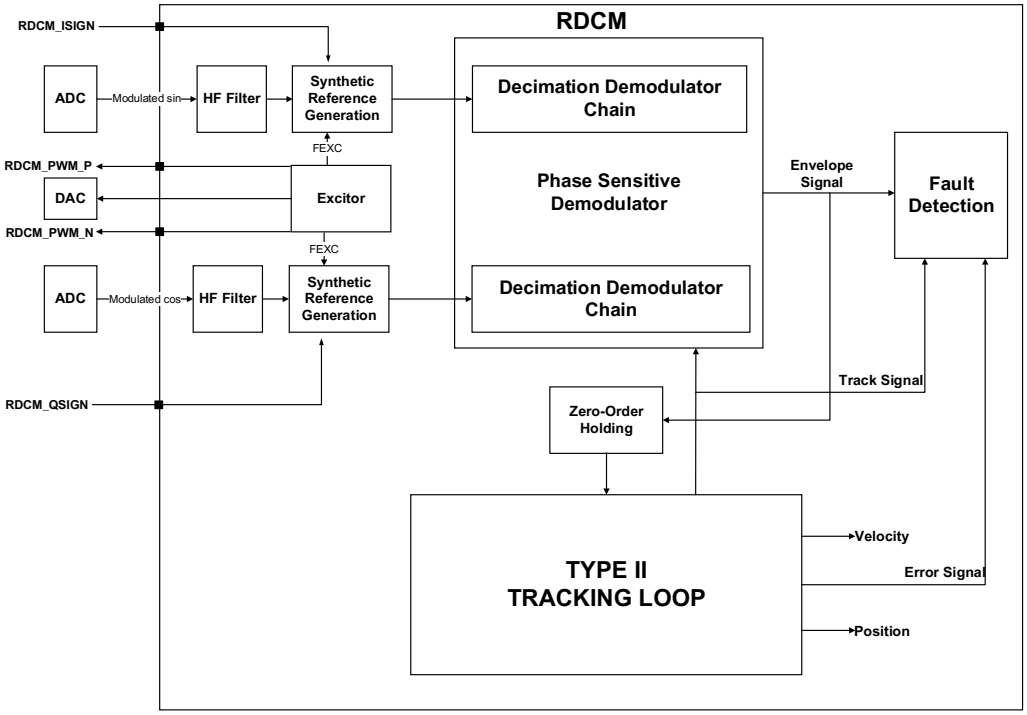
39.2. Characteristics

- Complete monolithic resolver-to-digital converter
- High tracking rate
- Up to 14-bit resolution
- Programmable on-chip excitor
  21. Analog sinusoidal waveform
  22. Digital PWM waveform
- Calculation support for ADC data or simulation data
- Configurable gain adjustment for I/Q signals
- Configurable high-pass filter for I/Q signals
- Configurable Synthetic Reference Signal for I/Q Rectification
  23. Manual mode, phase error adjustment range is 0 to 2pi
  24. Calibration mode (on-chip excitor reference), phase error extraction range is 0 to 2pi
  25. Calibration mode (external reference sign signal), phase error extraction and adjustment range is 0 to 2pi
  26. Auto mode, phase error extraction and adjustment range is ϕ to ϕ + pi (ϕ is configurable)
  27. Moving average function for carrier phase extraction
- Demodulator (concatenated hardware filter stages)
  28. Configurable CIC3 filter with decimation rates of 4 to 512
  29. FIR0 filter with 8 coefficients (10-bit) with decimation rate 2
  30. FIR1 filter with 28 coefficients (10-bit) with decimation rate 1 or 2
  31. Pass band 1/3 of envelope data rate (FIR1 decimation rate 2)
  32. Pass band 1/6 of envelope data rate (FIR1 decimation rate 1)
  33. Pass band ripple: -0.2 dB to 0.2 dB
  34. Stop band attenuation: (0.5 to 1) * f_d > 40 dB, (1 to 1.5) * f_d > 45 dB, (1.5 to 2) * f_d > 50 dB, (2 to 2.5) * f_d > 55 dB, (2.5 to total decimation rate / 2) * f_d > 60 dB
- Nonorthogonal phase mismatch compensation

- Configurable tracking loop for different resolution (default 14bit) and tracking rate
- Absolute position and velocity detection
- Smooth filter function for velocity extraction
- System fault detection support
  35. I channel envelope signal threshold match function
  36. Q channel envelope signal threshold match function
  37. Monitor signal threshold match function
  38. Track error signal threshold match function
- Programmable fault detection thresholds
- Interrupt generation
- Support DMA mode
- Support timestamp
- Support result gate
- Support result data FIFO

39.3. Block diagram

Figure 39-1. RDCM block diagram



For resolver application, RDCM provides an on-chip programmable sinusoidal oscillator that provides sine wave excitor to drive resolver. The I/Q input signals will be sampled by two 14-bit ADC0/1 as the RDC digital inputs. The synthetic reference generation module is provided for I/Q digital signals rectification. The phase sensitive demodulator is provided for I/Q digital (rectified) signals demodulation and envelope extraction. The type II tracking loop is provided for resolver position/velocity tracking. The fault detection module is provided for signal loss/degradation and system fault detection.

Note: Routine parallel mode of ADC is recommended for application of RDCM.

Table 39-1. RDCM pins definition gives the RDCM pin description.

Table 39-1. RDCM pins definition

Name	Description
RDCM_PWM_P	Positive PWM of RDCM exciter output
RDCM_PWM_N	Negative PWM of RDCM exciter output
RDCM_ISIGN	Carrier sign signal of I channel
RDCM_QSIGN	Carrier sign signal of Q channel

39.4. Function overview

39.4.1. RDCM clock

The function clock for RDCM is CK_RDCM, with a maximum frequency of 80MHz.

39.4.2. Sinusoidal excitor

The excitor uses a cordic core for generating programmable sinusoidal digital waveform.

If NCOETM[1:0] in RDCM_CTL register is set to 0, the software trigger mode is selected. The cordic core is calculated at every CK_RDCM clock cycle. If NCOETM[1:0] is not set to 0, the external trigger mode is selected and the trigger signal is from TRIGSEL. The cordic core is calculated at every selected trigger edge. The cordic core is 14 pipeline-depth. The first digital output is generated after 14 calculation cycles.

The excitor frequency of the sinusoidal wave is set by FEXC[13:0] in RDCM_EXCFG0 register. The excitor frequency is the calculation frequency * FEXC[13:0] / 16384.

When NCOETM[1:0] is set to 0, the calculation frequency of NCO (excitror, I/Q reference frequency mixer) will be CK_RDCM / (NCODIV[13:0] + 1), and NCODIV[13:0] is stored in RDCM_EXCFG1 register. In other external trigger modes, the calculation cycle is decided by the external trigger frequency divided by (NCODIV[13:0] + 1). The external trigger is configured in TRIGSEL.

The initial phase of the sinusoidal wave is set by PEXC[13:0] in RDCM_EXCFG0 register. The initial phase of the sinusoidal wave is 2pi * PEXC[13:0] / 16384.

Analog sinusoidal waveform

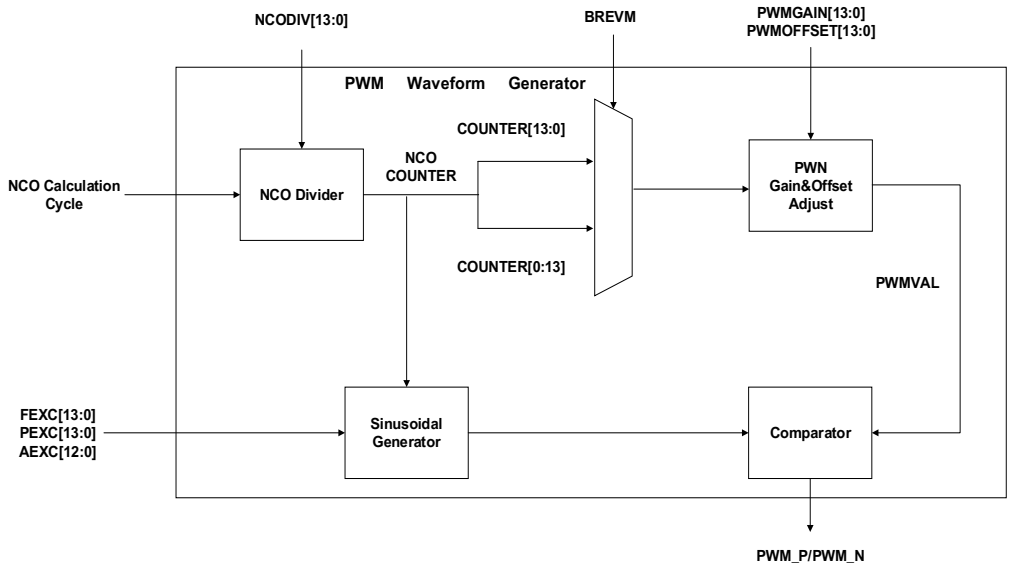
The amplitude of the sinusoidal wave is set by AEXC[12:0] in RDCM_EXCFG1 register. The range of digital sinusoidal wave is 0 ~ AEXC[12:0] / 2 to 12-bit DACs.

A positive sinusoidal wave (DAC0_OUT0, initial phase is 2pi * PEXC[13:0] / 16384) and a negative sinusoidal wave (DAC0_OUT1, initial phase is -2pi * PEXC[13:0] / 16384) will be both generated to drive the resolver.

PWM waveform

The block diagram of PWM waveform generator is shown in ***Figure 39-2. PWM waveform generator block diagram***.

Figure 39-2. PWM waveform generator block diagram



The PWM waveform generator, operating under the clock CK_RDCM, outputs a PWM signal that induces a sine signal in the resolver's excitation coil. When the PWMEN bit in the RDCM_PWMCFG register is set to 1, it can generate triangular or square wave patterns.

When BREVM bit in RDCM is 1, the PWMVAL will be bit-reversed. The bit-reverse generation mode increases the frequency spectrum to yield a smoother induced sine signal.

The NCO counter is generated by NCODIV[13:0] in RDCM_EXCFG1 register. The PWM value (PWMVAL) can be adjusted by PWMGAIN[13:0] and PWMOFFSET[13:0] in RDCM_PWNCFG1 register.

When BREVM bit in RDCM_PWMCFG register is set to 1, the bit-reversed mode is used. The PWMVAL is calculated by the follow formula:

PWMVAL = (COUNTER[0:13] - PWMOFFSET[13:0]) * PWMGAIN[13:0] / 16384 (39-1)

It is recommended to set PWMGAIN[13:0] to be 2 * AEXC[12:0], PWMOFFSET[13:0] to be 8192.

When BREVM bit in RDCM_PWMCFG register is set to 0, the bit-reversed mode is not used. The PWMVAL is calculated by the follow formula:

PWMVAL = (COUNTER[13:0]) * PWMGAIN[13:0] - PWMOFFSET[13:0] (39-2)

It is recommended to set PWMGAIN[13:0] to be roundup(2 * AEXC[12:0] / (NCODIV[13:0] + 1)), PWMOFFSET[13:0] to be AEXC[12:0].

Note: The recommended value of NCODIV[13:0] is roundup(16384 / FEXC[13:0] -1).

The 2's complement signed PWMVAL will be compared to the 2's complement signed digital sinusoidal wave (range: - AEXC[12:0]~ AEXC[12:0]) at each NCO calculation cycle. The PWM duty cycle in one NCO divider cycle indicates the current step value of the digital sinusoidal wave.

39.4.3. High-pass filter

The DC component of the I/Q channel digital data from the ADC can be eliminated by a high-pass filter for each channel. The offset component present in most input signals can be removed to obtain the original AC signal. The offset compensation can operate in two modes.

When I/QHFM[2:0] in RDCM_HFCFG0/1 register is set to 0, the offset compensation can remove the offset component of a DC signal by subtracting a predefined value from the raw results. The high-pass filter is disabled in this mode.

When I/QHFM[2:0] in RDCM_HFCFG0/1 register is set to other value, the offset compensation can automatically remove the DC component of a cyclic differential signal. This is achieved by the implemented high-pass filter. The compensation value in I/QOFFSET[14:0] of RDCM_HFCFG0/1 register is automatically updated by the filter. This attenuates the signal by at least -3 dB for frequencies below the selected cutoff frequency.

The cutoff frequency of high-pass filter for different values of I/QHFM[2:0] is shown in ***Table 39-2. High-pass filter cutoff frequency***.

Table 39-2. High-pass filter cutoff frequency

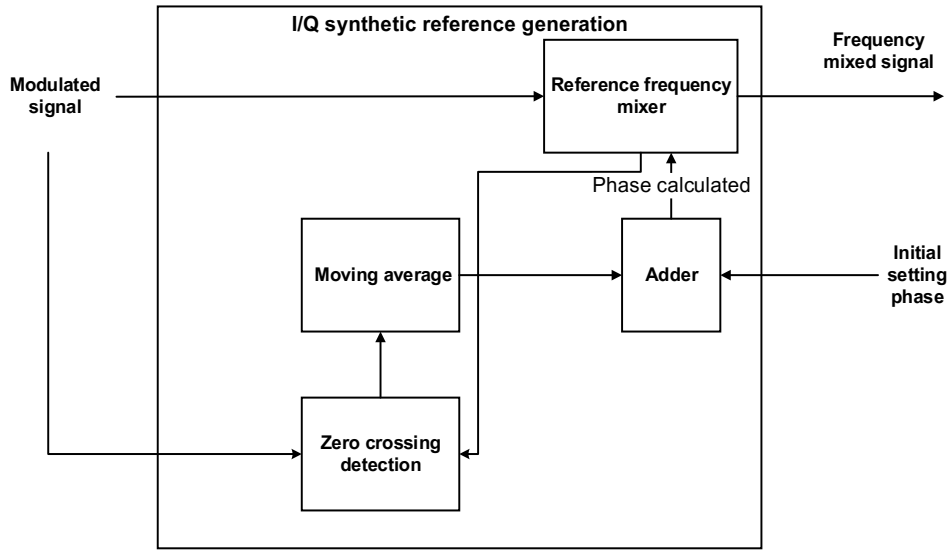
I/QHFM[2:0]	Cutoff frequency	Note
000	-	High-pass filter disable
001	$1 * 10^{-5} * f_d$	Lowest cutoff frequency, rate 1
010	$4 * 10^{-5} * f_d$	Rate 2
011	$16 * 10^{-5} * f_d$	Rate 3
100	$62 * 10^{-5} * f_d$	Rate 4
101	$25 * 10^{-4} * f_d$	Rate 5
110	$1 * 10^{-2} * f_d$	Rate 6
111	$3.5 * 10^{-2} * f_d$	Highest cutoff frequency, rate 7

Note: f_d is data output rate from ADCs, meaning the sampling rate of ADCs.

39.4.4. Synthetic Reference Generation

The synthetic reference generation is divided into two separate I and Q channels, with each channel having the same structure, as shown in ***Figure 39-3. I/Q synthetic reference generation block diagram***.

Figure 39-3. I/Q synthetic reference generation block diagram



When a resolver undergoes a high rotation rate, the resolver produces speed voltages, along with the ideal sine and cosine outputs. These speed voltages are in quadrature to the main signal waveform. Moreover, nonzero resistance in the resolver windings causes a nonzero phase shift between the reference input and the sine and cosine outputs.

To compensate for the described phase error between the excitor and the sine/cosine signals, the following adjust mode is supplied. These mode can be configured in I/QZCDMS[1:0] of RDCM_CTL register.

Manual mode: Users can configure the initial phase of the I/Q reference frequency mixer in I/QPHASE[13:0] of RDCM_RFMCFG0/1 register to manually compensate for phase errors. Phase error compensation can achieve a full range of adjustment.

Calibration mode contain two types, configured in I/QSIGNSEL bit of RDCM_DISWSEL register.

When I/QSIGNSEL bit is set to 0, use on-chip excitor sign as I/Q channel RDCM frequency mixer reference. The resolver shaft angle should be fixed at a positive half-cycle of the envelope position for this mode. The hardware will capture and statistical averaging the positive /negative zero crossing mismatch counter. The zero crossing detection mode is set in I/QZCDMS[1:0] of RDCM_CTL register. The phase error extraction can achieve full range. The I/Q calibrated phase value will update in I/QPHASE[13:0] of RDCM_RFMCFG0/1 register and the mixer phase will be adjusted automatically.

Note: If the resolver shaft angle is in the negative half-cycle of the envelope, the calibrated phase will have a pi phase error.

When I/QSIGNSEL bit is set to 1, Use external-chip excitor sign as I/Q channel RDCM frequency mixer reference. The hardware will capture and statistical averaging the positive / negative zero crossing mismatch counter. The phase error extraction and adjustment can achieve full range. The I/Q calibrated phase value will update I/QPHASE[13:0] of

RDCM_RFMCFG0/1 register and the mixer phase will be adjusted automatically.

Automatic mode: An internal synthetic reference signal is generated in phase with the reference frequency carrier in I/Q channel. The synthetic reference is derived using the internally filtered sine and cosine signals. It is generated by statistical average of evaluating the zero crossing delay of the sine, cosine and the reference frequency mixer in separate. By configuring the different initial phase of excitor and reference frequency mixer, the automatic phase error extraction and adjustment range can achieve ϕ ~ ϕ + pi (ϕ is configurable). The I/Q calibrated phase will update in I/QPHASE[13:0] of RDCM_RFMCFG0/1 register and the mixer phase will be adjusted automatically.

Note: The synchronization of the excitor and reference frequency mixer should be always guaranteed.

Example:

1. Set the PEXC[13:0] in RDCM_EXCFG0 register to 0.
2. Set the IPHASE[13:0] in RDCM_RFMCFG0 register to 12288.
3. Set the QPHASE[13:0] in RDCM_RFMCFG1 register to 0.

The phase shift in the range of -pi/2 ~ pi/2 between the excitor and the sine can be compensated for I channel.

The phase shift in the range of 0 ~ pi between the excitor and the cosine can be compensated for Q channel.

If the phase shift is out of the range, the signal after the reference frequency mixer will be opposite.

Reference frequency mixer

The I/Q signal after high-pass filter is fed through to the reference frequency mixer for signal rectification. To compensate for the described phase error between the resolver reference excitor and the sine/cosine signals, the frequency and phase of the I/Q channel frequency mixer can be configured respectively in RDCM_RFMCFG0/1 register by users.

Note: It's recommended to configure the frequency of the excitor, I/Q reference frequency mixer to the same value. I/QRFMEN bit in RDCM_CTL register is only valid when EXCEN bit in RDCM_CTL register is set. I/Q channel reference frequency mixer will not work when EXCEN bit is reset.

The reference frequency mixer is realized by a is 14 pipeline-depth cordic core. The first digital output of frequency mixer signal is generated after 14 calculation cycles.

Zero crossing detection

To detect the delay of zero crossing point of the high frequency part of modulated signal (offset-compensated) and reference frequency mixer, a mismatch timer is used to generate the counter value in NCO (excitor and reference frequency mixer) calculation cycle. To remove the influence of the envelope in modulated signal to zero crossing detection, the mismatch

counter value threshold should be configured. The recommended value is roundup(16384 / FEXC[13:0]) for calibration mode and roundup(8192 / FEXC[13:0] for automatic mode, stored in I/QZCTH[13:0] bits of RDCM_ZTCFG0 register. The mismatch counter value between I/QZCTH[13:0] - I/QZTVAR[13:0] ~ I/QZCTH[13:0] + I/QZTVAR[13:0] will be captured for evaluating.

Statistical average

The zero crossing detection method is sensitive to the signal quality and are precision-lismited. The precision of phase error adjustment is pi * FEXC[13:0] / 8192. The mismatch counter value that captured may be unstable. The hardware statistical average is used to get a stable mismatch counter value.

The transfer function of the statistical average is:

$$H(z) = 2^{-I/QAVEM[3:0]} / (1 - \left(1 - 2^{-I/QAVEM[3:0]}\right) * Z^{-1}) \quad (39\text{-}3)$$

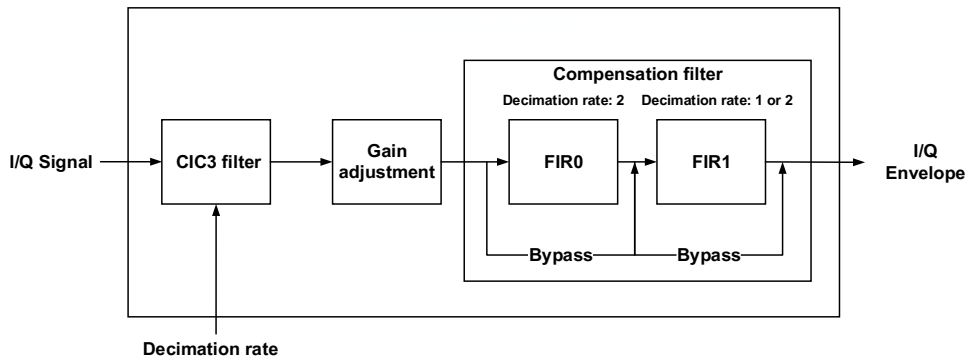
When I/QAVEM[3:0] in RDCM_MCACFG register is set to 4'b0000, the statistical average function is disabled, and the mismatch counter value that captured will update the phase adjustment of the reference frequency mixer each time. The statistical average strength is larger for a larger I/QAVEM[3:0] value, but the time of convergence is longer.

The phase adjustment value is the statistical averaged mismatch timer counter value multiplied by FEXC[13:0] and will be added to the original phase value of the reference frequency mixer to compensate the phase error described above.

39.4.5. Phase sensitive demodulator

The phase sensitive demodulator is divided into two separate I and Q channels, with each channel having the same structure The demodulator consists of three parts: the CIC3 filter, gain adjustment, and compensation filter, as shown in ***Figure 39-4. Phase sensitive demodulator block diagram***.

Figure 39-4. Phase sensitive demodulator block diagram



The I/Q envelopes are generated by feeding the I/Q data stream through a chain of filter

elements and decimating it by a calculated decimation ratio respectively. The I/Q envelopes can be transferred by DMA for further software tracking of velocity and position, but the group delay of filter elements of the demodulator is involved. To eliminate the high-frequency part, the total decimation ratio of the demodulator should be set as follow:

$$DEC_{CIC3} >= F_{data_input} / (2 * F_{excitor}) \quad (39\text{-}4)$$

DEC_{CIC3}: The decimation ratio of the demodulator.

F_{data_input}: The data input rate.

$F_{excitor}$: The frequency of excitor.

Note: In practical applications, the data input rate is equal to the sampling rate of the ADC. In simulation mode, the data input rate is determined by the frequency of the external trigger signal from TRIGSEL.

Most elements of the filter can be bypassed. The filter chain is configurable and its behavior can be adapted to the requirements of the actual application. This comprises the frequency attenuation and the total decimation rate.

CIC3 filter

The cascaded integrator comb filter (CIC filter) is a simple and efficient low-pass filter. CIC3 filter has three comb filter stages connected to boost its frequency performance. Users can set the decimation rate from 4 to 512, and this setting is stored in the I/QCICDFAC[8:0] of RDCM_CICCFG0/1 register.

To synchronize filters of different channels precisely, the decimation counter starts from an arbitrary start value. This value can be different from the decimation factor and is loaded only when the counter is started. The CIC3 filter properties is shown in ***Table 39-3. CIC3 filter properties (N = I/QCICDFAC[8:0] + 1)***

Table 39-3. CIC3 filter properties (N = I/QCICDFAC[8:0] + 1)

Filter	Maximum delay	Max output value gain
CIC3	3N	$[-N^3, N^3]$

The valid data bits portion of the CIC3 filter output is influenced by the CIC3 decimation factor I/QCICDFAC[8:0] and the employed modulator. The data shifter is controlled by I/QCICSHIFT[5:0] of RDCM_CSGCFG0/1 register.

The value for I/QCICSHIFT[5:0] is determined by following formula:

$$I/QCICSHIFT[5:0] = roundup(11 - \log_2(A_s / (N^3 * A_t)) \quad (39\text{-}5)$$

N: This value is equal to I/QCICDFAC[8:0] + 1.

A_s: The amplitude value of the signal after high-pass filter.

A_t: The target amplitude value after calibration.

The rounding error in the above formula can be corrected by calculating a gain correction

factor I/QGAIN[12:0], stored in RDCM_CSGCFG0/1 register.

$$I/QGAIN[12:0] = truncate(((A_s / (N^3 * A_t)) * 2^{(I/QCICSHIFT[5:0] -11)}) * 4096) \quad (39\text{-}6)$$

Example:

For N = 15, the amplitude value of the signal after high-pass filter A_s is 7000, we expect the signal amplitude after CIC3 gain adjustment A_t is also 7000.

The value of I/QCICSHIFT[5:0] is:

I/QCICSHIFT[5:0] = roundup(11 - $\log_2$(7000 / (N^3 * 7000))) = 23.

The gain correction factor is:

I/QGAIN[12:0] = truncate(((7000 / (N^3 * 7000)) × $2^{(I/QCICSHIFT[5:0]-11)}$) * 4096) = 4971.

Gain adjustment

I/Q channel gain adjustment is done by multiplying the raw results with a resulting correction factor decided by software. The resulting correction factor is I/QGAIN[12:0] / 4096. The resulting correction factor can be in the range of 0.000 to 1.999, with an initial value of 1.000.

The resulting correction factor is used to compensate the reduced digital gain that results from decimation factors in the CIC3 filter that are not 2^N.

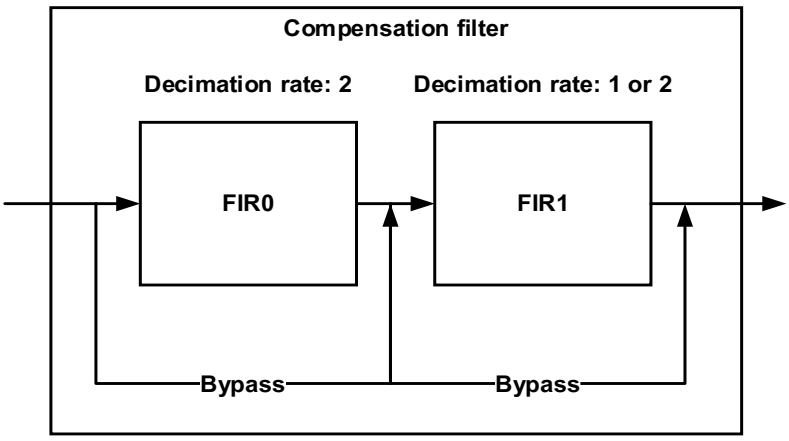
The value of I/QGAIN[5:0] is calculated from application parameters either during compile time for static configurations, or during run time for dynamic configurations.

Note: If the I/Q channel phase sensitive demodulator function is disabled, the gain adjustment will be used for the data output of high-pass filter in I/Q channel filter straightly. The normalized gain adjustment of the offset-compensated signal is I/QGAIN[12:0] / ($2^{I/QCICSHIFT[5:0]}$). The I/Q channel phase sensitive demodulator function enable is configured in I/QDEMEN bit in RDCM_CTL register.

Compensation filter

The compensation filter consists of two FIR modules. The decimation rate of FIR0 is 2, and the decimation rate of FIR1 can be set to either 1 or 2. The decimation rate of FIR1 is controlled by I/QFIR1DEC bit in RDCM_DEMCFG register. The total decimation rate for the phase sensitive demodulator is 2 * N or 4 * N. The structure of compensation filter is shown in ***Figure 39-5. Compensation filter block diagram***.

Figure 39-5. Compensation filter block diagram



The coefficients (signed 10 bits, must be symmetric) of both FIR filters are fixed. The filters are symmetric and have linear phase. The FIR0 is an order-8 symmetric filter and FIR1 is an order-28 symmetric filter.

The default coefficients for FIR0/FIR1 are in the following table.

Table 39-4. FIR0 default coefficients

a0	a1	a2	a3	a4	a5	a6	a7
-29	-43	143	441	441	143	-43	-29

Table 39-5. FIR1 default coefficients

a0, a27	a1, a26	a2, a25	a3, a24	a4, a23	a5, a22	a6, a21	a7, a20	a8, a19	a9, a18	a10, a17	a11, a16	a12, a15	a13, a14
-5	-3	3	14	-2	-22	-12	30	39	-20	-86	-19	196	399

39.4.6. Group delay summary

Data entering a digital filter require a specific number of filter clock cycles to pass through before they emerge at the output, ready for system use. When the digital filter chain reaches a steady state, the delay is known as the group delay, which is determined by the filter's configuration. *Table 39-6. Group delay of filter elements* shows the delays introduced by each element in the filter chain. The delays are measured in RDCM calculation cycles, which correspond to ADC data update cycles.

Table 39-6. Group delay of filter elements

Filter element	Delay	Description
High-pass filter	0	Delay of high-pass filter
Gain stage	1	Pineline delay of gain stage calculation
Nonorthogonal phase mismatch compensation	0	Pineline delay of nonorthogonal phase mismatch compensation
CIC3	3 * (N - 1) / 2	Delay of CIC3 stage in PSD (N is the decimation ratio of CIC3)

Filter element	Delay	Description
FIR0	3.5 * N	Delay of FIR0 stage in PSD
FIR1	13.5 * N * 2^{F0}	Delay of FIR1 stage in PSD F0 = 0 for FIR0 is off, F0 = 1 for FIR0 is on
Zero-order holding	N * 2^{F0} * 2^{F1} / 2	Delay of zero-order holding after PSD F0 = 0 for FIR0 is off, F0 = 1 for FIR0 is on, F1 = 0 for FIR1 is off or FIR1 decimation is off, F1 = 1 for FIR1 on and FIR1 decimation is on

Note: PSD is phase sensitive demodulator.

39.4.7. Nonorthogonal phase mismatch compensation

In the ideal cases, the I/Q signal envelopes are orthogonal. Actually, due to non-ideal factors like assembly of the resolver, the nonorthogonal phase mismatch is involved.

We assume the envelope of I channel is sin(θ), the envelope of Q channel is cos(θ+σ). θ is the angle of resolver position, σ is the nonorthogonal phase mismatch. The cos(θ+σ) can be expanded using the follow formula:

$$\cos(\theta+\sigma)=\cos(\theta)\cos(\sigma)-\sin(\theta)\sin(\sigma) \quad (39\text{-}7)$$

Using the above formula, we can derive the follow formula:

$$\cos(\theta) = \cos(\theta+\sigma) / \cos(\sigma) + \sin(\theta)\tan(\sigma) \quad (39\text{-}8)$$

let compensation factor X = 1/cos(σ), compensation factor Y = tan(σ)

$$Qcompensated = Qsignal * X + Isignal * Y = \cos(\theta+\sigma)/\cos(\sigma) + \sin(\theta)\tan(\sigma) = \cos(\theta) \quad (39\text{-}9)$$

To compensate the nonorthogonal phase mismatch, the XFAC[15:0] and YFAC[15:0] in RDCM_NPMCFG register can be configured by software manually.

The XFAC[15:0] should be configured as 1/cos(σ)*16384

The YFAC[15:0] should be configured as tan(σ)*16384

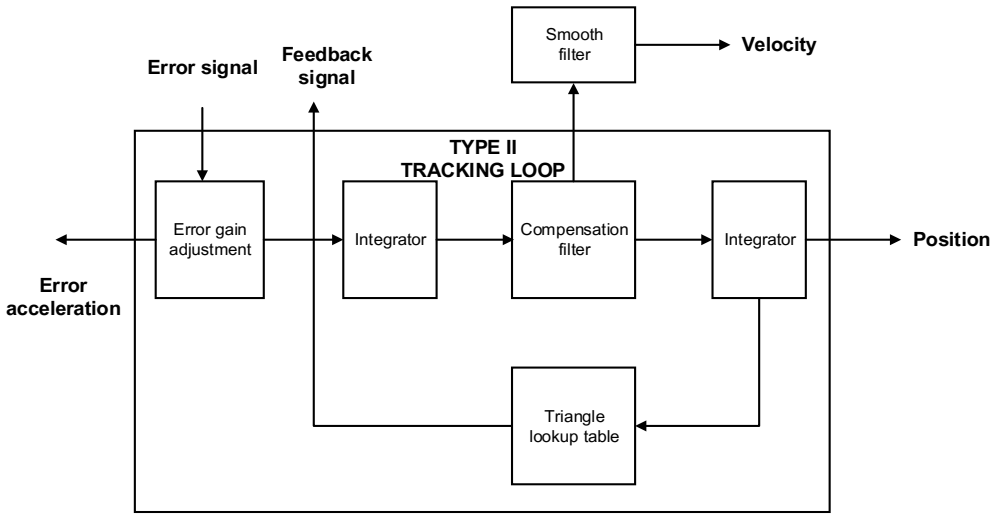
The range of nonorthogonal phase mismatch compensation is -pi/3 ~ pi/3.

Note: Nonorthogonal phase mismatch compensation is only valid after I/Q signal is demodulated.

39.4.8. Type II tracking loop

RDCM hardware decoding is based on a type II tracking closed-loop principle. Whenever the resolver moves through an angle equivalent to the weight of the least significant bit (LSB), the output value increases by one. The structure of type II tracking loop is shown in ***Figure 39-6. Type II tracking loop block diagram***.

Figure 39-6. Type II tracking loop block diagram



RDCM hardware decoding tracks the resolver shaft angle θ by generating an output angle ϕ. This output is fed back and compared to the input angle θ. If the RDCM hardware decoding accurately follows the input angle, the error between the two angles is minimized towards zero.

The type II tracking loop converts the digital signals after carrier frequency mixed or demodulation to digital position and velocity. The digital position and velocity are stored in POSITION[15:0] and VELOCITY[15:0] of RDCM_TRKDATA register.

The first gain stage consists of the gain of the error signal into the first integrator. The first integrator generates a signal proportional to velocity. The error gain can be configured in TRKERRGAIN[28:0] of RDCM_TRKGAINCFG register.

The second integrator is the same as the first and generates the position output from the velocity signal. The sine/cosine lookup has a configurable gain. The velocity and position integrator gain factor can be configured in VINTGAIN[4:0] and PINTGAIN[4:0] of RDCM_TRKINFCFG0 register.

The compensation filter contains a pole and a zero that are used to provide phase margin and reduce high frequency noise gain. The zero and pole of the compensation filter can be configured in ZERO[14:0] and POLE[14:0] of RDCM_TRKCMPCFG register.

The actual velocity of the resolver is ADC data rate multiplied by VELOCITY[15:0] / (16384 * 1024). The unit is rps.

The angle of the resolver is 2pi * POSITION[15:0] / 16384. The unit is rad.

The mismatch like nonorthogonality, gain mismatch of sine/cosine and so on will cause vibration of the extraction of velocity value. A smooth filter is used to eliminate the instability of velocity value. The smooth filter strength can be configured in VSFM[3:0] of RDCM_VSFCFG register.

The transfer function of the smooth filter is:

$$H(z) = 2^{-VSFM[3:0]} / (1 - (1 - 2^{-VSFM[3:0]}) * Z^{-1}) \quad (39\text{-}10)$$

When VSFM[3:0] is set to 4'b0000, the smooth filter function is disabled, and the velocity is extracted from the compensation filter. The smooth filter strength is larger for a larger VSFM[3:0] value but the time of convergence is longer.

39.4.9. Data output rate

If the I/Q channel demodulator is disabled, the output data of high-pass filter/reference frequency mixer/ tracking loop is generated at the ADC sampling rate. If the demodulator is enabled, the output data of demodulator/monitor is generated at the ADC sampling rate decimated by a configurable ratio.

Note: If main chain (high-pass filter, demodulator and tracking loop) calculation trigger is enabled in CALETM[1:0] in RDCM_CTL register, the ADC sampling rate will be substituted by main chain calculation trigger frequency.

39.4.10. Timestamp support

The result data, configured in RESDSEL[2:0] of RDCM_CTL register, is generated by feeding the input data stream through a chain of filter elements and decimating it by a configurable ratio.

Some applications require determining the result at specific times between two regular output values. The interpolation algorithm must ascertain the position of the desired time point relative to the regular results.

This interpolation is supported by providing a timestamp that marks the delay since the last regular output value. This timestamp counts the RDCM clock cycles since the last result data was produced. The fraction of an output cycle is defined by the configured decimation factors, which enables the interpolation.

The timestamp information is generated by a timestamp counter. The 16-bit timestamp counter is clocked with the configurable prescaled RDCM clock. The prescaled RDCM clock can be configured in TSCLK[1:0] of RDCM_TSCTL register. It covers the following cycles:

1. CIC3 filter decimation: up to 512
2. FIR filter decimation: up to 4

The timestamp information is updated in TSVAL[15:0] of RDCM_TSDATA register upon a trigger event. The trigger signal is configured in TRIGSEL. The trigger edge can be selected by TSCAPETM[1:0] of RDCM_CTL register.

When timestamp counter capture is valid, meaning TSCAPIF bit in RDCM_CS register is set, an timestamp interrupt will be generated if TSCAPIE bit in RDCM_CS register is set.

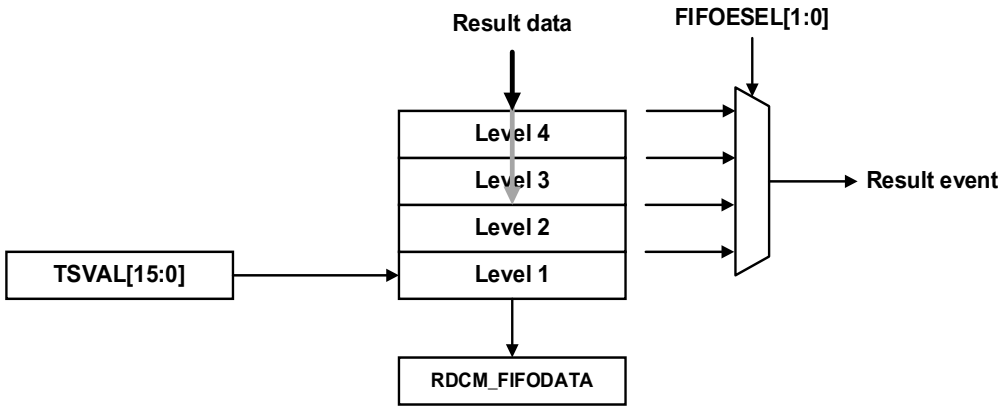
The TSTRGF bit and TSCAPIF bit are set to 1 when timestamp information is stored and is cleared by hardware after TSVAL[15:0] is read. These bits indicate if or not a timestamp trigger has occurred before the corresponding result value was generated.

39.4.11. Result data

The result data register is selected in RESDSEL[2:0] in RDCM_CTL register.

When FIFOEN bit in RDCM_CTL register is set to 1, the result data is not only stored in the corresponding data registers but also in a FIFO structure, which can be accessed via the RDCM_FIFODATA register. The FIFO structure is shown in *Figure 39-7. FIFO structure*.

Figure 39-7. FIFO structure



This FIFO has the following advantages:

1. Buffering allows for increased latency in read result data.
2. Data from both I/Q channels can be read with a single read operation.
3. The previous result value remains available for calculating interpolation values needed for timestamp operations.

Timestamp information can be accessed not only through the RDCM_FIFODATA register but also through the TSVAL[15:0] in RDCM_TSDATA register.

By using the FIFO function, all related data can be transmitted using a single DMA channel.

Result data storage without FIFO enable

If FIFOEN bit is set to 0, the FIFO is disabled and the result data is written to RDCM_FIFODATA register directly. The timestamp information can be only obtained by read the TSVAL[15:0] of RDCM_TSDATA register in this mode.

Result data storage with FIFO enable

If FIFOEN bit is set to 1, the FIFO is enabled and the result data is not directly written to

RDCM_FIFODATA register but are stored in a FIFO structure.

When the gate signal is invalid, MCU can't read data from the RDCM_FIFODATA register. The gate signal is the timestamp trigger signal from TRIGSEL and its valid level is configured in GATEVSEL[1:0] of RDCM_CS register.

When TSCAPEN bit in RDCM_CTL register is set to 1, the timestamp is enabled. In timestamp mode, all result data is directly forwarded to level 2 of the FIFO if the gate signal is invalid. When the gate is valid (this is the trigger event for timestamp). A timestamp is generated and inserted to level 1 of the FIFO. Subsequent result data is then piled into the FIFO. If timestamp mode is disabled, all result data is directly forwarded to level 1 of the FIFO if the gate signal is invalid. When the gate is valid (this is the trigger event for timestamp), subsequent result data is then piled into the FIFO.

The result event (DMA request or result data interrupt) is generated when a certain number of data has been stored in the FIFO and the result flag RESDIF bit in RDCM_CS register is set to 1. The result event is configured in FIFOESEL[1:0] in RDCM_FIFOCS register.

39.4.12. Fault Detection

For different fault detection applications, the below cases are given for threshold checking. The different fault detection configurations are in RDCM_CS register and RDCM_CTL register.

I/Q envelope signal threshold checking

The I/Q channel envelope signal is given as below:

$$I_{envelope\ signal} = E_0 * \sin(\theta) \quad (39\text{-}11)$$

$$Q_{envelope\ signal} = E_0 * \cos(\theta) \quad (39\text{-}12)$$

$I_{envelope\ signal}$: I channel signal after demodulation (I channel envelope signal).

$Q_{envelope\ signal}$: Q channel signal after demodulation (Q channel envelope signal).

E_0: The signal amplitude of I/Q channel (assuming the amplitude of I/Q channel signal is same here).

θ: Resolver shaft angle after group delay.

A comparator provides automatic threshold checking by comparing the envelope signal to two configurable reference values in I/Q channel. This feature can be used to supervise the input signal and detect abnormal input values.

The I/Q envelope signal threshold checking enable is configured in I/QETHMEN bit in RDCM_CTL register. The checking modes include double side threshold checking and single side threshold checking, configured in I/QETHMS[1:0] in RDCM_CS register.

Double side threshold checking:

- Lower threshold < Envelope signal value < Upper threshold.

■ Envelope signal value < Lower threshold or Upper threshold < Envelope signal value

Single side threshold checking:

1. Lower threshold < Envelope signal value
2. Lower threshold > Envelope signal value

The lower threshold is configured in I/QETHL[14:0] of RDCM_ETHCFG0/1 register and the upper threshold is configure in I/QETHU[14:0] of RDCM_ETHCFG0/1 register.

When I/QETHMIF bit in RDCM_CS register is set to 1, the envelope signal threshold match event will generate interrupts in I/Q channel separately if I/QETHMIE bit in RDCM_CS register is set to 1.

Tracking error signal threshold checking

The tracking error signal of the two channel envelope signals after tracking loop operation is given as:

$$E_0 * \sin(\theta) * \cos(\phi) - E_0 * \cos(\theta) * \sin(\phi) = E_0 * \sin(\theta - \phi) \quad (39\text{-}13)$$

ϕ: Tracking angle of θ.

The tracking error signal approximately equal to $E_0 * (\theta - \phi)$ for small values of $(\theta - \phi)$.

The tracking error signal threshold checking enable is configured in TRKERRTHMEN bit in RDCM_CTL register. The checking modes include double side threshold checking and single side threshold checking, configured in TRKERRTHMS[1:0] in RDCM_CS register.

Double side threshold checking:

1. Lower threshold < Tracking error signal value < Upper threshold.
2. Tracking error signal value < Lower threshold or Upper threshold < Tracking error signal value

Single side threshold checking:

1. Lower threshold < Tracking error signal value
2. Lower threshold > Tracking error signal value

The lower threshold is configured in TRKERRTHL[14:0] of RDCM_TRKERRTHCFG register and the upper threshold is configure in TRKERRTHU[14:0] of RDCM_TRKERRTHCFG register.

When TRKERRTHMIF bit in RDCM_CS register is set to 1, the tracking error threshold match event will generate an interrupt if TRKERRTHMIE bit in RDCM_CS register is set to 1.

The error signal can be read by accessing the TRKERRDATA[15:0] in RDCM_MONDATA register.

Note: The tracking error threshold match event only be valid after meeting the required number of times (TRKERRTHCNT[13:0]+1) continuously. The TRKERRTHCNT[15:0] is in RDCM_THCNT register.

Monitor signal threshold checking

The monitor signal is given as:

$$A_1 * \sin(\theta) * \sin(\phi) + A_2 * \cos(\theta) * \cos(\phi) \quad (39\text{-}14)$$

A_1: A_1 is the amplitude of I channel envelope signal.

A_2: A_2 is the amplitude of Q channel envelope signal.

When A_1 is equal to A_2 and the RDCM is tracking (θ is ϕ), the monitor signal output has a constant magnitude of A1, which is independent of resolver shaft angle.

When A1 is not equal to A2, the monitor signal magnitude varies between A1 and A2 at twice the velocity of shaft rotation. The monitor signal is used to detect degradation or loss of input signals.

The monitor signal threshold checking enable is configured in MTHMEN bit in RDCM_CTL register. The checking modes include double side threshold checking and vibration checking, configured in MTHMS[1:0] in RDCM_CS register.

Double side threshold checking mode:
1. Lower threshold < Monitor signal value < Upper threshold.
2. Monitor signal value < Lower threshold or Upper threshold < Monitor signal value

Vibration checking mode:
1. Lower threshold < Monitor signal vibration value
2. Lower threshold > Monitor signal vibration value

The monitor signal vibration value is the difference of maximum value and minimum value (Absolute value, will be updated automatically in hardware).

The maximum value and minimum value will also be updated when this function is enabled (MTHMEN bit in RDCM_CTL register is set) and will be cleared to zero when this function is disabled. It's recommended to enable this function after tracking enters a state of convergence.

When MTHMIF bit in RDCM_CS register is set to 1, the monitor signal checking event will generate an interrupt if MTHIE bit in RDCM_CS register is set to 1.

The monitor signal can be read by accessing the MONDATA[15:0] in RDCM_MONDATA register.

Note: The monitor signal threshold match event only be valid after meeting the required number of times (MTHCNT[15:0] +1) continuously. The MTHCNT[15:0] is in RDCM_THCNT register.

39.4.13. DMA mode

The DMA request is used to transfer the result data of RDCM. The result data is selected by RESDSEL[2:0] of RDCM_CTL register. The DMA request is enabled by DMAEN bit of RDCM_CTL register. When this bit is set to 1, the RDCM generates a DMA request when the

result data is updated or result data FIFO reaches the level configured in FIFOESE[1:0] of RDCM_FIFOCS register. When this request is received, the DMA will transfer the result data from the according data register to the destination address which is specified by the user.

39.4.14. Overflow detection

Overflow detection is enabled only when DMA is enabled. An overflow error occurs when the FIFO is full and new result data is written before the previous data is read out. The OVIF bit of the RDCM_CS register is set to 1. Overflow interrupt is generated if OVIE bit in RDCM_CS register is set to 1.

It is recommended to reinitialize the DMA module to recover the RDCM from overflow state. To ensure the result data are transferred correctly, the RDCM calculation is stopped. The RDCM calculation will be stalled until the OVIF bit is cleared.

If the result data FIFO is disabled (FIFOEN = 0), software procedure for recovering the DMA from overflow state:

1. Clear DMAEN bit of RDCM_CTL register to 0
2. Clear RDCMEN bit of RDCM_CS register to 0
3. Clear CHEN bit of DMA_CHxCTL register to 0 with reinit DMA module
4. Clear OVIF bit of RDCM_CS register to 0
5. Configure the RDCM
6. Set DMAEN bit of RDCM_CTL register to 1
7. Set RDCMEN bit of RDCM_CS register to 1

If the result data FIFO is enabled (FIFOEN =1), software procedure for recovering the DMA from overflow state:

1. Clear DMAEN bit of RDCM_CTL register to 0
2. Clear RDCMEN bit of RDCM_CS register to 0
3. Clear CHEN bit of DMA_CHxCTL to 0 with reinit DMA module
4. Reset the result data FIFO (FIFOFH = 1) or read all the left results out.
5. Clear WERRF bit of RDCM_FIFOCS register to 0.
6. Clear OVIF bit of RDCM_CS register to 0
7. Configure the RDCM
8. Set DMAEN bit of RDCM_CTL register to 1
9. Set RDCMEN bit of RDCM_CS register to 1

39.4.15. Interrupts

The interrupt can be generated on one of the events:

- Result data valid.
- DMA transfer overflow.
- Timestamp counter capture.
- I channel zero crossing detection.
- Q channel zero crossing detection.

- I channel envelope signal threshold match.
- Q channel envelope signal threshold match.
- Monitor signal threshold match.
- Track error signal threshold match.

Separate interrupt enable bits, configured in RDCM_CS register, are available for flexibility.

39.4.16. Simulation mode

When users need to debug certain configuration parameters within the RDCM, such as filter parameters or tracking loop parameters, they can use virtual data for simulation without the need for an external resolver. By setting the DISEL bit to 0 in the RDCM_DISWSEL register, users can input virtual data into the I/Q channels. This virtual data is stored in I/QSIMDI[13:0] of RDCM_SIMDI register and can be transferred there by DMA.

Note: When using virtual data for simulation, the data input rate depends on the update rate of the virtual data rather than the sampling rate of the ADC.

39.5. Registers definition

RDCM base address: 0x4001 6400

39.5.1. Control and status register (RDCM_CS)

Address offset: 0x00

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESDIF	OVIF	TSCAPIF	IZCDIF	QZCDIF	IETHMIF	QETHF	MTHMIF	TRKERRTHMIF	IETHMS[1:0]		QETHMS[1:0]		MTHMS[1:0]		Reserved
rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rw		rw		rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESDIE	OVIE	TSCAPIE	IZCDIE	QZCDIE	IETHMIE	QETHMIE	MTHIE	TRKERRTHMIE	TRKERRTHMS[1:0]		GATEVSEL[1:0]		CALSEL[1:0]		RDCMEN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw		rw		rw

Bits	Fields	Descriptions
31	RESDIF	Result data interrupt flag. If FIFOEN = 0 0: Result data register not update 1: Result data register update Cleared by software writing 0 to it or reading the data register selected by RESDSEL[2:0] in RDCM_CTL register. If FIFOEN = 1 0: Result FIFO level not reach the level configured by FIFOESEL[1:0] 1: Result FIFO level reach the level configured by FIFOESEL[1:0]
30	OVIF	DMA transfer data register overflow interrupt flag If FIFOEN = 0 0: DMA transfer data register not overflow 1: DMA transfer data register overflow If FIFOEN =1 0: No result FIFO write error occurred 1: Result FIFO write error occurred This bit is set by hardware when the DMA transfer data register are overflow. This flag is only set when DMA is enabled. The recent data is lost when this bit is set.
29	TSCAPIF	Timestamp counter capture interrupt flag. 0: Timestamp counter capture not occurred 1: Timestamp counter capture occurred.
28	IZCDIF	I channel zero crossing detection interrupt flag.

		0: I channel zero crossing detection not occurred 1: I channel zero crossing detection occurred
27	QZCDIF	Q channel zero crossing detection interrupt flag. 0: Q channel zero crossing detection not occurred 1: Q channel zero crossing detection occurred
26	IETHMIF	I channel envelope signal threshold match interrupt flag. 0: I channel envelope signal threshold not match 1: I channel envelope signal threshold match
25	QETHMIF	Q channel envelope signal threshold match interrupt flag. 0: Q channel envelope signal threshold not match 1: Q channel envelope signal threshold match
24	MTHMIF	Monitor signal threshold match interrupt flag. 0: Monitor signal threshold not match 1: Monitor signal threshold match
23	TRKERRTHMIF	Track error signal threshold match interrupt flag. 0: Track error signal threshold not match 1: Track error signal threshold match
22:21	IETHMS[1:0]	I channel envelope signal threshold mode selection 00: IETHL[14:0] < I channel envelope signal value < IETHU[14:0]. 01: I channel envelope signal value < IETHL[14:0] or IETHU[14:0] < I channel envelope signal value 10: IETHL[14:0] < I channel envelope signal value 11: IETHL[14:0] > I channel envelope signal value
20:19	QETHMS[1:0]	Q channel envelope signal threshold mode selection 00: QETHL[14:0] < Q channel envelope signal value < QETHU[14:0]. 01: Q channel envelope signal value < QETHL[14:0] or QETHU[14:0] < Q channel envelope signal value 10: QETHL[14:0] <Q channel envelope signal value 11: QETHL[14:0] > Q channel envelope signal value
18:17	MTHMS[1:0]	Monitor signal mode selection 00: MTHL[14:0] < monitor signal value < MTHU[14:0]. 01: monitor signal value < MTHL[14:0] or MTHU[14:0] < monitor signal value 10: MTHL[14:0] > monitor signal vibration value 11: MTHL[14:0] < monitor signal vibration value
16	Reserved	Must be kept at reset value.
15	RESDIE	Interrupt enable for RESDIF 0: Interrupt disable 1: Interrupt enable
14	OVIE	Interrupt enable for OVIF

		0: Interrupt disable 1: Interrupt enable
13	TSCAPIE	Interrupt enable for TSCAPIF 0: Interrupt disable 1: Interrupt enable
12	IZCDIE	Interrupt enable for IZCDIF 0: Interrupt disable 1: Interrupt enable
11	QZCDIE	Interrupt enable for QZCDIF 0: Interrupt disable 1: Interrupt enable
10	IETHMIE	Interrupt enable for IETHMIF 0: Interrupt disable 1: Interrupt enable
9	QETHMIE	Interrupt enable for QETHMIF 0: Interrupt disable 1: Interrupt enable
8	MTHIE	Interrupt enable for MTHMIF 0: Interrupt disable 1: Interrupt enable
7	TRKERRTHMIE	Interrupt enable for TRKERRTHMIF 0: Interrupt disable 1: Interrupt enable
6:5	TRKERRTHMS	Tracking error signal threshold mode selection. 00: TRKERRTHL[14:0] < Tracking error signal value < TRKERRTHU[14:0]. 01: Tracking error signal value < TRKERRTHL[14:0] or TRKERRTHU[14:0] < Tracking error signal value 10: TRKERRTHL[14:0] < Tracking error signal value 11: TRKERRTHL[14:0] > Tracking error signal value
4:3	GATEVSEL[1:0]	Gate signal (timestamp trigger) valid selection 00: Always valid 01: Valid level is high 10: Valid level is low 11: Reserved
2:1	CALSEL[1:0]	RDCM main chain calculation event selection. 00: ADC0 end of conversion event. 01: ADC1 end of conversion event. 10: ADC0 or ADC1 end of conversion event. 11: External trigger from TRIGSEL for RDC main chain calculation (high-pass filter,

demodulator, tracking loop).

0	RDCMEN	RDCM enable. 0: Stop RDCM 1: RDCM is enabled and runs

39.5.2. Control register (RDCM_CTL)

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
DMAEN	DDSM	RESDSEL[2:0]			TRKEN	EXCEN	NCORST	NCOETM[1:0]		Reserved	QZCDMS[1:0]		QDEMRST	QDEMEN	QRFMEN
rw	rw	rw			rw	rw	w	rw			rw		w	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVDEN	FIFOEN	TSCAPEN	IETHMEN	QETHMEN	MTHMEN	TRKERRTHMEN	TSCAPETM[1:0]		CALETM[1:0]		IZCDMS[1:0]		IDEMRST	IDEMEN	IRFMEN
rw	rw	rw	rw	rw	rw	rw	rw		rw		rw		w	rw	rw

Bits	Fields	Descriptions
31	DMAEN	DMA request enable 0: DMA request is disabled 1: DMA request is enabled
30	DDSM	DMA disable mode 0: The DMA engine is disabled after the end of transfer signal from DMA controller is detected. 1: When DMAEN=1, the DMA issues a request according to RESDIF bit.
29:27	RESDSEL[2:0]	Result data register selection. 000: RDCM_TRKDATA 001: RDCM_HFDATA 010: RDCM_RFMDATA 011: RDCM_DEMDATA 100: RDCM_MONDATA 101: RDCM_POSDATA 110: RDCM_VELDATA Others: Reserved
26	TRKEN	Track loop enable 0: Track loop is disabled 1: Track loop is enabled and runs
25	EXCEN	Sinusoidal exciter enable

		0: Sinusoidal exciter is disabled 1: Sinusoidal exciter is enabled and runs
24	NCORST	Reset the phase of all NCOs (Exciter and I/Q reference frequency mixer) to the initial setting phase It is set by software and cleared by hardware immediately after the phase resets.
23:22	NCOETM[1:0]	External trigger mode for all NCOs (Excitor, I/Q Reference Frequency Mixer) calculation. 00: Calculation according to the NCO division of CK_RDCM. 01: Rising edge of external trigger for all NCOs calculation. 10: Falling edge of external trigger for all NCOs calculation. 11: Rising and falling edge of external trigger for all NCOs calculation.
21	Reserved	Must be kept at reset value.
20:19	QZCDMS[1:0]	Zero crossing detection mode selection in Q channel 00: Zero crossing detection is disabled. (Manual reference frequency mixer phase adjust) 01: Positive zero crossing detection mode. (Only for calibration) 10: Negative zero crossing detection mode. (Only for calibration) 11: Both positive and negative zero crossing detection mode. (Automatic reference frequency mixer phase adjust)
18	QDEMRST	Q channel demodulator reset Every non-recursive filter element of the Q channel demodulator (CIC3, FIR0, FIR1) is cleared to zero and the related decimation counters are loaded with their start value. It is set by software and cleared by hardware immediately after the demodulator is reset.
17	QDEMEN	Q channel demodulator enable 0: Q channel demodulator is disabled 1: Q channel demodulator is enabled and runs
16	QRFMEN	Q channel reference frequency mixer enable 0: Q channel reference frequency mixer is disabled 1: Q channel reference frequency mixer is enabled and runs.
15	OVDEN	DMA transfer overflow detection enable. 0: DMA transfer overflow detection disable 1: DMA transfer overflow detection enable
14	FIFOEN	Result data FIFO enable. 0: Result data FIFO is disabled 1: Result data FIFO is enabled
13	TSCAPEN	Timestamp counter capture enable. 0: Timestamp counter capture function is disabled

		1: Timestamp counter capture function is enabled
12	IETHMEN	I channel envelope signal threshold match enable. 0: I channel envelope signal threshold match function is disabled 1: I channel envelope signal threshold match function is enabled
11	QETHMEN	Q channel envelope signal threshold match enable. 0: Q channel envelope signal threshold match function is disabled 1: Q channel envelope signal threshold match function is enabled
10	MTHMEN	Monitor signal threshold match enable. 0: Monitor signal threshold match function is disabled 1: Monitor signal threshold match function is enabled
9	TRKERRTHMEN	Track error signal threshold match enable. 0: Track error signal threshold match function is disabled 1: Track error signal threshold match function is enabled
8:7	TSCAPETM[1:0]	External trigger mode for timestamp counter capture. 00: Disable trigger for timestamp counter capture. 01: Rising edge of external trigger for timestamp counter capture. 10: Falling edge of external trigger for timestamp counter capture. 11: Rising and falling edge of external trigger for timestamp counter capture.
6:5	CALETM[1:0]	External trigger mode for RDCM main chain calculation (high-pass filter, demodulator, tracking loop). 00: Disable trigger for RDCM main chain calculation. 01: Rising edge of external trigger for RDCM main chain calculation. 10: Falling edge of external trigger for RDCM main chain calculation. 11: Rising and falling edge of external trigger for RDCM main chain calculation.
4:3	IZCDMS[1:0]	Zero crossing detection mode selection in I channel 00: Zero crossing detection is disabled. (Manual reference frequency mixer phase adjust) 01: Positive zero crossing detection mode. (Only for calibration) 10: Negative zero crossing detection mode. (Only for calibration) 11: Both positive and negative zero crossing detection mode. (Automatic reference frequency mixer phase adjust)
2	IDEMRST	I channel demodulator reset Every non-recursive filter element of the I channel demodulator (CIC3, FIR0, FIR1) is cleared to zero and the related decimation counters are loaded with their start value. It is set by software and cleared by hardware immediately after the demodulator is reset.
1	IDEMEN	I channel demodulator enable 0: I channel demodulator is disabled 1: I channel demodulator is enabled and runs

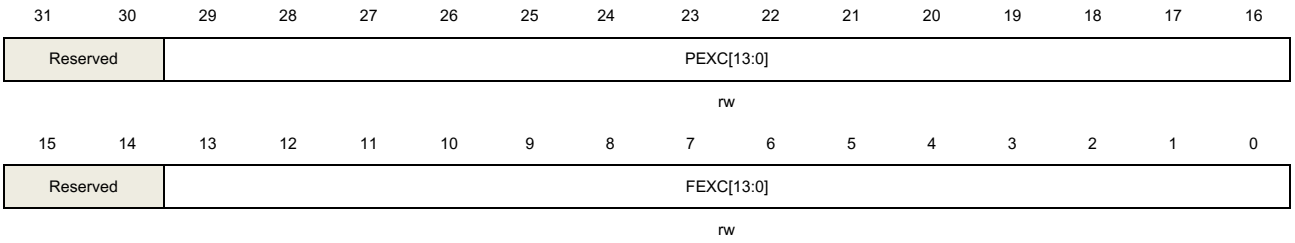
0	IRFMEN	I channel reference frequency mixer enable 0: I channel reference frequency mixer is disabled 1: I channel reference frequency mixer is enabled and runs

39.5.3. Exciter configuration register 0 (RDCM_EXCFG0)

Address offset: 0x08

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)



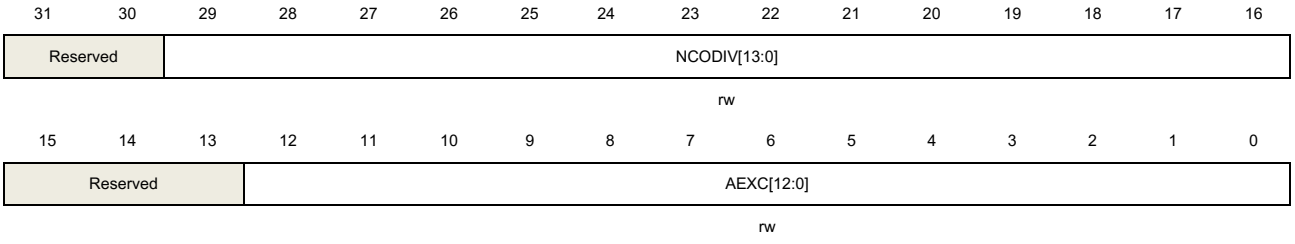
Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29:16	PEXC[13:0]	Initial phase of the exciter The initial phase of the sinusoidal wave is 2 * pi * PEXC[13:0] / 16384.
15:14	Reserved	Must be kept at reset value.
13:0	FEXC[13:0]	Frequency of the exciter The exciter frequency is calculation frequency * FEXC[13:0] / 16384.

39.5.4. Exciter configuration register 1 (RDCM_EXCFG1)

Address offset: 0x0C

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.
29:16	NCODIV[13:0]	NCO(Excitror,I/Q reference frequency mixer) calculation frequency division factor

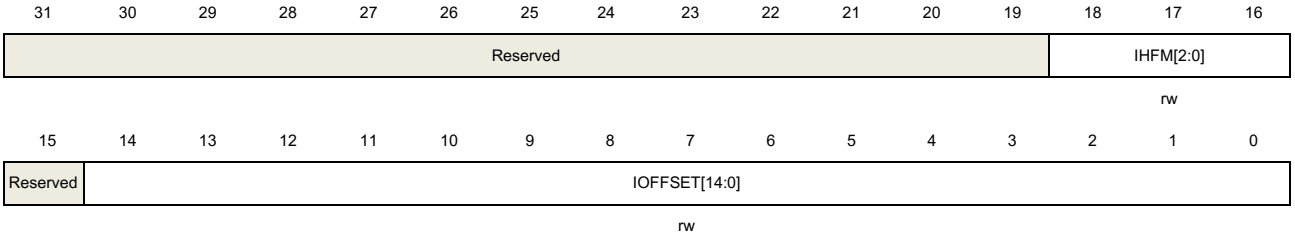
		The NCO(Excitror,I/Q reference frequency mixer) calculation frequency will be divided by (NCODIV[13:0] + 1).
15:13	Reserved	Must be kept at reset value.
12:0	AEXC[12:0]	Amplitude of the exciter The range of sinusoidal wave is 0~AEXC[12:0] / 2 to DACs.

39.5.5. High pass filter configuration register 0 (RDCM_HFCFG0)

Address offset: 0x10

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)



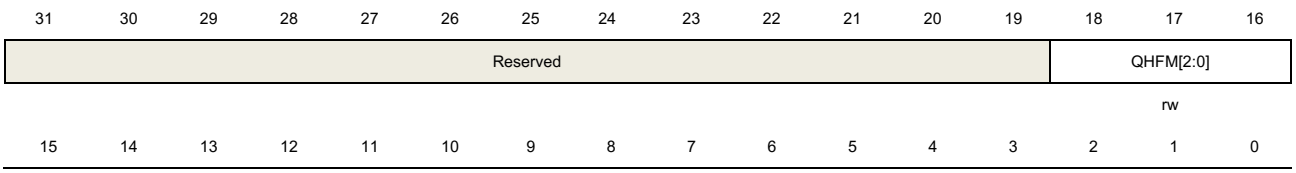
Bits	Fields	Descriptions
31:19	Reserved	Must be kept at reset value.
18:16	IHFM[2:0]	I channel high-pass (offset compensation) filter mode 000: Offset compensation filter disabled 001: Enable offset compensation filter, set cutoff frequency to rate 1 … 111: Enable offset compensation filter, set cutoff frequency to rate 7
15	Reserved	Must be kept at reset value.
14:0	IOFFSET[14:0]	I channel offset factor The signed value of I channel data will be subtracted by IOFFSET (IHFM[2:0] = 000). The data is stored in a 15-bit two's complement format.

39.5.6. High pass filter configuration register 1 (RDCM_HFCFG1)

Address offset: 0x14

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)



Reserved	QOFFSET[14:0]
	rw

Bits	Fields	Descriptions
31:19	Reserved	Must be kept at reset value.
18:16	QHFM[2:0]	Q channel high-pass (offset compensation) filter mode 000: Offset compensation filter disabled, register not changed 001: Enable offset compensation filter, set cutoff frequency to rate 1 … 111: Enable offset compensation filter, set cutoff frequency to rate 7
15	Reserved	Must be kept at reset value.
14:0	QOFFSET[14:0]	Q channel offset factor The signed value of I channel data will be subtracted by QOFFSET (QHFM[2:0] = 000). The data is stored in a 15-bit two's complement format.

39.5.7. Reference frequency mixer configuration register 0 (RDCM_RFMCFG0)

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		IFRE[13:0]													
		rw													

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		IPHASE[13:0]													
		rw													

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29:16	IFRE[13:0]	The frequency control word of the I channel reference frequency mixer. The frequency of the sinusoidal wave is calculation frequency multiplied by IFRE[13:0] / 16384.
15:14	Reserved	Must be kept at reset value.
13:0	IPHASE[13:0]	Initial phase of the I channel reference frequency mixer. The initial phase of the sinusoidal wave is 2 * pi * IPHASE[13:0] / 16384.

39.5.8. Reference frequency mixer configuration register 1 (RDCM_RFMCFG1)

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		QFRE[13:0]													
		rw													

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		QPHASE[13:0]													
		rw													

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29:16	QFRE[13:0]	The frequency control word of the Q channel reference frequency mixer. The frequency of the sinusoidal wave is calculation frequency multiplied by QFRE[13:0] / 16384.
15:14	Reserved	Must be kept at reset value.
13:0	QPHASE[13:0]	Initial phase of the Q channel reference frequency mixer. The initial phase of the sinusoidal wave is 2 * pi * QPHASE / 16384.

39.5.9. Zero crossing threshold configuration register 0 (RDCM_ZTCFG0)

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		QZCTH[13:0]													
		Rw													

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		IZCTH[13:0]													
		rw													

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29:16	QZCTH[13:0]	The threshold of mismatch counter value in zero crossing detection of Q channel.
15:14	Reserved	Must be kept at reset value.
13:0	IZCTH[13:0]	The threshold of mismatch counter value in zero crossing detection of I channel.

39.5.10. Zero crossing threshold configuration register 1 (RDCM_ZTCFG1)

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		QZTVAR[13:0]													
		rw													

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		IZTVAR[13:0]													
		rw													

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29:16	QZTVAR[13:0]	The threshold variation range of mismatch counter value in zero crossing detection of Q channel.
15:14	Reserved	Must be kept at reset value.
13:0	IZTVAR[13:0]	The threshold variation range of mismatch counter value in zero crossing detection of I channel.

39.5.11. Mismatch counter average configuration register (RDCM_MCACFG)

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												QAVEM[3:0]			
												Rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												IAVEM[3:0]			
												rw			

Bits	Fields	Descriptions
31:20	Reserved	Must be kept at reset value.
19:16	QAVEM[3:0]	The mismatch counter average factor of Q channel. 0000: Mismatch counter average function is disabled, and the mismatch counter value that captured will update the phase adjustment of the reference frequency mixer each time. Others: Mismatch counter average function is enabled. The average strength is larger for a larger QAVEM[3:0] value, but the time of convergence is longer.
15:4	Reserved	Must be kept at reset value.
3:0	IAVEM[3:0]	The mismatch counter average factor of I channel.

0000: Mismatch counter average function is disabled, and the mismatch counter value that captured will update the phase adjustment of the reference frequency mixer each time.
Others: Mismatch counter average function is enabled. The average strength is larger for a larger IAVEM[3:0] value, but the time of convergence is longer.

39.5.12. Demodulator configuration register (RDCM_DEMCFG)

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												QCIC3EN	QFIR1DEC	QFIR1EN	QFIR0EN
												rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												ICIC3EN	IFIR1DEC	IFIR1EN	IFIR0EN
												rw	rw	rw	rw

Bits	Fields	Descriptions
31:20	Reserved	Must be kept at reset value.
19	QCIC3EN	Q channel CIC3 filter enable 0: CIC3 is disabled 1: CIC3 is enabled
18	QFIR1DEC	Q channel FIR1 filter decimation rate 0: Decimation rate 2 for FIR1 1: FIR1 does not decimate
17	QFIR1EN	Q channel FIR1 filter enable 0: FIR1 is disabled and bypassed 1: FIR1 is enabled
16	QFIR0EN	Q channel FIR0 filter enable 0: FIR0 is disabled and bypassed 1: FIR0 is enabled
15:4	Reserved	Must be kept at reset value.
3	ICIC3EN	I channel CIC3 filter enable 0: CIC3 is disabled 1: CIC3 is enabled
2	IFIR1DEC	I channel FIR1 filter decimation rate 0: Decimation rate 2 for FIR1

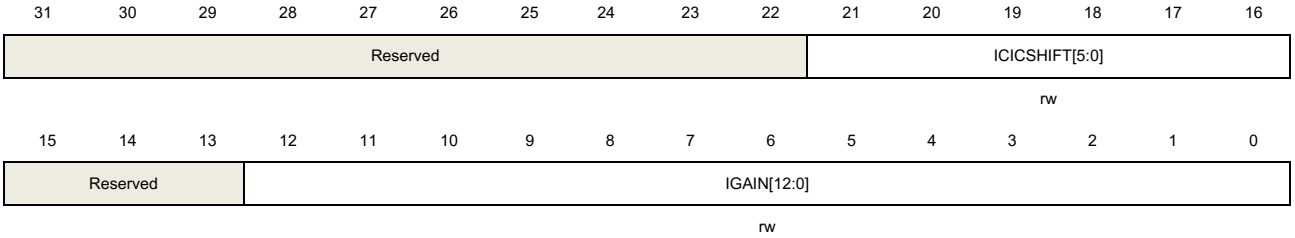
		1: FIR1 does not decimate
1	IFIR1EN	I channel FIR1 filter enable 0: FIR1 is disabled and bypassed 1: FIR1 is enabled
0	IFIR0EN	I channel FIR0 filter enable 0: FIR0 is disabled and bypassed 1: FIR0 is enabled

39.5.13. CIC3 shift and gain configuration register 0 (RDCM_CSGCFG0)

Address offset: 0x30

Reset value: 0x0000 1000

This register has to be accessed by word (32-bit)



Bits	Fields	Descriptions
31:22	Reserved	Must be kept at reset value.
21:16	ICICSHIFT[5:0]	I channel output shift If I channel demoluator is enabled (IDEMEN = 1): During calibration, these bits selects the valid output bits from the CIC3 filter, depending on the chosen decimation factor. 000000: Use output bits [14:0] 000001: Use output bits [15:1] … 101001: Use output bits [55:41] Others: Reserved. If I channel demoluator is disabled (IDEMEN = 0): 000000: Use output bits [14:0] 000001: Use output bits [15:1] … 001110: Use output bits [28:14] Others: Reserved.
15:13	Reserved	Must be kept at reset value.
12:0	IGAIN[12:0]	I channel gain factor I channel multiplication factor for gain correction during calibration. The resulting

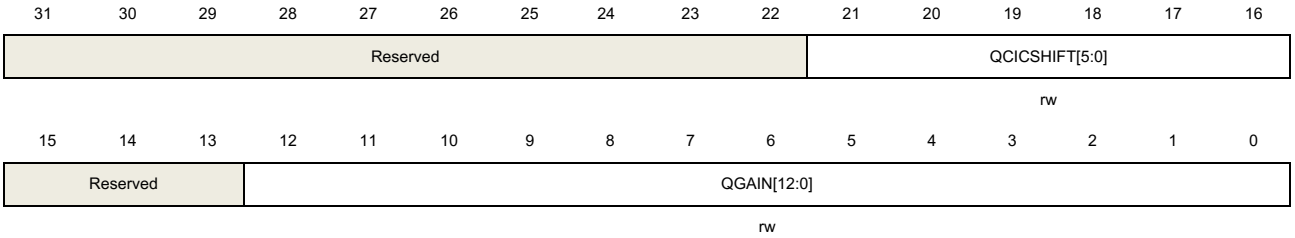
factor is IGAIN[12:0] / 4096.

39.5.14. CIC3 shift and gain configuration register 1 (RDCM_CSGCFG1)

Address offset: 0x34

Reset value: 0x0000 1000

This register has to be accessed by word (32-bit)



Bits	Fields	Descriptions
31:22	Reserved	Must be kept at reset value.
21:16	QCICSHIFT[5:0]	Q channel output shift If Q channel demoluator is enabled (QDEMEN = 1): During calibration, these bits selects the valid output bits from the CIC3 filter, depending on the chosen decimation factor. 000000: Use output bits [14:0] 000001: Use output bits [15:1] … 101001: Use output bits [55:41] Others: Reserved. If Q channel demoluator is disabled (QDEMEN = 0): 000000: Use output bits [14:0] 000001: Use output bits [15:1] … 001110: Use output bits [28:14] Others: Reserved.
15:13	Reserved	Must be kept at reset value.
12:0	QGAIN[12:0]	Q channel gain factor Q channel multiplication factor for gain correction during calibration. The resulting factor is QGAIN[12:0] / 4096.

39.5.15. CIC3 configuration register 0 (RDCM_CICCFG0)

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							ICICSVAL[7:0]								
							rw								

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							ICICDFAC[8:0]								
							rw								

Bits	Fields	Descriptions
31:25	Reserved	Must be kept at reset value.
24:16	ICICSVAL[8:0]	I channel CIC filter start value The decimation counter begins counting at value ICICSVAL[8:0], when started or restarted. Valid values are 003H to ICICDFAC[8:0] (4 to selected IDCR)
15:9	Reserved	Must be kept at reset value.
8:0	ICICDFAC[8:0]	I channel CIC filter decimation factor These bits define the decimation rate of the CIC filter: IDCR = ICICDFAC[8:0] + 1. Valid values are 003H to 1FFH (IDCR = 4 to 512).

39.5.16. CIC3 configuration register 1 (RDCM_CICCFG1)

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							QCICSVAL[7:0]								
							rw								

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							QCICDFAC[8:0]								
							rw								

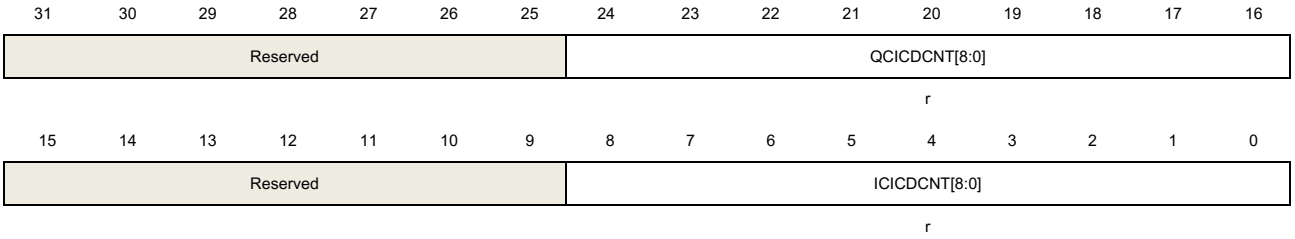
Bits	Fields	Descriptions
31:25	Reserved	Must be kept at reset value.
24:16	QCICSVAL[8:0]	Q channel CIC filter start value The decimation counter begins counting at value QCICSVAL[8:0], when started or restarted. Valid values are 003H to QCICDFAC[8:0] (4 to selected QDCR)
15:9	Reserved	Must be kept at reset value.
8:0	QCICDFAC[8:0]	Q channel CIC filter decimation factor

These bits define the decimation rate of the CIC filter: QDCR = QCICDFAC[8:0] + 1. Valid values are 003H to 1FFH (QDCR = 4 to 512).

39.5.17. CIC3 decimation counter register (RDCM_CDCNT)

Address offset: 0x40
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)

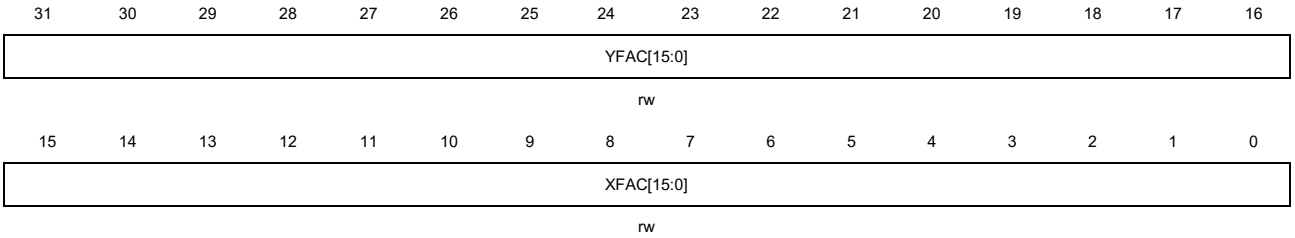


Bits	Fields	Descriptions
31:25	Reserved	Must be kept at reset value.
24:16	QCICDCNT[8:0]	Q channel CIC3 filter decimation counter The decimation counter counts the filter cycles until an output is generated.
15:9	Reserved	Must be kept at reset value.
8:0	ICICDCNT[8:0]	I channel CIC3 Filter decimation counter The decimation counter counts the filter cycles until an output is generated.

39.5.18. Nonorthogonal phase mismatch configuration register (RDCM_NPMCFG)

Address offset: 0x44
Reset value: 0x0000 4000

This register has to be accessed by word (32-bit)



Bits	Fields	Descriptions
24:16	YFAC[15:0]	Q channel nonorthogonal phase mismatch compensation factor Y. The YFAC[15:0] should be configured as tan(σ) * 16384.

∑ is the nonorthogonal phase mismatch relative to I channel.

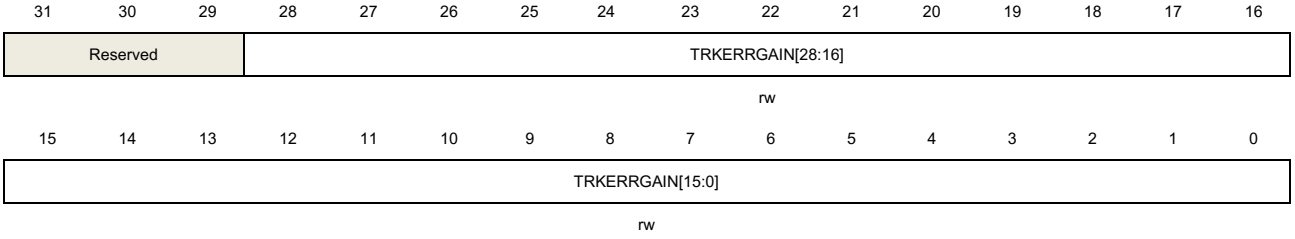
15:0	XFAC[15:0]	Q channel nonorthogonal phase mismatch compensation factor X. The XFAC[15:0] should be configured as 1 / cos(σ) * 16384. ∑ is the nonorthogonal phase mismatch relative to I channel.

39.5.19. Tracking gain configuration register (RDCM_TRKGAINCFG)

Address offset: 0x48

Reset value: 0x1394 FF16

This register has to be accessed by word (32-bit)



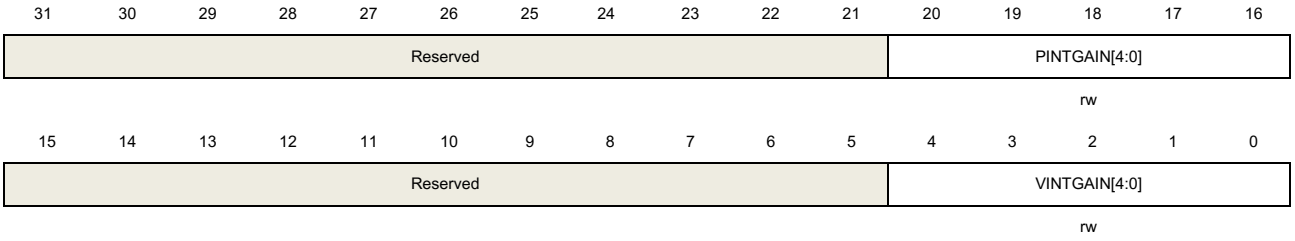
Bits	Fields	Descriptions
31:29	Reserved	Must be kept at reset value.
28:0	TRKERRGAIN[28:0]	The error acceleration gain factor.

39.5.20. Tracking integrator configuration register (RDCM_TRKINTCFG)

Address offset: 0x4C

Reset value: 0x0018 0018

This register has to be accessed by word (32-bit)



Bits	Fields	Descriptions
31:21	Reserved	Must be kept at reset value.
20:16	PINTGAIN[4:0]	The position gain factor of integrator The position gain factor is 1 / ($2^{PINTGIN[4:0]}$)
15:5	Reserved	Must be kept at reset value.
4:0	VINTGAIN[4:0]	The velocity gain factor of integrator

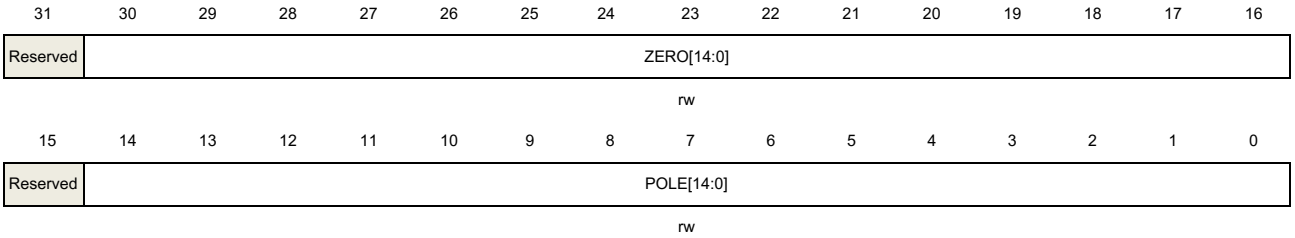
The velocity gain factor is 1 / ($2^{VINTGAIN[4:0]}$)

39.5.21. Tracking compensation configuration register (RDCM_TRKCMPCFG)

Address offset: 0x50

Reset value: 0x3FFE 3FE7

This register has to be accessed by word (32-bit)



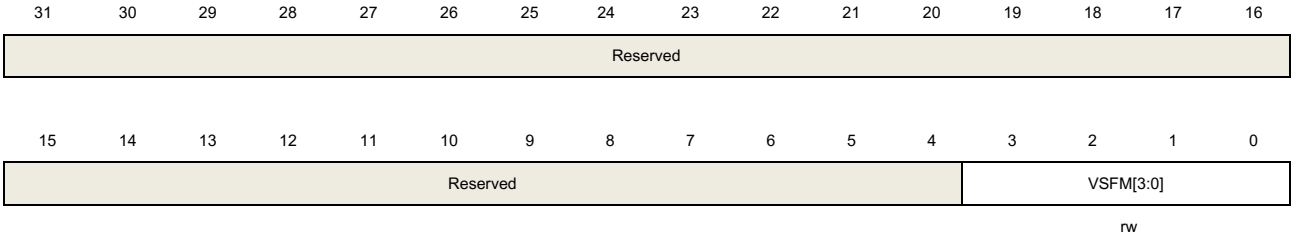
Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30:16	ZERO[14:0]	The zero of compensation filter.
15	Reserved	Must be kept at reset value.
14:0	POLE[14:0]	The pole of compensation filter.

39.5.22. Velocity smooth filter configuration register (RDCM_VSFCFG)

Address offset: 0x54

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)



Bits	Fields	Descriptions
31:19	Reserved	Must be kept at reset value.
3:0	VSFM[3:0]	The velocity smooth filter mode in tracking loop. 0000: Smooth filter is disabled and the velocity is extracted from the compensation filter. Others: Smooth filter is enabled. The smooth filter strength is larger for a larger VSFM[3:0] value, but the time of convergence is longer.

39.5.23. High pass filter data register (RDCM_HFDATA)

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
QHFDATA[15:0]															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IHFDATA[15:0]															
r															

Bits	Fields	Descriptions
31:16	QHFDATA[15:0]	Data output from Q channel high pass filter. The data is stored in a 16-bit two's complement format.
15:0	IHFDATA[15:0]	Data output from I channel high pass filter. The data is stored in a 16-bit two's complement format.

39.5.24. Reference frequency mixer data register (RDCM_RFMDATA)

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
QRFMDATA[15:0]															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IRFMDATA[15:0]															
r															

Bits	Fields	Descriptions
31:16	QRFMDATA[15:0]	Data output from Q channel reference frequency mixer. The data is stored in a 16-bit two's complement format.
15:0	IRFMDATA[15:0]	Data output from I channel reference frequency mixer. The data is stored in a 16-bit two's complement format.

39.5.25. Demodulator data register (RDCM_DEMDATA)

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
QDEMDATA[15:0]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDEMDATA[15:0]															
r															

Bits	Fields	Descriptions
31:16	QDEMDATA[15:0]	Data output from Q channel phase sensitive demodulator. The data is stored in a 16-bit two's complement format.
15:0	IDEMDATA[15:0]	Data output from I channel phase sensitive demodulator. The data is stored in a 16-bit two's complement format.

39.5.26. Tracking data register (RDCM_TRKDATA)

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
POSITION[15:0]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VELOCITY[15:0]															
r															

Bits	Fields	Descriptions
31:16	POSITION[15:0]	Position data The data is stored in a 16-bit two's complement format.
15:0	VELOCITY[15:0]	Velocity data The data is stored in a 16-bit two's complement format.

39.5.27. Monitor data register (RDCM_MONDATA)

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
MONDATA[15:0]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRKERRDATA[15:0]															
r															

Bits	Fields	Descriptions
31:16	MONDATA[15:0]	Monitor signal data The data is stored in a 16-bit two's complement format.
15:0	TRKERRDATA[15:0]	Tracking error signal data The data is stored in a 16-bit two's complement format.

39.5.28. Timestamp control register (RDCM_TSCTL)

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														TSCLK[1:0]	
														rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSCNT[15:0]															
r															

Bits	Fields	Descriptions
31:18	Reserved	Must be kept at reset value.
17:16	TSCLK[1:0]	Timestamp counter clock selection 00: $f_{TS} = f_{RDCM}$ 01: $f_{TS} = f_{RDCM} / 2$ 10: $f_{TS} = f_{RDCM} / 4$ 11: $f_{TS} = f_{RDCM} / 8$
15:0	TSCNT[15:0]	Timestamp counter value TSCNT[15:0] is clocked with the RDCM clock and is cleared when a new result value has been generated.

39.5.29. Timestamp data register (RDCM_TSDATA)

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
TSTRGF	Reserved														

rc_w0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSVAL[15:0]															
r															

Bits	Fields	Descriptions
31	TSTRGF	Timestamp trigger flag 0: No timestamp trigger occurred since last read access 1: Timestamp information has been stored after a timestamp trigger. Cleared by software writing 0 to it or by reading the RDCM_TSDATA register.
30:16	Reserved	Must be kept at reset value.
15:0	TSVAL[15:0]	The most recent captured timestamp value This value is copied from TSCNT[15:0] in RDCM_TSCTL register.

39.5.30. FIFO control and status register (RDCM_FIFOCS)

Address offset: 0x74

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											WERRF	RERRF	FIFONUM[2:0]		
											rc_w0	rc_w0	r		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													FIFOFH	FIFOESEL[1:0]	
													w	rw	

Bits	Fields	Descriptions
31:21	Reserved	Must be kept at reset value.
20	WERRF	Write error flag. 0: No problem encountered 1: A write access error occurred while the FIFO was full. Set by hardware and cleared by writing 0 to it.
19	RERRF	Read error flag. 0: No problem encountered 1: A read access error occurred while the FIFO was empty Set by hardware and cleared by writing 0 to it.
18:16	FIFONUM[2:0]	FIFO valid data number 000: FIFO is empty 001: FIFO contains 1 valid value 010: FIFO contains 2 valid values

		011: FIFO contains 3 valid values 100: FIFO contains 4 valid values Others: Reserved
15:3	Reserved	Must be kept at reset value.
2	FIFOFH	FIFO flush 0: No action 1: Remove all values from result FIFO
1:0	FIFOESEL[1:0]	FIFO result event selection 00: Generate result data event when FIFO contains 1 value 01: Generate result data event when FIFO contains 2 values 10: Generate result data event when FIFO contains 3 values 11: Generate result data event when FIFO contains 4 values

39.5.31. FIFO data register (RDCM_FIFODATA)

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
FIFODATA[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIFODATA[15:0]															
r															

Bits	Fields	Descriptions
31:0	FIFODATA[31:0]	FIFO data from result data register

39.5.32. Envelop threshold configuration register 0 (RDCM_ETHCFG0)

Address offset: 0x7C

Reset value: 0x3FFF 4000

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	IETHU[14:0]														
	rw														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	IETHL[14:0]														
	rw														

Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30:16	IETHU[14:0]	Upper threshold for envelope signal of I channel. This value is compared to the upper bits of the envelope signal. The data is stored in a 15-bit two's complement format.
15	Reserved	Must be kept at reset value.
14:0	IETHL[14:0]	Lower threshold for envelope signal of I channel. This value is compared to the upper bits of the envelope signal. The data is stored in a 15-bit two's complement format.

39.5.33. Envelop threshold configuration register 1 (RDCM_ETHCFG1)

Address offset: 0x80
Reset value: 0x3FFF 4000

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								QETHU[14:0]							
								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								QETHL[14:0]							
								rw							

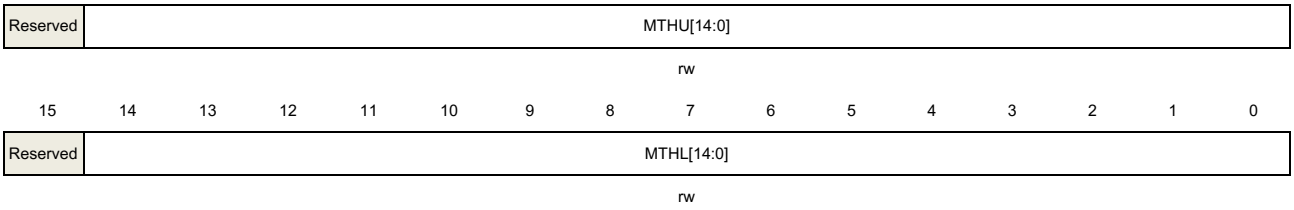
Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30:16	QETHU[14:0]	Upper threshold for envelope signal of Q channel. This value is compared to the upper bits of the envelope signal. The data is stored in a 15-bit two's complement format.
15	Reserved	Must be kept at reset value.
14:0	QETHL[14:0]	Lower threshold for envelope signal of Q channel. This value is compared to the upper bits of the envelope signal. The data is stored in a 15-bit two's complement format.

39.5.34. Monitor signal threshold configuration register (RDCM_MTHCFG)

Address offset: 0x84
Reset value: 0x3FFF 4000

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16



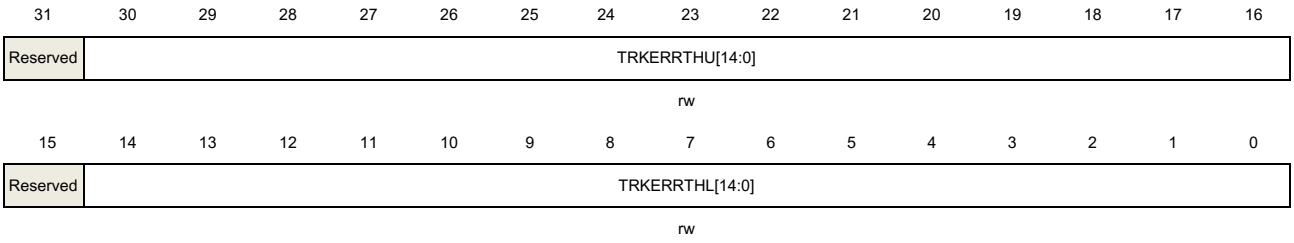
Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30:16	MTHU[14:0]	Upper threshold for monitor signal. This value is compared to the upper bits of the monitor signal. The data is stored in a 15-bit two's complement format.
15	Reserved	Must be kept at reset value.
14:0	MTHL[14:0]	Lower threshold for monitor signal. This value is compared to the upper bits of the monitor signal. The data is stored in a 15-bit two's complement format.

39.5.35. Tracking error signal threshold configuration register (RDCM_TRKERRTHCFG)

Address offset: 0x88

Reset value: 0x3FFF 4000

This register has to be accessed by word (32-bit)



Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30:16	TRKERRTHU[14:0]	Upper threshold for tracking error signal. This value is compared to the upper bits of the error signal. The data is stored in a 15-bit two's complement format.
15	Reserved	Must be kept at reset value.
14:0	TRKERRTHL[14:0]	Lower threshold for tracking error signal. This value is compared to the upper bits of the error signal. The data is stored in a 15-bit two's complement format.

39.5.36. Threshold match counter register (RDCM_THCNT)

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
Reserved		MTHCNT[13:0]													
		rw													

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		TRKERRTHCNT[13:0]													
		rw													

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29:16	MTHCNT[13:0]	The monitor signal threshold match counter value. The monitor signal threshold match event should only be valid after meeting the required number of times (MTHCNT[13:0] + 1) continuously.
15:14	Reserved	Must be kept at reset value.
13:0	TRKERRTHCNT[13:0]	The tracking error signal threshold match counter value. The tracking error threshold match event should only be valid after meeting the required number of times (TRKERRTHCNT[13:0] + 1) continuously.

39.5.37. Simulation data input register (RDCM_SIMDI)

Address offset: 0x100

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		QSIMDI[13:0]													
		rw													

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		ISIMDI[13:0]													
		rw													

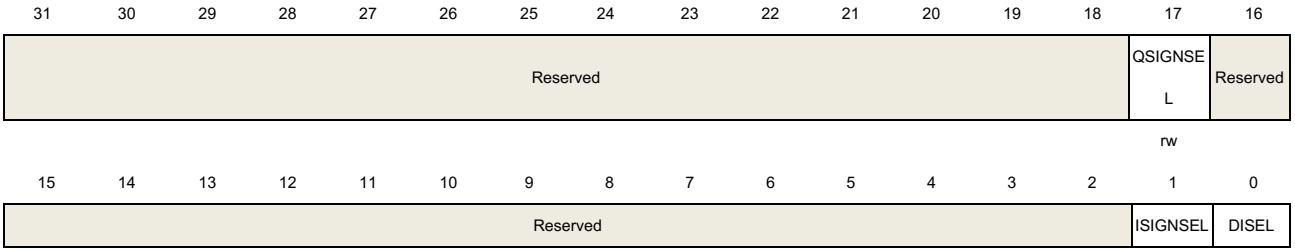
Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29:16	QSIMDI[13:0]	Simulation Q channel Data input.
15:14	Reserved	Must be kept at reset value.

13:0	ISIMDI[13:0]	Simulation I channel Data input.

39.5.38. Data input software selection register (RDCM_DISWSEL)

Address offset: 0x104
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)

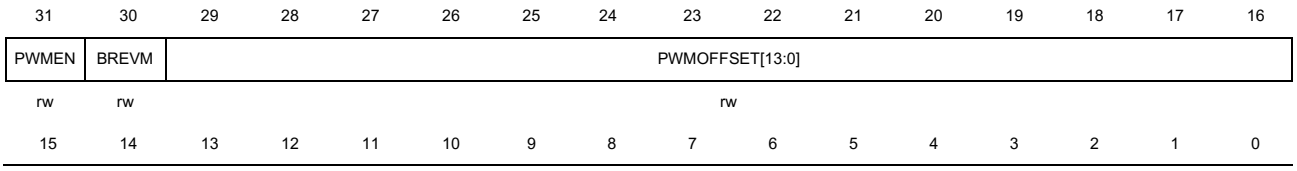


Bits	Fields	Descriptions
31:18	Reserved	Must be kept at reset value.
17	QSIGNSEL	Q channel RDCM frequency mixer reference sign source selection. 0: Use on-chip excitor sign as Q channel RDCM frequency mixer reference. 1: Use external excitor sign signal pin input as Q channel RDCM frequency mixer reference.
16:2	Reserved	Must be kept at reset value.
1	ISIGNSEL	I channel RDCM frequency mixer reference sign source selection. 0: Use on-chip excitor sign as I channel RDCM frequency mixer reference. 1: Use external excitor sign signal pin input as I channel RDCM frequency mixer reference.
0	DISEL	RDCM data input selection 0: RDCM I/Q channel calculation use ADC conversion data. 1: RDCM I/Q channel calculation use ISIMDI[13:0] and QSIMDI[13:0].

39.5.39. PWM configuration register (RDCM_PWMCFG)

Address offset: 0x108
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)



Reserved	PWMGAIN[13:0]
	rw

Bits	Fields	Descriptions
31	PWMEN	PWM waveform generator enable. 0: PWM waveform generator disabled. 1: PWM waveform generator enabled.
30	BREVM	Bit-reversed mode for PWM value. 0: Not bit-reversed. 1: Bit-reversed.
29:16	PWMOFFSET[13:0]	PWM offset for PWM value.
15:14	Reserved	Must be kept at reset value.
13:0	PWMGAIN[13:0]	PWM gain for PWM value.

39.5.40. Position data register (RDCM_POSDATA)

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															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POSITION[15:0]															
r															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	POSITION[15:0]	Position data.

39.5.41. Velocity data register (RDCM_VELDATA)

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
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

VELOCITY[15:0]

r

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	VELOCITY[15:0]	Velocity data.

40. Universal synchronous / asynchronous receiver / transmitter (USART)

40.1. 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_IRC64MDIV) 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.

40.2. 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 37.5 Mbits/s (applicable to GD32H77x) when the clock frequency is 300 MHz and oversampling is by 8.
- Programmable baud-rate generator allowing speed up to 43.75 Mbits/s (applicable to the general mode of GD32H78x) when the clock frequency is 350 MHz and oversampling is by 8.
- Programmable baud-rate generator allowing speed up to 46.875 Mbits/s (applicable to the high-performance mode of GD32H78x) when the clock frequency is 375 MHz and oversampling is by 8.
- Fully programmable serial interface characteristics:
  - A data word (7 / 8 / 9 or 10 bits) LSB or MSB first.
  - Even, odd or no-parity bit generation / detection.

  - 0.5, 1, 1.5 or 2 stop bit generation.
- Swappable Tx / Rx pin.
- Configurable data polarity.
- Hardware modem operations (CTS / RTS) and RS485 drive enable.
- Configurable multibuffer communication using centralized DMA.
- Separate enable bits for transmitter and receiver.
- Parity control:
  - Transmits parity bit.
  - Checks parity of received data byte.
- LIN break generation and detection.
- IrDA support.
- Synchronous mode and transmitter clock output for synchronous transmission.
- ISO 7816-3 compliant smartcard interface:
  - Character mode (T = 0).
  - Block mode (T = 1).
  - Direct and inverse convention.
- Multiprocessor communication:
  - Enter into mute mode if address match does not occur.
  - Wake up from mute mode by idle line or address match detection.
- Support for ModBus communication:
  - Timeout feature.
  - CR / LF character recognition.
- Wake up from deep-sleep mode:
  - By standard RBNE interrupt.
  - By WUF interrupt.
- Various status flags:
  - Flags for transfer detection: receive buffer not empty (RBNE), receive FIFO full (RFF), receive FIFO empty (RFE), receive FIFO threshold reached(RFT), receive FIFO not empty(RFNE), transmit buffer empty (TBE), transfer complete (TC), transmit FIFO not full (TFNF), transmit FIFO threshold reached (TFTIF), transmit FIFO empty (TFE), transmit FIFO full (TFF).
  - Flags for error detection: overrun error (ORERR), noise error (NERR), frame error (FERR) and parity error (PERR).
  - Flag for hardware flow control: CTS changes (CTSF).
  - Flag for LIN mode: LIN break detected (LBDF) and send break (SBF).
  - Flag for multiprocessor communication: IDLE frame detected (IDLEF).
  - Flag for ModBus communication: address/character match (AMF0 / AMF1) and receiver timeout (RTF).
  - Flags for smartcard block mode: end of block (EBF) and receiver timeout (RTF).
  - Wakeup from deep-sleep mode flag (WUF).
  - Receiver wakeup from mute mode(RWU).
  - Interrupt occurs at these events when the corresponding interrupt enable bits are set.

While USART0 / USART1 / USART2 / USART5 is fully implemented, UART3 / UART4 /

UART6 / UART7 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.

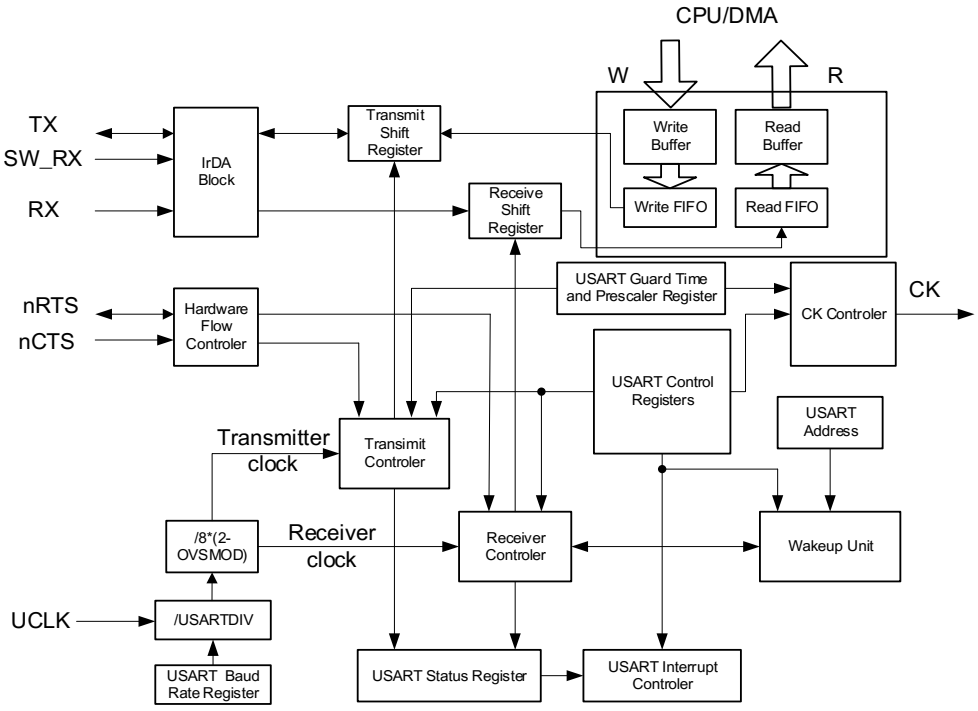
40.3. Function overview

The interface is externally connected to another device by the main pins listed in *Table 40-1. Description of USART important pins*.

Table 40-1. 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

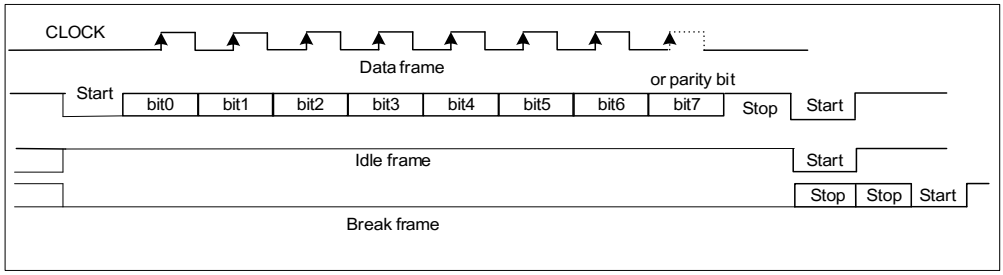
Figure 40-1. USART module block diagram



40.3.1. 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 40-2. 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 40-2. 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.

In an break frame, all the frame bits are logic 0 and 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.

40.3.2. 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:

$$\text{USARTDIV} = \frac{\text{UCLK}}{16 \times \text{Baud Rate}} \quad (40\text{-}1)$$

In case of oversampling by 8, the equation is:

$$\text{USARTDIV} = \frac{\text{UCLK}}{8 \times \text{Baud Rate}} \quad (40\text{-}2)$$

For example, when oversampled by 16:

1. Get USARTDIV by calculating 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.

40.3.3. 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.

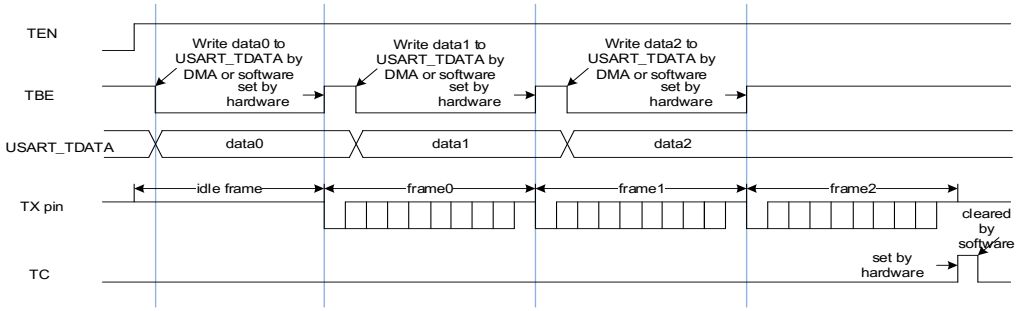
If a frame is transmitted and the TBE bit is asserted, the TC bit of the USART_STAT register will be set. An interrupt will be generated if the corresponding interrupt enable bit (TCIE) is set in the USART_CTL0 register.

The USART transmit procedure is shown in ***Figure 40-3. 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 40-3. 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.

40.3.4. 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.

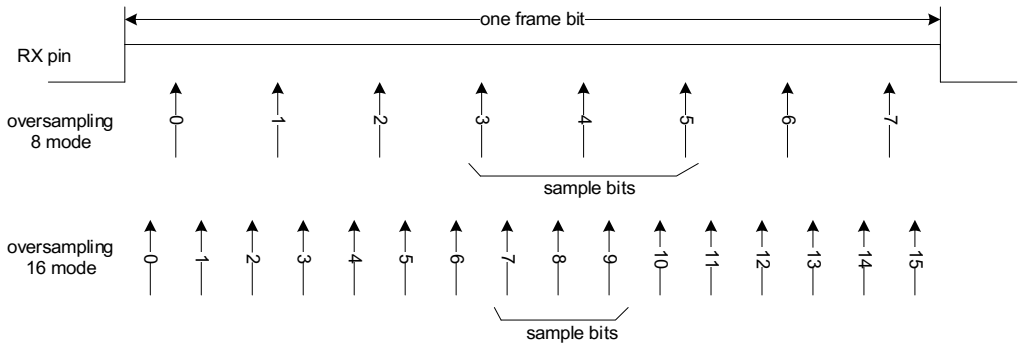
When a frame is received, the RBNE bit in USART_STAT is asserted, an interrupt is generated if the corresponding interrupt enable bit (RBNEIE) is set in the USART_CTL0 register. The status of the reception are stored in the USART_STAT register.

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 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 40-4. 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)

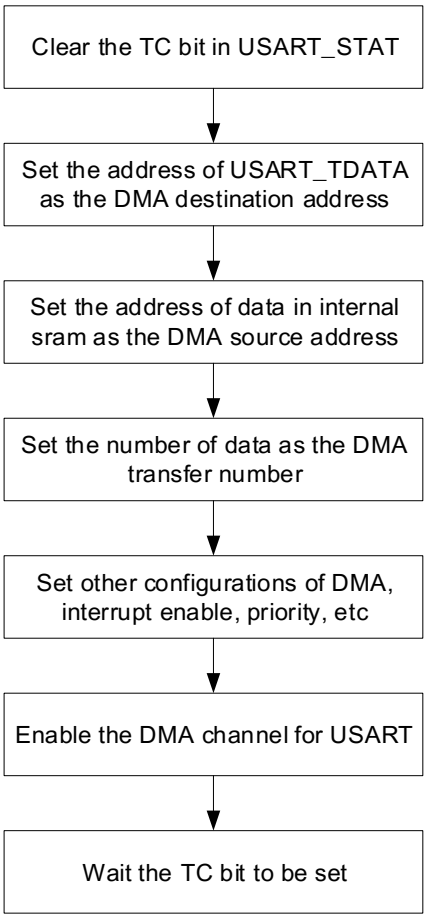
occurs during reception, NERR, PERR, FERR or ORERR will be set at the same time with RBNE. If the receive DMA is not enabled, when the RBNE interrupt occurs, software need to check whether there is a noise error, parity error, frame error or overrun error.

40.3.5. 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 40-5. Configuration step when using DMA for USART transmission***.

Figure 40-5. Configuration step when using DMA for USART transmission

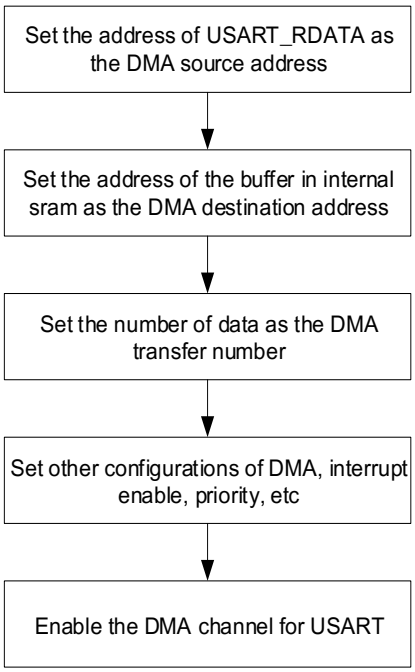


After all of the data frames are transmitted, the TC bit in USART_STAT is set. An interrupt occurs if the TCIE bit in USART_CTL0 is set.

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 40-6. 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 40-6. Configuration step when using DMA for USART reception

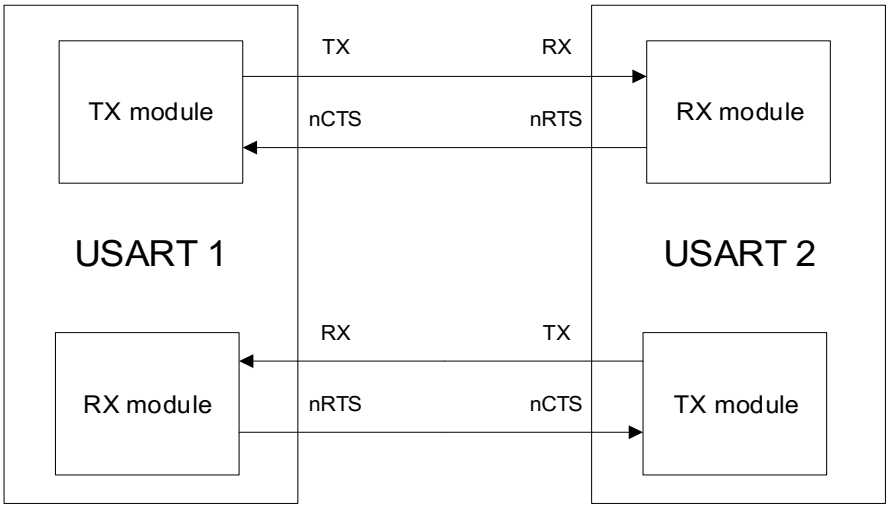


When the number of the data received by USART reaches the DMA transfer number, an end of transfer interrupt can be generated in the DMA module.

40.3.6. 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 40-7. Hardware flow control between two USARTs



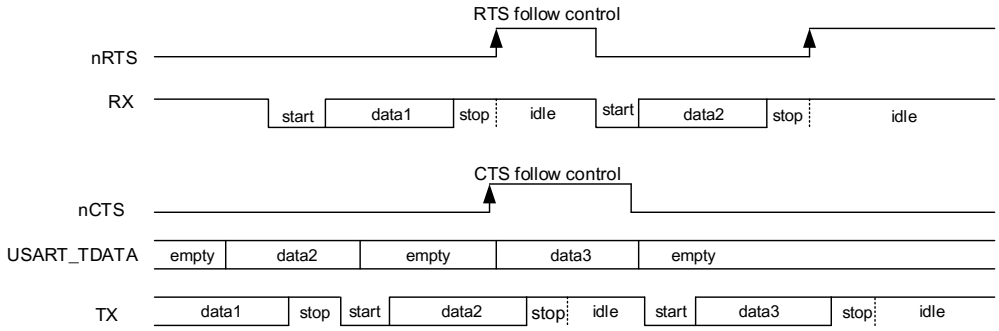
RTS flow control

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 40-8. 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.

40.3.7. 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 position of the address mark is the same as the Parity Bit When parity is enabled for 8 bit and 9 bit data formats.

If only one of AMEN0 and AMEN1 bit is set, a match address is compared only with the associate ADDR0 or ADDR1 and data is transferred to the receive data buffer only on a match.

If AMEN0 and AMEN1 are all set, a match address is compared with both ADDR0 and

ADDR1 and data is transferred only on a match with either ADDR0 or ADDR1. So, a second match address can be used for broadcast or general call address to serial bus.

Note: If the MEN bit is set, the WM bit is reset and the RWU bit is reset, an idle frame is detected on the RX pin, the IDLEF bit will be set. If the RWU bit is set, the IDLEF is not set. Otherwise, AMEN0 or AMEN1 must be set when using address match method to wakeup USART from mute mode.

40.3.8. LIN mode

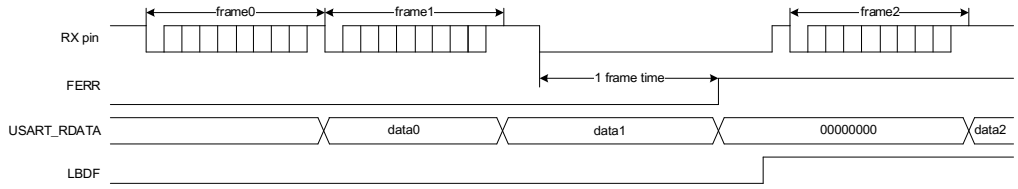
The local interconnection network mode is enabled by setting the LMEN bit in USART_CTL1. The CKEN, STB[1:0] bit in USART_CTL1 and the SCEN, HDEN, IREN bits in USART_CTL2 should be cleared in LIN mode.

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.

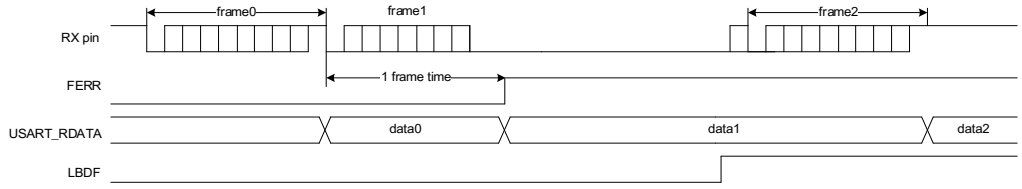
As shown in ***Figure 40-9. Break frame occurs during idle state***, if a break frame occurs during the idle state on the RX pin, the USART receiver will receive an all '0' frame, with an asserted FERR status.

Figure 40-9. Break frame occurs during idle state



As shown in ***Figure 40-10. 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 40-10. Break frame occurs during a frame



40.3.9. 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 40-11. Example of USART in synchronous mode

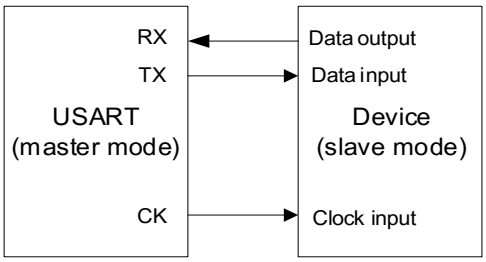
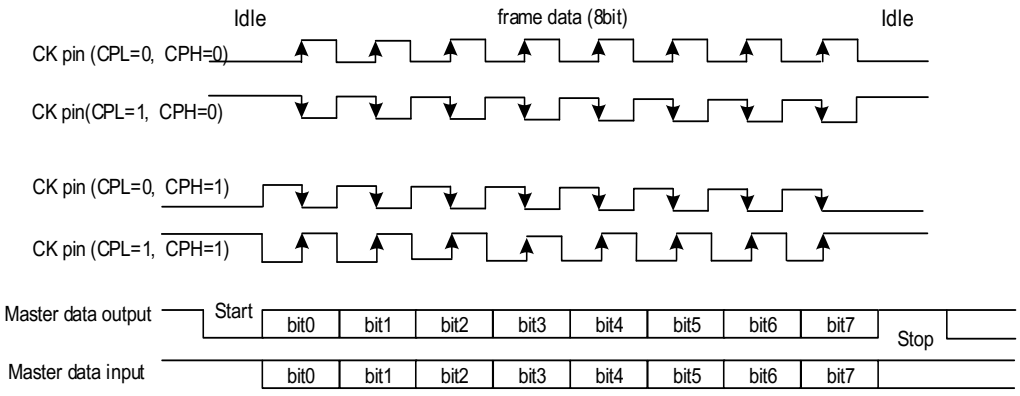


Figure 40-12. 8-bit format USART synchronous waveform (CLEN = 1)



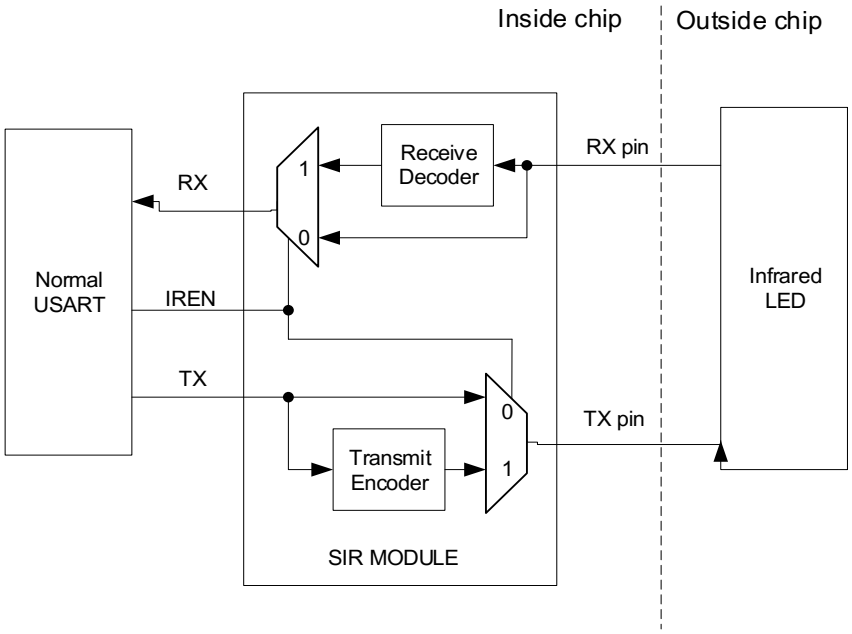
40.3.10. IrDA SIR ENDEC mode

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.

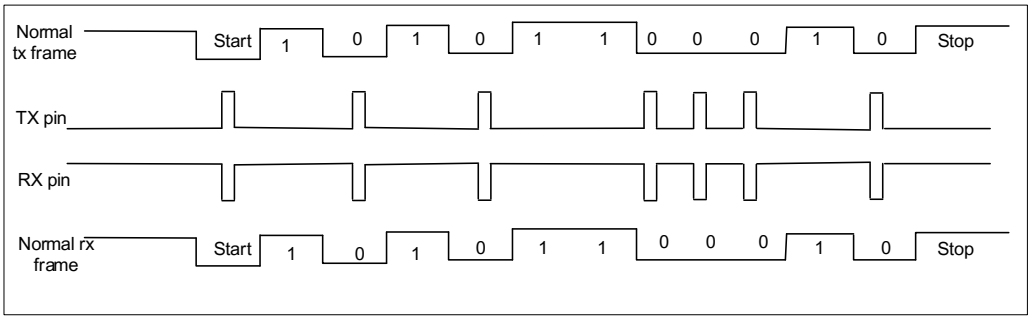
Figure 40-13. IrDA SIR ENDEC module



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 40-14. 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, USART BAUD register needs to be configured as 16*PSC[7:0]. The pulse width on the TX pin is 3 times of this low speed period. The receiver decoder works in the same manner as the normal IrDA mode.

40.3.11. 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..

40.3.12. Internal loopback mode

The USART support internal loopback mode that is enabled by setting the LBEN bit in USART_CTL1. The STRP, TINV and RINV bits in USART_CTL1 should be cleared in loopback mode. The TX and RX pins are connected together internally.

40.3.13. Smartcard (ISO7816-3) mode

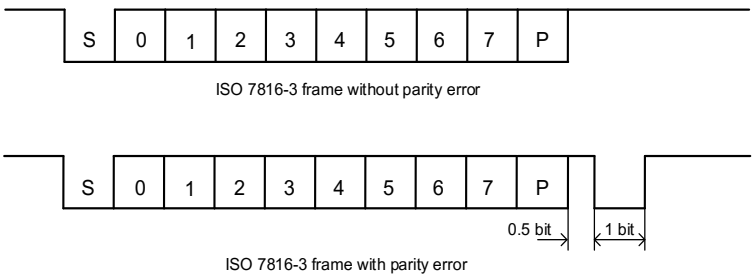
The smartcard mode is an asynchronous mode, which is designed to support the ISO7816-3 protocol. Both the character (T = 0) mode and the block (T = 1) mode are supported. The smartcard mode is enabled by setting the SCEN bit in USART_CTL2. The LMEN bit in USART_CTL1 and HDEN, IREN bits in USART_CTL2 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 40-15. 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 bits time will be inserted before the start of a resented frame. At the end of the last repeated character the TC bit is set immediately without guard time. The USART will stop transmitting and assert the frame error status if it still receives the NACK signal after the programmed number of retries. The USART will not take the NACK signal as the start bit.

During USART reception, if the parity error is detected in the current frame, the TX pin is pulled low during the last 1 bit time of the stop bits. This signal is the NACK signal to smartcard. Then a frame error occurs in smartcard side. The RBNE / receive DMA request is not activated if the received character is erroneous. According to the protocol, the smartcard can resend the data. The USART stops transmitting the NACK and the error is regarded as a parity error if the received character is still erroneous after the maximum number of retries which is specified in the SCRTNUM bit field. The NACK signal is enabled by setting the NKEN bit in USART_CTL2.

The idle frame and break frame are not supported in the Smartcard mode.

Block (T = 1) mode

In block (T = 1) mode, the NKEN bit in the USART_CTL2 register should be cleared to deactivate the NACK transmission.

When requesting a read from the smartcard, the RT[23:0] bits in USART_RT register should be programmed with the BWT (block wait time) - 11 value and RBNEIE must be set. A timeout interrupt will be generated, if no answer is received from the card before the expiration of this period. If the first character is received before the expiration of the period, it is signaled by the RBNE interrupt. If DMA is used to read from the smartcard in block mode, the DMA must be enabled only after the first character is received.

In order to allow the automatic check of the maximum wait time between two consecutive characters, the USART_RT register must be programmed to the CWT (character wait time) - 11 value, which is expressed in baudtime units, after the reception of the first character (RBNE interrupt). The USART signals to the software through the RT flag and interrupt (when RTIE bit is set), if the smartcard doesn't send a new character in less than the CWT period after the end of the previous character.

The USART uses a block length counter, which is reset when the USART is transmitting (TBE = 0), to count the number of received characters. The length of the block, which must be programmed in the BL[7:0] bits in the USART_RT register, is received from the smartcard in the third byte of the block (prologue field). This register field must be programmed to the minimum value (0x0), before the start of the block, when using DMA mode. With this value, an interrupt is generated after the 4th received character. The software must read the third byte as block length from the receive buffer.

In interrupt driven receive mode, the length of the block may be checked by software or by programming the BL value. However, before the start of the block, the maximum value of BL (0xFF) may be programmed. The real value will be programmed after the reception of the third character.

The total block length (including prologue, epilogue and information fields) equals BL+4. The end of the block is signaled to the software through the EBF flag and interrupt (when EBIE bit is set). The RT interrupt may occur in case of an error in the block length.

Direct and inverse convention

The smartcard protocol defines two conventions: direct and inverse.

The direct convention is defined as: LSB first, logical bit value of 1 corresponds to H state of the line and parity is even. In this case, the following control bits must be programmed: MSBF = 0, DINV = 0.

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.

40.3.14. 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.

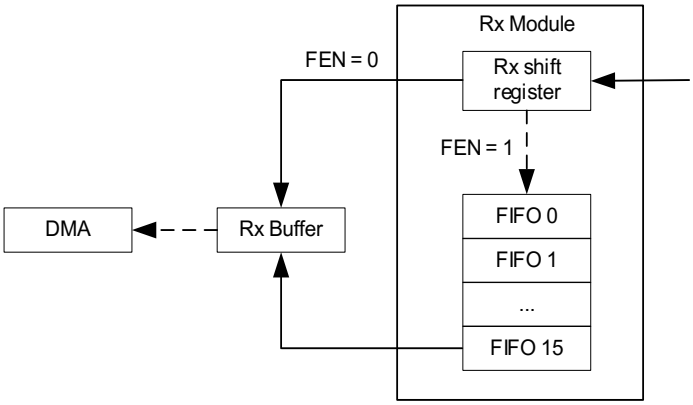
In the ModBus / ASCII mode, the end of a block is recognized by a specific (CR / LF) character sequence. The USART manages this mechanism using the character match function by programming the LF ASCII code in the ADDR0 or ADDR1 field and activating the address match interrupt (AMIE0 = 1 or AMIE1 = 1). When a LF has been received or can check the CR / LF in the DMA buffer, the software will be informed.

40.3.15. 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 40-16. USART receive FIFO structure

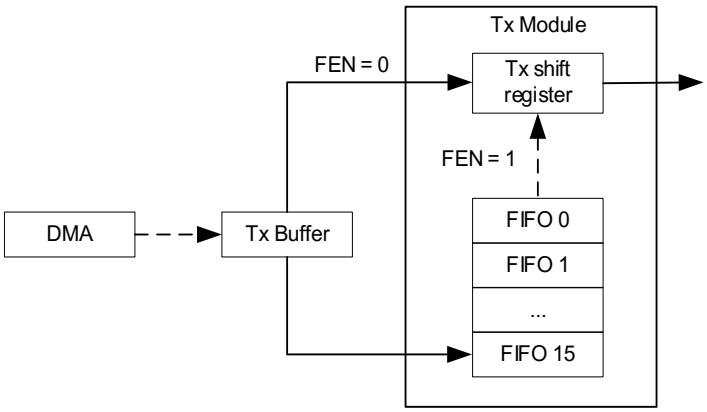


If the software read receive data buffer in the routing of the RBNE interrupt, the RBNEIE bit should be reset at the beginning of the routing and set after all of the receive data is read out. The PERR / NERR / FERR / EBF flags should be cleared before reading a receive data out.

Transmit FIFO

The transmitter FIFO can be enabled by setting the FEN bit of the USART_FCS register. Up to 17 frames transmitter data can be stored in the transmitter FIFO and transmitter buffer. The TFE flag will be set when the transmitter FIFO is empty, an interrupt is generated if the TFEIE bit is set. The TFTIF flag will be set (or TFT flag will be clear) when the available space in transmitter FIFO reaches the threshold configured in the TFTCFG[2:0] bits, an interrupt is generated if the TFTIE bit is set. TFNF flag will be set when transmitter FIFO is not full, an interrupt will be generated if TFNFIE bit is set.

Figure 40-17. USART transmitter FIFO structure



40.3.16. 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_IRC64MDIV 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.

40.3.17. USART interrupts

The USART interrupt events and flags are listed in *Table 40-3. USART interrupt requests*.

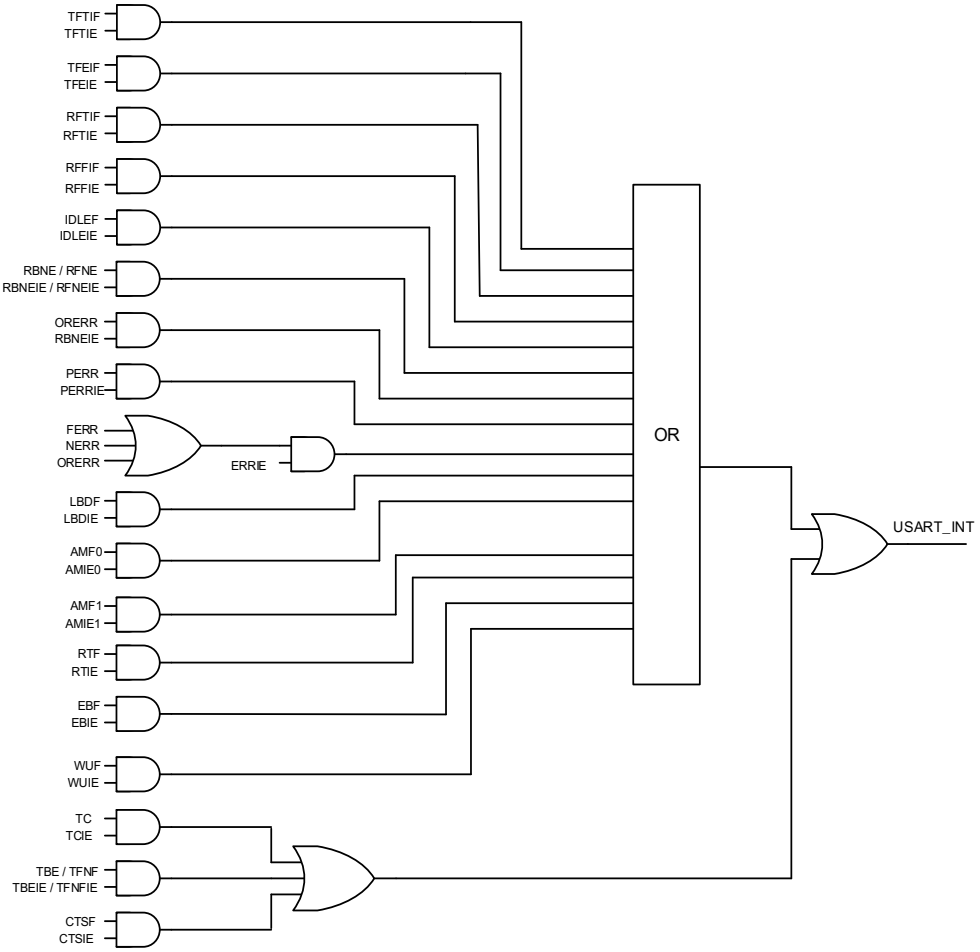
Table 40-3. USART interrupt requests

Interrupt event	Event flag	Enable Control bit
Transmit data register empty or Transmit FIFO not full	TBE / TFNF	TBEIE / TFNFIE

Interrupt event	Event flag	Enable Control bit
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	RFT	RFTIE
Transmit FIFO empty	TFE	TFEIE
Transmit FIFO threshold reaches	TFTIF	TFTIE
Idle line detected	IDLEF	IDLEIE
Parity error flag	PERR	PERRIE
Break detected flag in LIN mode	LBDF	LBDIE
Reception errors (noise flag, overrun error, framing error)	NERR or ORERR or FERR	ERRIE
ADDR0 match	AMF0	AMIE0
ADDR1 match	AMF1	AMIE1
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 40-18. USART interrupt mapping diagram



40.4. Register definition

USART0 base address: 0x4001 1000

USART1 base address: 0x4000 4400

USART2 base address: 0x4000 4800

UART3 base address: 0x4000 4C00

UART4 base address: 0x4000 5000

USART5 base address: 0x4001 1400

UART6 base address: 0x4000 7800

UART7 base address: 0x4000 7C00

40.4.1. 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: ADDR1 character match interrupt is disabled 1: ADDR1 character match interrupt is enabled
30:29	Reserved	Must be kept at reset value.
28	WL1	Word length 1 This bit and WL0 bit determine the word length 00: 8 Data bits 01: 9 Data bits 10: 7 Data bits 11: 10 Data bits This bit field cannot be written when the USART is enabled (UEN = 1).
27	EBIE	End of block interrupt enable

		0: End of block interrupt is disabled 1: End of block interrupt is enabled This bit is reserved in UART3 / UART4 / UART6 / UART7.
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 / UART6 / UART7.
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_IRC64MDIV or CK_LXTAL. This bit is reserved in UART3 / UART4 / UART6 / UART7.
0	UEN	USART enable 0: USART prescaler and outputs disabled 1: USART prescaler and outputs enabled

40.4.2. 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		LBEN	AMEN0
rw	rw	rw		rw	rw	rw	rw		rw	rw	rw			rw	rw

Bits	Fields	Descriptions
31:24	ADDR0[7:0]	Address 0 of the USART terminal These bits give the address 0 of the USART terminal. In multiprocessor communication during mute mode or deep-sleep mode, this is used for wakeup with address match detection. The received frame, the MSB of which is equal to 1, will be compared to these bits. When the ADDM0 bit is reset, only the ADDR0[3:0] bits are used to compare. In normal reception, these bits are also used for character detection. The whole received character (8-bit) is compared to the ADDR0[7:0] value and AMF0 flag is set on matching. This bit field cannot be written when both reception (REN = 1) and USART (UEN = 1) are enabled.
23	RTEN	Receiver timeout enable 0: Receiver timeout function disabled 1: Receiver timeout function enabled This bit is reserved in UART3 / UART4 / UART6 / UART7.
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 / UART6 / UART7.
13:12	STB[1:0]	STOP bits length 00: 1 Stop bit 01: 0.5 Stop bit 10: 2 Stop bits 11: 1.5 Stop bit This bit field cannot be written when the USART is enabled (UEN = 1). **Note:** 0.5 and 1.5 stop bit are not used in UART3 / UART4 / UART6 / UART7.
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 / UART6 / UART7.
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 / UART6 / UART7.
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 / UART6 / UART7.
4	ADDM0	Address 0 detection mode This bit is used to select between 4-bit address detection and full-bit address 0 detection. 0: 4-bit address detection 1: Full-bit address detection. In 7-bit, 8-bit and 9-bit data modes, the address detection is done on 6-bit, 7-bit and 8-bit address (ADDR0[5:0], ADDR0[6:0] and ADDR0[7:0]) respectively This bit field cannot be written when the USART is enabled (UEN = 1).
3:2	Reserved	Must be kept at reset value.
1	LBEN	Loopback mode 0: Loopback mode disable 1: Loopback mode enable This bit field cannot be written when the USART is enabled (UEN = 1).
0	AMEN0	Address 0 match mode enable 0: Address 0 match mode disable 1: Address 0 match mode enable.

40.4.3. 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 / UART6 / UART7.
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 / UART6 / UART7.
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 / UART6 / UART7.
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). **Note:** When FIFO is enabled and OVRD is set (overrun functionality is disabled), the received data will enter FIFO first instead of RDATA, resulting in the RTS pin not being pulled high in hardware flow control mode.
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 / UART6 / UART7.
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 / UART6 / UART7.
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 / UART6 / UART7.
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

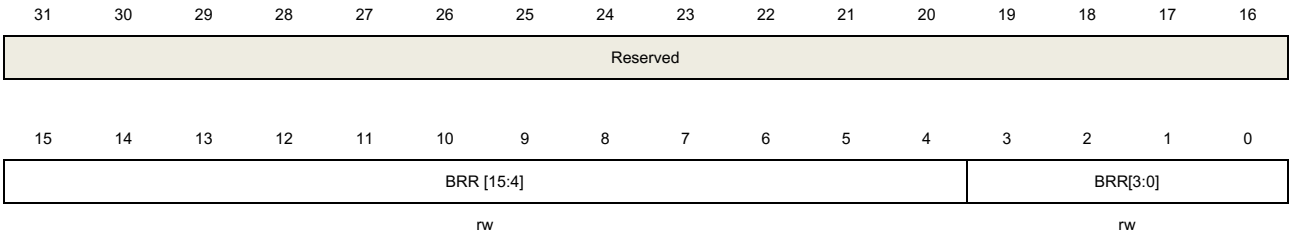
40.4.4. Baud rate generator register (USART_BAUD)

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.

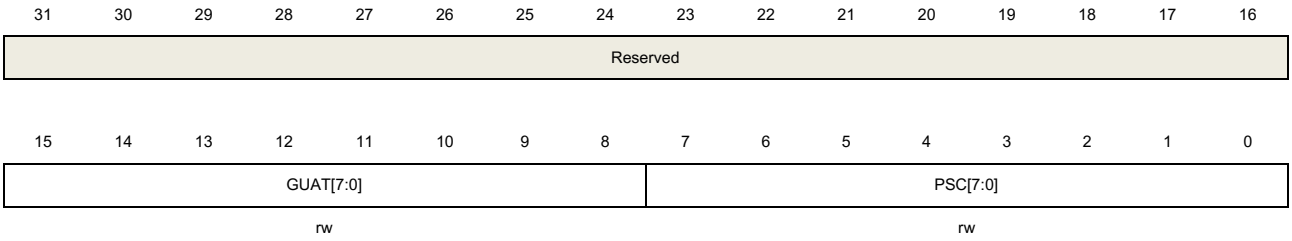
40.4.5. Prescaler and guard time configuration register (USART_GP)

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 / UART6 / UART7.

This register has to be accessed by word (32-bit).



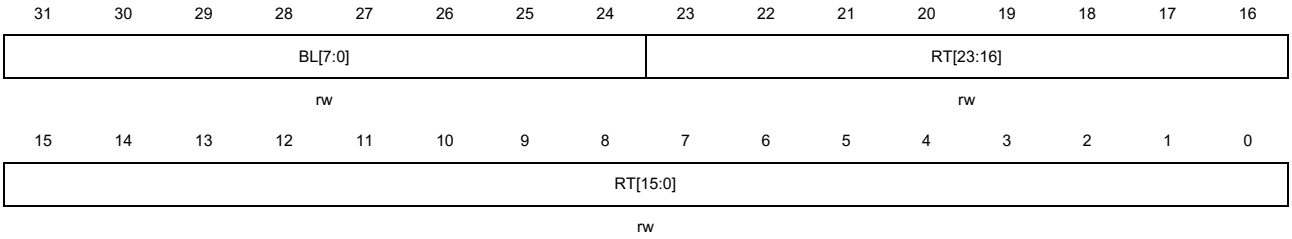
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:8	GUAT[7:0]	Guard time value in smartcard mode This bit field cannot be written when the USART is enabled (UEN=1).
7:0	PSC[7:0]	Prescaler value for dividing the system clock In IrDA Low-power mode, the division factor is the prescaler 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).

40.4.6. Receiver timeout register (USART_RT)

Address offset: 0x14
Reset value: 0x0000 0000

This bit is reserved in UART3 / UART4 / UART6 / UART7.

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:24	BL[7:0]	Block Length These bits specify the block length in smartcard T=1 reception. Its value equals the number of information characters + the length of the Epilogue Field (1-LEC / 2-CRC) - 1. This value, which must be programmed only once per received block, can be programmed after the start of the block reception (using the data from the LEN character in the Prologue Field). The block length counter is reset when TBE = 0 or TFE = 0 (FIFO is enabled) in smartcard mode. In other modes, when REN = 0 (receiver disabled) and / or when the EBC bit is written to 1, the block length counter is reset.
23:0	RT[23:0]	Receiver timeout threshold These bits are used to specify receiver timeout value in terms of number of baud clocks. In standard mode, the RTF flag is set if no new start bit is detected for more than the RT value after the last received character. In smartcard mode, the CWT and BWT are implemented by this value. In this case, the timeout measurement is started from the start bit of the last received character. These bits can be written on the fly. The RTF flag will be set if the new value is lower than or equal to the counter. These bits must only be programmed once per received character. **Note:** When oversampling by 16, RT[23:0] cannot be configured as 0XFFFFFF.

40.4.7. 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, during the flush command, TFNF is reset until Transmit FIFO is empty.
3	RXFCMD	Receive data flush command When FIFO is disabled, Writing 1 to this bit clears the RBNE flag to discard the received data without reading it. When FIFO is enabled, Writing 1 to this bit empties the Receive FIFO and sets RFNE flag in USART_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.

40.4.8. 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 This bit is reserved in UART3 / UART4 / UART6 / UART7.
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 / UART6 / UART7.
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 AMC0 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 AMC1 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 / UART6 / UART7.
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 / UART6 / UART7.
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 / UART6 / UART7.

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.

40.4.9. 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 / UART6 / UART7.
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

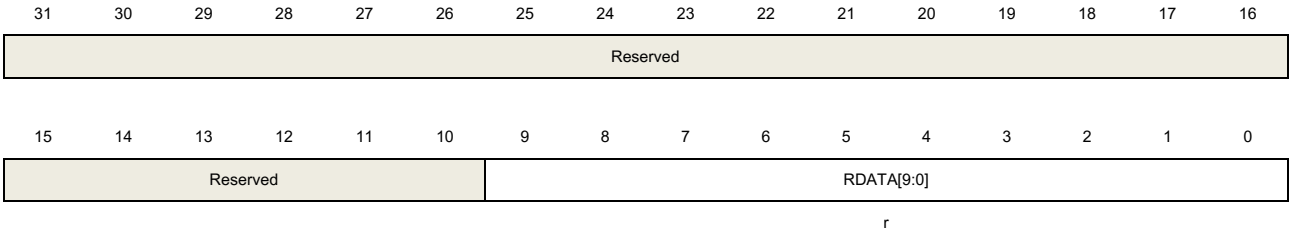
		Writing 1 to this bit clears the EBF bit in the USART_STAT register. This bit is reserved in UART3 / UART4 / UART6 / UART7.
11	RTC	Receiver timeout clear Writing 1 to this bit clears the RTF flag in the USART_STAT register. This bit is reserved in UART3 / UART4 / UART6 / UART7.
10	Reserved	Must be kept 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 / UART6 / UART7.
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.

40.4.10. Receive data register (USART_RDATA)

Address offset: 0x24

Reset value: Undefined

This register has to be accessed by word (32-bit)



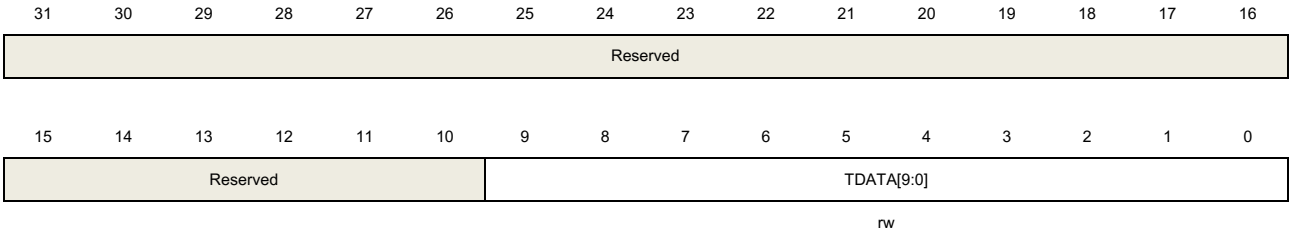
Bits	Fields	Descriptions
31:10	Reserved	Must be kept at reset value.
9:0	RDATA[9:0]	Receive data value The received data character is contained in these bits. The value read in the MSB (bit6, bit7, bit8 or bit9 depending on the data length) will be the received parity bit, if receiving with the parity is enabled (PCEN bit set to 1 in the USART_CTL0 register).

40.4.11. Transmit data register (USART_TDATA)

Address offset: 0x28

Reset value: Undefined

This register has to be accessed by word (32-bit).



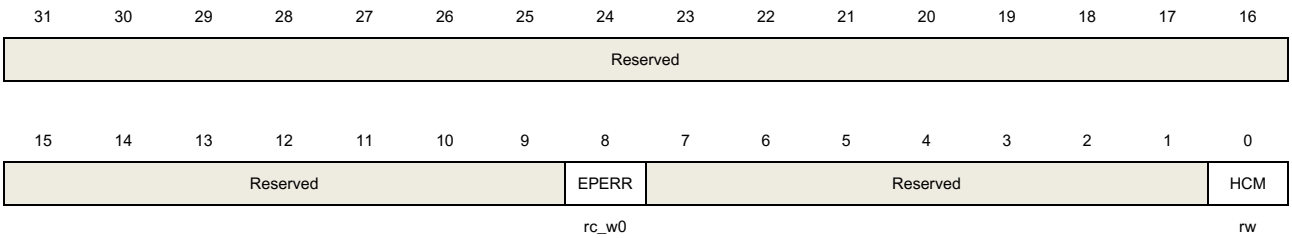
Bits	Fields	Descriptions
31:10	Reserved	Must be kept at reset value
9:0	TDATA[9:0]	Transmit Data value The transmit data character is contained in these bits. The value written in the MSB (bit6, bit7, bit8 or bit9 depending on the data length) will be replaced by the parity, when transmitting with the parity is enabled (PCEN bit set to 1 in the USART_CTL0 register). This register must be written only when TBE bit in USART_STAT register is set.

40.4.12. 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.

40.4.13. USART FIFO control and status register (USART_FCS)

Address offset: 0xD0

Reset value: 0x0300 0440

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TFEIE	Reserved	TFTIE	Reserved	RFTIE	TFEC	TFTIF	TFEIF	Reserved	RFTIF	TFTCFG[2:0]			RFTCFG[2:0]		
rw		rw		rw	w	r	r		rc_w0	rw			rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RFFIF	RFCNT[2:0]			RFF	RFE	RFFIE	FEN	TFF	TFE	TFT	RFT	Reserved	RFCNT[4:3]		ELNACK
rc_w0	r			r	r	rw	rw	r	r	r	r		r		rw

Bits	Fields	Descriptions
31	TFEIE	Transmit FIFO empty interrupt enable If this bit is set, an interrupt occurs whenever the TFE bit is set. 0: Transmit FIFO empty interrupt is disabled 1: Transmit FIFO empty interrupt is enabled
30	Reserved	Must be kept at reset value.
29	TFTIE	Transmit FIFO threshold interrupt enable If this bit is set, an interrupt will occur whenever the available space in transmit FIFO reached the threshold configured in TFTCFG[2:0]. 0: Transmit FIFO threshold interrupt is disabled 1: Transmit FIFO threshold interrupt is enabled
28	Reserved	Must be kept at reset value.
27	RFTIE	Receive FIFO threshold interrupt enable If this bit is set, an interrupt will occur whenever the receive FIFO reached the

		threshold configured in RFTCFG. 0: Receive FIFO threshold interrupt is disabled 1: Receive FIFO threshold interrupt is enabled
26	TFEC	Transmit FIFO empty flag clear **Note:** Writing 1 to this bit sets the TFEIF flag.
25	TFTIF	Transmit FIFO threshold interrupt flag The bit is valid when TFTIE bit is set. 0: The available space in transmit FIFO does not reach the programmed threshold 1: The available space in transmit FIFO reached the programmed threshold. An interrupt will occur if the TFTIE bit is set. Set by hardware when the the available space in transmit FIFO reaches the threshold configured in TFTCFG[2:0].
24	TFEIF	Transmit FIFO empty interrupt flag The bit is valid when TFEIE bit is set. 0: Transmit FIFO is not empty 1: Transmit FIFO is empty
23	Reserved	Must be kept at reset value.
22	RFTIF	Receive FIFO threshold interrupt flag The bit is valid when RFTIE bit is set. 0: Receive FIFO does not reach the programmed threshold 1: Receive FIFO reached the programmed threshold
21:19	TFTCFG[2:0]	Transmit FIFO threshold configuration 000:Transmit FIFO reaches 1/8 of its depth 001:Transmit FIFO reaches 1/4 of its depth 010:Transmit FIFO reaches 1/2 of its depth 011:Transmit FIFO reaches 3/4 of its depth 100:Transmit FIFO reaches 7/8 of its depth 101:Transmit FIFO becomes empty 11x: Reserved
18:16	RFTCFG[2:0]	Receive FIFO threshold configuration 000:Receive FIFO reaches 1/8 of its depth 001:Receive FIFO reaches 1/4 of its depth 010:Receive FIFO reaches 1/2 of its depth 011:Receive FIFO reaches 3/4 of its depth 100:Receive FIFO reaches 7/8 of its depth 101:Receive FIFO becomes full 11x: Reserved
15	RFFIF	Receive FIFO full interrupt flag The bit is valid when RFFIE bit is set. 0: Receive FIFO is not full

		1: Receive FIFO is full
14:12	RFCNT[2:0]	Receive FIFO counter number These bits and RFCNT[4:3] bits determine the receive FIFO counter number.
11	RFF	Receive FIFO full flag 0: Receive FIFO is not full 1: Receive FIFO is full. An interrupt will occur if the RFFIE bit is set. Set by hardware when the number of received data is RXFIFO size + 1.
10	RFE	Receive FIFO empty flag 0: Receive FIFO is not empty 1: Receive FIFO is empty.
9	RFFIE	Receive FIFO full interrupt enable If this bit is set, an interrupt occurs when the RFF bit is set. 0: Receive FIFO full interrupt is disable 1: Receive FIFO full interrupt is enable
8	FEN	FIFO enable 0: FIFO is disable 1: FIFO is enable This bit field cannot be written when the USART is enabled (UEN = 1). Note: Do not change the FEN bit when receiving or transmitting data is not accomplished. When UEN is cleared and reconfigure the UEN without changing the FEN bit, please flush the FIFO if do not need the pervious FIFO value.
7	TFF	Transmit FIFO full flag 0: Transmit FIFO is not full 1: Transmit FIFO is full
6	TFE	Transmit FIFO empty flag 0: Transmit FIFO is not empty 1: Transmit FIFO is empty. An interrupt will occur if the TFEIE bit is set. Set by hardware when the transmit FIFO is empty. The flag is cleared when the transmit FIFO has at least one data. The bit can also be set by setting TXFCMD bit in USART_CMD.
5	TFT	Transmit FIFO threshold flag 0: The available space in transmit FIFO reached the programmed threshold. 1: The available space in transmit FIFO does not reach the programmed threshold. Reset by hardware when the available space in transmit FIFO reaches the threshold configured in TFTCFG[2:0].
4	RFT	Receive FIFO threshold flag 0: Receive FIFO does not reach the programmed threshold 1: Receive FIFO reached the programmed threshold. An interrupt will occur if the RFTIE bit is set.

		Set by hardware when the receive FIFO reaches the threshold configured in RFTCFG[2:0]. This means that there are (RFTCFG[2:0] - 1) data in the Receive FIFO and one data in the USART_RDATA register **Note:** When the RFTCFG[2:0] = 0b101 and 16 data are available, RFT flag will be set.
3	Reserved	Must be kept at reset value.
2:1	RFCNT[4:3]	Receive FIFO counter number These bits and RFCNT[2:0] bits determine the receive FIFO counter number.
0	ELNACK	Early NACK when smartcard mode is selected. The NACK pulse occurs 1/16 bit time earlier when the parity error is detected. 0:Early NACK disable when smartcard mode is selected 1:Early NACK enable when smartcard mode is selected This bit is reserved in UART3 / UART4 / UART6 / UART7.

41. Serial peripheral interface/Inter-IC sound (SPI/I2S)

41.1. Overview

The SPI/I2S module can communicate with external devices using the SPI protocol or the I2S audio protocol.

The serial peripheral interface (SPI) provides a SPI protocol of data transmission and reception function in master or slave mode. Both full-duplex, half-duplex and simplex communication modes are supported, with hardware CRC calculation and checking. Quad-SPI master mode is also supported in SPI3 / 4.

The inter-IC sound (I2S) supports four audio standards: I2S Philips standard, MSB justified standard, LSB justified standard, and PCM standard. I2S works at either master or slave mode for transmission and reception.

41.2. Characteristics

41.2.1. SPI characteristics

- Master or slave operation with full-duplex, half-duplex or simplex mode.
- Separate transmit and receive 32-bit FIFO.
- Data frame size can be 4 to 32 bits.
- Bit order can be LSB or MSB.
- Software and hardware NSS management, MOSI and MISO pin switch alternate function.
- Hardware CRC calculation, transmission and checking.
- Transmission and reception using DMA.
- SPI TI mode supported.
- Multi-master or multi-slave mode function.
- Protect configurations and settings.
- Both the minimum delay between data frames and the minimum delay between NSS and data stream are adjustable.
- Master mode failures can trigger interrupts, overrun or underrun flags, and CRC error detection.
- Adjustable main device receiver sampling time.
- Configurable FIFO thresholds (data packing).
- In slave mode, the underrun condition is configurable.
- Quad-SPI configuration available in master mode (in SPI3 / 4).

41.2.2. I2S characteristics

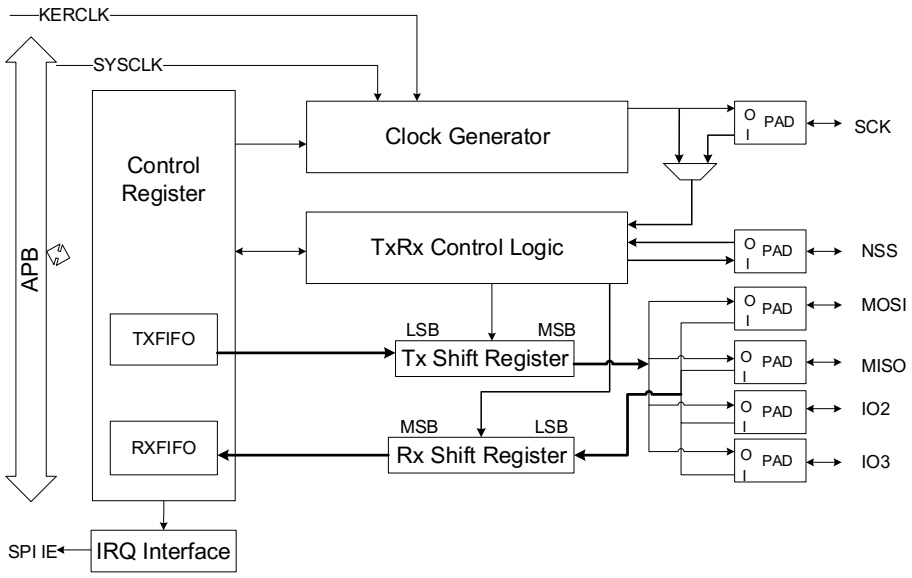
- Master or slave operation for transmission/reception.

- Four I2S standards supported: Philips, MSB justified, LSB justified and PCM standard.
- Data length can be 16 bits, 24 bits or 32 bits.
- Channel length can be 16 bits or 32 bits.
- Error signals improve reliability: underrun, overrun, and frame errors.
- Transmission and reception use a 32 bits wide buffer.
- Audio sample frequency can be 8 kHz to 192 kHz using I2S clock divider.
- Programmable idle state clock polarity.
- Master clock (MCK) can be output.
- Transmission and reception support DMA function.
- Separate transmit and receive 32-bit FIFO.

41.3. SPI function overview

41.3.1. SPI block diagram

Figure 41-1. Block diagram of SPI



- SYSCLK: system clock, provided by APB bus.
- KERCLK: kernel clock, provided by RCU. It has an asynchronous relationship with the system clock.
- The clock signal frequency must be consistent with user conditions and data transmission speed to prevent data loss. (**NOTE**: suggest the frequency of SYSCLK is greater than or equal to KERCLK)
- The SCK signal of the SPI slave is provided by the SPI master.

41.3.2. SPI signal description

Normal configuration (Not Quad-SPI Mode)

Table 41-1. SPI internal input/output signals

Signal name	Signal type	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 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 (available in SPI3 / 4). Quad-SPI mode can only work in master mode.

The SPI is connected to external devices through 6 pins in Quad-SPI mode:

Table 41-2. Quad-SPI internal input/output signals

Signal name	Signal type	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

Serial data line switching configuration

SPI can exchange the functions of MOSI and MISO pins through the SWPMIO bit in the SPI_CFG1 register.

Table 41-3. MISO / MISO signal switching description

MODE	SWPMIO	MOSI	MISO
Master transmit	0	Transmission	-
	1	-	Transmission
Slave transmit	0	-	Transmission
	1	Transmission	-
Master receive	0	-	Reception
	1	Reception	-
Slave receive	0	Reception	-
	1	-	Reception
Master Full-duplex	0	Transmission	Reception
	1	Reception	Transmission
Slave Full-duplex	0	Reception	Transmission
	1	Transmission	Reception

41.3.3. SPI clock timing and data format

CKPL and CKPH bits in SPI_CFG1 register decide the timing of SPI clock and data signal. The CKPL bit decides the SCK level when idle and CKPH bit decides either first or second clock edge is a valid sampling edge. These bits take no effect in TI mode.

Figure 41-2. SPI timing diagram in normal mode

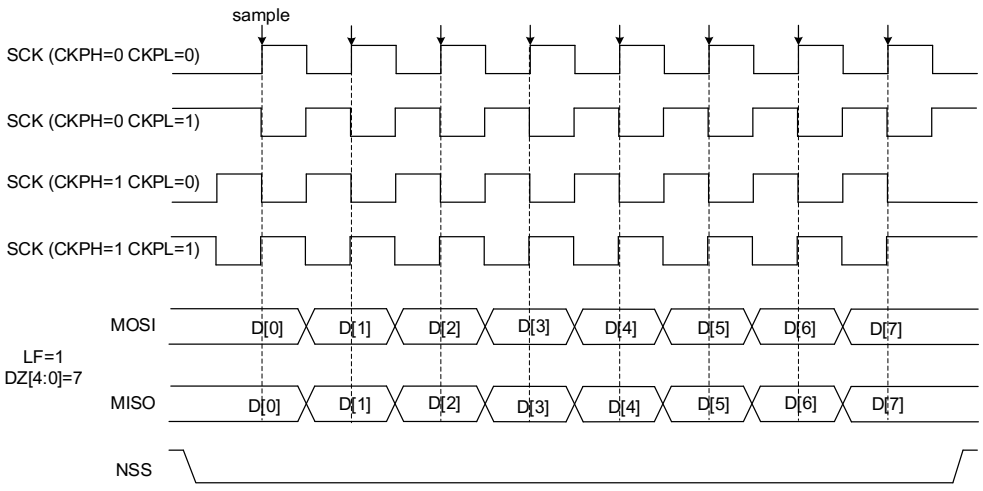
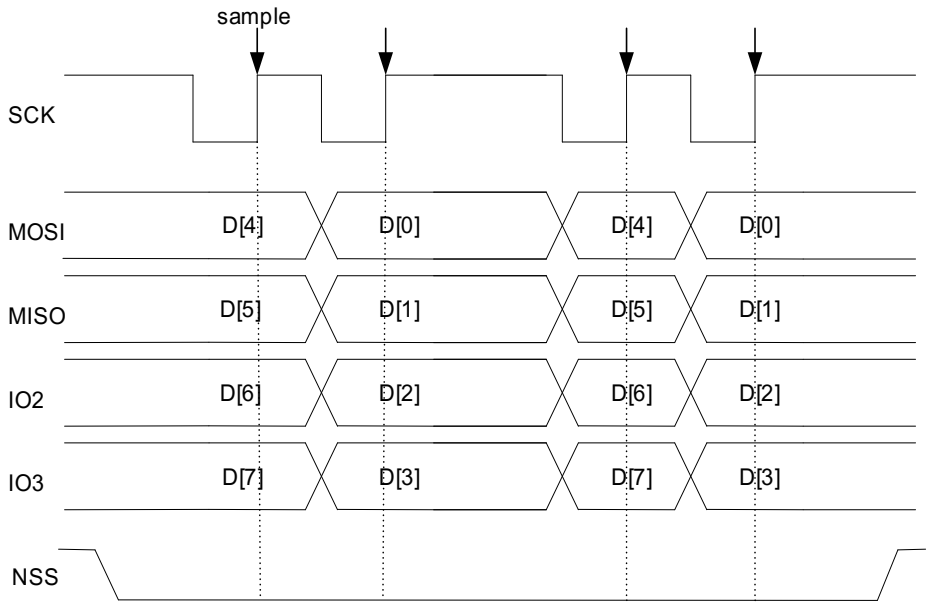


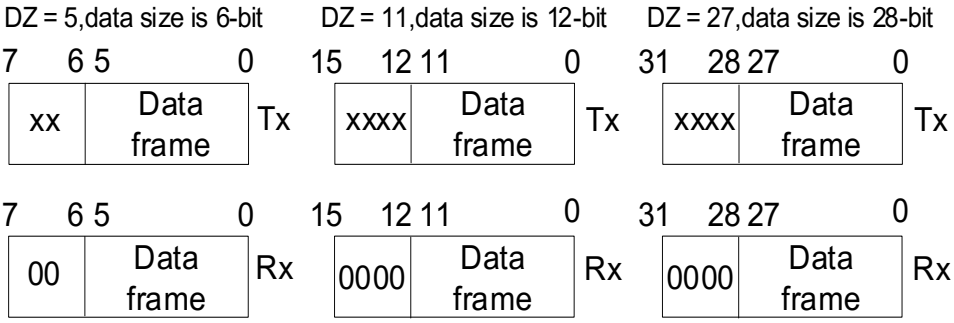
Figure 41-3. SPI3 / 4 timing diagram in Quad-SPI mode (CKPL = 1, CKPH = 1, LF = 0)



In SPI normal mode, the length of data is configured by the DZ[4:0] bits in the SPI_CFG0 register. It can be set from 4-bit up to 32-bit length and the setting applies for both transmission and reception. Data order is configured by LF bit in SPI_CFG1 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. The data frame length is fixed to 8 bits in Quad-SPI mode.

When the SPI_TDATA / SPI_RDATA register is accessed, data frames are always right-aligned into either a byte (if the data fits into a byte), a half-word or a word. During communication, only bits within the data frame are clocked and transferred.

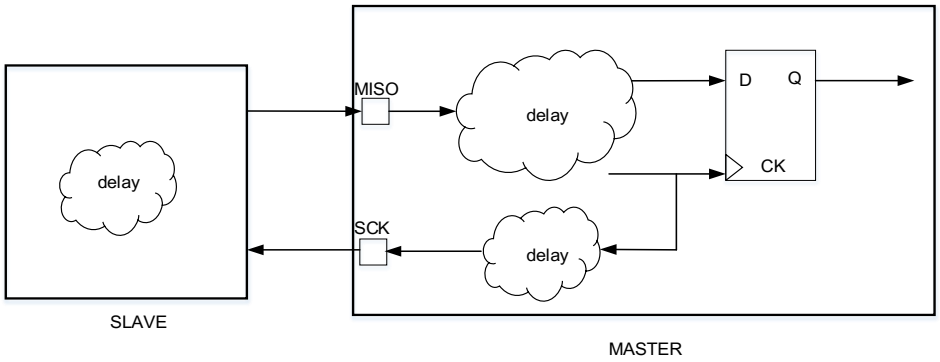
Figure 41-4. SPI data frame right-aligned diagram



41.3.4. SPI clock delay mode

SPI can be configured as master or slave mode. When SPI is configured as master, the SCK signal is sent to slave after delay. The slave send data by MISO, the data is sent to master sampling side after delay. This series of delays causes a phase difference between the data and clock received by the SPI master, resulting in data sampling errors. This issue becomes more pronounced at higher bit rates.

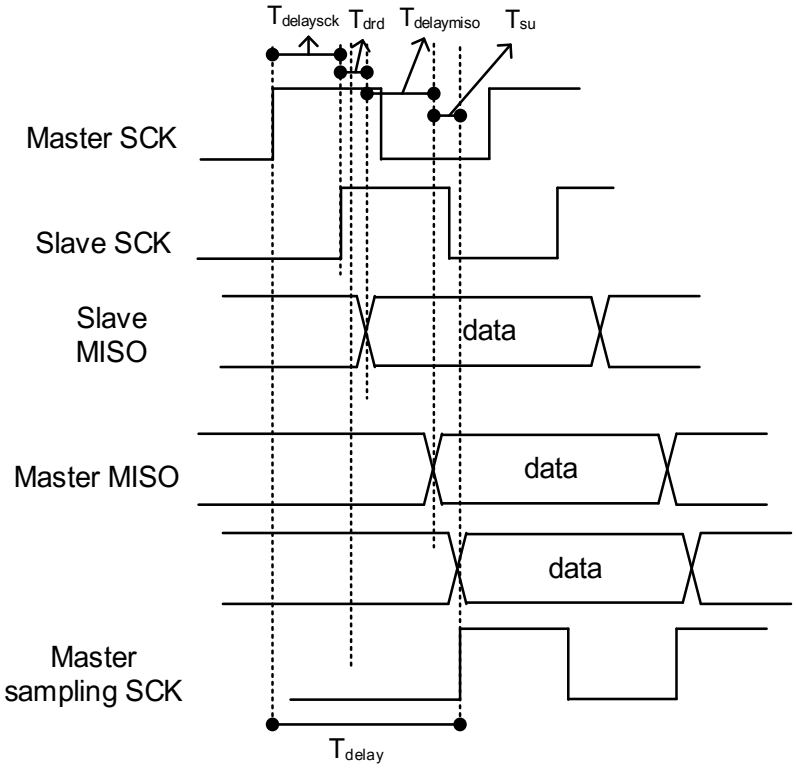
Figure 41-5. SPI data and clock transmission path diagram



In order to solve this problem, can be configured SPI_RXDLYCK register to adjust SPI internal receive clock phase that meet the correct sampling sequence(this configuration is optional, requires a combination of the actual scenario).

If the MRXDEN bit in the SPI_RXDLYCK register is set to 0, the delay function is opened, otherwise the delay function is closed. The MRXD[4:0] bits is used to configure delay units, the delay length can be configured to 1 ~ 32 units(one delay unit is 0.5ns in room temperature). User should configure T_{delay} latency according to their own scenario(T_{delay} > $T_{delaysck}$ + T_{drd} + $T_{delaymiso}$ + T_{su}).

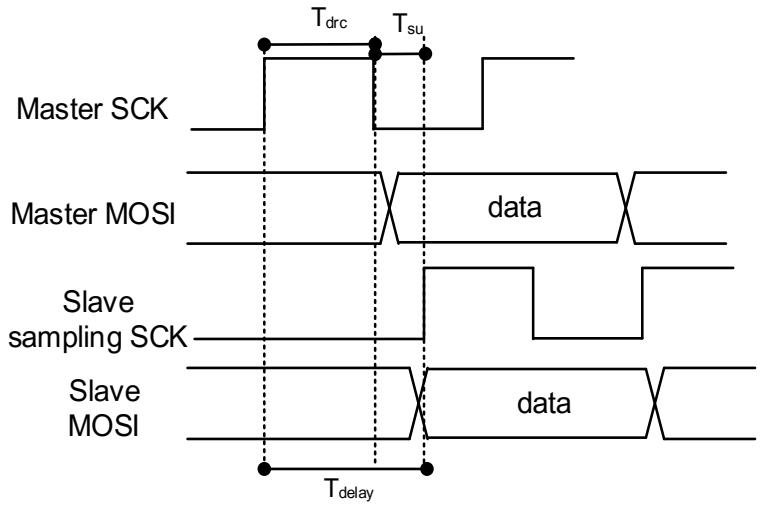
Figure 41-6. SPI master rx delay configuration diagram



If the SPI is configured as slave mode. If the SRXDEN bit in the SPI_RXDLYCK register is set to 0, the delay function is opened, otherwise the delay function is closed. The SRXD[4:0] bits is used to configure delay units, the delay length can be configured to 1 ~ 32 units(one

delay unit is 0.5ns in room temperature). User should configure T_{delay} latency according to their own scenario($T_{delay} > T_{drc} + T_{su}$).

Figure 41-7. SPI slave rx delay configuration diagram



41.3.5. RxFIFO and TxFIFO

RxFIFO and TxFIFO are used in different directions for SPI data transactions, and they can enable the SPI to work in a continuous flow, and can prevent short data frame length or interrupt / DMA overrun occurs when the delay is too long.

A write access to the SPI_TDATA register stores the written data in the end of TxFIFO, while a read access to the SPI_RDATA returns the oldest value in RxFIFO which has not been read. FIFO processing depends on the data exchange mode (duplex and simplex), data frame format (DZ value), the size of the access to the FIFO register (8, 16, or 32 bits) and how the data is organized in the packet. The size of TxFIFO / RxFIFO is 16 x 32 bits and the maximum access data frame length is 32 bits. According to different frame length, the maximum number of frames that can be stored in FIFO is described in

Table 41-4. The maximum number of data frame stored in SPIX FIFO. (N = FIFO size / 32 = 16 x 32 / 32 = 16)

Table 41-4. The maximum number of data frame stored in SPIX FIFO

Frame size(DZ)	DZ <= 8bit	8bit < DZ <= 16bit	16bit < DZ <= 24bit	DZ > 24bit
Frame numbers (BYTEN = 1, WORDEN = 0)	N	-	-	-
Frame numbers (BYTEN = 0, WORDEN = 0)	2N	N		

Frame numbers (WORDEN = 1)	4N	2N	N	N

NOTE: Both RxFIFO and TxFIFO content is kept flushed when SPI is disable (SPIEN = 0).

RxFIFO reception

A read access to SPI_RDATA is managed by the RP event. This is triggered when RxFIFO is not considered to be empty (at least a complete data packet in RxFIFO). When RP is cleared, the RxFIFO is considered to be empty (or the data packet in RxFIFO is incomplete). RP triggers interrupts at RPIE position 1, or DMA requests at DMAREN position 1.

TxFIFO transmission

Write access to SPI_TDATA is managed by the TP event. This event is triggered when TxFIFO has enough available space to receive a packet. If the TxFIFO is filled by software or DMA, the TP flag cleared. If there is not enough space to store at least one data packet when TXF set to 1 or SPI disable, write to TxFIFO is will be ignored. TP triggers interrupts at TPIE position 1, or DMA requests at DMATEN position 1. TPIE shield is cleared by hardware when TXF flag is set to 1.

Duplex packet handling

In full-duplex mode, the DP bit can monitor the TP and RP events. With the DP flag set to 1, the application writes an appropriate amount of data to the SPI_TDATA register to upload one complete packet, and then reads an equal number of data from the SPI_RDATA register to download one complete packet. After a packet is uploaded and downloaded, the application checks the DP value to see if it can be ordered push in and pop up other packets, and if possible, upload/download them packet by packet until DP reads 0. DP triggers interrupts with DPIE set 1, or DMA requestion when both DMATEN and DMAREN set to 1. DPIE shield is cleared by hardware when TXF flag is set to 1.

If the next data is received while the RxFIFO is full, the reception overrun event occurs. Overrun events can be polled or handled by interrupts. This may occur in slave mode or master mode (full-duplex or receive-only mode, MASP = 0). The main device is in receive-only mode, when MASP = 1, if the RxFIFO is full, the generated clock will stop automatically to prevent the overrun events.

Data packing

When the data frame size (DZ) <= 8 bits, data packing mode is automatically enabled when any 16-bit or 32-bit read or write access in SPI_RDATA or SPI_TDATA (BYTEN = 0 or WORDEN = 1). The multi-data frames pattern is handled in parallel in this case. At the sending terminal, if FIFOLVL = 1 (2 data frames in a packet) or FIFOLVL = 3 (4 data frame in a packet), two or four data frames are sent after the single 16-bit or 32-bit access the SPI_TDATA register of the transmitter. At the receiving terminal, if FIFOLVL = 1 (2 data frames in a packet) or FIFOLVL = 3 (4 data frames in a packet), two or four data frames are

received simultaneously after the single 16-bit or 32-bit access the SPI_RDATA register of the receiver, which can generate just one RP event in the receiver. The receiver then has to read all data frames by 16-bit or 32-bit from SPI_RDATA. If FIFOLVL = 0 (1 data frames in a packet), the receiver will generate two or four RP events when reading a data frame by 16-bit or 32-bit from SPI_RDATA.

If 9-bit <= DZ <= 16-bit, data packing mode is automatically enabled when any 32-bit read or write access in SPI_RDATA or SPI_TDATA. The least significant halfword will be used to store significant data. At the sending terminal, if FIFOLVL = 1 (2 data frames in a packet), two data frames are sent after the single 32-bit access the SPI_TDATA register of the transmitter. At the receiving terminal, if FIFOLVL = 1 (2 data frames in a packet), two data frames are received simultaneously after the single 32-bit access the SPI_RDATA register of the receiver, which can generate just one RP event in the receiver. The receiver then has to read all data frames by 32-bit from SPI_RDATA. If FIFOLVL = 0 (1 data frames in a packet), the receiver will generate two RP events when reading a data frame by 32-bit from SPI_RDATA.

When short data frame size (<= 8-bit or <= 16-bit) are paired with large data access patterns (16-bit or 32-bit), the FIFOLVL value must be configured as a multiple of the number of data frames , multiple of 4 if 32-bit access is used for frames up to 8 bits, multiple of 2 if 16-bit access is used for frames up to 8-bit or multiple of 2 while 32-bit access is used for frames up to 16-bit.

The FIFOLVL setting must always be higher than the subsequent read access size, otherwise additional pseudo-data will be read. FIFO data access that is smaller than the configured data size is not allowed (Data frame size is configured by DZ, the size of FIFO data access is configured by BYTEN and WORDEN). Always ensure that at least one complete data frame is accessed.

NOTE: TXSER and TXSIZE must be set to multiples of the number of data frames in a packet in DMA mode.

Sequential transaction handling

Users can processes multiple numbers of data in a message according to TXSIZE and TXSER value. The transaction of a message starts when SPI is enabled by setting MSTART bit and finishes when the number of data required has been transferred. If TXSIZE remains zero when MSTART is set to 1, the infinite transaction is initialized. Transactions can be suspended at any time by setting the MSPDR bit, which clears MSTART bit.

In master mode, after the number of data in TXSIZE has been transferred, if TXSER contains a non-zero value, the value of TXSER will be copied to TXSIZE and the TXSER value is cleared automatically. The transmission will then increase the number of data corresponding to the newly loaded value in TXSIZE. After the reload operation, if TXSERFIE is set to 1, the TXSERF flag is set to 1 and will triggers interrupt. The user can write the next non-zero value to the TXSER before the next reload, so it can handle multiple data. In this case, ET events do not occur because transmission continues.

If data amount of TXSIZE or TXSER (number of data frames) is not aligned with packet length defined in FIFOLVL, then the last incomplete packet before the end of sending needs to be packaged. ***Data packing*** detail describe the principle of packaging.

Note: in order to prevent transmission underrun, the SPI_URDATA register of slave can be written to a specific value. When the TxFIFO of slave becomes empty, the value in its SPI_URDATA will be the next data is sent out automatically, and after the host receives this value which can be parsed by the software, so the host can suspend its receiver through software operation.

Transaction delay handling

If the reception speed of slave is less than the transmission speed of master, the master must be to reduce the transmission speed, by lowering the clock frequency or increase the time delay between data frames. The MDFD[3:0] bits in SPI_CFG1 register is used to add delay between data frames in master mode, and MSSD[3:0] is used to add delay between active edge of NSS and start transfer or receive data in master mode. ***NSS signal timing*** shows the detail description.

41.3.6. NSS function

Slave mode

When slave mode is configured (MSTMOD = 0), SPI gets NSS level from NSS pin in hardware NSS mode (NSSIM = 0) or from NSSI bit in software NSS mode (NSSIM = 1) and transmits / receives data only when NSS level is active. The user can set NSSIOPL bit to decide the active level of input/output external signals (on NSS pins).

Table 41-5. NSS function in slave mode

Mode	Register configuration	Description
Slave hardware NSS mode	MSTMOD = 0 NSSIM = 0	SPI slave gets NSS level from NSS pin.
Slave software NSS mode	MSTMOD = 0 NSSIM = 1	SPI slave NSS level is determined by the NSSI bit. NSSI = 0: NSS level is low NSSI = 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 (NSSIM = 0, NSSDRV = 0) or software mode (NSSIM = 1). Then, once the NSS pin (in hardware NSS mode) or the NSSI bit (in software NSS mode and NSSDRV = 0) goes non-active, 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 (NSSIM = 0, NSSDRV = 1). After enabling SPI and setting MSTART, NSS remains active until the current amount of data to be transmitted is completed (ET flag is set), at which point NSS becomes inactive.

Applications can use a generic I/O port as an NSS pin for more flexible NSS applications.

Table 41-6. NSS function in master mode

Mode	Register configuration	Description
Master hardware NSS output mode	MSTMOD = 1 NSSIM = 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 become active level after enabling SPI and setting MSTART.
Master hardware NSS input mode	MSTMOD = 1 NSSIM = 0 NSSDRV = 0	Applicable to multi-master mode. At this time, NSS is configured as hardware input mode. Once the NSS pin become non-active level, 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 NSSIM = 1 NSSI = 0 NSSDRV: Don't care	Applicable to multi-master mode. When NSSDRV = 0, once NSSI become non-active level, SPI will automatically enter slave mode, and a master configuration error will occur and the CONFERR bit will be 1.
	MSTMOD = 1 NSSIM = 1 NSSI = 1 NSSDRV: Don't care	The slave can use hardware or software NSS mode.

NSS signal timing

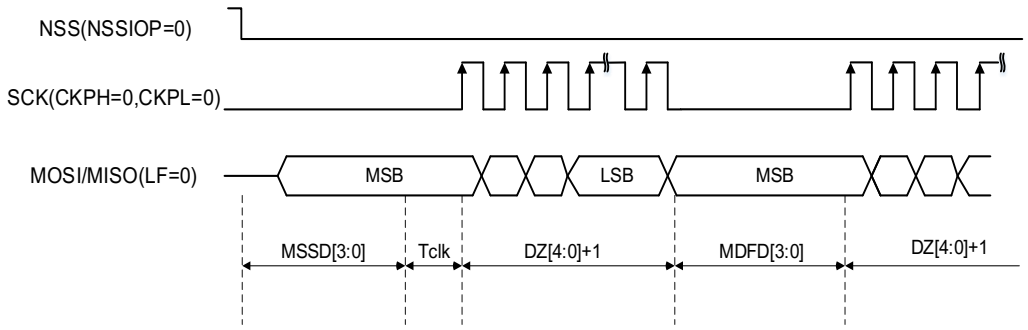
When applying hardware output NSS control (NSSIM = 0, NSSDRV = 1), users can configure MDFD[3:0] and MSSD[3:0] bit fields to control NSS signal timing between data frames and insert additional delay at the start of each transaction (to separate NSS and clock start).

Figure 41-8. NSS signal delay timing diagram (MSSD[3:0] = 0011 (3 × Tclk), MDFD = 0011 (3 x T_{clk})) shows the effective delay time between data signal and NSS signal with MSSD[3:0] = 3, and the effective delay time between data frames with MDFD[3:0] = 3.

Note: When NSSDRV is 0, MSSD must be configured as 0.

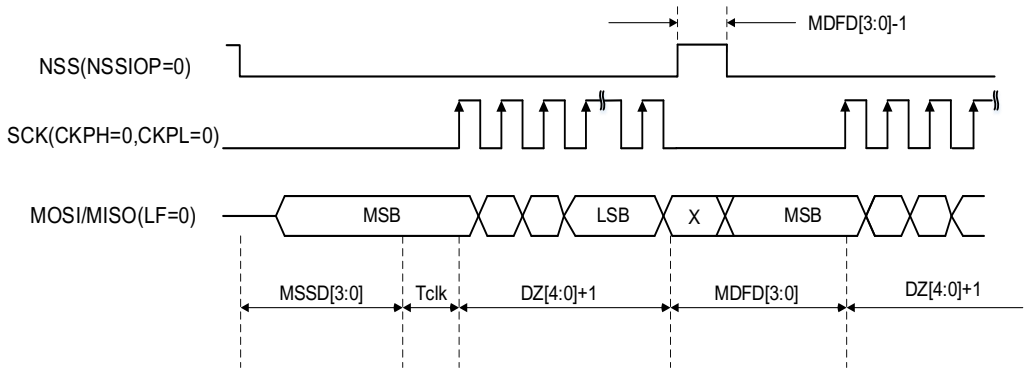
Figure 41-8. NSS signal delay timing diagram (MSSD[3:0] = 0011 (3 × Tclk), MDFD =

0011 (3 x T_{clk}))



When NSSCTL = 1 and MDFD[3:0] > 1, interlaced pulses can be inserted between SPI data frames. ***Figure 41-9. NSS interlaced pulses timing diagram (MSSD[3:0] = 0011 (3 x T_{clk}), MDFD = 0011 (3 x T_{clk}))*** shows the NSS signal pulse status with MDFD[3:0] > 1.

Figure 41-9. NSS interlaced pulses timing diagram (MSSD[3:0] = 0011 (3 x T_{clk}), MDFD = 0011 (3 x T_{clk}))



41.3.7. SPI operation modes

Table 41-7. SPI operation modes

Mode	Description	Register configuration	Data pin usage
MFD	Master full-duplex	MSTMOD = 1 RO = 0 BDEN = 0 BDOEN: Don't care	MOSI: Transmission MISO: Reception
MTU	Master transmission with unidirectional connection	MSTMOD = 1 RO = 0 BDEN = 0 BDOEN: Don't care	MOSI: Transmission MISO: Not used
MRU	Master reception with unidirectional connection	MSTMOD = 1 RO = 1 BDEN = 0 BDOEN: Don't care	MOSI: Not used MISO: Reception

Mode	Description	Register configuration	Data pin usage
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 41-10. A typical full-duplex connection

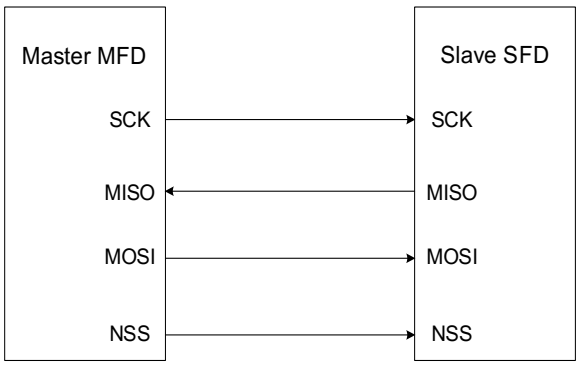


Figure 41-11. A typical simplex connection (Master: Receive, Slave: Transmit)

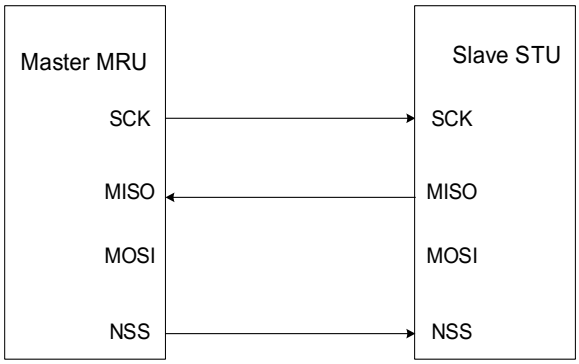


Figure 41-12. A typical simplex connection (Master: Transmit only, Slave: Receive)

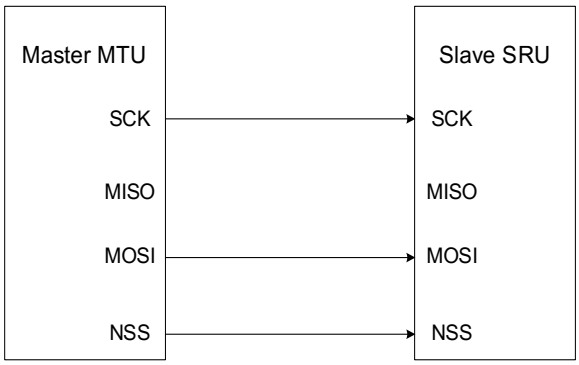
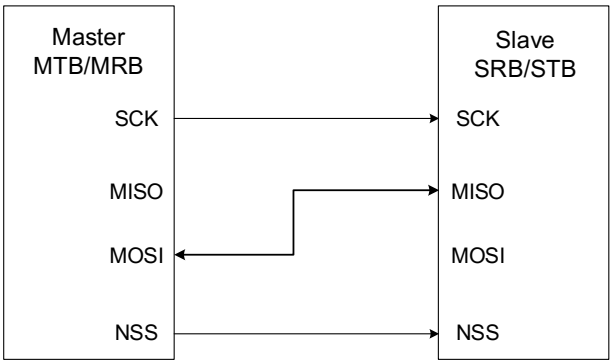


Figure 41-13. A typical bidirectional connection



SPI initialization sequence

Before transmitting or receiving data, application should follow the SPI initialization sequence described below:

1. If master mode or TI mode is used, program the PSC [2:0] bits in SPI_CFG0 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_CFG1 register).
3. Program the frame format (LF bit in the SPI_CFG1 register).

4. Program data format (DZ[4:0] bits in the SPI_CFG0 register).
5. Program the FIFO level (FIFOLVL[3:0] bits in the SPI_CFG0 register), and FIFO access size (WORDEN and BYTEN).
6. Program the NSS mode (NSSIM / NSSDRV / NSSIOPL / NSSCTL / MDFD[3:0] / MSSD[3:0] bits in the SPI_CFG1 register, NSSI bit in the SPI_CTL0 register) according to the application's demand as described above in ***NSS function*** section.
7. If slave mode, program the TXURDT[1:0] and TXUROP[1:0] bits in SPI_CFG0 register.
8. If TI mode is used, set TMOD bit in SPI_CFG1 register, otherwise, ignore this step.
9. Configure MSTMOD, RO, BDEN and BDOEN depending on the operating modes described in ***SPI operation modes*** section.
10. Program the SPI_CTL1 register to select the length of the transmission, and if the value is unknown, TXSIZE must be programmed to zero.
11. Program the SPI_CRCPOLY register, and program the CRCSZ[4:0] bit fields and CRCFS bit to configure CRC polynomials and CRC calculations as needed, described in ***CRC function*** section.
12. Initialize DMATEN / DMAREN bits if they are needed when DMA is used, described in ***DMA function*** section.
13. If Quad-SPI mode is used, set the QMOD bit in SPI_QCTL register. Ignore this step if Quad-SPI mode is not used.(only used in SPI3 / 4)
14. If you need to configure protection, program the IOAFEN bit of the SPI_CTL0 register.
15. Enable the SPI (set the SPIEN bit).
16. If master mode (MSTMOD =1), when SPIEN=1, program the MSTART bit in SPI_CTL0 to transfer data. Ignore this step if no data transfer is required.

Note: During communication, CKPH, CKPL, MSTMOD, PSC[2:0] and LF bits should not be changed.

SPI basic transmission and reception sequence

Transmission sequence

After the initialization sequence, the SPI is enabled and stays at idle state. In master mode, the transmission starts when the application writes a data into the TxFIFO. In slave mode the transmission starts when SCK clock signal begins to toggle at SCK pin and NSS level is active, so application should ensure that data is already written into TxFIFO before the transmission starts in slave mode.

When SPI begins to send a data frame, it first loads this data frame from the TxFIFO to the shift register and then begins to transmit the loaded data frame. The related operation is described in ***RxFIFO and TxFIFO*** sections.

Write access to SPI_TDATA is managed by the TP event. With the TP flag set to 1, the application performs an appropriate number of SPI data register writes to transfer the contents of the data package. After uploading the new complete package, the application checks the TP value to check whether TxFIFO can receive additional data packet, if TP = 1, they are uploaded packet by packet until TP reads 0. If the transmission size and packet size are not

aligned, when the last number of data packets to be transferred cannot reach the configured size (set by FIFOLVL). The application can still write standard number of previous complete packets to TxFIFO without adverse effects: only consistent data (complete data frames) will pull down to TxFIFO, while redundant write times (or any incomplete data) will be ignored.

In master mode, software should write the next data into SPI_TDATA register before the transmission of current data frame is completed if it desires to generate continuous transmission. As long as there is data in TxFIFO, data delivery continues until TxFIFO becomes empty.

NOTE: In polling mode, TXFIFO must not be written before SPIEN is enabled; in DMA mode, DMATEN must not be enabled before SPIEN is enabled.

Reception sequence

After the last valid sample clock, the incoming data will be moved from shift register to the RxFIFO and RP will be set to 1. The application should read SPI_RDATA register to get the received data and this will clear the RP flag automatically when the number of data less than FIFOLVL in RxFIFO. 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 TxFIFO is not empty. The related operation is described in ***RxFIFO and TxFIFO*** sections.

A read access to SPI_RDATA is managed by the RP event. With the RP flag set to 1, the application performs an appropriate number of SPI data register reads to download a single piece of data contents of the package. After the complete packet is downloaded, the application checks the RP value to see whether there are other packets in the RxFIFO, if any, they are downloaded packet by packet until RP reads 0. At the end of the receive, it may occur that some data is still available in the RxFIFO without reaching the FIFOLVL level, so RP will not be set to 1. In this case, the number of remaining RX data frames in the RxFIFO will be indicated by RWNE and RPLVL in the SPI_STAT register. If the transmission size and packet size are not aligned, the above condition occurs when the last number of data packets to be received cannot reach the configured size (set by FIFOLVL). However, the application can still read standard number of previous complete packets from RxFIFO without adverse effects: only consistent data (complete data frames) will pull up from RxFIFO, while redundant read times (or any incomplete data) will read 0.

When receiving data, the master provides clock signal, and the receiving process is stopped when the master stops or suspends the SPI interface. The master initiates the process by setting MSTART to 1, which can be suspended by writing 1 to the MSPDR of the SPI_CTL0 register, or by writing 1 to the MASP bit. The receiving process also complete when the data frames number in TXSIZE and TXSER are transferred.

SPI operation sequence in different modes (Not Quad-SPI, TI mode)

In full-duplex mode, either MFD or SFD, the RP and TP 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 RP and RXORERR 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 TP flag and read out RxFIFO in time after the RP 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 TP flag.

NOTE: When the SPI operates as a receiver, transmit-related register bits such as TC and TXF are also set.

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, CKPH, LF, NSSIM, NSSIOPL, NSSDRV bits take no effect and the SCK sample edge is falling edge.

Figure 41-14. Timing diagram of TI master mode with discontinuous transfer

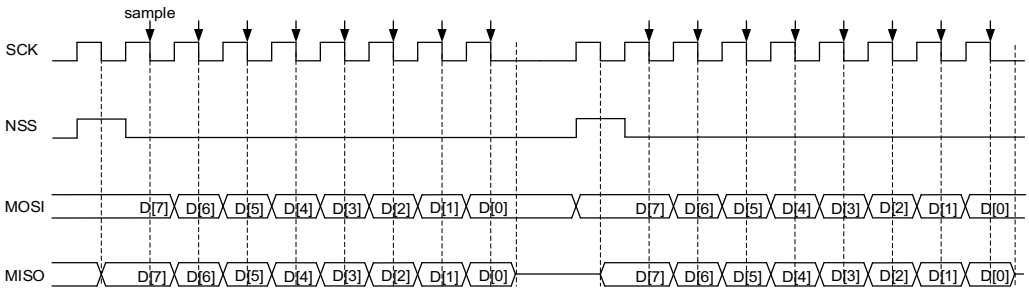
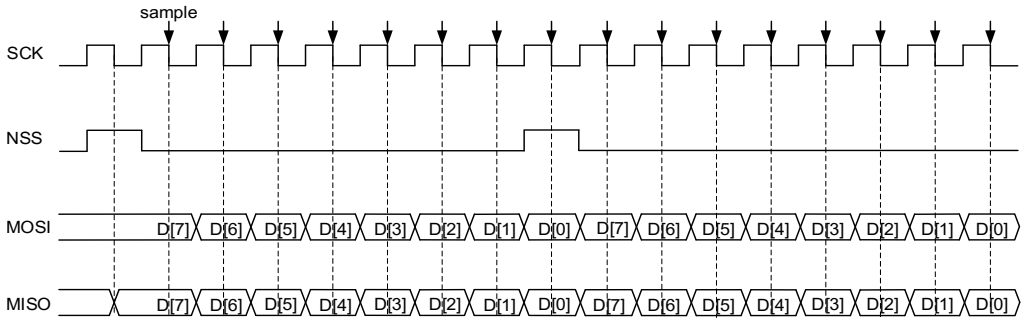


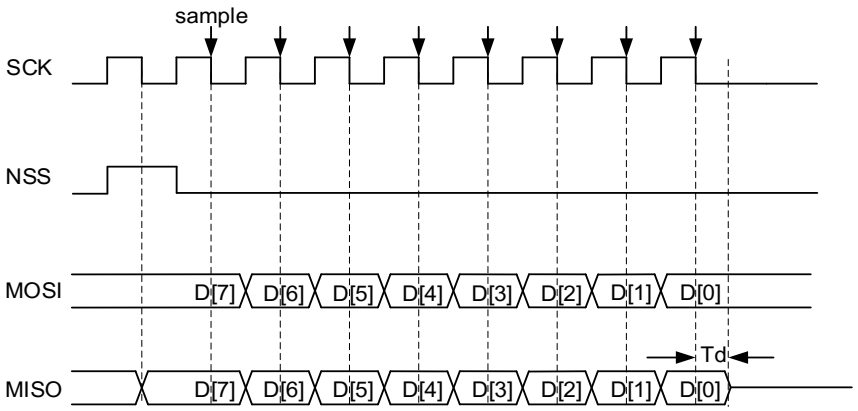
Figure 41-15. Timing diagram of TI master mode with continuous transfer



In master TI mode, SPI can perform continuous or non-continuous transfer. If the master writes SPI_TDATA 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 41-16. 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 MISO pin. This time is called T_d. T_d is decided by PSC [2:0] bits in SPI_CFG0 register.

$$\frac{T_{bit}}{2} + 2 * T_{kerclk} <= T_d \quad (41\text{-}1)$$

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.

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 TP and TC bits is set, then set QMOD bit in SPI_QCTL register. In Quad-SPI mode, BDEN, BDOEN, CRCEN, CRCSZ, RO and LF bits should be kept cleared and DZ[4:0] should be set to ensure that SPI data size is 8-bit, MSTMOD should be set to 1 for SPI is used in master mode. WORDEN should be set to 1. SPIEN, MSTART, TXSIZE, TXSER, PSC, CKPL and CKPH should be configured as desired.

Note: The CRC function is not supported in Quad-SPI mode. The PSC cannot be configured with two or four division frequencies.

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

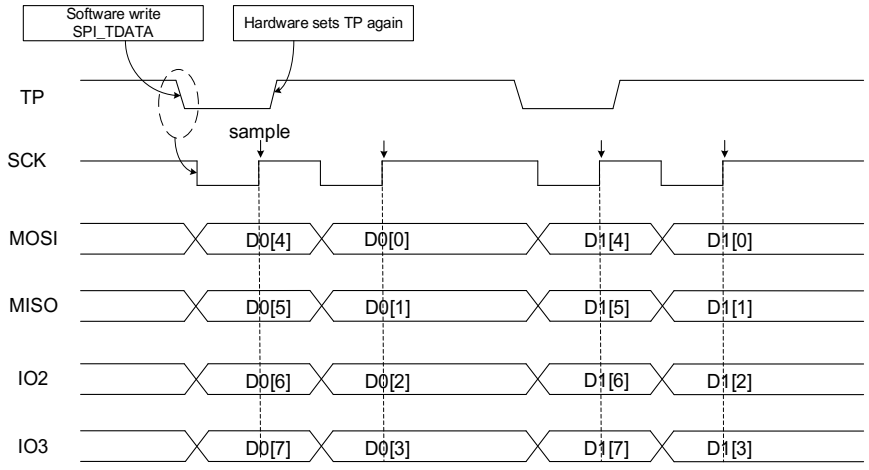
SPI works in quad write mode when QMOD is set and QRD is cleared in SPI_QCTL register. In this mode, MOSI, MISO, IO2 and IO3 are all used as output pins. SPI begins to generate

clock on SCK line and transmit data on MOSI, MISO, IO2 and IO3 as soon as data is written into SPI_TDATA (TP is cleared) with SPIEN and MSTART bits are set. Once SPI starts transmission, it always checks TP 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 / SPI_CTL1 / SPI_CFG0 / SPI_CFG1 based on your application requirements.
2. Set QMOD bit in SPI_QCTL register and then enable SPI by setting SPIEN in SPI_CTL0 register.
3. Write a byte to SPI_TDATA register and the TP will be cleared.
4. Wait until TP is set by hardware again before writing the next byte.

Figure 41-17. Timing diagram of quad write operation in Quad-SPI mode



Quad read operation

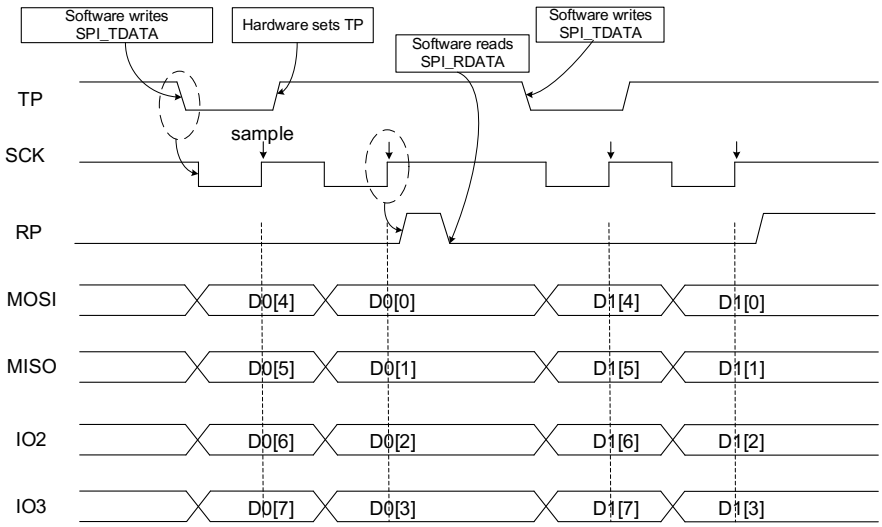
SPI works in quad read mode when QMOD and QRD are both set in SPI_QCTL register. In this mode, MOSI, MISO, IO2 and IO3 are all used as input pins. SPI begins to generate clock on SCK line as soon as a data is written into SPI_TDATA (TP is cleared) and SPIEN is set. Writing data into SPI_TDATA is only to generate SCK clocks, so the written data can be any value. Once SPI starts transmission, it always checks SPIEN and TP status at the end of a frame and stops when condition is not met. So, dummy data should always be written into SPI_TDATA 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 / SPI_CTL1 / SPI_CFG0 / SPI_CFG1 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_TDATA register.
4. Wait until the RP flag is set and read SPI_RDATA to get the received byte.

5. Write an arbitrary byte (for example, 0xFF) to SPI_TDATA to receive the next byte.

Figure 41-18. 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 MTU MTB SFD STU STB

Any transaction can be terminated when the device in full duplex or send only mode which stops providing data to be sent. In this In this case, the clock stops after the last data has been transferred. The TC flag can be polled (or enable interrupt through ESTCIE = 1) to wait for the last data frame to be sent. Wait for TC = 1 or ET = 1 (no more data to be sent and send the last data frame). If enable CRC function, CRC will be sent automatically after the last data is processed. In this case TC/ET will be set to 1 after CRC frame completion. When sending is suspended, the software must wait until the MSTART bit is cleared. Then disable the SPI by clearing SPIEN bit.

MRU MRB

To stop peripherals, SPI communication must first be suspended by setting MSPDR to 1 or wait for data transmission complete according to ET flag when the master device is in receive-only mode. If the receive flow is paused, please wait for SPD = 1. When the SPI is suspended, the data received but not read is always stored in the RxFIFO (When SPI is disable, RxFIFO will be cleared). Read all RxFIFO data (until RWNE = 0 and RPLVL = 0). Then disable the SPI by clearing SPIEN bit.

SRU SRB

Application can disable the SPI when it doesn't want to receive data, and any ongoing data will be lost.

TI mode

The disabling sequence of TI mode is the same as the sequences described above.

Quad-SPI mode

Application can operate as MFD mode, then the QMOD bit in SPI_QCTL register and SPIEN bit in SPI_CTL0 register are cleared.

41.3.8. 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_CFG0 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 initialization finish, If DMATEN is set, SPI will generate a DMA request each time when TP = 1, then DMA will acknowledge to this request and write data into the SPI_TDATA register automatically. If the data to transfer is not ready, TP and TXURERR will be set to 1. In this case, data will be sent based on the TXUROP bit selection. If DMAREN is set, SPI will generate a DMA request each time when RP = 1, then DMA will acknowledge to this request and read data from the SPI_RDATA register automatically. If the ET is set to 1 at the end of the transaction and the last packet is incomplete, DMA requests are automatically activated to read the rest of the data according to RWNE and RPLVL[1:0] settings(in SPI_STAT register).

Data packing with DMA

If the transfers are managed by DMA (DMATEN = 1 or DMAREN = 1), when DZ[4:0] <= 8-bit, and SPI_TDATA register is accessed in 16-bits or 32-bit, or 8-bit < DZ[4:0] <= 16-bit , and SPI_TDATA register is accessed in 32-bit, the DMA data packing mode is enabled, the DMA should automatically manages the write operations to the SPI_TDATA register.

Regardless of the data packing mode used, and regardless of whether the number of data to be transferred is multiple of DMA data size (16 or 32 bits). When the frame size is small, DMA will automatically complete the transfer based on the TXSIZE field setting. To configure DMA, DMA data access that is smaller than the configured data size is forbidden. Always be sure the data to access at least one complete data frames.

41.3.9. 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. The polynomial length is defined by the most significant bit of SPI_CRCPOLY register. If DZ <= 32-bit, CRC polynomial length supports 5 - 33 bits. If DZ <= 16-bit, CRC polynomial length supports 5 - 17

bits. The CRC polynomial length must be greater than the DZ value. If DZ = 32-bit, the CRCFS bit in SPI_CTL0 register must be set to 1 to make sure CRC polynomial used in full scale mode. The CRCSZ bit fields in SPI_CFG0 register can define the most significant bit number processed by the CRC calculator and comparing with CRC frame.

Application can enable the CRC function by setting CRCEN bit in SPI_CFG0 register, it must be configured before SPI is enabled. The CRCPOLY value can only be modified when there is no data transmission, and it is recommended to configure it when CRCEN=0. 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. Sending and receiving of CRC value is implemented in the form of frame, the frame length is equal to the length of the CRCSZ bit fields in SPI_CFG0 register.

In transmission phase, hardware send the calculated CRC value stored in SPI_TCRC register automatically after the last data is written to the TxFIFO. In reception phase, the CRC value is stored into SPI_RCRC register after the last data is read from the RxFIFO. The CRC calculator will get CRC value by performing CRC calculation to the received data, and the CRC value will be compare with the SPI_RCRC value. When CRC checking fails, the CRCERR flag will be set. Writing 1 to CRCERRC bit in SPI_STATC register can clear the CRCERR bit.

Note: when the SPI is disable, or at the beginning of the new data sampling after the last data transmission is completed, the SPI_TCRC and SPI_RCRC registers will be initialized, the value of the initialization can be defined by TXCRCI and RXCRCI bit in SPI_CTL0 register.

When SPI is configured as slave mode and the CRC function is used, the CRC calculation will still be performed even when the NSS pin is high (when the NSS signal is high, as long as there is a clock pulse on the SCK pin, the CRC calculation will continue).This situation occurs when the master device alternately communicates with multiple slave devices. At this time, it is recommended to restart the CRC function when the NSS signal is low.When the slave device is not selected (NSS signal is high) and switches to being selected as a new slave device (NSS signal is low), in order to maintain the synchronization of the next CRC calculation results between the master and slave devices, the CRC values at both ends should be cleared. It is recommended to clear the CRC value according to the following steps:

1. Disable the SPI module (SPIEN=0);
2. Clear the CRCEN bit (CRCEN=0);
3. Set the CRCEN bit (CRCEN=1);
4. Enable the SPI module (SPIEN=1).

41.3.10. SPI interrupts

Status flags

- **Transmit packet space available flag (TP)**

This bit is set when the TxFIFO have enough available position to accommodate a packet, the software can write the next data packet to the TxFIFO by writing the SPI_TDATA register. This bit is cleared when the TxFIFO don't have enough space to place in the next packet, the software can not write the next data packet to the TxFIFO by writing the SPI_TDATA register.

■ **Receive packet space available flag (RP)**

This bit is set when the RxFIFO is not empty, which means that at least one data packet is received and stored in the receive buffer, and software can read the data packet by reading the SPI_RDATA register. This bit is cleared when the RxFIFO is empty or the data stored in the RxFIFO can not reach the FIFOLVL. So software can not read the data packet by reading the SPI_RDATA register when RxFIFO is empty. Or in this case, the number of remaining RX data frames in the RxFIFO will be indicated by RWNE and RPLVL in the SPI_STAT register, the application can still read standard number of previous complete packets from RxFIFO without adverse effects.

■ **End of transmission/reception flag (ET)**

ET is a status flag to indicate whether the transmission/reception is ongoing or end. After complete transmission, i.e., when TXSIZE data volume is sent or received based on SPI, this flag is set by hardware and can be cleared by software set ETC bit in SPI_STATC register. The ET flag triggers an interrupt at ESTCIE is set to 1.

■ **Duplex packet flag (DP)**

If the TP and RP flags are set to 1, the DP flag is set to 1, which means TxFIFO has space for write operations and RxFIFO contains at least one packet for read operations. DP is useful for full-duplex communication, optimizing data upload/download performance, thus minimizing the need for CPU bandwidth and system power, especially when SPI is operating in stop mode.

■ **TxFIFO transmission has been filled flag (TXF)**

When all packets of one transmission are sent by the application or DMA, which means the TXSIZE data volume has been pushed into TxFIFO, TXF flag will be set to 1 by hardware. This bit can be cleared by the software writes 1 to the TXFC bit of SPI_STATC register. The TXF flag triggers an interrupt at TXFIE is set to 1.

■ **Additional number of SPI data to be transacted was reload flag (TXSERF)**

After processing the number of data in TXSIZE, if TXSER contains non-zero value, the contents of TXSER are copied to TXSIZE and the TXSER value is automatically cleared. The transmission will then increase the amount of data corresponding to the newly loaded value in TXSIZE. After the amount of data have been sent to TxFIFO, the TXSERF flag is set to 1 and triggers an interrupt at TXSERFIE is set to 1.

■ **Suspend flag (SPD)**

In master mode, the device automatically suspends the receive mode when the current frame is completed or the RxFIFO is full (MASP is set to 1 in SPI_CTL0 register), after MSPDR is

executed, SPD is set to 1 by hardware. The SPD flag can be cleared by writing 1 to SPDC bit in SPI_STATC register.

■ **Transmission complete flag (TC)**

This flag is changed by hardware. If TXSIZE = 0, TxFIFO is empty, TC is set to 1, and there is no activity on the bus. If TXSIZE > 0, TC will be set to 1 at the end of the transmission regardless of TxFIFO usage. When TC is set to 1, the transmission is finish. The CRC mode is enabled, TC will be set to 1 after CRC is sent. The TC flag triggers an interrupt at ESTCIE is set to 1.

Error conditions

■ **Configuration 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 and NSSDRV = 0, the CONFERR is set when the NSSI 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 CONFERR can be cleared by writing 1 to the CONFERRC bit of SPI_STATC register. The CONFERR flag triggers an interrupt at CONFEIE is set to 1.

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.

■ **Receive overrun error (RXORERR)**

The RXORERR bit is set if a data is received when the RxFIFO has not enough space to store this received data. The RxFIFO contents won't be covered with the newly incoming data, so the newly incoming data is lost. The RXORERR flag triggers an interrupt at RXOREIE is set to 1. The RXORERR can be cleared by writing 1 to the RXORERRC bit of SPI_STATC register.

■ **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. The FERR flag triggers an interrupt at FEIE is set to 1. The FERR can be cleared by writing 1 to the FERRC bit of SPI_STATC register.

■ **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. The CRCERR flag triggers an interrupt at CRCERIE is set to 1. The CRCERR can be cleared by writing 1 to the CRCERRC bit of SPI_STATC register.

■ **Transmit underrun error (TXURERR)**

In slave transfer mode, if the TxFIFO is empty, and need to send new data into the transfer shift register, the TXURERR bit is set. After an underrun is caught, the next data supplied for sending depends on the TXUROP, WORDEN, BYTEN bit. The TXURERR flag triggers an interrupt at TXUREIE is set to 1. The TXURERR can be cleared by writing 1 to the TXURERRC bit of SPI_STATC register.

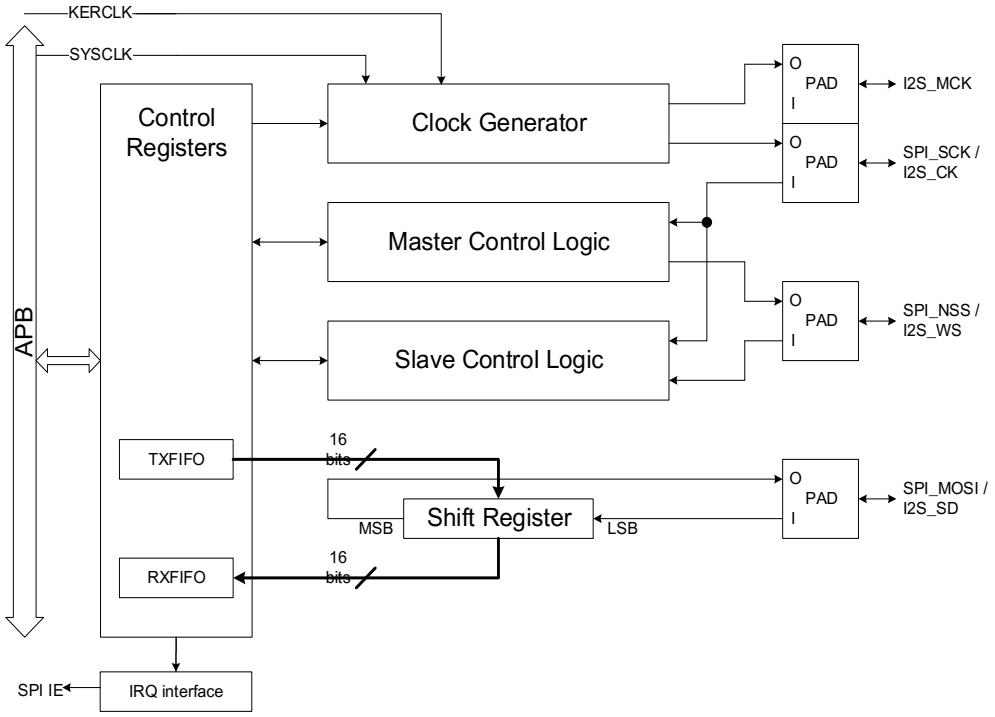
Table 41-8. SPI interrupt requests

Flag	Description	Clear method	Interrupt enable bit
TP	Transmit packet space available flag	TP is cleared by hardware when TxFIFO space less than FIFOLVL	TPIE
RP	Receive packet space available flag	RP is cleared by hardware when the data in RxFIFO less than FIFOLVL	RPIE
ET	End of transmission/reception flag	ETC set to 1	ESTCIE
DP	Duplex packet flag	DP is cleared by hardware when TP and RP are cleared	DPIE
TXF	TxFIFO transmission has been filled flag	TXFC set to 1	TXFIE
TXSERF	Additional number of SPI data to be transacted was reload flag	TXSERFC set to 1	TXSERFIE
TC	Transmission complete flag	TC is cleared by hardware when transmission start	ESTCIE
CONFERR	Configuration error	CONFERRC set to 1	CONFEIE
RXORERR	Receive overrun error	RXORERRC set to 1	RXOREIE
FERR	Format error	FERRC set to 1	FEIE
CRCERR	CRC error	CRCERRC set to 1	CRCERIE
TXURERR	Transmit underrun error	TXURERRC set to 1	TXUREIE

41.4. I2S function overview

41.4.1. I2S block diagram

Figure 41-19. Block diagram of I2S



- SYSCLK: system clock, provided by APB bus.
- KERCLK: kernel clock, provided by RCU. It has an asynchronous relationship with the system clock.
- The clock signal frequency must be consistent with user conditions and data transmission speed to prevent data loss. (**NOTE**: suggest the frequency of SYSCLK is greater than or equal to KERCLK)
- The SCK signal of the I2S slave is provided by the I2S master.

There are five sub modules to support I2S function, including control registers, clock generator, master control logic, slave control logic and shift register. All the user configuration registers are implemented in the control registers module, including the TxFIFO and RxFIFO. The clock generator is used to produce I2S communication clock in master mode. This clock generator is the source of MCK. The master control logic is implemented to generate the I2S_WS signal and control the communication in master mode. The slave control logic is implemented to control the communication in slave mode according to the received I2SCK and I2S_WS. The shift register handles the serial data transmission and reception on I2S_SD.

41.4.2. I2S signal description

There are four pins on the I2S interface, including I2S_CK, I2S_WS, I2S_SD and I2S_MCK. I2S_CK is the serial clock signal, which shares the same pin with SPI_SCK. I2S_WS is the frame control signal, which shares the same pin with SPI_NSS. I2S_SD is the serial data signal, which shares the same pin with SPI_MOSI. I2S_MCK is the master clock signal. It produces a frequency rate equal to 256 x Fs, and Fs is the audio sampling frequency.

41.4.3. I2S audio standards

The I2S audio standard is selected by the I2SSTD bits in the SPI_I2SCTL register. Four audio standards are supported, including I2S Philips standard, MSB justified standard, LSB justified standard, and PCM standard. All standards except PCM handle audio data time-multiplexed on two channels (the left channel and the right channel). For these standards, the I2S_WS signal indicates the channel side. For PCM standard, the I2S_WS signal indicates frame synchronization information.

The data length and the channel length are configured by the DTLEN bits and CHLEN bit in the SPI_I2SCTL register. Since the channel length must be greater than or equal to the data length, four packet types are available. They are 16-bit data packed in 16-bit frame, 16-bit data packed in 32-bit frame, 24-bit data packed in 32-bit frame, and 32-bit data packed in 32-bit frame.

For all standards and packet types, the most significant bit (MSB) is always sent first. For all standards based on two channels time-multiplexed, the channel left is always sent first followed by the channel right.

I2S Philips standard

For I2S Philips standard, I2S_WS and I2S_SD are updated on the falling edge of I2S_CK, and I2S_WS becomes valid one clock before the data. The timing diagrams for each configuration are shown below.

Figure 41-20. I2S Philips standard timing diagram (DTLEN = 00, CHLEN = 0, CKPL = 0)

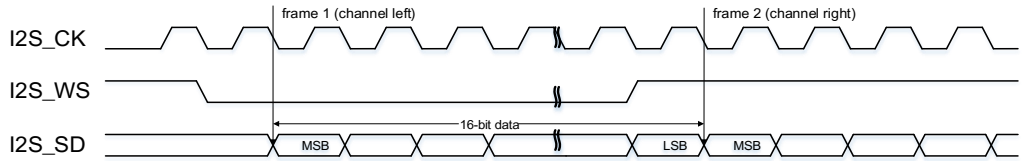


Figure 41-21. I2S Philips standard timing diagram (DTLEN = 00, CHLEN = 0, CKPL = 1)

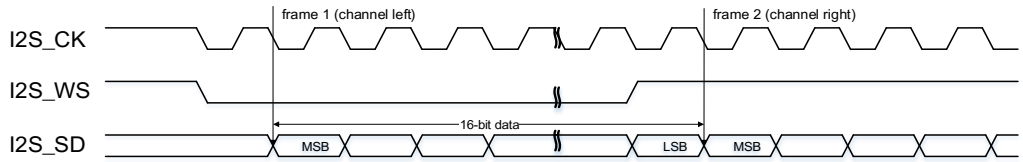


Figure 41-22. I2S Philips standard timing diagram (DTLEN = 10, CHLEN = 1, CKPL = 0)

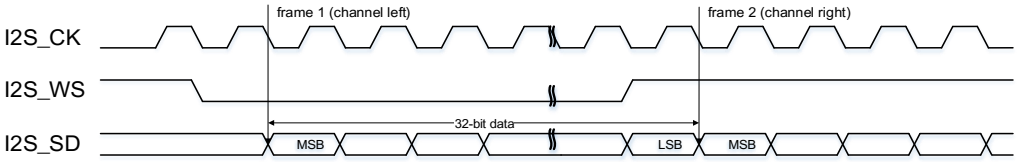


Figure 41-23. I2S Philips standard timing diagram (DTLEN = 10, CHLEN = 1, CKPL = 1)

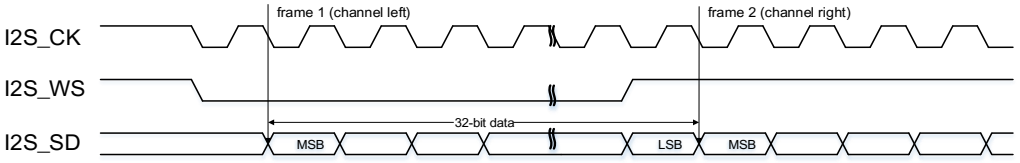


Figure 41-24. I2S Philips standard timing diagram (DTLEN = 01, CHLEN = 1, CKPL = 0)

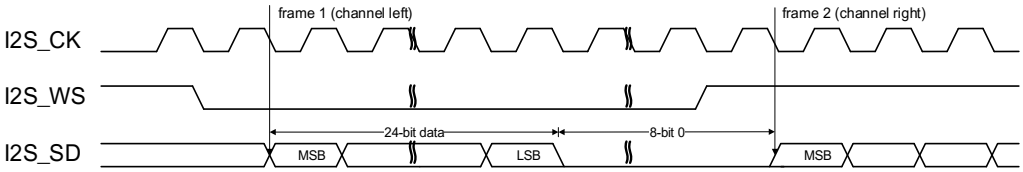


Figure 41-25. I2S Philips standard timing diagram (DTLEN = 01, CHLEN = 1, CKPL = 1)

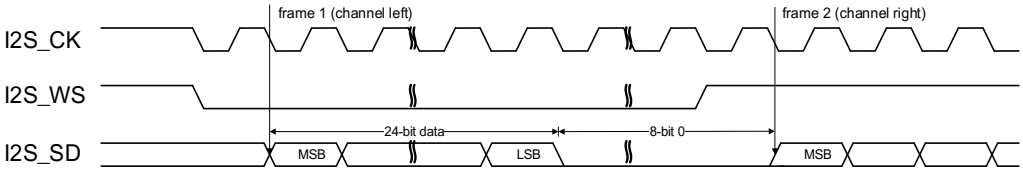


Figure 41-26. I2S Philips standard timing diagram (DTLEN = 00, CHLEN = 1, CKPL = 0)

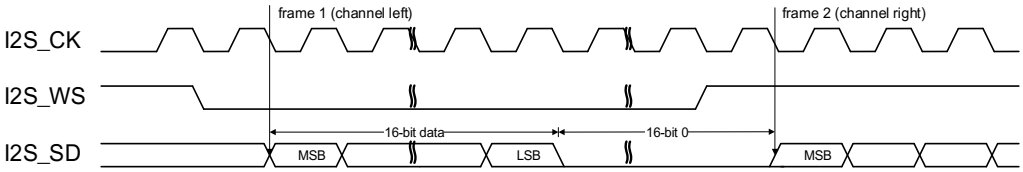
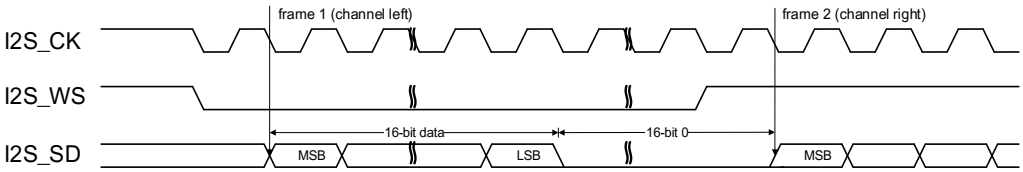


Figure 41-27. I2S Philips standard timing diagram (DTLEN = 00, CHLEN = 1, CKPL = 1)



MSB justified standard

For MSB justified standard, I2S_WS and I2S_SD are updated on the falling edge of I2S_CK. The timing diagrams for each configuration are shown below.

Figure 41-28. MSB justified standard timing diagram (DTLEN = 00, CHLEN = 0, CKPL = 0)

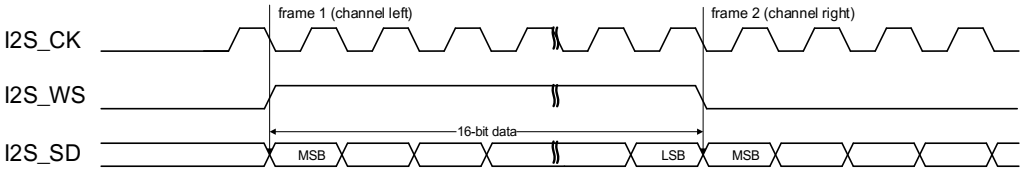


Figure 41-29. MSB justified standard timing diagram (DTLEN = 00, CHLEN = 0, CKPL = 1)

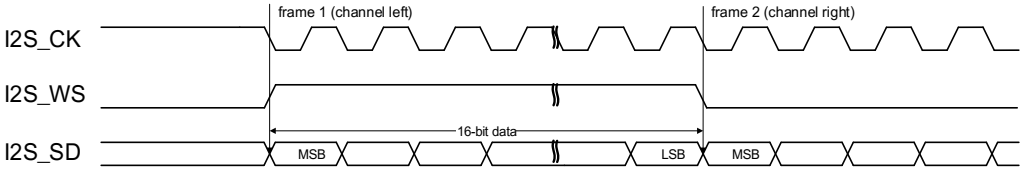


Figure 41-30. MSB justified standard timing diagram (DTLEN = 10, CHLEN = 1, CKPL = 0)

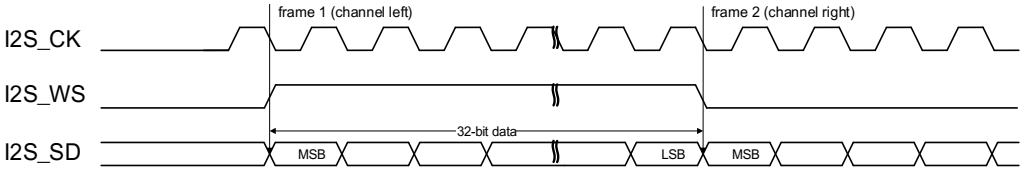


Figure 41-31. MSB justified standard timing diagram (DTLEN = 10, CHLEN = 1, CKPL = 1)

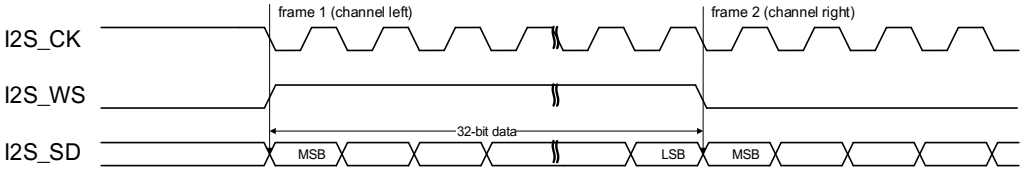


Figure 41-32. MSB justified standard timing diagram (DTLEN = 01, CHLEN = 1, CKPL = 0)

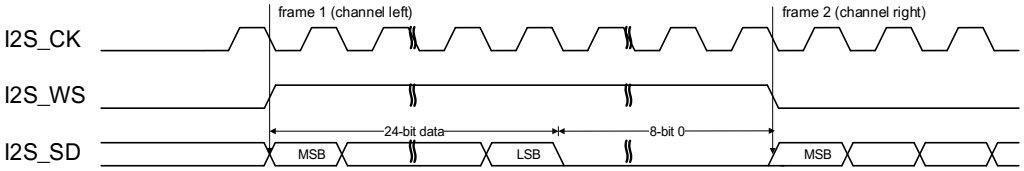


Figure 41-33. MSB justified standard timing diagram (DTLEN = 01, CHLEN = 1, CKPL =

1)

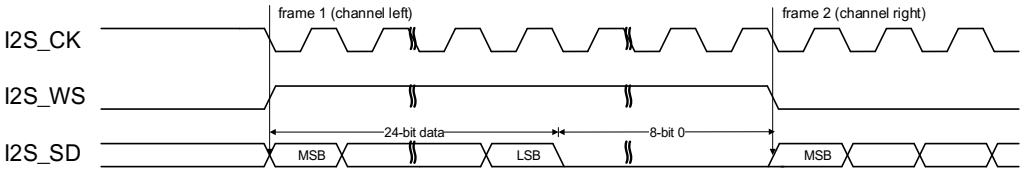


Figure 41-34. MSB justified standard timing diagram (DTLEN = 00, CHLEN = 1, CKPL = 0)

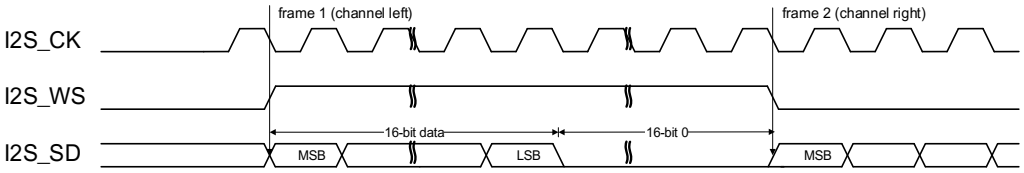
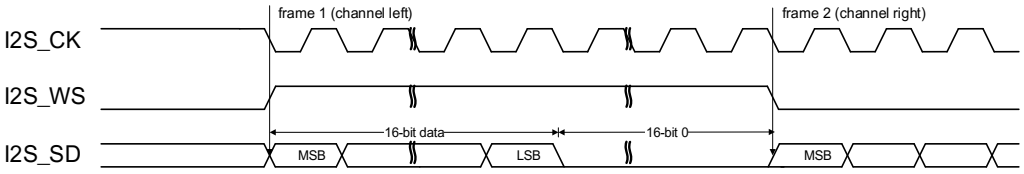


Figure 41-35. MSB justified standard timing diagram (DTLEN = 00, CHLEN = 1, CKPL = 1)



LSB justified standard

For LSB justified standard, I2S_WS and I2S_SD are updated on the falling edge of I2S_CK. In the case that the channel length is equal to the data length, LSB justified standard and MSB justified standard are exactly the same. In the case that the channel length is greater than the data length, the valid data is aligned to LSB for LSB justified standard while the valid data is aligned to MSB for MSB justified standard. The timing diagrams for the cases that the channel length is greater than the data length are shown below.

Figure 41-36. LSB justified standard timing diagram (DTLEN = 01, CHLEN = 1, CKPL = 0)

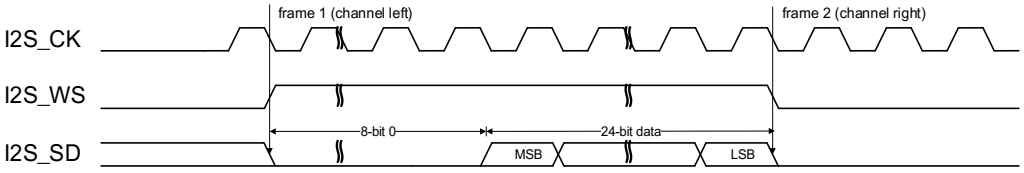


Figure 41-37. LSB justified standard timing diagram (DTLEN = 01, CHLEN = 1, CKPL =

1)

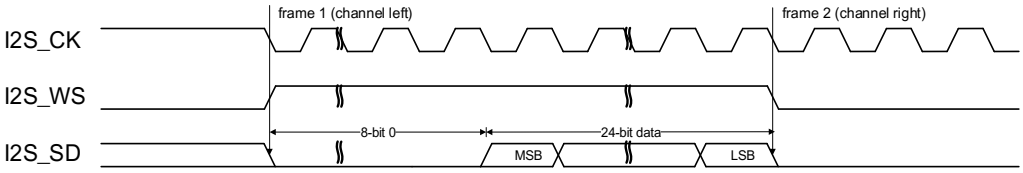


Figure 41-38. LSB justified standard timing diagram (DTLEN = 00, CHLEN = 1, CKPL = 0)

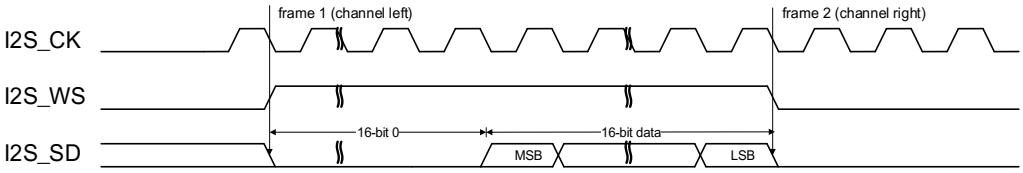
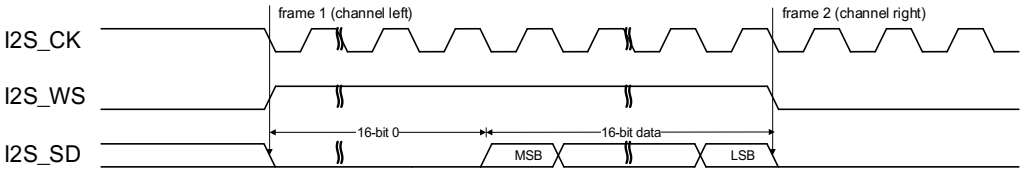


Figure 41-39. LSB justified standard timing diagram (DTLEN = 00, CHLEN = 1, CKPL = 1)



PCM standard

For PCM standard, I2S_WS and I2S_SD are updated on the rising edge of I2S_CK, and the I2S_WS signal indicates frame synchronization information. Both the short frame synchronization mode and the long frame synchronization mode are available and configurable using the PCMSMOD bit in the SPI_I2SCTL register. The SPI_TDATA / SPI_RDATA register is handled in the exactly same way as that for I2S Philips standard. The timing diagrams for each configuration of the short frame synchronization mode are shown below.

Figure 41-40. PCM standard short frame synchronization mode timing diagram (DTLEN = 00, CHLEN = 0, CKPL = 0)

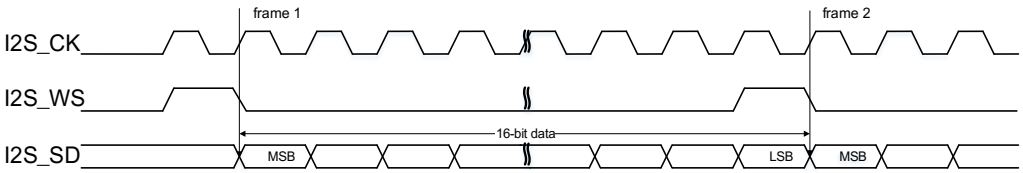


Figure 41-41. PCM standard short frame synchronization mode timing diagram (DTLEN

= 00, CHLEN = 0, CKPL = 1)

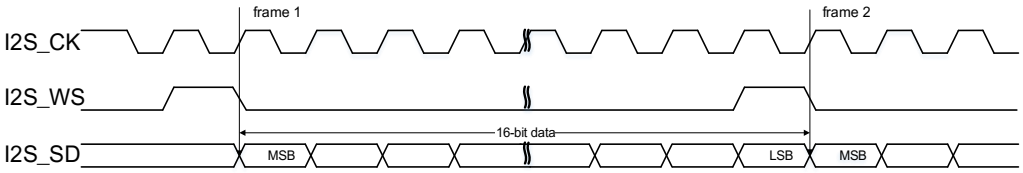


Figure 41-42. PCM standard short frame synchronization mode timing diagram (DTLEN = 10, CHLEN = 1, CKPL = 0)

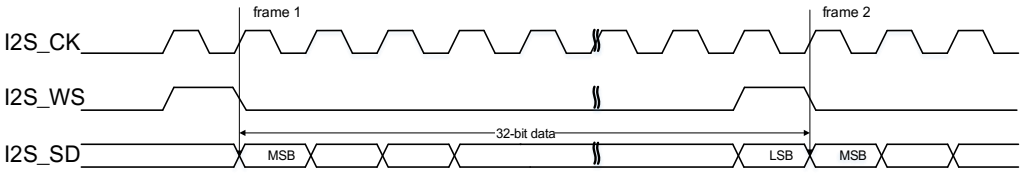


Figure 41-43. PCM standard short frame synchronization mode timing diagram (DTLEN = 10, CHLEN = 1, CKPL = 1)

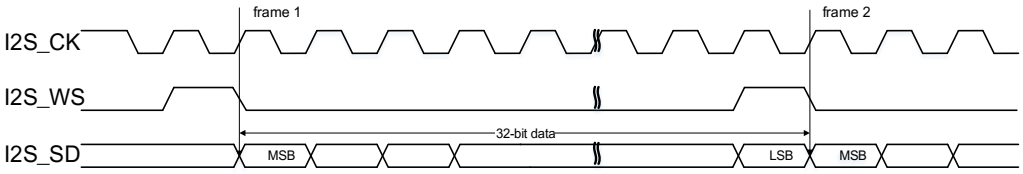


Figure 41-44. PCM standard short frame synchronization mode timing diagram (DTLEN = 01, CHLEN = 1, CKPL = 0)

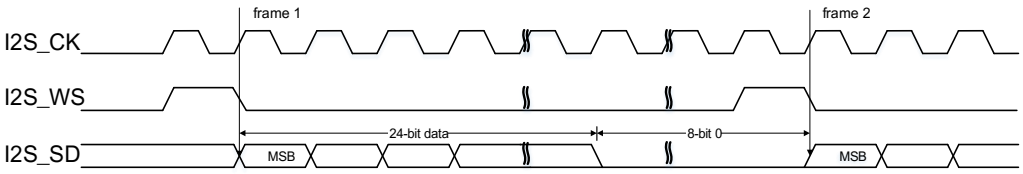


Figure 41-45. PCM standard short frame synchronization mode timing diagram (DTLEN = 01, CHLEN = 1, CKPL = 1)

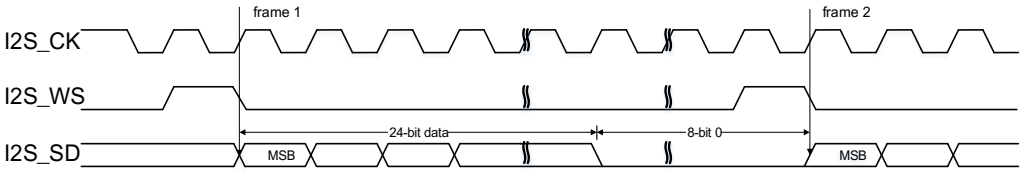


Figure 41-46. PCM standard short frame synchronization mode timing diagram (DTLEN = 00, CHLEN = 1, CKPL = 0)

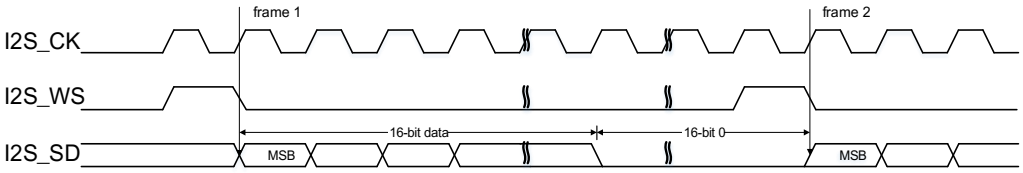
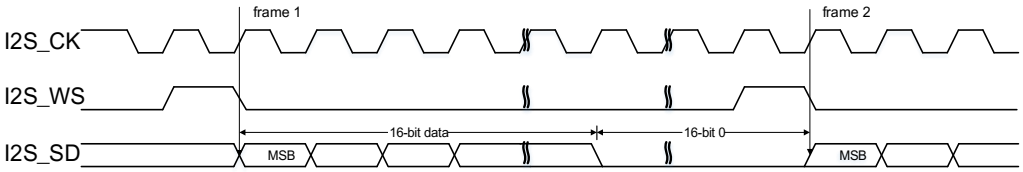


Figure 41-47. PCM standard short frame synchronization mode timing diagram (DTLEN = 00, CHLEN = 1, CKPL = 1)



The timing diagrams for each configuration of the long frame synchronization mode are shown below.

Figure 41-48. PCM standard long frame synchronization mode timing diagram (DTLEN = 00, CHLEN = 0, CKPL = 0)

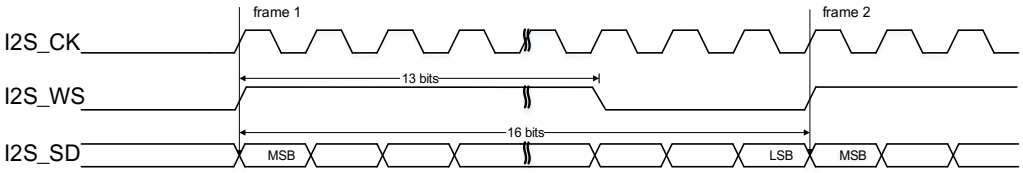


Figure 41-49. PCM standard long frame synchronization mode timing diagram (DTLEN = 00, CHLEN = 0, CKPL = 1)

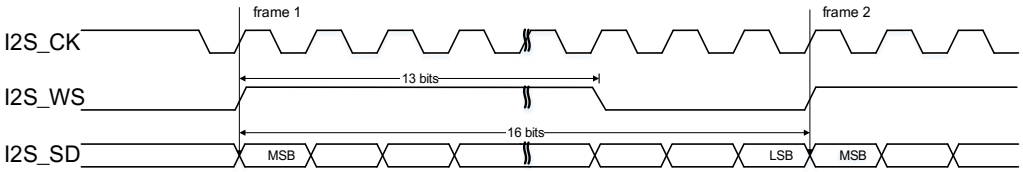


Figure 41-50. PCM standard long frame synchronization mode timing diagram (DTLEN = 10, CHLEN = 1, CKPL = 0)

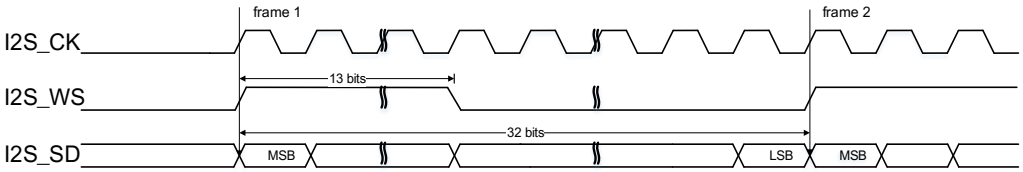


Figure 41-51. PCM standard long frame synchronization mode timing diagram (DTLEN = 10, CHLEN = 1, CKPL = 1)

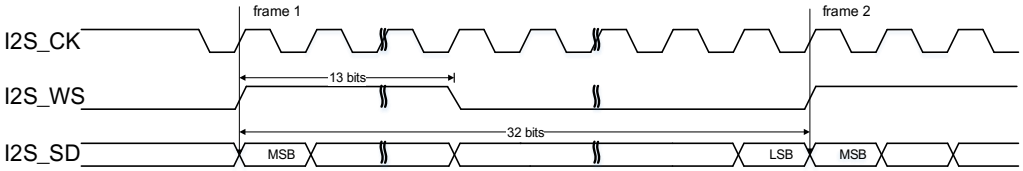


Figure 41-52. PCM standard long frame synchronization mode timing diagram (DTLEN

= 01, CHLEN = 1, CKPL = 0)

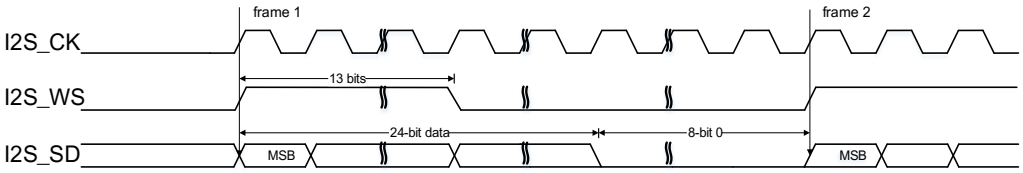


Figure 41-53. PCM standard long frame synchronization mode timing diagram (DTLEN = 01, CHLEN = 1, CKPL = 1)

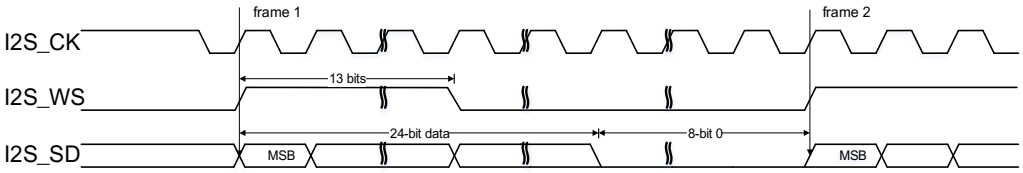


Figure 41-54. PCM standard long frame synchronization mode timing diagram (DTLEN = 00, CHLEN = 1, CKPL = 0)

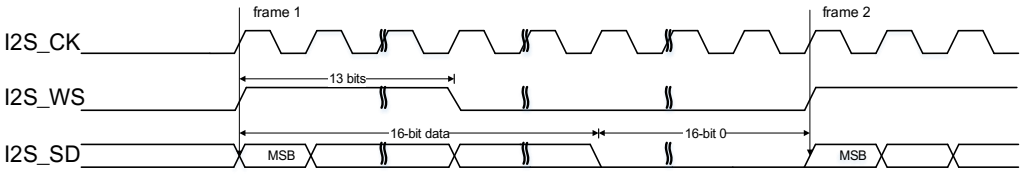
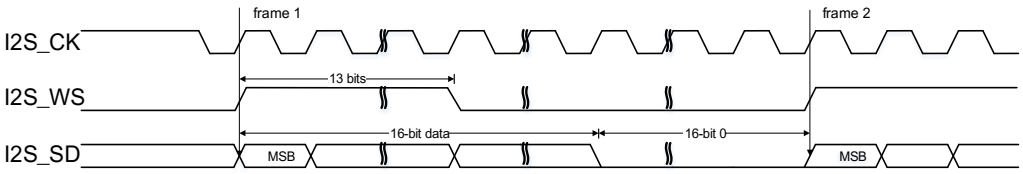
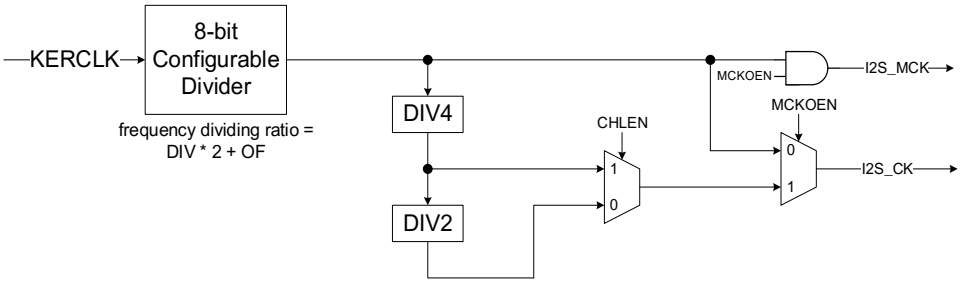


Figure 41-55. PCM standard long frame synchronization mode timing diagram (DTLEN = 00, CHLEN = 1, CKPL = 1)



41.4.4. I2S clock

Figure 41-56. Block diagram of I2S clock generator



The block diagram of I2S clock generator is shown as ***Figure 41-56. Block diagram of I2S clock generator***. The I2S interface clocks are configured by the DIV bits, the OF bit, the MCKOEN bit and the CHLEN bit in the SPI_I2SCTL register. The source clock is the kernel clock (KERCLK). The I2S bitrate can be calculated by the formulas shown in ***Table 41-9. I2S***

bitrate calculation formulas.

Table 41-9. I2S bitrate calculation formulas

MCKOEN	CHLEN	Formula
0	0	KERCLK / (DIV * 2 + OF)
0	1	KERCLK / (DIV * 2 + OF)
1	0	KERCLK / (8 * (DIV * 2 + OF))
1	1	KERCLK / (4 * (DIV * 2 + OF))

The relationship between audio sampling frequency (Fs) and I2S bitrate is defined by the following formula:

Fs = I2S bitrate / (number of bits per channel * number of channels)

So, in order to get the desired audio sampling frequency, the clock generator needs to be configured according to the formulas listed in *Table 41-10. Audio sampling frequency calculation formulas*.

Note: It is recommended that the configuration value of the I2S serial clock be set to less than 1/6 of the PCLK clock corresponding to the APB to which the I2S is mounted (excluding 1/6).

Table 41-10. Audio sampling frequency calculation formulas

MCKOEN	CHLEN	Formula
0	0	KERCLK / (32 * (DIV * 2 + OF))
0	1	KERCLK / (64 * (DIV * 2 + OF))
1	0	KERCLK / (256 * (DIV * 2 + OF))
1	1	KERCLK / (256 * (DIV * 2 + OF))

41.4.5. RxFIFO and TxFIFO

RxFIFO and TxFIFO are used in different directions for I2S data transactions, and they can enable the I2S to work in a continuous flow, and can prevent short data frame length or interrupt/DMA overrun occurs when the delay is too long.

A write access to the SPI_TDATA register stores the written data in the end of TxFIFO, while a read access to the SPI_RDATA returns the oldest value in RxFIFO which has not been read. In I2S mode, left audio sampling and right audio sampling are interleaved in the FIFO. This means that for the send operation, the user must start filling TxFIFO with left samples and then right samples, and so on. For receive mode, the first data read from RxFIFO is left channel, the next data is the right channel, and so on.

FIFO processing depends on the data length (DTLEN value), the size of the access to the FIFO register (16 or 32 bits). The size of TxFIFO / RxFIFO is 16 x 32 bits and the maximum access data frame length is 32 bits. According to different frame length, the maximum number of frames that can be stored in FIFO is described in *Table 41-11. The maximum number of*

data frame stored in I2SX FIFO. (N = FIFO size / 32 = 16 x 32 / 32 = 16)

Table 41-11. The maximum number of data frame stored in I2SX FIFO

Data length (DTLEN)	DTLEN = 16 bit	DTLEN = 24 bit	DTLEN = 32 bit
Frame numbers(WORDEN = 0)	N	-	-
Frame numbers(WORDEN = 1)	2N	N	N

According to the FIFO threshold of programmable can generate interrupts or DMA requests. The influence of FIFOLVL is consistent with SPI.

NOTE: SPI_TDATA and SPI_RDATA data is the default right aligned. Both RxFIFO and TxFIFO content is kept flushed when I2S is disable (I2SEN = 0).

41.4.6. Operation

Operation modes

The operation mode is selected by the I2SOPMOD bits in the SPI_I2SCTL register. There are four available operation modes, including master transmission mode, master reception mode, slave transmission mode, and slave reception mode. The direction of I2S interface signals for each operation mode is shown in the *Table 41-12. Direction of I2S interface signals for each operation mode*.

Table 41-12. Direction of I2S interface signals for each operation mode

Operation mode	I2S_MCK	I2S_CK	I2S_WS	I2S_SD
Master transmission	Output or NU(1)	Output	Output	Output
Master reception	Output or NU(1)	Output	Output	Input
Slave transmission	Input or NU(1)	Input	Input	Output
Slave reception	Input or NU(1)	Input	Input	Input

(1) NU means the pin is not used by I2S and can be used by other functions.

I2S initialization sequence

I2S initialization sequence contains seven steps shown below. In order to initialize I2S working in master mode, all the seven steps should be done. In order to initialize I2S to slave mode, only step 2, step 3, step 4, step 5 and step 6 should be done.

■ Step 1: Configure the DIV [7:0] bits, the OF bit, and the MCKOEN bit in the SPI_I2SCTL

register, in order to define the I2S bitrate and determine whether I2S_MCK needs to be provided or not.

- Step 2: Configure the CKPL in the SPI_I2SCTL register, in order to define the idle state clock polarity.
- Step 3: Configure the FIFO level (FIFOLVL[3:0] bits in the SPI_CFG0 register).
- Step 4: Configure the I2SSEL bit, the I2SSTD [1:0] bits, the PCMSMOD bit, the I2SOPMOD [1:0] bits, the DTLEN [1:0] bits, and the CHLEN bit in the SPI_I2SCTL register to define the I2S feature.
- Step 5: Configure the TPIE bit, the RPIE bit, the TXUREIE bit, the RXOREIE bit, the FEIE bit, the DMATEN bit, and the DMAREN bit, in order to select the potential interrupt sources and the DMA capabilities. This step is optional.
- Step 6: Set the I2SEN bit in the SPI_I2SCTL register to enable I2S.
- Step 7: Activate the serial interface by setting MSTART in the SPI_CTL0 register to 1.

I2S basic transmission and reception sequence

Transmission sequence

After the initialization sequence, the I2S is enabled and stays at idle state. In master mode, the transmission starts when the application writes a data into the TxFIFO. The application should ensure that data is already written into TxFIFO before the transmission starts in slave mode.

When I2S begins to send a data frame, it first loads this data frame from the TxFIFO to the shift register and then begins to transmit the loaded data frame.

Write access to SPI_TDATA is managed by the TP event. With the TP flag set to 1, the application performs an appropriate number of I2S data register writes to transfer the contents of the data package. After uploading the new complete package, the application checks the TP value to check whether TxFIFO can receive additional data packet, if TP = 1, they are uploaded packet by packet until TP reads 0. If the transmission size and packet size are not aligned, when the last number of data packets to be transferred cannot reach the configured size (set by FIFOLVL). The application can still write standard number of previous complete packets to TxFIFO without adverse effects: only consistent data (complete data frames) will pull down to TxFIFO, while redundant write times (or any incomplete data) will be ignored.

In master mode, software should write the next data into SPI_TDATA register before the transmission of current data frame is completed if it desires to generate continuous transmission. As long as there is data in TxFIFO, data delivery continues until TxFIFO becomes empty.

Reception sequence

After the last valid sample clock, the incoming data will be moved from shift register to the RxFIFO and RP will be set to 1. The application should read SPI_RDATA register to get the received data and this will clear the RP flag automatically when the number of data less than FIFOLVL in RxFIFO.

A read access to SPI_RDATA is managed by the RP event. With the RP flag set to 1, the application performs an appropriate number of SPI data register reads to download a single piece of data contents of the package. After the complete packet is downloaded, the application checks the RP value to see whether there are other packets in the RxFIFO, if any, they are downloaded packet by packet until RP reads 0. At the end of the receive, it may occur that some data is still available in the RxFIFO without reaching the FIFOLVL level, so RP will not be set to 1. In this case, the number of remaining RX data frames in the RxFIFO will be indicated by RWNE and RPLVL in the SPI_STAT register. If the transmission size and packet size are not aligned, the above condition occurs when the last number of data packets to be received cannot reach the configured size (set by FIFOLVL). However, the application can still read standard number of previous complete packets from RxFIFO without adverse effects: only consistent data (complete data frames) will pull up from RxFIFO, while redundant read times (or any incomplete data) will read 0.

I2S disabling sequence

I2S master disabling sequence:

- Step 1: Set the MSPDR bit to 1 to request that the data transfer be suspended.
- Step 2: Monitor the MSTART bit until it becomes 0.
- Step 3: Stop the bus clock and DMA function.
- Step 4: Clear the I2SEN bit in the SPI_I2SCTL to stop the I2S peripheral.

I2S slave disabling sequence:

- Step 1: Clear the I2SEN bit in the SPI_I2SCTL to stop the I2S peripheral.
- Step 2: Stop the bus clock and DMA function.

41.4.7. DMA function

DMA function is the same as SPI mode. The only difference is that the CRC function is not available in I2S mode.

41.4.8. I2S interrupts

Status flags

There are two status flags implemented in the SPI_STAT register (TP, RP), There is one status flag implemented in the SPI_I2SCTL register (I2SCH). The user can use them to fully monitor the state of the I2S bus.

- **Transmit packet space available flag (TP)**

This bit is set when the TxFIFO have enough available position to accommodate a packet, the software can write the next data packet to the TxFIFO by writing the SPI_TDATA register. This bit is cleared when the TxFIFO don't have enough space to place in the next packet, the software can not write the next data packet to the TxFIFO by writing the SPI_TDATA register.

■ **Receive packet space available flag (RP)**

This bit is set when the RxFIFO is not empty, which means that at least one data packet is received and stored in the receive buffer, and software can read the data packet by reading the SPI_RDATA register. This bit is cleared when the RxFIFO is empty or the data stored in the RxFIFO can not reach the FIFOLVL. So software can not read the data packet by reading the SPI_RDATA register when RxFIFO is empty. Or in this case, the number of remaining RX data frames in the RxFIFO will be indicated by RWNE and RPLVL in the SPI_STAT register, the application can still read standard number of previous complete packets from RxFIFO without adverse effects.

■ **I2S channel flag (I2SCH)**

This flag indicates the channel side information of the current transfer and has no meaning in PCM mode. It is updated when channel switch in transmission mode or in reception mode. This flag doesn't generate any interrupt.

Note: because of the existence of FIFO, the change of this bit have no relevant with TP / RP flag, and the change will happen at the end of channel transmission (channel transmission complete, which doesn't mean data transmission complete. For example, channel with 32 bits, data with 16 bits, channel transmission complete on behalf of the 32-bit data completed).

Error conditions

There are three error flags:

■ **Receive overrun error (RXORERR)**

The RXORERR bit is set if a data is received when the RxFIFO has not enough space to store this received data. The RxFIFO contents won't be covered with the newly incoming data, so the newly incoming data is lost. The RXORERR flag triggers an interrupt at RXOREIE is set to 1. The RXORERR can be cleared by writing 1 to the RXORERRC bit of SPI_STATC register.

Note: I2S mode has a hardware mechanism to prevent the error of data interchange between left and right channels due to overrun. For example, data receive sequence is L0->R0->L1->R1->L2->R2->L3->R3... LN->RN (L for left-channel data, R for right-channel data). When the overrun occurs after R1 is received, L2 data is lost. When RxFIFO recovers (which can receive data), the hardware will automatically discard R2 data, and receive L3 data to the left channel, and receive R3 data to the right channel. When the overrun occurs after L2 is received, R2 data is lost. When RxFIFO recovers (which can receive data), the hardware will automatically discard L3 data and receive R3 data to the right channel and receive L4 data to the left channel.

■ **Format error (FERR)**

In slave I2S mode, the slave also monitors the I2S_WS signal and set an error flag if the I2S_WS signal toggles at an incorrect position, for example: toggles at the middle bit of a byte. The FERR flag triggers an interrupt at FEIE is set to 1. The FERR can be cleared by writing

1 to the FERRC bit of SPI_STATC register.

■ **Transmit underrun error (TXURERR)**

In slave transfer mode, if the TxFIFO is empty, and need to send new data into the transfer shift register, the TXURERR bit is set. When this happens, at least one data is lost. The TXURERR flag triggers an interrupt at TXUREIE is set to 1. The TXURERR can be cleared by writing 1 to the TXURERRC bit of SPI_STATC register.

Note: I2S mode has a hardware mechanism to prevent the left and right channel data interchange error due to underrun. For example, data transfer sequence is L0->R0->L1->R1->L2->R2->L3->R3... LN->RN (L for left-channel data, R for right-channel data). Current underrun occurs after R1 is sent, L2 data is not timely transferred into TxFIFO resulting in TxFIFO is empty, the hardware will automatically transfer R1 data into the left channel, and then transfer R1 data to the right channel. When L2 data is transferred into TxFIFO, and then transfer L2 data to the left channel, and transfer R2 data to the right channel. When the underrun occurs after L2 is sent, R2 data is not timely transferred into TxFIFO resulting in TxFIFO is empty, the hardware will automatically transfer L2 data to the right channel, and then transfer L2 data to the left channel, when R2 data is transferred into TxFIFO, and then transfer R2 data to the right channel, and then transfer L3 data to the left channel.

I2S interrupt events and corresponding enabled bits are summed up in the ***Table 41-13. I2S interrupt***.

Table 41-13. I2S interrupt

Interrupt flag	Description	Clear method	Interrupt enable bit
TP	Transmit packet space available flag	TP is cleared by hardware when TxFIFO contains less than FIFOLVL empty locations	TPIE
RP	Receive packet space available flag	RP is cleared by hardware when RxFIFO contains less than FIFOLVL empty locations	RPIE
TXURERR	Transmit underrun error	TXURERRC set to 1	TXUREIE
RXORERR	Receive overrun error	RXORERRC set to 1	RXOREIE
FERR	Format error	FERRC set to 1	FEIE

41.5. Register definition

SPI0/I2S0 base address: 0x4001 3000

SPI1/I2S1 base address: 0x4000 3800

SPI2/I2S2 base address: 0x4000 3C00

SPI3 base address: 0x4001 3400

SPI4 base address: 0x4001 5000

SPI5/I2S5 base address: 0x4001 3800

41.5.1. Control register 0 (SPI_CTL0)

Address offset: 0x00
Reset value: 0x0000 0000

This register can be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															IOAFEN
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXCRCI	RXCRCI	CRCFS	NSSI	Reserved	MSPDR	MSTART	MASP	Reserved							SPIEN
rw	rw	rw	rw		w	rw	rw								rw

Bits	Fields	Descriptions
31:17	Reserved	Must be kept at reset value.
16	IOAFEN	Related Ios AF configuration enable 0: Related Ios AF configuration is enable 1: Related Ios AF configuration is disable This bit is set by software and cleared by hardware whenever SPIEN bit is changed from 1 to 0, or it can be cleared to 0 by software when the SPIEN bit and the CONFERR bit are 0. It is cleared and cannot be set when the CONFERR bit is set. This bit will be write protected when SPIEN is set. The SPI_CFG1 register cannot be changed when this bit is set.
15	TXCRCI	The transmitter CRC initialization configuration 0: All 0 mode 1: All 1 mode
14	RXCRCI	The receiver CRC initialization configuration 0: All 0 mode 1: All 1 mode

13	CRCFS	Full scale CRC polynomial configuration 0: Not use full scale CRC polynomial 1: Use full scale CRC polynomial
12	NSSI	The input level of internal NSS signal 0: NSS pin is pulled low 1: NSS pin is pulled high Only when the NSSIM position 1, this bit is valid. This bit values will effect to the peripheral NSS input, and ignore the I/O values of NSS pin.
11	Reserved	Must be kept at reset value.
10	MSPDR	Suspend request in SPI master mode 0: Do not request suspend 1: Request suspend Reading this bit returns zero. In SPI master mode, if this bit is set by software, SPI communication will be suspended and the MSTART bit will reset when the current frame transfer over. Application should check SPD flag in SPI_STAT register to check the end of transaction. Before SPI disabled, master communication must be suspended (either by setting this bit or emptying the SPI_TDATA register).
9	MSTART	Master start transfer 0: The master transfer is in idle status 1: The master transfer is occurring, or has been temporarily suspended by automatic suspend This bit can be set by software, and can be cleared by hardware when ET =1 in SPI_STAT register or when receiving suspend request. In SPI mode, only when SPIEN = 1 and MSTMOD = 1 (in SPI_CFG1 register), this bit can be set. In I2S / PCM mode, only when I2SEN = 1, this bit can be set.
8	MASP	The master is suspended automatically in receive mode 0: SPI stream/clock generation is continuous whether or not an overrun occurs 1: Before the overrun condition is reached, the SPI stream is suspended in the full RxFIFO state. SPD flag (in SPI_STAT register) will set 1 When SPI communication is suspended to prevent an overrun condition, several bits of the next frame may be synchronized out due to an internal synchronization delay. After reading the RxFIFO, communication resumes and subsequent bits transfers continue without any restrictions. For the same reason, automatic suspension is not very reliable when the data size is less than 8 bits. In this case, by combining the insert delay between data frames applied when MDFD parameters remain non-zero value to achieve safe suspension; The sum of data size and interleaving SPI cycles should always result in an interval of at least 8 SPI clock cycles. **NOTE:** MASP can be set only in receiving mode, otherwise it may cause RXORERR (reception overrun) error.

7:1	Reserved	Must be kept at reset value.
0	SPIEN	SPI enable 0: SPI peripheral is disabled 1: SPI peripheral is enabled This bit can be set and cleared by software and can not be set when CONFERR bit is set in SPI_STAT register.

41.5.2. Control register 1 (SPI_CTL1)

Address offset: 0x04
Reset value: 0x0000 0000

This register can be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXSER[15:0]															
rs															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXSIZE[15:0]															
rw															

Bits	Fields	Descriptions
31:16	TXSER[15:0]	When previous number of data stored in the TXSIZE has been transferred, it will reload the transmission expansion data amount in the TXSER to TXSIZE. These bits can only be set by the software when its value is zero. After the TXSIZE reload, it is cleared by the hardware. If it is the last time TXSIZE reload, the TXSER counter must to be previously written before CTXSIZE (in SPI_STAT register) counter reaches 1, in addition to this last reload, must program the TXSER value in advance before the CTXSIZE (in SPI_STAT register) counter reaches 1 (or reaches 2 if the CRCEN is set) and CTXSIZE counter is less than the last TXSER value minus 1, otherwise reloads will not be considered and communication will be terminated normally. **NOTE:** TXSER value need to be set greater than 1.
15:0	TXSIZE[15:0]	The current number of data to transfer These bits can be modified by software, and can not be modified when MSTART bit set 1. When TXSIZE is 0, and MSTART is set to 1, it will start endless transmission. When CRC is enable, TXSIZE cannot be set to 0xFFFF / 0x0001.

41.5.3. Configuration register 0 (SPI_CFG0)

Address offset: 0x08
Reset value: 0x0007 0007

This register can be accessed by word (32-bit). When SPI is enable, this register is write

protected, except DMATEN and DMAREN bits.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	PSC[2:0]			Reserved			WORDEN	BYTEN	CRCEN	Reserved	CRCSZ[4:0]				
	rw						rw	rw	rw				rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMATEN	DMAREN	Reserved	TXURDT[1:0]		TXUROP[1:0]		FIFOLVL[3:0]				DZ[4:0]				
rw	rw		rw		rw		rw				rw				

Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30:28	PSC[2:0]	Master clock prescaler selection 000: KERCLK / 2 001: KERCLK / 4 010: KERCLK / 8 011: KERCLK / 16 100: KERCLK / 32 101: KERCLK / 64 110: KERCLK / 128 111: KERCLK / 256 **NOTE:** the 000 / 001 configuration of PSC should not be used in Quad-SPI mode. **NOTE:** When PSC is configured as 000, data sizes below 8 bits should not be used in TI mode; when PSC is configured as 000, 4- to 5-bit data sizes should not be used in normal mode.
27:25	Reserved	Must be kept at reset value.
24	WORDEN	Word access enable This bit is used to indicate the width of access to the FIFO, and to generate the threshold of the RxFIFO for the RWNE. 0: Access according to BYTEN 1: Word access enable In I2S mode, in order to ensure the stability of channel data, WORDEN is associated with DTLEN. If DTLEN = 0, WORDEN must be 0. If DTLEN > 0, WORDEN must be 1.
23	BYTEN	Byte access enable This bit is used to indicate the width of access to the FIFO, and to generate the threshold of the RxFIFO for the RWNE. 0: Halfword access enable 1: Byte access enable in I2S mode, BYTEN must always be 0 to ensure stable channel data..
22	CRCEN	CRC calculation enable 0: CRC calculation is disabled

		1: CRC calculation is enabled
21	Reserved	Must be kept at reset value.
20:16	CRCSZ[4:0]	CRC size This bit fields must be equal to DZ value or multiples of DZ value. 00000: Not use 00001: Not use 00010: Not use 00011: 4-bit 00100: 5-bit 00101: 6-bit ….. 11101: 30-bit 11110: 31-bit 11111: 32-bit
15	DMATEN	Transmit buffer DMA enable 0: Transmit buffer DMA is disabled 1: Transmit buffer DMA is enabled
14	DMAREN	Receive buffer DMA enable 0: Receive buffer DMA is disabled 1: Receive buffer DMA is enabled
13	Reserved	Must be kept at reset value.
12:11	TXURDT[1:0]	Detection of underrun error at slave transmitter 00: The underrun detected condition is set at the beginning of the data frame (no first bit protection) 01: The underrun detected condition is set at the end of last data frame 10: The underrun detected condition is set at the beginning of NSS signal 11: Reserved
10:9	TXUROP[1:0]	Operation of slave transmitter when underrun detected 00: Slave send a constant value defined by the SPI_URDATA register 01: Slave send the data frame received from master lastly 10: Slave send the data frame which is lastly transmitted by itself. (This data frame is stored in its TxFIFO) 11: Reserved
8:5	FIFOLVL[3:0]	FIFO threshold level These bits show the number of data frames in a single packet. The size of the packet should not exceed half of the FIFO space. 0000: 1-data frame 0001: 2-data frame 0010: 3-data frame 0011: 4-data frame

….
1101: 14-data frame
1110: 15-data frame
1111: 16-data frame
The SPI interface is more efficient if the configured packet size is aligned with the parallel bits of data register access. It is best to choose FIFOLVL from 2, 4, 6 etc when DZ <= 8 bits and the SPI data register is accessed by half-word. If DZ > 8 bits and the SPI data register is accessed by word, it is best to choose FIFOLVL from 2, 4, 6 etc. While if DZ <= 8 bits, it is best to choose FIFOLVL from 4, 8, 12 etc.

4:0 DZ[4:0] Date size
These bits configure the number of bits in a data frame:
00000: Reserved
00001: Reserved
00010: Reserved
00011: 4-bit (When data width is 4 bit, must use the word / half word access FIFO, otherwise there is a risk of data disorder)
00100: 5-bit
00101: 6-bit
00110: 7-bit
……
11101: 30-bit
11110: 31-bit
11111: 32-bit

41.5.4. Configuration register 1 (SPI_CFG1)

Address offset: 0x0C
Reset value: 0x0000 0000

This register can be accessed by word (32-bit). When SPI is enable or IOAFEN bit is set to 1, this register is write protected.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFCTL	NSSCTL	NSSDRV	NSSIOPL	Reserved	NSSIM	CKPL	CKPH	LF	MSTMOD	TMOD	Reserved		BDEN	BDOEN	RO
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
SWPMIO	Reserved							MDFD[3:0]					MSSD[3:0]		
rw								rw					rw		

Bits	Fields	Descriptions
31	AFCTL	AF GPIOs control This bit can be set or cleared when SPI is disable. 0: Peripherals do not control GPIO pins when disabled

		1: Peripherals always control all associated GPIO pins When the SPI master must be temporarily disabled for specific configuration reasons (such as CRC reset, or CKPH change), setting this bit to 1 forces the relevant output configured for alternate function mode to the state corresponding to the current SPI configuration to prevent an exit burr. In slave mode, this bit should not be used, because any slave transmitter cannot enforce its MISO output once the SPI is disabled. **NOTE:** AFCTL is prohibited in MFD, MRU, and MRB modes.
30	NSSCTL	NSS pin output control in master mode 0: The NSS remains active level until the data transfer is complete, after that becomes inactive level via the ET flag 1: When MDFD[3:0] > 1, SPI data frames are interleaved with NSS invalid pulses
29	NSSDRV	NSS pin output enable in master mode 0: Output disable 1: Output enable
28	NSSIOPL	NSS pin input/output polarity selection 0: Low level is active 1: High level is active
27	Reserved	Must be kept at reset value.
26	NSSIM	NSS input signal manage mode 0: The NSS PAD decides the NSS input value 1: The NSSI bit of SPI_CTL0 decides the NSS input value
25	CKPL	Clock polarity selection 0: CLK pin is pulled low when SPI is in idle status 1: CLK pin is pulled high when SPI is in idle status
24	CKPH	Clock phase selection 0: Capture the first data at the first clock transition 1: Capture the first data at the second clock transition
23	LF	LSB first mode 0: Transmit MSB first 1: Transmit LSB first This bit has no meaning in SPI TI mode.
22	MSTMOD	Master mode enable 0: Slave mode 1: Master mode
21	TMOD	SPI TI mode enable 0: SPI TI mode disabled 1: SPI TI mode enabled

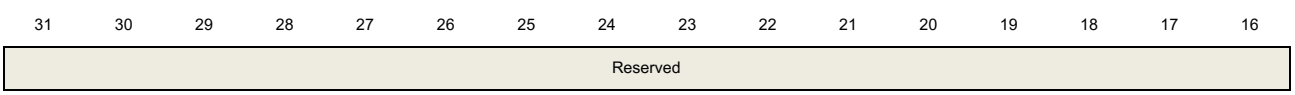
20:19	Reserved	Must be kept at reset value.
18	BDEN	Bidirectional enable 0: 2 line unidirectional transmit mode 1: 1 line bidirectional transmit mode. The information transfers between the MOSI pin in master and the MISO pin in slave
17	BDOEN	Bidirectional transmit output enable When BDEN is set, this bit determines the direction of transfer. 0: Work in receive-only mode 1: Work in transmit-only mode
16	RO	Receive only When BDEN is cleared, this bit determines the direction of transfer. 0: Full-duplex mode 1: Receive-only mode
15	SWPMIO	MOSI and MISO pin swap 0: No swap 1: Swap When this bit is set to 1, MISO and MOSI pin alternate functions are switched.
14:8	Reserved	Must be kept at reset value.
7:4	MDFD[3:0]	Delay between the data frames in SPI master mode 0000: No delay 0001: 1 clock cycle delay …. 1111: 15 clock cycle delay This bit has no meaning in SPI TI mode.
3:0	MSSD[3:0]	Delay between active edge of NSS and start transfer or receive data in SPI master mode 0000: No delay 0001: 1 clock cycle delay …. 1111: 15 clock cycle delay This bit has no meaning in SPI TI mode.

41.5.5. Interrupt register (SPI_INT)

Address offset: 0x10

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
Reserved					TXSERFIE	CONFEIE	FEIE	CRCERIE	RXOREIE	TXUREIE	TXFIE	ESTCIE	DPIE	TPIE	RPIE
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:11	Reserved	Must be kept at reset value.
10	TXSERFIE	TXSER reload interrupt enable 0: TXSER interrupt disable 1: TXSER interrupt enable
9	CONFEIE	SPI configuration error interrupt enable 0: SPI configuration error interrupt disable 1: SPI configuration error interrupt enable
8	FEIE	TI frame error interrupt enable 0: TI frame error interrupt disable 1: TI frame error interrupt enable
7	CRCERIE	CRC error interrupt enable 0: CRC error interrupt disable 1: CRC error interrupt enable
6	RXOREIE	Overrun error interrupt enable 0: Overrun interrupt disable 1: Overrun interrupt enable
5	TXUREIE	Underrun error interrupt enable 0: Underrun interrupt disable 1: Underrun interrupt enable
4	TXFIE	Transmission filled interrupt enable 0: TXF interrupt disable 1: TXF interrupt enable
3	ESTCIE	End of transfer or TxFIFO clear interrupt enable 0: ESTC interrupt disable 1: ESTC interrupt enable
2	DPIE	DP interrupt enable 0: DP interrupt disable 1: DP interrupt enable This bit can be set by software, and will be cleared when TXF is set to 1.
1	TPIE	TP interrupt enable 0: TP interrupt disable 1: TP interrupt enable

This bit can be set by software, and will be cleared when TXF is set to 1.

0	RPIE	RP interrupt enable 0: RP interrupt disable 1: RP interrupt enable

41.5.6. Status register (SPI_STAT)

Address offset: 0x14

Reset value: 0x0000 1002

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
CTXSIZE[15:0]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RWNE	RPLVL[1:0]		TC	SPD	TXSERF	CONFERR	FERR	CRCERR	RXORERR	TXURERR	TXF	ET	DP	TP	RP
r	r		r	r	r	r	r	r	r	r	r	r	r	r	r

Bits	Fields	Descriptions
31:16	CTXSIZE[15:0]	These bits show the number of data frames remaining in the TXSIZE bit field (in the SPI_CTL1 register). This value is not very reliable when there is data transmission on the bus.
15	RWNE	The word of RxFIFO is not empty 0: Data takes RxFIFO space less than a word (32-bit) 1: Data takes RxFIFO space at least a word (32-bit)
14:13	RPLVL[1:0]	RxFIFO packet level These bits define the number of data frames stored in the last 32-bit word area of RxFIFO. If the size of data frame <= 8 bit (DZ[4:0] <= 7): 00: there are no frame (RWNE = 0) or multiple of 4 frames (RWNE = 1) stored in RxFIFO 01: there is 1 frame stored in RxFIFO (RWNE = 0) 10: there are 2 frames stored in RxFIFO (RWNE = 0) 11: there are 3 frames stored in RxFIFO (RWNE = 0) If the size of data frame > 8 bit and <= 16 bit (7 < DZ[4:0] <= 15): invalid If the size of data frame > 16 bit (DZ[4:0] > 15): 00: only read Other: not used.
12	TC	TxFIFO transmission complete flag 0: There are data stored in TxFIFO, or the last frame is sent ongoing (including

		CRC) 1: The last data or CRC frame is sent complete
11	SPD	Suspend flag 0: SPI is not suspended 1: SPI master mode is suspended
10	TXSERF	The additional SPI data has been reloaded 0: no data accepted 1: The current transaction continues after receiving an additional number of data This bit can be cleared by write 1 to TXSERFC bit (in SPI_STATC register) or program TXSER bits (in SPI_CTL1 register).
9	CONFERR	SPI configuration error 0: No configuration error 1: Configuration error occurred
8	FERR	SPI TI format error 0: No TI format error 1: TI format error occurs
7	CRCERR	SPI CRC error 0: No CRC error 1: CRC error occurs
6	RXORERR	Reception overrun error 0: No reception overrun error 1: Reception overrun error occurs
5	TXURERR	Transmission underrun error 0: No transmission underrun error 1: Transmission underrun error occurred
4	TXF	TxFIFO transmission has been filled 0: TxFIFO upload ongoing or not start 1: TxFIFO upload complete
3	ET	End of transmission / reception flag 0: Transmission / reception ongoing or not start 1: Transmission / reception complete
2	DP	Duplex packet 0: TxFIFO is full and / or RxFIFO is empty 1: TxFIFO has space to write a complete data frame (TP = 1) and RxFIFO contains at least one packet to read (RP = 1)
1	TP	TxFIFO packet space available flag 0: TxFIFO don't have enough space to receive the next packet 1: TxFIFO have enough available space to receive a packet

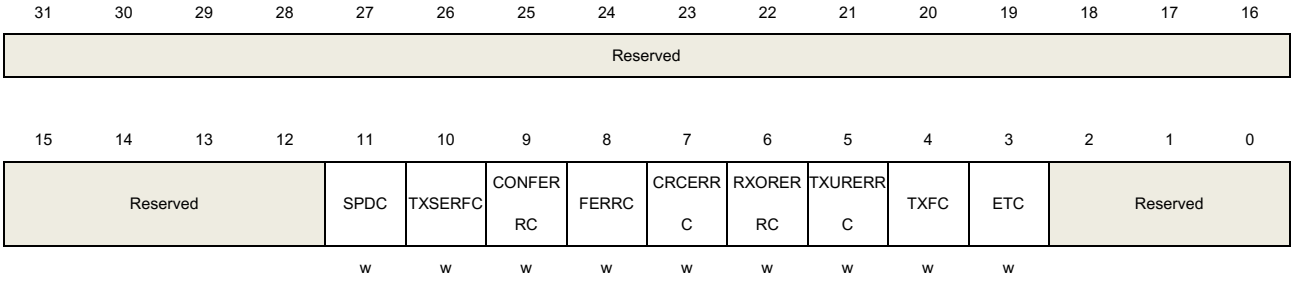
0	RP	RxFIFO packet space available flag 0: The RxFIFO is empty or the received packet is incomplete (not reach at FIFOLVL) 1: RxFIFO contains at least one complete packet

41.5.7. Interrupt/Status flags clear register (SPI_STATC)

Address offset: 0x18

Reset value: 0x0000 0000

This register can be accessed by word (32-bit).



Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11	SPDC	Clear the suspend flag Write 1 to this bit can clear the SPD bit in the SPI_STAT register.
10	TXSERFC	Clear the TXSERF flag Write 1 to this bit can clear the TXSERF bit in the SPI_STAT register.
9	CONFERRC	Clear the configuration error flag Write 1 to this bit can clear the CONFERR bit in the SPI_STAT register.
8	FERRC	Clear the SPI TI format error flag Write 1 to this bit can clear the FERR bit in the SPI_STAT register.
7	CRCERRC	Clear the CRC error flag Write 1 to this bit can clear the CRCERR bit in the SPI_STAT register.
6	RXORERRC	Clear the reception overrun error flag Write 1 to this bit can clear the RXORERR bit in the SPI_STAT register.
5	TXURERRC	Clear the transmission underrun error flag Write 1 to this bit can clear the TXURERR bit in the SPI_STAT register.
4	TXFC	Clear the TxFIFO transmission filled flag Write 1 to this bit can clear the TXF bit in the SPI_STAT register.
3	ETC	Clear the end of transmission/reception flag

		Write 1 to this bit can clear the ET bit in the SPI_STAT register.
2:0	Reserved	Must be kept at reset value.

41.5.8. Data Transfer register (SPI_TDATA)

Address offset: 0x20

Reset value: 0x0000 0000

This register can be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TDATA[31:16]															
w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDATA[15:0]															
w															

Bits	Fields	Descriptions
31:0	TDATA[31:0]	Data transfer register The hardware has two FIFOs, including TxFIFO and RxFIFO. The SPI_TDATA register serves as an interface with TxFIFO. Write data to SPI_TDATA will save the data to TxFIFO. Data is always right-aligned, and according to WORDEN, BYTEN, DZ value to place data. For example: if WORDEN is set 1, DZ is 8-bit, TDATA [7:0] is data 0, TDATA [15:8] is data 1, TDATA [23:16] is data 2, TDATA [31:24] is data 3. If WORDEN is set 0, BYTEN is set 0, DZ is 8-bit, TDATA [7:0] is data 0, TDATA [15:8] is data 1, TDATA [31:16] is invalid data. If WORDEN is set 0, BYTEN is set 1, DZ is 8-bit, TDATA [7:0] is data 0, TDATA [31:8] is invalid data. If DZ is more than 8-bit, only according to the word or half word access.

41.5.9. Data Receive register (SPI_RDATA)

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
RDATA[31:16]															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDATA[15:0]															
r															

Bits	Fields	Descriptions
31:0	RDATA[31:0]	Data receive register

The hardware has two FIFOs, including TxFIFO and RxFIFO. The SPI_RDATA register serves as an interface with RxFIFO. Read data from SPI_RDATA will get the data from RxFIFO. Data is always right-aligned, and according to WORDEN, BYTEN, DZ value to place data. For example: if WORDEN is set 1, DZ is 8-bit, RDATA [7:0] is data 3, RDATA [15:8] is data 2, RDATA [23:16] is data 1, RDATA [31:24] is data 0. If WORDEN is set 0, BYTEN is set 0, DZ is 8-bit, RDATA [15:0] is invalid data , RDATA [23:16] is data 1, RDATA [31:24] is data 0. If WORDEN is set 0, BYTEN is set 1, DZ is 8-bit, RDATA [23:0] is invalid data , RDATA [31:24] is data 0. If DZ is more than 8-bit, only according to the word or half word access.

41.5.10. CRC polynomial register (SPI_CRCPOLY)

Address offset: 0x40
Reset value: 0x0000 0107

This register can be accessed by word (32-bit). When SPI is enable, this register is write protected.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CRCPOLY[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRCPOLY[15:0]															
rw															

Bits	Fields	Descriptions
31:0	CRCPOLY[31:0]	CRC polynomial register This register contains the CRC polynomial and it is used for CRC calculation. The default value of 0x107 is corresponding to DZ = 8-bit. It is compatible with some other GD products with fixed length polynomial string (use the default value 0x07). The polynomial length is defined by the most significant bit of SPI_CRCPOLY register. The length value should be greater than the DZ value. In addition, if DZ = 32-bit, the CRCFS must be set to 1 to make sure the size of polynomial greater than the size of data.

41.5.11. TX CRC register (SPI_TCRC)

Address offset: 0x44
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
TCRC[31:16]															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

TCRC[15:0]

r

Bits	Fields	Descriptions
31:0	TCRC[31:0]	Tx CRC register. When the CRCEN bit of SPI_CFG0 is set, the hardware computes the CRC value of the transmitted bytes and saves them in SPI_TCRC register. These bits have no meaning in I2S mode.

41.5.12. RX CRC register (SPI_RCRC)

Address offset: 0x48

Reset value: 0x0000 0000

This register can be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RCRC[31:16]															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RCRC[15:0]															
r															

Bits	Fields	Descriptions
31:0	RCRC[31:0]	Rx CRC register. When the CRCEN bit of SPI_CFG0 is set, the hardware computes the CRC value of the received bytes and saves them in SPI_RCRC register. These bits have no meaning in I2S mode.

41.5.13. Underrun Data register (SPI_URDATA)

Address offset: 0x4C

Reset value: 0x0000 0000

This register can be accessed by word (32-bit). When SPI is enable, this register is write protected.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
URDATA[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
URDATA[15:0]															
rw															

Bits	Fields	Descriptions
31:0	URDATA[31:0]	Transmission underrun data at slave mode.

This register is considered only in slave mode and underrun conditions. The number of bits considered depends on the DZ bits setting of SPI_CFG0 register. The treatment of the underrun condition depends on the TXURDT and TXUROP bits of the SPI_CFG0 register.

41.5.14. I2S control register (SPI_I2SCTL)

Address offset: 0x50
Reset value: 0x0000 0000

This register can be accessed by word (32-bit). For SPI0 / I2S0, SPI1 / I2S1, SPI2 / I2S2, and SPI5 / I2S5, the reset value is 0x0002 0000.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I2SCH	Reserved					MCKOEN	OF	DIV[7:0]							
r						rw	rw	rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				I2SSEL	I2SEN	I2SOPMOD[1:0]		PCMSMOD	Reserved	I2SSTD[1:0]		CKPL	DTLEN[1:0]		CHLEN
				rw	rw	rw		rw		rw		rw	rw		rw

Bits	Fields	Descriptions
31	I2SCH	I2S channel side 0: The next data needs to be transmitted or the data just received is channel left. 1: The next data needs to be transmitted or the data just received is channel right. This bit is set and cleared by hardware. This bit is not used in SPI mode, and has no meaning in the I2S PCM mode. For TX, this bit is meaningful only if FIFOLVL = 15, that is, TxFIFO is only used for sending 1 data. For RX, this bit is meaningful only if FIFOLVL = 0, that is, RXFIFO is used for receiving only one data. For other configurations, the bit has no clear meaning.
30:26	Reserved	Must be kept at reset value.
25	MCKOEN	I2S_MCK output enable 0: I2S_MCK output is disabled 1: I2S_MCK output is enabled This bit should be configured when I2S mode is disabled. This bit is not used in SPI mode.
24	OF	Odd factor for the prescaler 0: Real divider value is DIV * 2 1: Real divider value is DIV * 2 + 1 This bit should be configured when I2S mode is disabled. This bit is not used in SPI mode.

23:16	DIV[7:0]	Dividing factor for the prescaler Real divider value is DIV * 2 + OF. DIV must not be 0. These bits should be configured when I2S mode is disabled. These bits are not used in SPI mode.
15:12	Reserved	Must be kept at reset value.
11	I2SSEL	I2S mode selection 0: SPI mode 1: I2S mode This bit should be configured when SPI/I2S is disabled.
10	I2SEN	I2S enable 0: I2S is disabled 1: I2S is enabled This bit is not used in SPI mode.
9:8	I2SOPMOD[1:0]	I2S operation mode 00: Slave transmission mode 01: Slave reception mode 10: Master transmission mode 11: Master reception mode This bit should be configured when I2S mode is disabled. This bit is not used in SPI mode.
7	PCMSMOD	PCM frame synchronization mode 0: Short frame synchronization 1: long frame synchronization This bit has a meaning only when PCM standard is used. This bit should be configured when I2S mode is disabled. This bit is not used in SPI mode.
6	Reserved	Must be kept at reset value.
5:4	I2SSTD[1:0]	I2S standard selection 00: I2S Philips standard 01: MSB justified standard 10: LSB justified standard 11: PCM standard These bits should be configured when I2S mode is disabled. These bits are not used in SPI mode.
3	CKPL	Idle state clock polarity 0: The idle state of I2S_CK is low level 1: The idle state of I2S_CK is high level This bit should be configured when I2S mode is disabled.

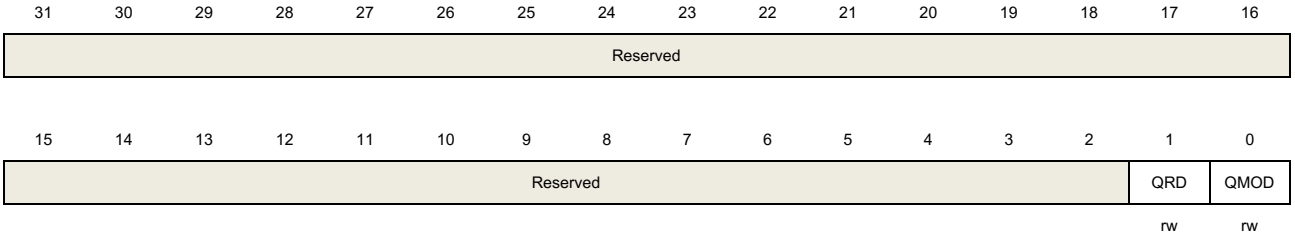
		This bit is not used in SPI mode.
2:1	DTLEN[1:0]	Data length 00: 16-bit 01: 24-bit 10: 32-bit 11: Reserved These bits should be configured when I2S mode is disabled. These bits are not used in SPI mode.
0	CHLEN	Channel length 0: 16-bit 1: 32-bit The channel length must be equal to or greater than the data length. This bit should be configured when I2S mode is disabled. This bit is not used in SPI mode.

41.5.15. Quad_SPI mode control register (SPI_QCTL)

Address offset: 0x80

Reset value: 0x0000 0000

This register can be accessed by word (32-bit).



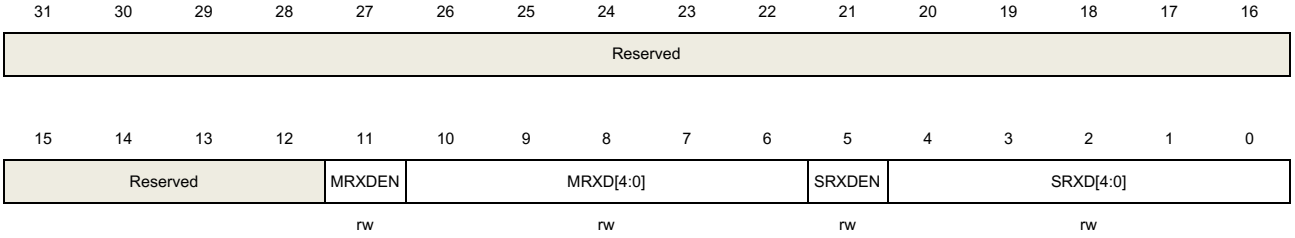
Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value
1	QRD	Quad-SPI mode read select 0: SPI is in quad wire write mode 1: SPI is in quad wire read mode This bit should be only be configured when SPI is not busy. This bit is only available in SPI3 / 4.
0	QMOD	Quad-SPI mode enable 0: SPI is in single wire mode 1: SPI is in Quad-SPI mode This bit should only be configured when SPI is not busy. This bit is only available in SPI3 / 4.

41.5.16. RX clock delay register (SPI_RXDLYCK)

Address offset: 0xFC

Reset value: 0x0000 0000

This register can be accessed by word (32-bit).



Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value
11	MRXDEN	When master receive, sampling clock delay enable 0: Sampling clock delay enable 1: Sampling clock delay disable
10:6	MRXD[4:0]	When master receive, sampling clock delay units 00000: Delay 1 unit 00001: Delay 2 units … 11111: Delay 32 units
5	SRXDEN	When slave receive, sampling clock delay enable 0: Sampling clock delay enable 1: Sampling clock delay disable
4:0	SRXD[4:0]	When slave receive, sampling clock delay units 00000: Delay 1 unit 00001: Delay 2 units … 11111: Delay 32 units

42. Inter-integrated circuit interface (I2C)

42.1. 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.

42.2. 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_PMCFG0).
- 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.

42.3. Function overview

Figure 42-1. I2C module block diagram below provides details on the internal configuration of the I2C interface.

Figure 42-1. I2C module block diagram

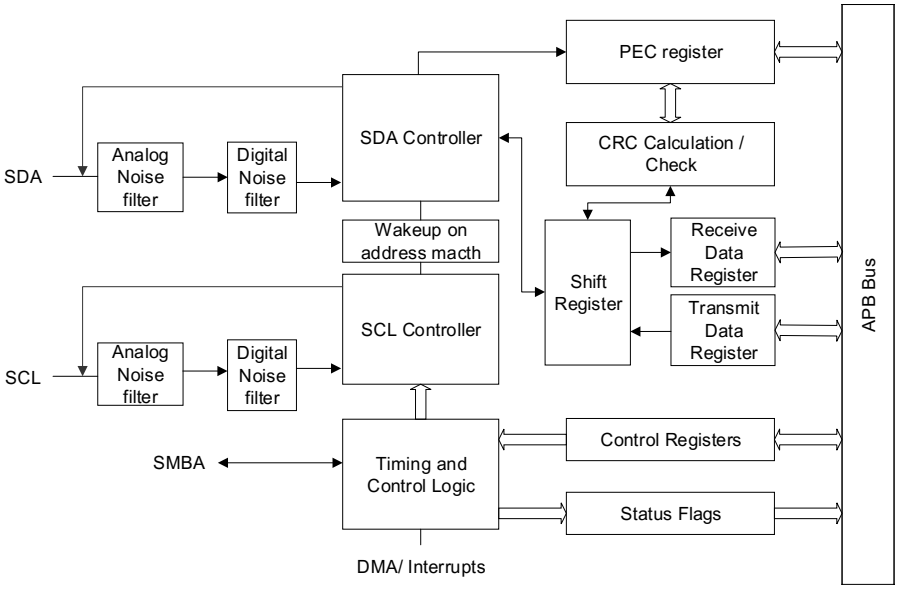


Table 42-1. Definition of I2C-bus terminology (refer to the I2C specification of Philips semiconductors)

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

42.3.1. 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 four clock sources:

- CK_APB1: APB1 clock (default value)
- CK_PLL2R: Phase Lock Loop
- CK_IRC64MDIV: Internal 64M RC oscillator
- CK_LPIRC4M: Low Power Internal 4M 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}.

42.3.2. I2C communication flow

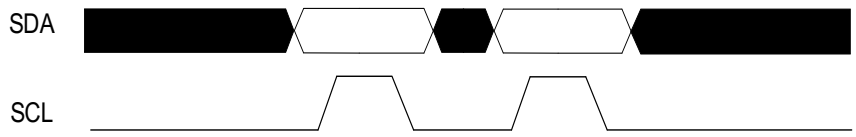
An I2C device is able to transmit or receive data whether it's a master or a slave, thus, there're 4 operation modes for an I2C device:

- 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 42-2. Data validation***). One clock pulse is generated for each data bit transferred.

Figure 42-2. Data validation

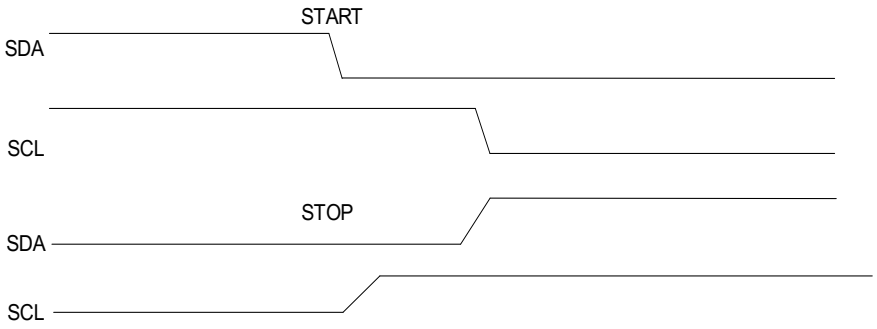


START and STOP signal

All transactions begin with a START and are terminated by a STOP (see ***Figure 42-3. START and STOP 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 42-3. 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 signal and it's also responsible for SCL clock generation.

In master mode, if AUTOEND=1, the STOP signals generated automatically by hardware. If AUTOEND=0, the STOP signal generated by software, or the master can generate a RESTART signal to start a new transfer.

Figure 42-4. I2C communication flow with 10-bit address (Master Transmit)

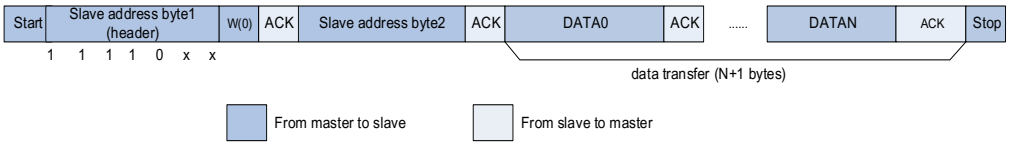


Figure 42-5. I2C communication flow with 7-bit address (Master Transmit)

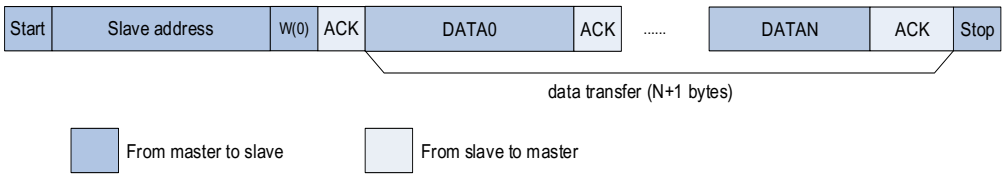
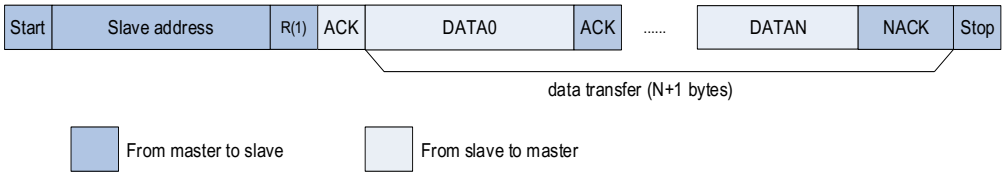


Figure 42-6. I2C communication flow with 7-bit address (Master Receive)



In 10-bit addressing mode, the HEAD10R bit can configured to decide whether the complete address sequence must be executed, or only the header to be sent. When HEAD10R=0, the complete 10 bit address read sequence must be executed with START + header of 10-bit address in write direction + slave address byte 2 + RESTART + header of 10-bit address in read direction, as is shown in ***Figure 42-7. 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 42-8. I2C communication flow with 10-bit address (Master Receive when HEAD10R=1)***.

Figure 42-7. I2C communication flow with 10-bit address (Master Receive when HEAD10R=0)

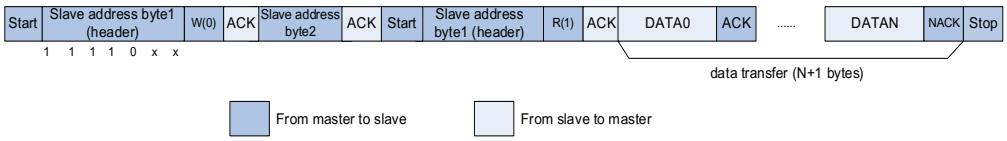
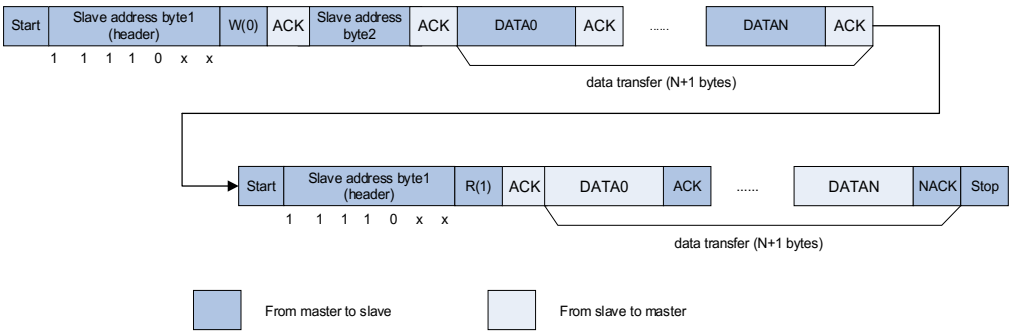


Figure 42-8. I2C communication flow with 10-bit address (Master Receive when HEAD10R=1)



42.3.3. 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 174ns in fast mode and fast mode plus.

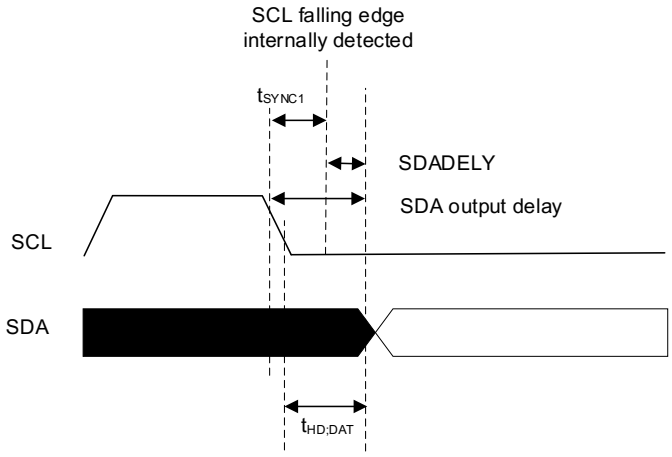
The digital noise filter can be used by configuring the DNF[3:0] bit in I2C_CTL0 register. The level of the SCL or the SDA will not be changed if the level is stable for no more than DNF[3:0]×t_{I2CCLK}. The length of spikes to be suppressed is configured by DNF[3:0].

42.3.4. 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.

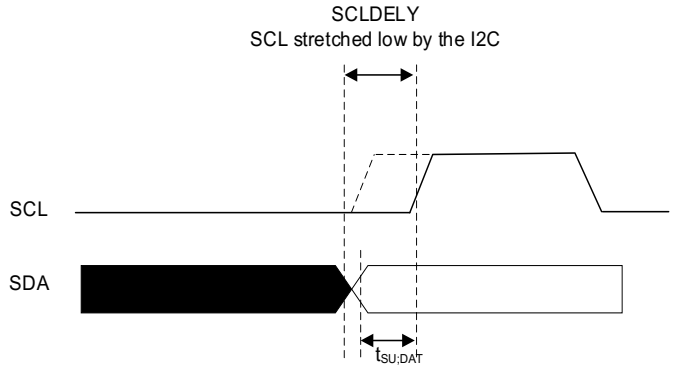
If the data is already available in I2C_TDATA register, the data will be sent on SDA after the SDADELY delay. As is shown in ***Figure 42-9. Data hold time***.

Figure 42-9. Data hold time



The SCLDELY counter starts when the data is sent on SDA output. As is shown in ***Figure 42-10. Data setup time***.

Figure 42-10. 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 42-2. 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	

42.3.5. 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:

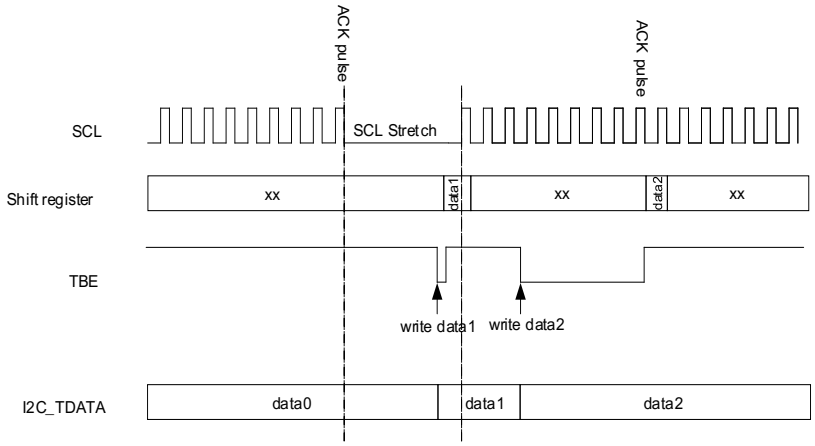
- Write I2CEN = 0
- Check I2CEN = 0
- Write I2CEN = 1

42.3.6. 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.

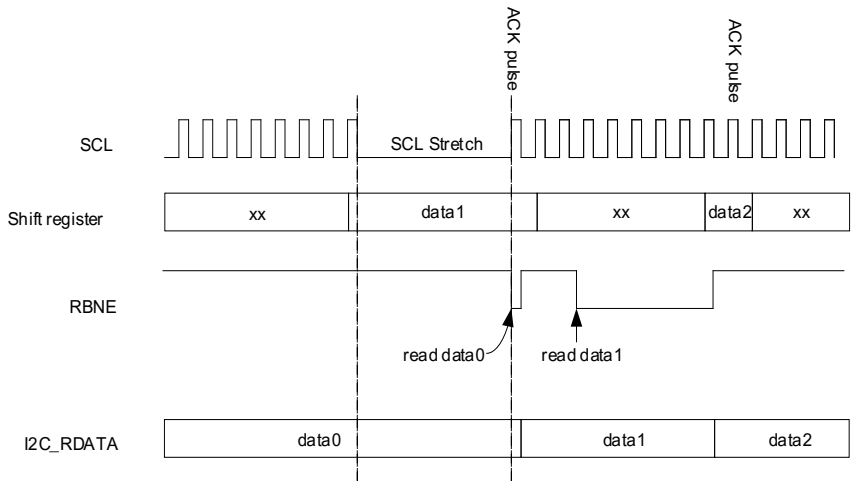
Figure 42-11. Data transmission



Data Reception

When receiving data, the data will be received in the shift register first. If RBNE is 0, the data in the shift register will move into I2C_RDATA register. If RBNE is 1, the SCL line will be stretched until the previous received data in I2C_RDATA is read. The stretch is inserted before the acknowledge pulse.

Figure 42-12. 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 42-3. Communication modes to be shut down***, the I2C embedded a byte counter in the hardware.

Table 42-3. 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.

42.3.7. 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.

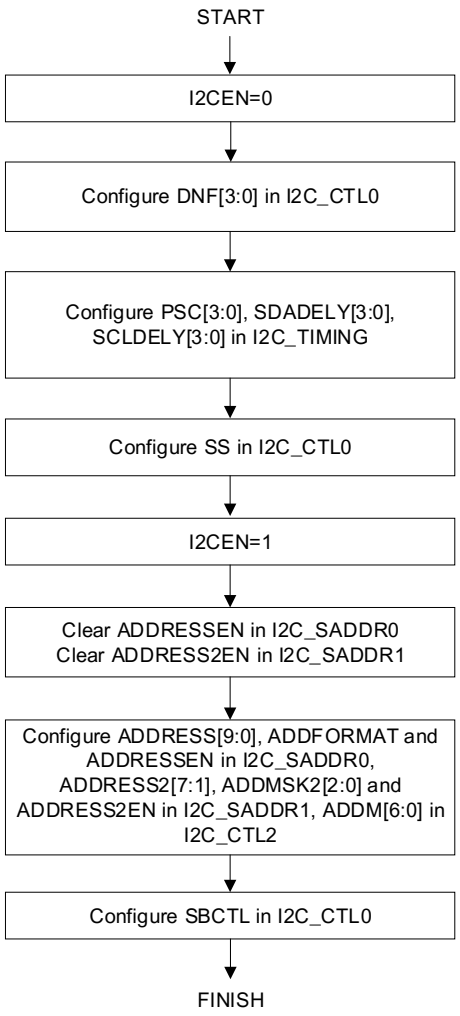
When the BYTENUM[7:0] is greater than 0x1, there is no stretch between the reception of two data bytes.

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 42-13. 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 choose 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 42-14. Programming model for slave transmitting when SS=0

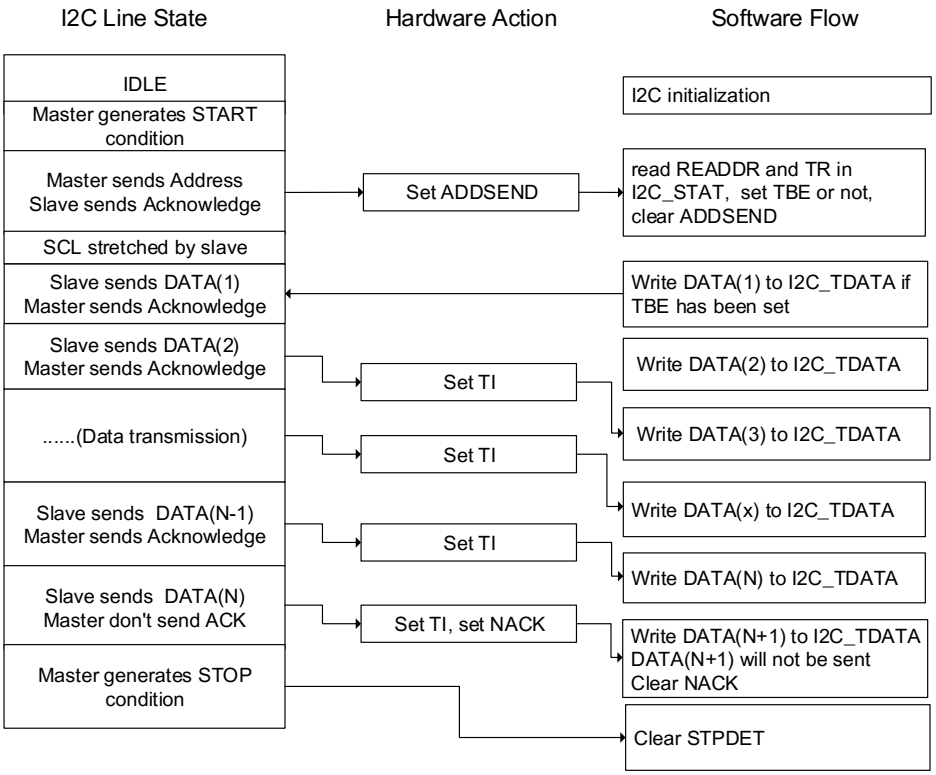
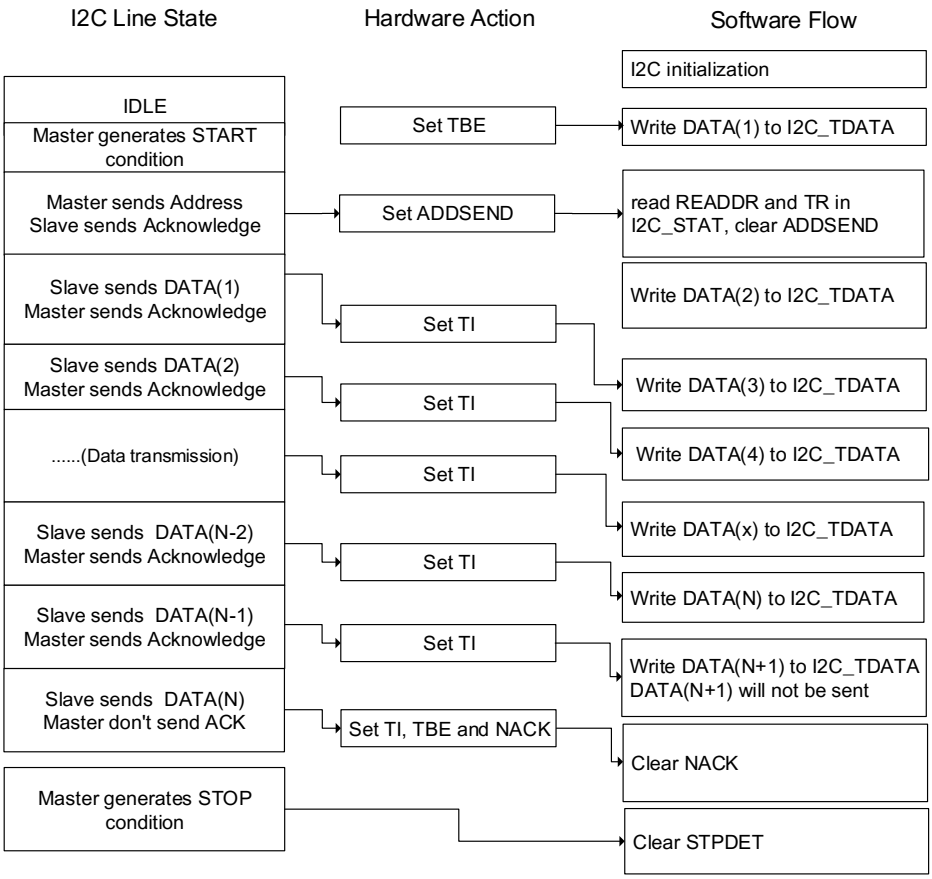


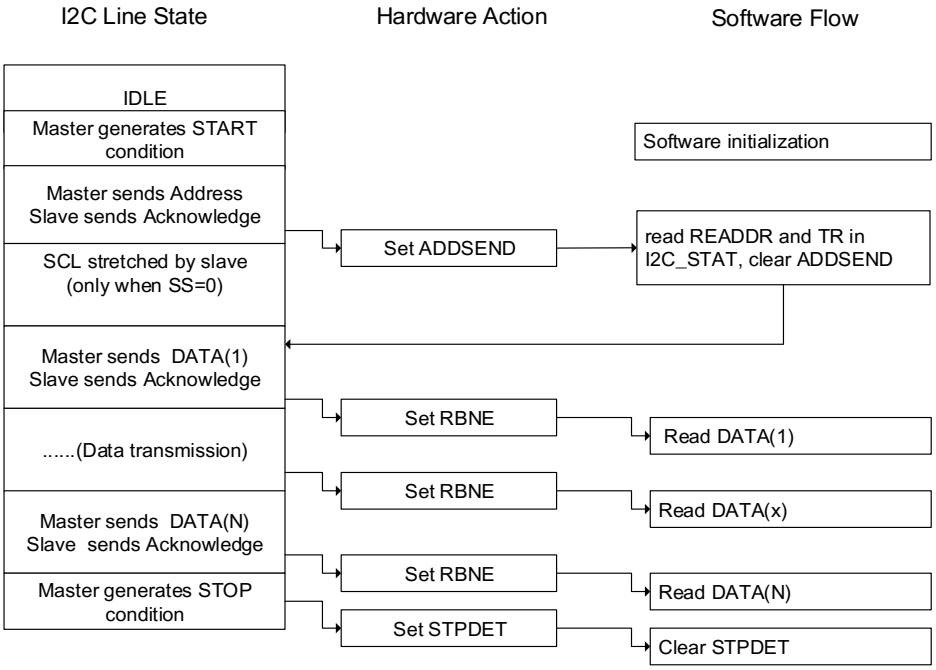
Figure 42-15. Programming model for slave transmitting when SS=1



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 42-16. Programming model for slave receiving



42.3.8. 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

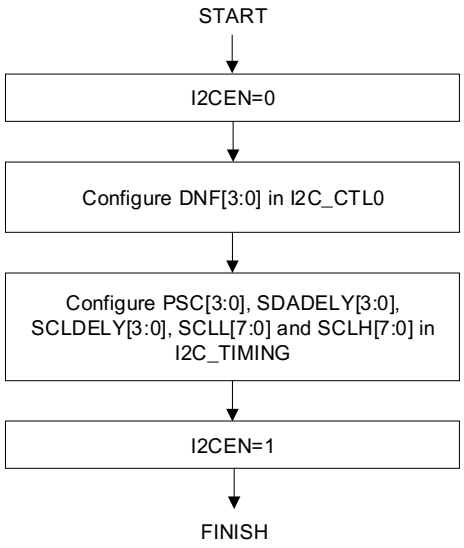
mode, the HEAD10R bit must be configured to decide whether the complete address sequence must be executed, or only the header to be sent. The number of bytes to be transferred should be configured in BYTENUM[7:0] in I2C_CTL1 register. If the number of bytes to be transferred is equal to or greater than 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.

Note: BYTENUM[7:0] can not be configured as 0.

In 10-bit addressing mode, if the master receives a NACK after the transmission of 10-bit header, the master will resend it until ACK is received. The ADDSENDC bit must be set to stop sending the slave address.

If the START bit is set, meanwhile the ADDSEND is set by addressing as a slave, the master changes to slave mode. The ADDSENDC bit must be set to clear the START bit.

Figure 42-17. 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 signal to start a new transfer. The TC bit is cleared when the START / STOP bit is set.
- If a NACK is received, a STOP signal is automatically generated, the NACK is set in I2C_STAT register, if the NACKIE bit is set, an interrupt will be generated.

Note: When the RELOAD bit is 1, the AUTOEND has no effect.

Figure 42-18. Programming model for master transmitting (N<=255)

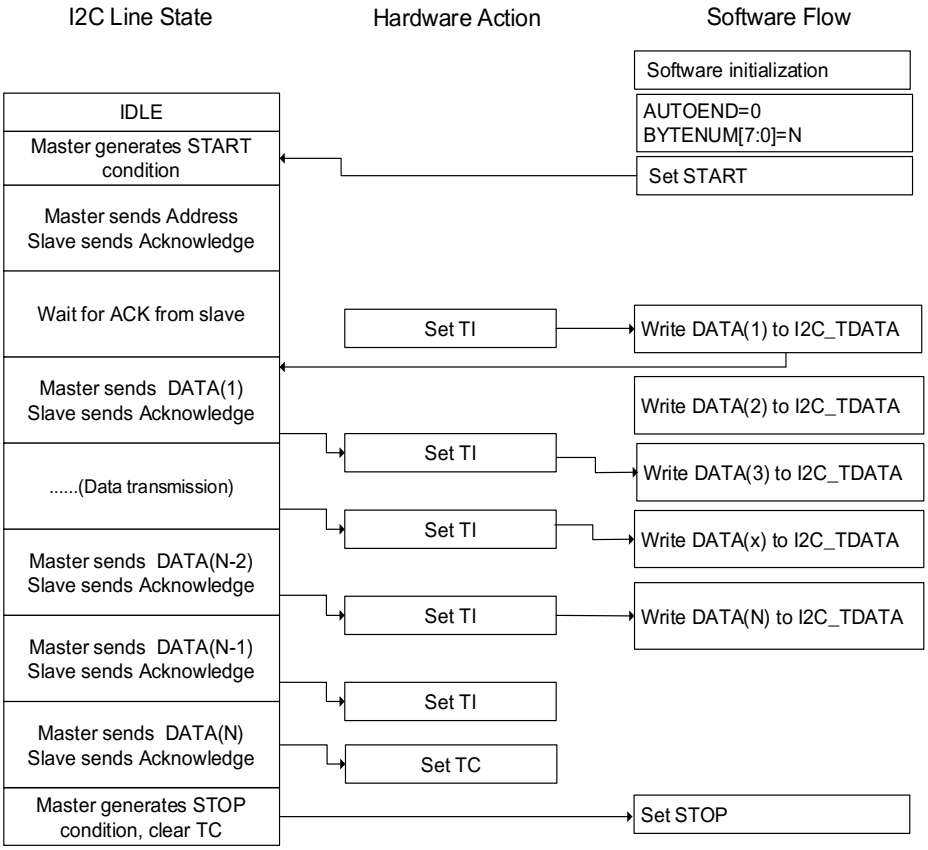
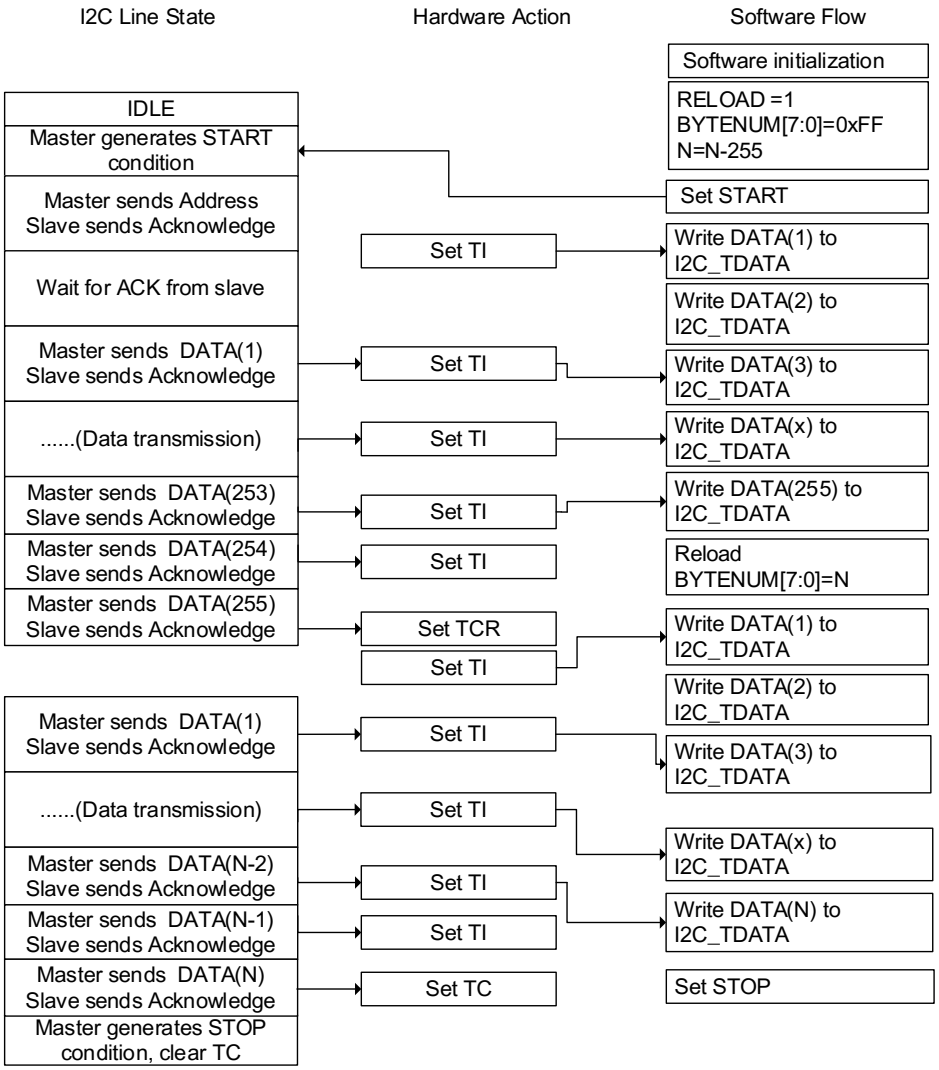


Figure 42-19. Programming model for master transmitting (N>255)



Programming model in master receiving mode

In master receiving mode, the RBNE bit in I2C_STAT register will be set when a byte is received. If the RBNEIE bit is set in I2C_CTL0 register, an interrupt will be generated. If the number of bytes to be received is greater than 255, RELOAD bit in I2C_CTL1 register must be set to enable the reload mode. In reload mode, when data of BYTENUM[7:0] bytes have been transferred, the TCR bit in I2C_STAT register will be set and the SCL stretches until BYTENUM[7:0] is modified with a non-zero value.

If data of BYTENUM[7:0] bytes have been transferred and RELOAD=0, the AUTOEND bit in I2C_CTL1 can be set to generate a STOP signal automatically. When AUTOEND is 0, the TC bit in I2C_STAT register will be set and the SCL is stretched. In this case, the master can generate a STOP signal by setting the STOP bit in the I2C_CTL1 register. Or generate a RESTART signal to start a new transfer. The TC bit is cleared when the START / STOP bit is set.

Figure 42-20. Programming model for master receiving (N<=255)

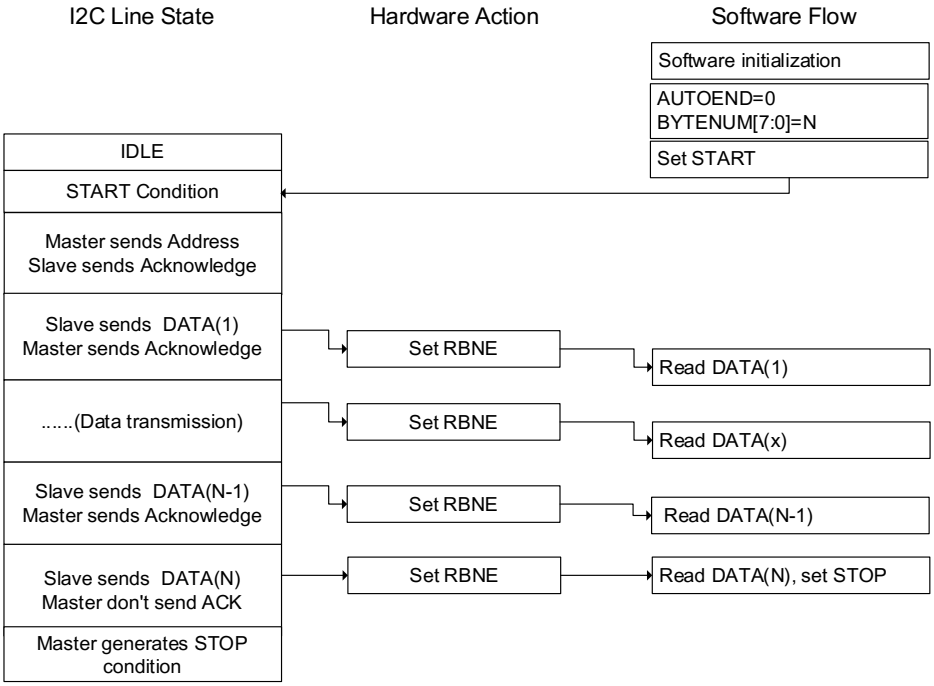
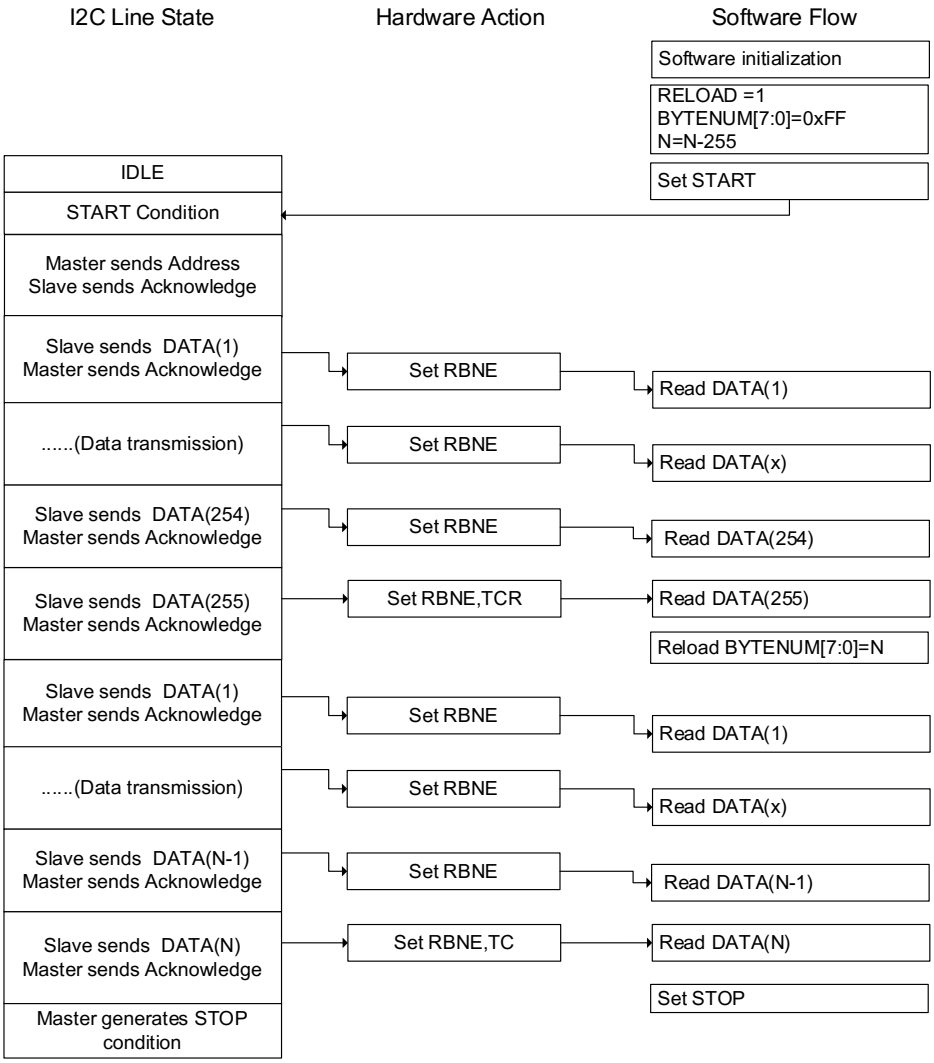


Figure 42-21. Programming model for master receiving (N>255)



42.3.9. 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.

42.3.10. SMBus mode

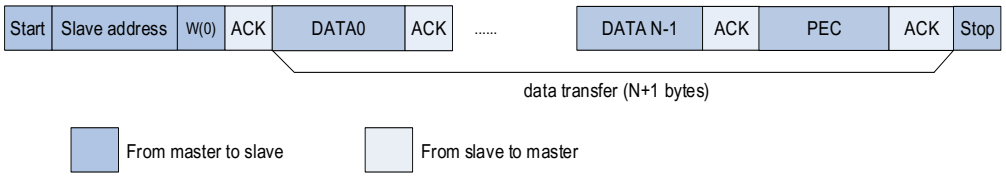
SMBus master transmitter and slave receiver

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 42-22. 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.

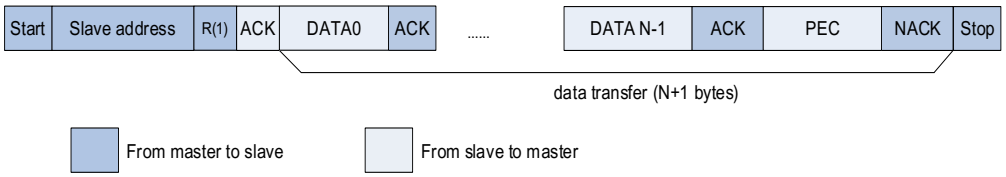
If the SMBus master receiver is required to generate a RESTART signal after receiving PEC byte, automatic end mode must be disabled. Before sending a START signal to the bus, PECTRANS bit must be set and slave addresses must be programmed. After receiving BYTENUM-1 data, the next received byte will be 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.
- When SMBus acts as a slave transmitter and uses the PEC function, BYTENUM[7:0] must be configured before detecting the START signal.

Figure 42-23. SMBus master receiver and slave transmitter communication flow



42.3.11. 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 IRC64M must be selected as the clock source for I2CCLK. During power saving mode, the IRC64M is switched off. The I2C interface switches the IRC64M on, and stretches SCL low until IRC64M is woken up when a START is detected.

Then the IRC64M 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, IRC64M 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, IRC64M 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).

42.3.12. 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.

42.3.13. I2C error and interrupts

The I2C error flags are listed in *Table 42-4. I2C error flags*.

Table 42-4. 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 42-5. I2C interrupt events*.

Table 42-5. I2C interrupt events

Interrupt event	Event flag	Enable control bit
I2C_RDATA is not empty during receiving	RBNE	RBNEIE
Transmit interrupt	TI	TIE
STOP signal detected in slave mode	STPDET	STPDETIE

Interrupt event	Event flag	Enable control bit
Transfer complete reload	TCR	TCIE
Transfer complete	TC	
Address match	ADDSEND	ADDMIE
Not acknowledge received	NACK	NACKIE
Bus error	BERR	ERRIE
Arbitration Lost	LOSTARB	
Overrun/Underrun error	OUERR	
PEC error	PECERR	
Timeout error	TIMEOUT	
SMBus Alert	SMBALT	

42.3.14. I2C debug mode

When the microcontroller enters the debug mode (Cortex®-M7 core halted), the SMBus timeout either continues to work normally or stops, depending on the I2Cx_HOLD configuration bits in the DBG module.

42.4. Register definition

I2C0 base address: 0x4000 5400

I2C1 base address: 0x4000 5800

I2C2 base address: 0x4000 C000

I2C3 base address: 0x4000 5C00

42.4.1. 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

42.4.2. 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:** When the START bit is set to 1, this bit cannot be modified, and the software cannot change this bit during transmission.
15	NACKEN	Generate NACK in slave mode 0: an ACK is sent after receiving a new byte. 1: a NACK is sent after receiving a new byte. **Note:** The bit can be set by software, and cleared by hardware when the NACK is sent, or when a STOP signal is detected or ADDSEND is set, or when I2CEN=0. When PEC is enabled, whether to send an ACK or a NACK is not depend on the NACKEN bit. When SS=1, and the OUERR bit is set, the value of NACKEN is ignored and a NACK will be sent.
14	STOP	Generate a STOP signal on I2C bus This bit is set by software and cleared by hardware when I2CEN=0 or STOP 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 to 1, this bit cannot be modified, and the software cannot change this bit during transmission..
11	ADD10EN	10-bit addressing mode enable in master mode 0: 7-bit addressing in master mode 1: 10-bit addressing in master mode **Note:** When the START bit is set to 1, this bit cannot be modified, and the software cannot change this bit during transmission..
10	TRDIR	Transfer direction in master mode 0: Master transmit 1: Master receive **Note:** When the START bit is set to 1, this bit cannot be modified, and the software cannot change this bit during transmission..
9:0	SADDRESS[9:0]	Slave address to be sent SADDRESS[9:8]: Slave address bit 9:8 If ADD10EN = 0, these bits have no effect. If ADD10EN = 1, these bits should be written with bits 9:8 of the slave address to be sent. SADDRESS[7:1]: Slave address bit 7:1 If ADD10EN = 0, these bits should be written with the 7-bit slave address to be sent. If ADD10EN = 1, these bits should be written with bits 7:1 of the slave address to be sent. SADDRESS0: Slave address bit 0 If ADD10EN = 0, this bit has no effect. If ADD10EN = 1, this bit should be written with bit 0 of the slave address to be sent **Note:** When the START bit is set to 1, this bit cannot be modified, and the software cannot change this bit during transmission..

42.4.3. 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:** 1. When ADDRESSEN is set, this bit should not be written. 2. When the slave address is set to the reserved address 0b'1111000, the slave will not respond if an address match occurs.
0	ADDRESS0	Bit 0 of a 10-bit address **Note:** When ADDRESSEN is set, this bit should not be written.

42.4.4. 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:** 1. When ADDRESSEN is set, this bit should not be written. 2. When the slave address is set to the reserved address 0b'1111000, the slave will not respond if an address match occurs.
0	Reserved	Must be kept at reset value.

42.4.5. Timing register (I2C_TIMING)

Address offset: 0x10

Reset value: 0x0000 0000

This register cannot be modified when the I2C is enabled (I2CEN = 1).

This register 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[3:0]				Reserved				SCLDELY[3:0]				SDADELY[3:0]			
rw								rw				rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCLH[7:0]								SCLL[7:0]							
rw								rw							

Bits	Fields	Descriptions
31:28	PSC[3:0]	Timing prescaler In order to generate the clock period t_{PSC} used for data setup and data hold counters, these bits are used to configure the prescaler for I2CCLK. The t_{PSC} is also used for

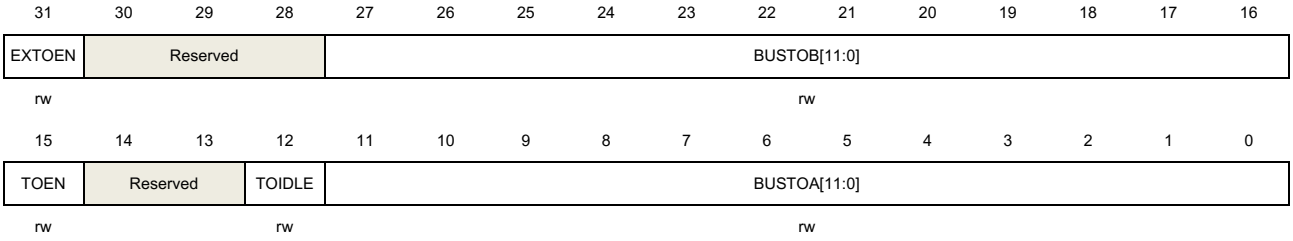
		SCL high and low level counters. $T_{PSC}=(PSC+1)*t_{I2CCLK}$
27:24	Reserved	Must be kept at reset value.
23:20	SCLDELY[3:0]	Data setup time A delay $t_{SCLDELY}$ between SDA edge and SCL rising edge can be generated by configuring these bits. And during $t_{SCLDELY}$, the SCL line is stretched low in master mode and in slave mode when SS = 0. $T_{SCLDELY}=(SCLDELY+1)*t_{PSC}$
19:16	SDADELY[3:0]	Data hold time A delay $t_{SDADELY}$ between SCL falling edge and SDA edge can be generated by configuring these bits. And during $t_{SDADELY}$, the SCL line is stretched low in master mode and in slave mode when SS = 0. 0x0: $t_{SDADELY}$= 1 * t_{PSC} 0x1-0xF: $t_{SDADELY}$= SDADELY * t_{PSC}
15:8	SCLH[7:0]	SCL high period SCL high period can be generated by configuring these bits. $T_{SCLH}=(SCLH+1)*t_{PSC}$ **Note:** These bits can only be used in master mode.
7:0	SCLL[7:0]	SCL low period SCL low period can be generated by configuring these bits. $T_{SCLL}=(SCLL+1)*t_{PSC}$ **Note:** These bits can only be used in master mode.

42.4.6. Timeout register (I2C_TIMEOUT)

Address offset: 0x14

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



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.

42.4.7. 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

42.4.8. Status clear register (I2C_STATC)

Address offset: 0x1C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		SMBALTC	TIMEOUTC	PECERRC	OUERRC	LOSTARBC	BERRC	Reserved		STPDETC	NACKC	ADDSENDC	Reserved		
		w	w	w	w	w	w			w	w	w			

Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value.
13	SMBALTC	SMBus alert flag clear. Software can clear the SMBALT bit of I2C_STAT by writing 1 to this bit.
12	TIMEOUTC	TIMEOUT flag clear. Software can clear the TIMEOUT bit of I2C_STAT by writing 1 to this bit.
11	PECERRC	PEC error flag clear. Software can clear the PECERR bit of I2C_STAT by writing 1 to this bit.
10	OUERRC	Overrun/Underrun flag clear. Software can clear the OUERR bit of I2C_STAT by writing 1 to this bit.
9	LOSTARBC	Arbitration Lost flag clear.

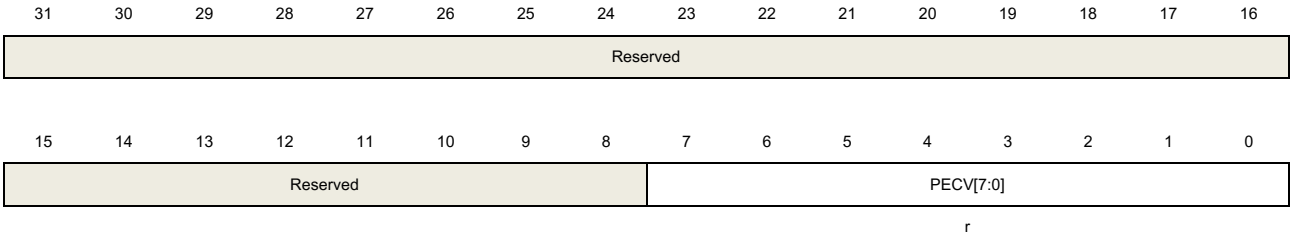
		Software can clear the LOSTARB bit of I2C_STAT by writing 1 to this bit.
8	BERRC	Bus error flag clear. Software can clear the BERR bit of I2C_STAT by writing 1 to this bit.
7:6	Reserved	Must be kept at reset value.
5	STPDETC	STPDET flag clear Software can clear the STPDET bit of I2C_STAT by writing 1 to this bit.
4	NACKC	Not Acknowledge flag clear Software can clear the NACK bit of I2C_STAT by writing 1 to this bit.
3	ADDSENDC	ADDSEND flag clear Software can clear the ADDSEND bit of I2C_STAT by writing 1 to this bit.
2:0	Reserved	Must be kept at reset value.

42.4.9. PEC register (I2C_PEC)

Address offset: 0x20

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



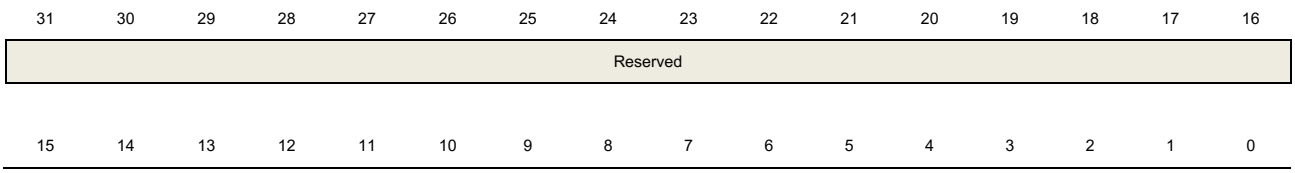
Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	PECV[7:0]	Packet Error Checking Value that calculated by hardware when PEC is enabled. PECV is cleared by hardware when I2CEN = 0.

42.4.10. Receive data register (I2C_RDATA)

Address offset: 0x24

Reset value: 0x0000 0000

This register can be accessed by word (32-bit).



Reserved	RDATA[7:0]
	r

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	RDATA[7:0]	Receive data value

42.4.11. Transmit data register (I2C_TDATA)

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								TDATA [7:0]							
								rw							

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	TDATA[7:0]	Transmit data value

42.4.12. Control register 2 (I2C_CTL2)

Address offset: 0x90

Reset value: 0x0000 FE00

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDM[6:0]							Reserved								
rw															

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.

43. Controller area network (CAN)

43.1. 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.

43.2. Characteristics

- Supports CAN protocol version 2.0A/B.
- Compliant with the ISO 11898-1:2015 standard.
- Supports CAN FD frames 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

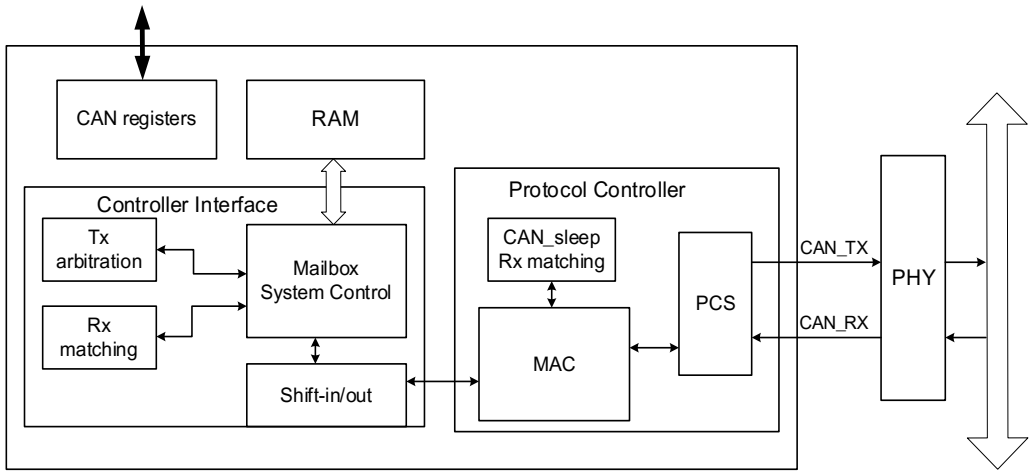
FIFO.

- 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.

43.3. Function overview

Figure 43-1. CAN module block diagram shows the CAN block diagram.

Figure 43-1. CAN module block diagram



As shown in *Figure 43-1. 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:

- Transmit 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.

43.3.1. Mailbox descriptor

The mailbox descriptor shown in *Table 43-1. 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 43-1. 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 43-3. Mailbox Rx CODE*** and ***Table 43-4. 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 43-2. 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 43-3. 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.
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

CODE	Meaning	CODE after reception	Serviced[1]	RRFRMS[2]	Description
					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.

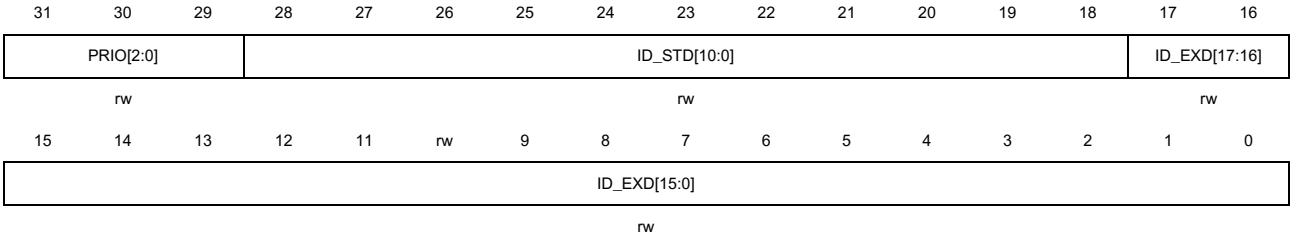
Table 43-4. Mailbox Tx CODE

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

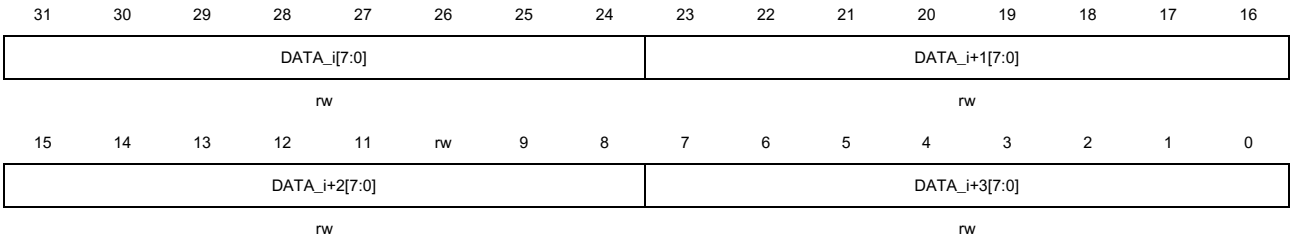
Address offset: 0x84



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)



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 43-5. Mailbox size

MDSZ[1:0]	Payload size in bytes	Mailbox size
0b00	8	32
0b01	16	21
0b10	32	12
0b11	64	7

43.3.2. Rx FIFO descriptor

The Rx FIFO descriptor is shown in *Table 43-6. 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 43-6. 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															

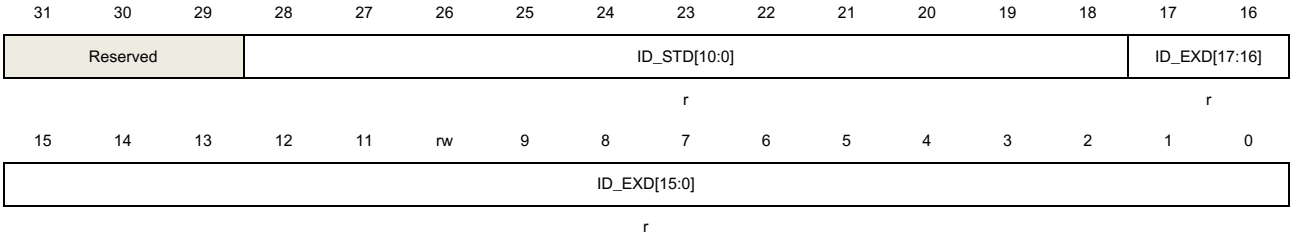
Bits	Fields	Descriptions
31:23	IDFMN[8:0]	Identifier filter matching number This bit field indicates which ID filter table element matches the received message that is in the output of the Rx FIFO.
22	SRR	Substitute remote request This bit is only used in extended format. It will be stored with the value 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

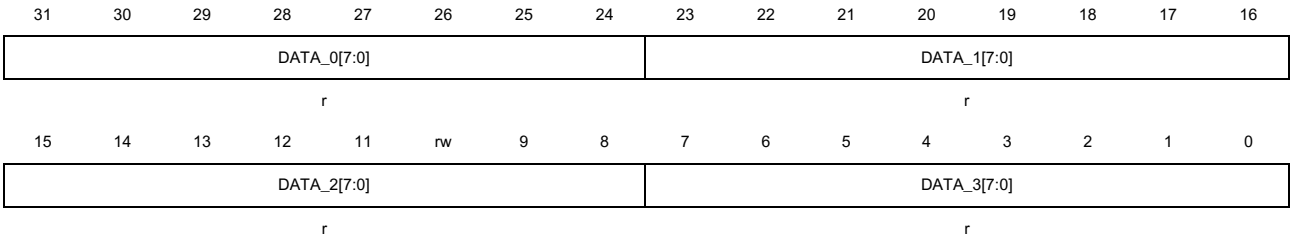
Address offset: 0x84



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.

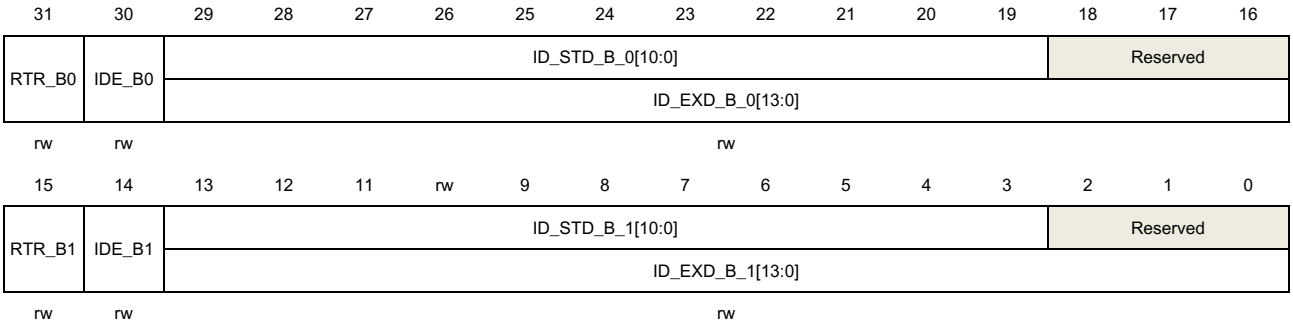
Note: The format is applied to all ID filter table elements. It is not possible to mix formats within the table.

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.
28:1	ID_STD_A[10:0]/ ID_EXD_A[28:0]	ID in format A This bit field specifies one full standard ID (standard or extended) for Rx FIFO matching process. If IDE_A is 0, the 18 to 28 bits are used for standard ID, and the rest bits are reserved; otherwise, all these bits are used for extended ID.

Format B mode:



Bits	Fields	Descriptions
31	RTR_B0	Remote frame 0 for format B This bit specifies whether remote frames can be stored into the FIFO or not when matches. 0: It indicates that remote frames are rejected and data frames can be stored. 1: It indicates that remote frames can be stored and data frames are rejected.
30	IDE_B0	ID Extended frame 0 for format B This bit specifies whether extended frames can be stored into the FIFO or not when matches. 0: Extended frames are rejected and standard frames can be stored. 1: Extended frames can be stored and standard frames are rejected.
29:16	ID_STD_B_0[10:0]/	ID for frame 0 in format B

	ID_EXD_B_0[13:0]	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.

43.3.3. 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).

43.3.4. 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 delay 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 HXTAL or IRC64MDIV in advance, otherwise 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 occurred. Clear PNMOD bit and PNEN bit in CAN_CTL0 register.
- Clear PNMOD bit and PNEN 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.

43.3.5. 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 43-5. 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 ***Rx mailbox inactivation*** or ***Tx 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 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 43-7. Mailbox arbitration value(32 bit) when local priority disabled*** and ***Table 43-8. 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 43-7. 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 43-8. 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 43-7. 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 43-8. 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 recovered 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 recovered, 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.

43.3.6. 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 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 ***Table 43-6. 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 either case. 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 43-9. 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 43-9. 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 43-10. 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 43-10. 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.

43.3.7. 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 43-11. 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 3, 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.

43.3.8. 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 the reception of CAN FD frames, these fixed stuffing bits will be ignored. 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. A node transmitting an CAN FD frame shall not resynchronize while it transmits the data phase of that frame.

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 43-2. 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:

$$t_{compensation} = t_{measure} + t_{offset} \quad (43\text{-}1)$$

with

$$t_{offset} = TDCO[6:0] \times t_{CANCLK} \quad (43\text{-}2)$$

$$t'_{offset} = t_{PBS1_FD} + t_{PTS_FD} + t_{SYNC_SEG} \quad (43\text{-}3)$$

$$t_{PBS1_FD} = (DPBS1[4:0] + 1) \times t_{q_FD} \quad (43\text{-}4)$$

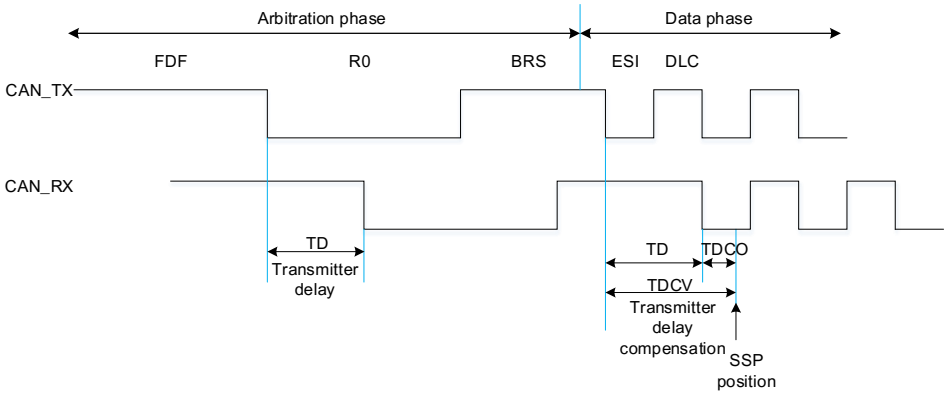
$$t_{PTS_FD} = DPTS[5:0] \times t_{q_FD} \quad (43\text{-}5)$$

$$t_{q_FD} = (DBAUDPSC[9:0] + 1) \times t_{CANCLK} \quad (43\text{-}6)$$

where the $t_{measure}$ is the measured transmitter delay, t_{offset} is the transmitter delay compensation offset which is saved in the TDCO[6:0] 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[7:0] bits of CAN_FDCTL register in unit of t_{CANCLK}.

In the equations, the DPBS1[4:0] bits, DPTS[5:0] bits, DBAUDPSC[9:0] bits are all configured in CAN_FDBT register.

Figure 43-2. 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.

43.3.9. 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 from 0 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 will 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.

43.3.10. 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 ($1 \times t_q$).

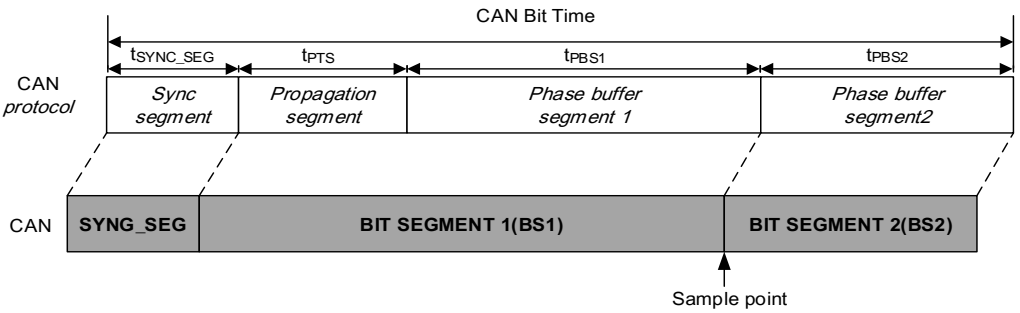
Bit segment 1 (BS1): It defines the location of the sample point. It includes the Propagation segment and Phase buffer segment 1 in the CAN standard. It may be automatically lengthened to compensate for positive phase drifts due to different frequency of the various nodes of the network.

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 43-3. CAN bit time***.

Figure 43-3. 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.

The valid edge transition is defined as the first transition from a recessive bit to a dominant

bit within a bit time, provided that the bus state detected at the previous sample point was recessive.

If a valid edge is detected in BS1, not in SYNC_SEG, BS1 is added 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.

Note: For detailed explanations of hard synchronization and resynchronization, please refer to the ISO 11898 standard.

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_CFG1 register, to derive from oscillator clock, or APB2 clock, or APB2 clock divided by 2, or IRC64MDIV internal clock.

The CAN calculates its baudrate as follows:

$$\text{BaudRate} = \frac{1}{\text{CAN Bit Time}} \tag{43-7}$$

$$\text{CAN Bit Time} = t_{SYNC_SEG} + t_{PTS} + t_{PBS1} + t_{PBS2} \tag{43-8}$$

with

$$t_{SYNC_SEG} = 1 \times t_q \tag{43-9}$$

$$t_{PTS} = (N_{PTS} + 1) \times t_q \text{ or } t_{PTS} = N_{DPTS} \times t_q \tag{43-10}$$

$$t_{PBS1} = (N_{PBS1} + 1) \times t_q \tag{43-11}$$

$$t_{PBS2} = (N_{PBS2} + 1) \times t_q \tag{43-12}$$

$$t_q = (N_{BAUDPSC} + 1) \times t_{CANCLK} \tag{43-13}$$

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[5:0] bits, DPBS1[4:0] bits, DPBS2[4:0] 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.

43.3.11. Interrupts

The CAN interrupt events and flags are list in *Table 43-11. Interrupt events*.

Table 43-11. 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			

43.4. 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 RCU_CFG1 register to select the CAN module clock source. Program the RCU_APB2EN register to enable the CAN module clock.
- Setup the communication interface
  Configure GPIO and AFIO module to select PADs to alternate functions.
- Enter Inactive mode
  Because INAMOD bit, HALT bit, NRDY bit and INAS bit are default set after power-on

reset or system reset, so CAN will automatically enters Inactive mode for configuration of CAN registers.

- Service the flags in CAN_STAT register
  Read the Rx mailbox or Rx FIFO descriptor contents, clear the corresponding asserted flag bit in CAN_STAT register, then read the CAN_TIMER register at last for a complete flag bit service. If Rx FIFO is enabled, do a clearing FIFO operation by setting MS0 bit in CAN_STAT register to 1. Also clear the asserted flags by Tx mailboxes.
- Initialize the physical memory space for mailbox and Rx FIFO descriptors
  Configure memory space for mailbox and Rx FIFO descriptors totally by MSZ[4:0] bits in CAN_CTL0 register.
- Configure the communication parameters
  1) Configure the CAN nominal bit rate by PTS[5:0] bits, PBS1[4:0] bits, PBS2[4:0] bits, SJW[4:0] bits and BAUDPSC[9:0] bits in CAN_BT register.
  2) Configure bit sampling mode by BSPMOD bit in CAN_CTL1 register if needed.
  3) Configure PREEN bit and EFDIS bit for bus integration state if needed.
- Configure the control parameters for transmission
  1) Configure arbitration priority by MTO bit in CAN_CTL1 register and LAPRIOEN bit in CAN_CTL0 register.
  2) Configure arbitration start delay by ASD[4:0] bits of CAN_CTL2 register if needed.
  3) Enable transmission abort function for Tx mailbox descriptor configuration by MST bit in CAN_CTL0 register.
- Configure the control parameters for reception
  1) Choose whether use Rx FIFO and Rx FIFO DMA for reception or not by RFEN bit and DMAEN bit in CAN_CTL0 register.
  2) Configure Rx private filter & Rx mailbox queue feature by RPFQEN bit in CAN_CTL0 register.
  3) Configure receive filter related parameters by RFO bit, RRFRMS bit and IDERTR_RMF bit of CAN_CTL2 register.
  4) Configure filter data of the Rx mailbox and Rx FIFO by CAN_RMPUBF, CAN_RFIFOPUBF and CAN_RFIFOMPFx (x = 0..31) registers. If Rx FIFO is enabled, configure Rx FIFO ID filter table element format by FS[1:0] bits of CAN_CTL0 register, configure Rx FIFO ID filter table element number by RFFN[3:0] bits of CAN_CTL2 register.
- If CAN FD operation is needed
  1) Select CAN FD operation protocol by ISO bit in CAN_CTL2 register.
  2) Enable CAN FD operation by FDEN bit in CAN_CTL0 register.
  3) Initialize the mailbox data size by MDSZ[1:0] bits of CAN_FDCTL register.
  4) Configure CAN FD related transmitter delay compensation feature by TDCEN bit and TDCO[4:0] bits of CAN_FDCTL register if needed.
  5) Configure the CAN data bit rate by DPTS[4:0] bits, DPBS1[2:0] bits, DPBS2[2:0] bits DSJW[2:0] bits and DBAUDPSC[9:0] bits in CAN_FDBT register.
- Configure interrupts
  Enable the needed interrupts in CAN_CTL0, CAN_CTL1, CAN_CTL2 and CAN_INTEN registers.

- Initialize the Tx / Rx mailbox descriptors

  1) If message transmission is needed, initialize the Tx mailbox descriptors.

  2) If message reception is needed, initialize the Rx mailbox descriptors, if Rx FIFO is enabled, also initialize the Rx FIFO descriptors including the ID filter table elements.

- If Pretended Networking mode is required, set PNEN bit and PNMOD bit in CAN_CTL0 register and configure the necessary registers for wakeup.

- Exit Inactive mode

  Clear HALT bit in CAN_CTL0 register to exit Inactive mode, and CAN starts to synchronize to the CAN bus.

43.5. CAN registers

CAN0 base address: 0x4001 A000

CAN1 base address: 0x4001 B000

CAN2 base address: 0x4001 C000

43.5.1. 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 automatically 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 warning 1: Enable Tx and Rx error warning
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

43.5.2. 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 stages synchronization for CAN bus sampling 1: one stage 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.

43.5.3. 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.

43.5.4. 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.

43.5.5. 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

43.5.6. Error register 1 (CAN_ERR1)

Address offset: 0x20

Reset value: 0x000X 000X

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BRFERR	BDFERR	Reserved	CRCFERR	FMFERR	STFFERR	Reserved				ERROVR	ERRFSF	BORF	SYN	TWERRIF	RWERRIF
rc	rc		rc	rc	rc					rc_w1	rc_w1	rc_w1	r	rc_w1	rc_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BRERR	BDERR	ACKERR	CRCERR	FMERR	STFERR	TWERRF	RWERRF	IDLEF	TS	ERRSI[1:0]		RS	BOF	ERRSF	MCE
rc	rc	rc	rc	rc	rc	r	r	r	r	r		r	rc_w1	rc_w1	r

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 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	MCE	Mode change enable

0: Mode change enable

1: Mode change disable

43.5.7. 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

43.5.8. 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***.

43.5.9. 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 43-12. 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: When a mailbox with CODE=RANSWER and a matching ID is found, a remote answer frame is automatically generated. 1: When a mailbox with CODE=RANSWER and a matching ID is found, the remote request frame is stored as if it were a data frame, without automatically transmitting a remote answer frame.
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.

43.5.10. 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.

43.5.11. 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"

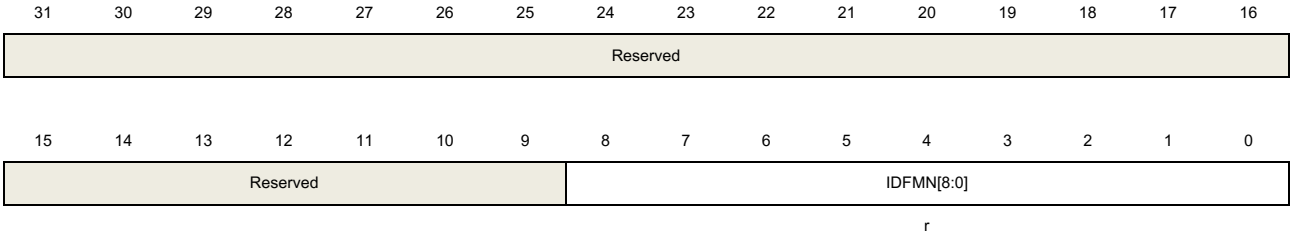
1: The bit is checked

43.5.12. 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.

43.5.13. Bit timing register (CAN_BT)

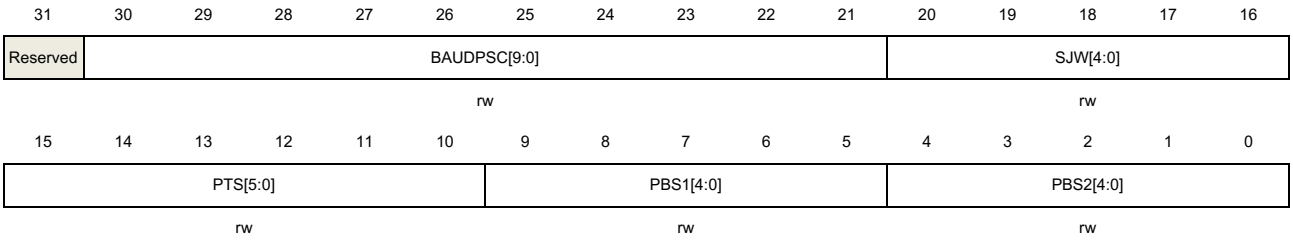
Address offset: 0x50

Reset value: 0x0100 0000

All bits of this register should be configured in Inactive mode only, because they are blocked by hardware in other modes.

This register is not affected by software reset bit SWRST in CAN_CTL0 register.

This register has to be accessed by word(32-bit).



Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30:21	BAUDPSC[9:0]	Baud rate prescaler The CAN baud rate prescaler = 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

43.5.14. 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

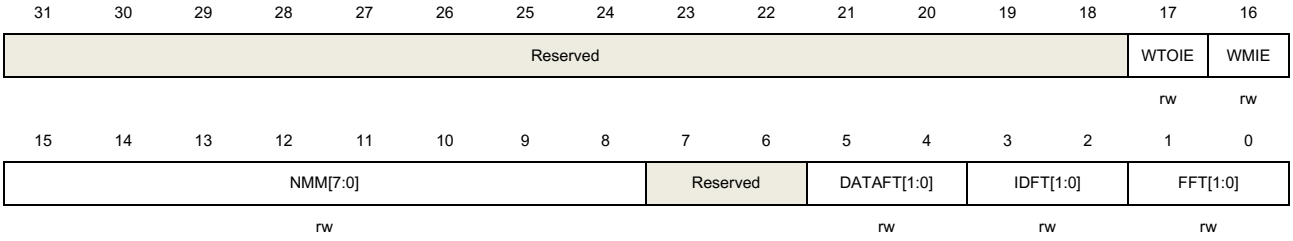
43.5.15. Pretended Networking mode control register 0 (CAN_PN_CTL0)

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).



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 transition 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

43.5.16. 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															

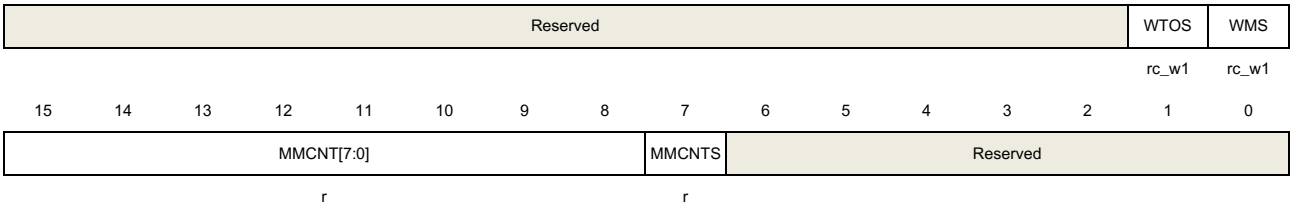
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	WTO[15:0]	Wakeup timeout The timeout is counted by step of 64 times the CAN Bit Time. Wakeup timeout is default disabled.

43.5.17. Pretended Networking mode status register (CAN_PN_STAT)

Address offset: 0xB08
Reset value: 0x0000 0080

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16



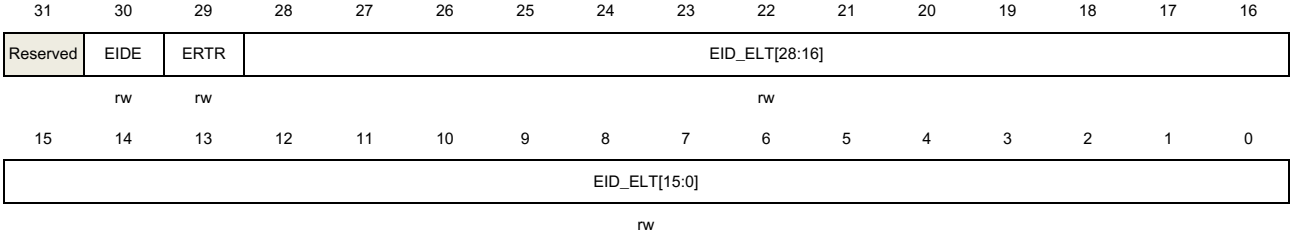
Bits	Fields	Descriptions
31:18	Reserved	Must be kept at reset value.
17	WTOS	Wakeup timeout flag status 0: No wakeup timeout event occurred 1: Wakeup timeout event occurred
16	WMS	Wakeup match flag status 0: No wakeup match event occurred 1: Wakeup match event occurred
15:8	MMCNT[7:0]	Matching message 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.

43.5.18. 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).



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.

43.5.19. 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

43.5.20. 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.

43.5.21. 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.

43.5.22. 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.

43.5.23. 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).

43.5.24. 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.

43.5.25. 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.

43.5.26. 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.

43.5.27. 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

43.5.28. 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

43.5.29. 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											TDCO[6:5]		MDSZ[1:0]	
rw												rw		rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDCEN	TDCS	Reserved	TDCO[4:0]					TDCV[7:0]							
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'
30:20	Reserved	Must be kept at reset value.
19:18	TDCO[6:5]	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. These bits serve as the 6th and 5th bits of the transmitter delay compensation offset (TDCO).
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.
7:0	TDCV[7:0]	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.

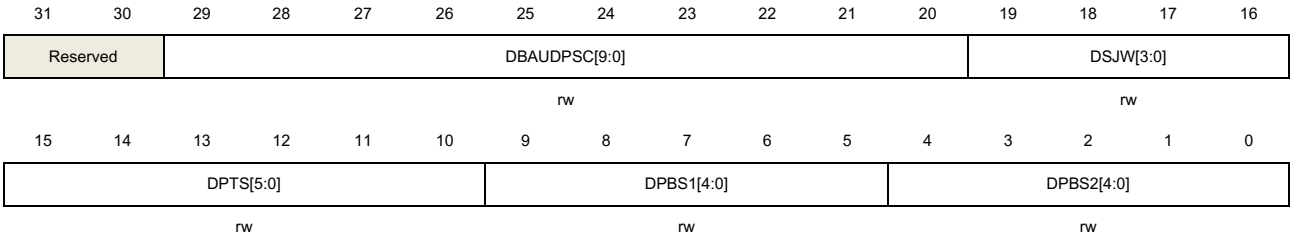
43.5.30. FD bit timing register (CAN_FDBT)

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.
19:16	DSJW[3:0]	Resynchronization jump width for data bit time Resynchronization jump width time quantum = DSJW[3:0] + 1
15:10	DPTS[5:0]	Propagation time segment for data bit time Propagation time segment time quantum = DPTS[5:0]
9:5	DPBS1[4:0]	Phase buffer segment 1 for data bit time Phase buffer segment 1 time quantum = DPBS1[4:0] + 1
4:0	DPBS2[4:0]	Phase buffer segment 2 for data bit time

Phase buffer segment 2 time quantum = DPBS2[4:0] + 1

43.5.31. 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).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			ANTM[4:0]					Reserved			CRCTCI[20:16]				
			r								r				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRCTCI[15:0]															
r															

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.

44. OSPI I/O manager(OSPIM)

44.1. Overview

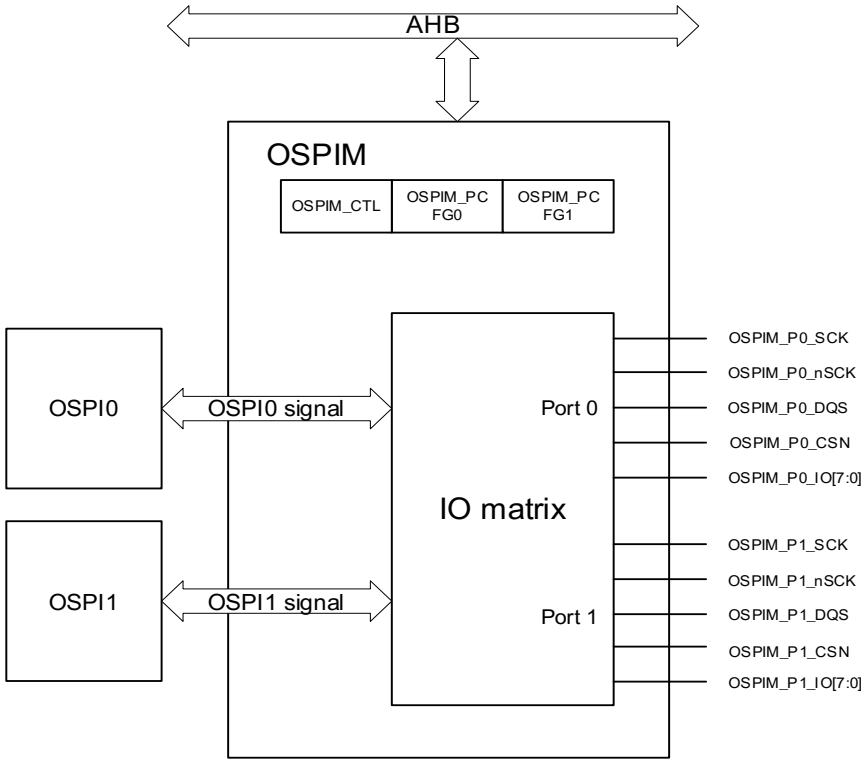
OSPIM supports OSPI pin assignment with full matrix.

44.2. Characteristics

- Support two OSPI(single-line, two-lines, four-lines, eight-lines) interfaces.
- Support two ports for pin assignment.
- Fully programmable IO matrix, can assign pins according to function.

44.3. Function overview

44.3.1. OSPIM block diagram



44.3.2. OSPIM matrix

The OSPIM matrix is fully programmable, and the pins can be pre-mapped according to function, as shown in the *Table 44-1 OSPIM matrix mapping*.

Table 44-1 OSPIM matrix mapping

Pins	Mapping
OSPIM_P0_SCK and OSPIM_P0_nSCK, OSPIM_P1_SCK and OSPIM_P1_nSCK	Can be independently mapped to OSPI0_SCK/OSPI0_nSCK or OSPI1_SCK/OSPI1_nSCK
OSPIM_P0_DQS, OSPIM_P1_DQS	Can be independently mapped to OSPI0_DQS or OSPI1_DQS
OSPIM_P0_CSN, OSPIM_P1_CSN	Can be independently mapped to OSPI0_CSN or OSPI1_CSN
OSPIM_P0_IO[3:0], OSPIM_P0_IO[7:4], OSPIM_P1_IO[3:0], OSPIM_P1_IO[7:4]	Can be independently mapped to OSPI0_IO[3:0], OSPI0_IO[7:4], OSPI1_IO[3:0] or OSPI1_IO[7:4]

By default, the signals of OSPI0 and OSPI1 are mapped to port 0 and port 1, respectively. Port 0 and port 1 of OSPIM can be independently configured through OSPIM_PCFGx register. If OSPIs are disabled, the OSPIM matrix must be configured to avoid unexpected transactions on the bus.

44.3.3. OSPIM multiplexer

Set the MULEN bit of the OSPIM_CTL register to enable the multiplexing mode. In multiplexing mode, two OSPIs are multiplexed on the bus. The two OSPI get the ownership of the bus in turn through the REQ and ACK signals.

In multiplexing mode, the following parts need to be noted:

- OSPIM_Px_CSN are not multiplexed.
- Due to arbitration and bus sharing, the status polling interval time of OSPI may increase when used.
- When OSPI is disabled, the MULEN bit of OSPIM_CTL must be set to avoid unexpected transactions on the bus.

Request to acknowledge time

The REQ to ACK time can be configured through the ACKTM[1:0] bit field of the OSPIM_CTL register, which is the time between the OSPI that is occupying the bus releases the bus and the bus is granted to the requested OSPI, that is, the time between the falling edge of the REQ signal of the OSPI occupying the bus and the falling edge of the ACK signal of the OSPI requesting the bus.

44.4. Register definition

OSPIM base address: 0x5200 B400

44.4.1. Control register (OSPIM_CTL)

Address offset: 0x00

Reset value: 0x0000 0000

This register can be accessed by word (32-bit).



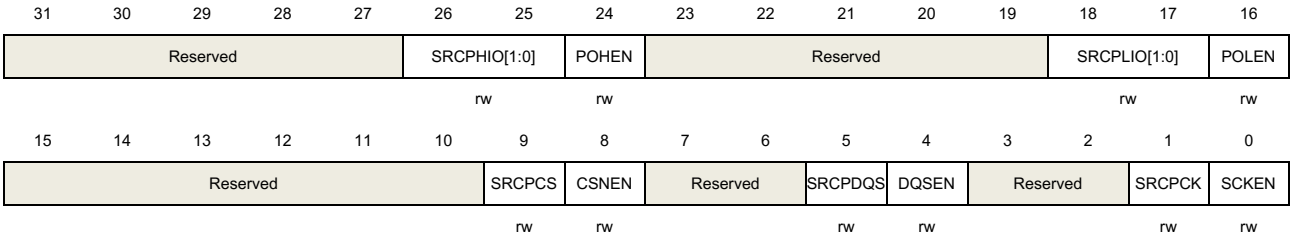
Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:16	ACKTM[7:0]	REQ to ACK time This bit field indicates that the time between two transactions is ACKTM[1:0]+1 OSPI clock cycles in Multiplex mode.
15:1	Reserved	Must be kept at reset value.
0	MULEN	Multiplexed mode enable 0: Disable multiplexed mode 1: Enable multiplexed mode

44.4.2. Port configuration register (OSPIM_PCFGx) (x = 0, 1)

Address offset: 0x04*(x+1)

Reset value: 0x0301 0111 (x = 0), 0x0705 0333 (x=1)

This register can be accessed by word (32-bit).



Bits	Fields	Descriptions
31:27	Reserved	Must be kept at reset value.
26:25	SRCPHIO[1:0]	Source selection for IO[7:4] of port x 00: Select OSPI0_IO[3:0]. 01: Select OSPI0_IO[7:4]. 10: Select OSPI1_IO[3:0].

		11: Select OSPI1_IO[7:4].
24	POHEN	Enable for IO[7:4] of port x 0: Disable IO[7:4] of port x. 1: Enable IO[7:4] of port x.
23:19	Reserved	Must be kept at reset value.
18:17	SRCPLIO[1:0]	Source selection for IO[3:0] of port x 00: Select OSPI0_IO[3:0]. 01: Select OSPI0_IO[7:4]. 10: Select OSPI1_IO[3:0]. 11: Select OSPI1_IO[7:4].
16	POLEN	Enable for IO[3:0] of port x 0: Disable IO[3:0] of port x. 1: Enable IO[3:0] of port x.
15:10	Reserved	Must be kept at reset value.
9	SRCPCS	Source selection for CSN of port x 0: The source of CSN is OSPI0_CSN 1: The source of CSN is OSPI1_CSN
8	CSNEN	Enable for CSN of port x 0: Disable CSN of port x 1: Enable CSN of port x
7:6	Reserved	Must be kept at reset value.
5	SRCPDQS	Source selection for DQS of port x 0: The source of DQS is OSPI0_DQS 1: The source of DQS is OSPI1_DQS
4	DQSEN	Enable for DQS of port x 0: Disable DQS of port x 1: Enable DQS of port x
3:2	Reserved	Must be kept at reset value.
1	SRCPCK	Source selection for SCK of port x 0: The source of SCK is OSPI0_SCK 1: The source of SCK is OSPI1_SCK
0	SCKEN	Enable for SCK of port x 0: Disable SCK of port x 1: Enable SCK of port x

45. Octal-SPI interface(OSPI)

45.1. Overview

The OSPI is a specialized interface that communicate with external memories such as serial PSRAM, serial NAND, serial NOR Flash, HyperRAM™ and HyperFlash™ memories (Only OSPI0 supports PSRAM and HyperRAM™). The interface support single, dual, quad, dual-quad and octal SPI mode.

45.2. Characteristics

- Three functional modes:
  - indirect mode: all operations are performed depends on OSPI registers.
  - status polling mode: the values of status registers in external memory are periodically read and check.
  - memory-mapped mode: the external memory is mapped to the microcontroller address space(OSPI0: 0x9000 0000 - 0x9FFF FFFF, OSPI1: 0x7000 0000 - 0x7FFF FFFF) and is accessed as an internal memory.
- Support read and write in memory-mapped mode.
- Support single, dual, quad, dual-quad and octal communication.
- OSPI0 supports unaligned access.
- Support DQS.
- Fully programmable command format for both indirect and memory-mapped mode.
- Support SDR(single data rate) and DTR(double transfer rate).
- Support 200 MHz DTR
- Support two frame format:
  - the standard frame format.
  - HyperBus™ frame format.
- Support wrap for both standard frame format and HyperBus™ frame format.
- Integrated FIFO for transmission/reception.
- MDMA channel for indirect mode.
- Interrupt generation on FIFO threshold, status match, timeout, transfer complete and access error.

Note:

1. In memory-mapped mode, write performance is approximately 34% of read performance. This may affect PSRAM write speed in certain applications, such as frame buffer updates in GUI applications.
2. When the OSPI communication rate needs to be 200 MHz, select CK_PLL0Q as the OSPI clock source and set CK_PLL0Q to 400 MHz. Then use the OSPI prescaler (PSC) to divide the clock by 2. When CK_PLL0Q is selected as the OSPI clock source, the system clock must also be set to 400 MHz.

45.3. Function overview

45.3.1. OSPI block diagram

The pins of OSPI are described in the table below.

Table 45-1 OSPI signal description

Pin name	Direction	Description
CSN	O	chip select output(active low)
SCK	O	clock output
nSCK	O	inverted clock used only in the 1.8V HyperBus protocol.
DQS	I/O	data strobe signal
IO0/SO	I/O	single mode: data output. Dual mode: data input or output. Quad mode: data input or output. Dual-quad mode: data input or output. Octal mode: data input or output.
IO1/SI	I/O	single mode: data input. Dual mode: data input or output. Quad mode: data input or output. Dual-quad mode: data input or output. Octal mode: data input or output.
IO2	I/O	single mode: output mode and forced to 0, connect WP pin of external memories, control "write protect" function. Dual mode: output mode and forced to 0, connect WP pin of external memories, control "write protect" function. Quad mode: data input or output. Dual-quad mode: data input or output. Octal mode: data input or output.
IO3	I/O	single mode: output mode and forced to 1, connect HOLD pin of external memories, control "hold" function. Dual mode: output mode and forced to 1, connect HOLD pin of external memories, control "hold" function. Quad mode: data input or output. Dual-quad mode: data input or output. Octal mode: data input or output.
IO4	I/O	single mode: output mode and forced to 0. Dual mode: output mode and forced to 0. Quad mode: output mode and forced to 0. Dual-quad mode: data input or output. Octal mode: data input or output.

Pin name	Direction	Description
IO5	I/O	single mode: output mode and forced to 0. Dual mode: output mode and forced to 0. Quad mode: output mode and forced to 0. Dual-quad mode: data input or output. Octal mode: data input or output.
IO6	I/O	single mode: output mode and forced to 0. Dual mode: output mode and forced to 0. Quad mode: output mode and forced to 0. Dual-quad mode: data input or output. Octal mode: data input or output.
IO7	I/O	single mode: output mode and forced to 0. Dual mode: output mode and forced to 0. Quad mode: output mode and forced to 0. Dual-quad mode: data input or output. Octal mode: data input or output.

Figure 45-1 OSPI octal communication mode block diagram

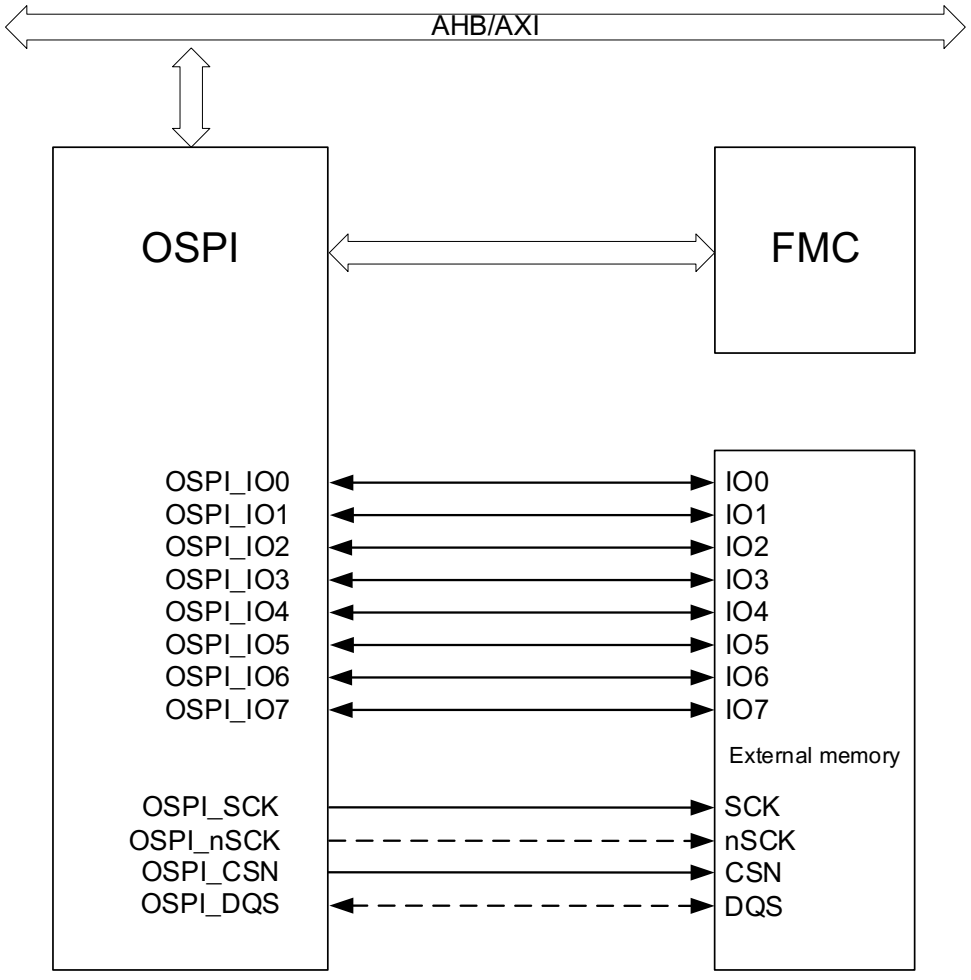


Figure 45-2 OSPI quad communication mode block diagram

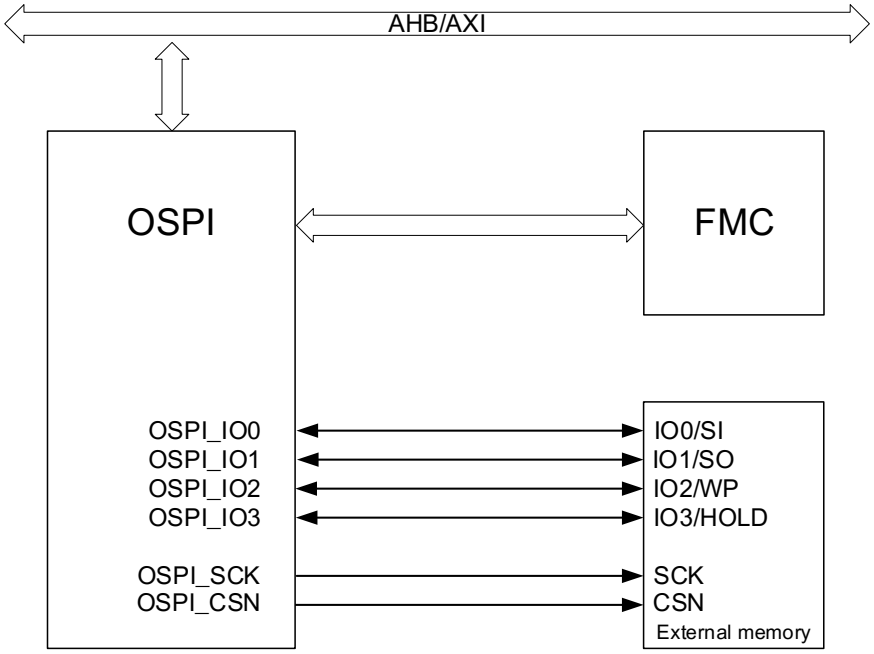
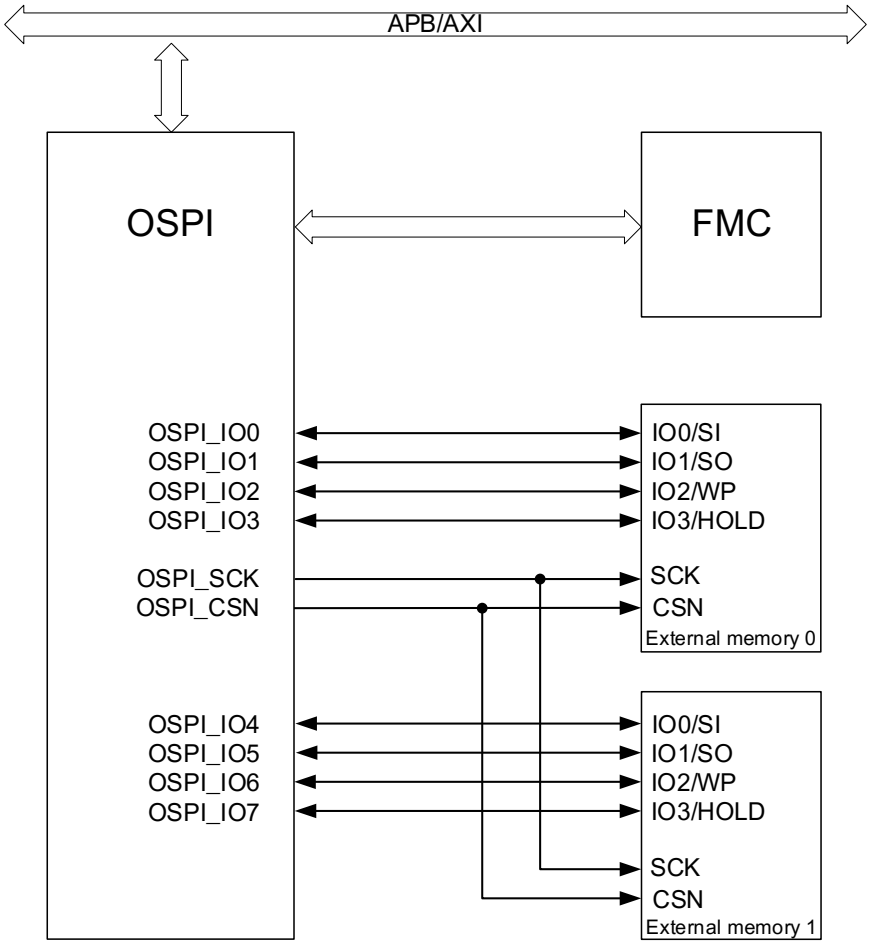


Figure 45-3 OSPI dual-quad communication mode block diagram

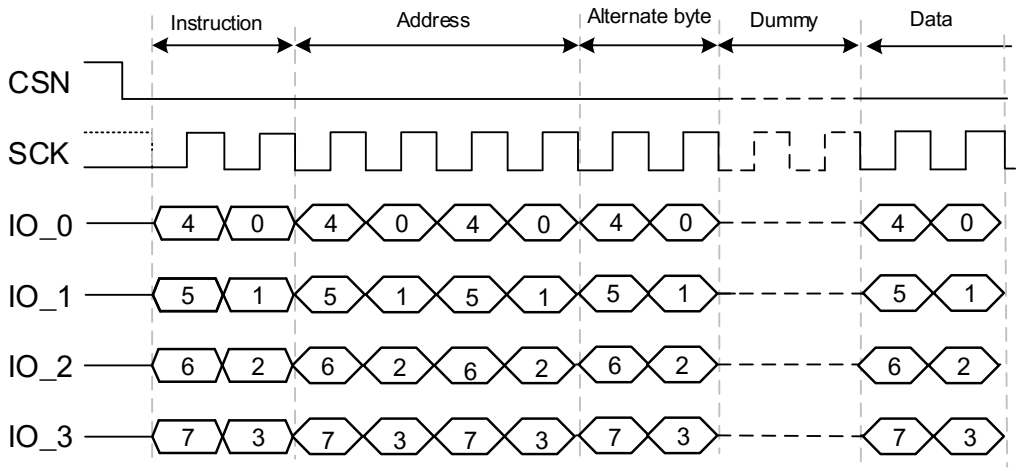


45.3.2. OSPI regular command format

In regular command, there are totally 5 phases which can be included or not: instruction, address, alternate-bytes, dummy and data. Any of these phases can be omitted or not, but at least one of the instructions phase, address phase, alternate byte phase and data phase must be present, this must be guaranteed by software, hardware is not designed to provide any protection methods. In addition, the most-signification-bit always occupies the highest IO line number.

Figure 45-4 OSPI command format in octal mode

Figure 45-5 OSPI command format in quad mode



Instruction phase

In instruction phase, configure the command size to be sent in the INSSZ[1:0] bit field of the OSPI_TCFG register(8, 16, 24 or 32 bits), configure the instructions to be sent to the external memory in the OSPI_INS register. The IMOD[2:0] field of the OSPI_TCFG register defines the instruction phase mode (no instruction, 1-line, 2-lines, 4-lines or 8-lines). Set the INSDTR bit of the OSPI_TCFG register to enable the DTR mode, and the command is sent on each edge of the clock.

When FMOD[1:0] bit field of the OSPI_CTL register is 0b11, the OSPI works in memory-mapped mode. In memory-mapped mode, define the write operation instruction in the OSPI_WINS register, configure the format of the instruction in the OSPI_WTCFG register (size, mode and whether to enable DTR). Configure the read operation instruction and the format of the instruction in the OSPI_INS register and OSPI_TCFG register.

Address phase

In address phase, configure the address size to be sent in the ADDRSZ[1:0] bit field of the OSPI_TCFG register(8, 16, 24 or 32 bits). The ADDRMOD[2:0] field of the OSPI_TCFG register defines the address phase mode (no address, 1-line, 2-lines, 4-lines or 8-lines). Set the ADDRDTR bit of the OSPI_TCFG register to enable the DTR mode, and the address is sent on each edge of the clock. In indirect mode and status polling mode, define the address information to be sent in the OSPI_ADDR register.

In memory-mapped mode, the address to be sent is directly given by AXI. Configure the format of the write operation address in the OSPI_WTCFG register (size, mode and whether to enable DTR). Configure the format of read operation address in the OSPI_TCFG register.

Alternate-bytes phase

In alternate-bytes phase, configure the alternate bytes size to be sent in ALTESZ[1:0] bit field of the OSPI_TCFG register(8, 16, 24 or 32 bits), configure the alternate bytes to be sent to the external memory in the OSPI_ALTE register. The ALTEMOD[2:0] field of the OSPI_TCFG register defines the alternate-bytes phase mode (no alternate-bytes, 1-line, 2-lines, 4-lines or 8-lines). Set the ABDTR bit of the OSPI_TCFG register to enable the DTR mode, and the alternate bytes is sent on each edge of the clock.

In memory-mapped mode, define the write operation alternate bytes in the OSPI_WALTE register, configure the format of the alternate bytes in the OSPI_WTCFG register (size, mode and whether to enable DTR). Configure the read operation alternate bytes and the format of the alternate bytes in the OSPI_ALTE register and OSPI_TCFG register.

Dummy phase

In dummy phase, 0-31 cycles, as specified by DUMYC[4:0] bit field in OSPI_TIMCFG register, are given without any data being transferred for external memory, in order to wait flash prepare data.

In memory-mapped mode, specify write operation dummy cycles in DUMYC[4:0] bit field of the OSPI_WTIMCFG register, and specify read operation dummy cycles in DUMYC[4:0] bit field of the OSPI_TIMCFG register.

Note:

6. At least 1 dummy cycle when OSPI is working in 2-lines, 4-lines or 8-lines mode to receive data from external memory.
7. At least 5 dummy cycles when using memory with DQS activated.

Data phase

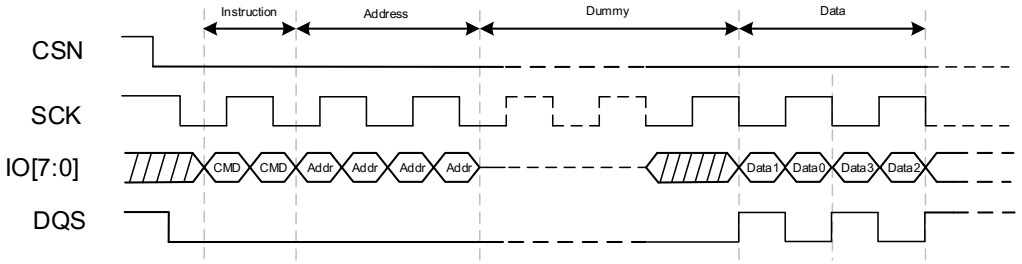
In data phase, any number of bytes can be transferred between the external memory and the OSPI interface. The DATAMOD[2:0] bit field of the OSPI_TCFG register defines the mode of the data phase(no data, 1-line, 2-lines, 4-lines or 8-lines, of which no data can only be used in indirect write mode). Set the DADTR bit of the OSPI_TCFG register to enable the DTR mode, and the data is sent on each edge of the clock. In indirect mode, the OSPI_DTLEN register defines the number of bytes to be sent or received. In write operation, data to be sent should be written to the OSPI_DATA register, while in read operation, received data is obtained by reading OSPI_DATA register. In status polling mode, OSPI_DTLEN register defines the number of bytes to be received, and the receive data is obtained from the OSPI_DATA register.

In memory-mapped mode, data to be sent is directly given by AXI. The number of bytes transmitted determines the access operation of the AXI bus, which can be read and written in 8-bit, 16-bit or 32-bit, and correspondingly transfer 1, 2, or 4 bytes. Configure the format of the data in the OSPI_WTCFG register(mode and whether to enable DTR) for sending. Configure the format of the data in the OSPI_TCFG register(mode and whether to enable DTR) for reading.

DQS signal

The DQS signal is typically used in high speed applications to indicate when the output data of memory is ready to be fetched by the MCU. When the DQS function is enabled, the toggling frequency is the same as the SCK frequency. For SDR Read operations, the data should only be latched on the rising edge of DQS signal. For DTR Read operations, the data should be latched on both rising edge and falling edge of the DQS signal.

Figure 45-6 DTR read timing using DQS in octal mode



Dual-quad mode

The dual-quad mode can be enabled by setting DQMOD to 1 in OSPI_CTL register. OSPI can access two devices at the same time in dual-quad mode, refer to ***Figure 45-3 OSPI dual-quad communication mode block diagram***. The total capacity of the two external devices is defined in the MESZ[4:0] bit field in the OSPI_DCFG0 register, and its value is twice the capacity of a single device. In dual-quad mode, the number of bytes accessed must be an even number, that is, the LSB of OSPI_DTLEN register must be 1.

In dual-quad mode, one of the devices is used to store even-numbered address data, and the other device is used to store odd-numbered address data. For example, when OSPI reads data from an external device, and the address X is an even number, the data read from the X/2 address of device A is assigned to address X, and the data read from the X/2 address of device B is assigned to address X+1.

In dual-quad mode, when OSPI read the status register of the external device, the amount of data that needs to be read is twice that of the non-this mode. When the status data given by device A and device B are both 1 byte, OSPI will receive 1 byte each from device A and device B, and OSPI needs to configure the data length to 2 bytes. When the status data given by device A and device B are both 2 bytes, OSPI needs to configure the data length to 4 bytes. At this time, the least significant byte of the OSPI data register is the least significant byte of the status data of device A, the second byte is the least significant byte of device B, and the third byte is the second byte of device A, the fourth byte is the second byte of device B.

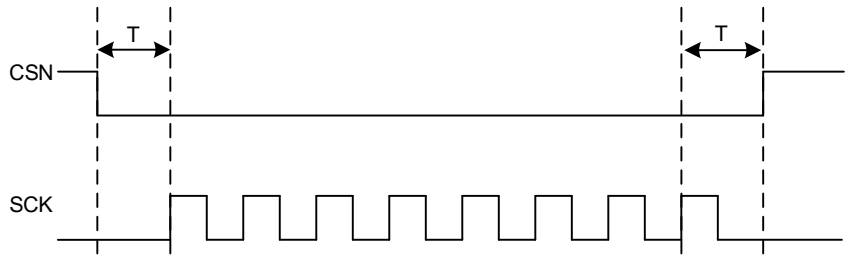
Note: Program execution is not possible in dual-quad mode.

CSN and SCK behavior

The default value of CSN is high, and it falls before a command begins and rises soon as it finishes. SCK output signal is a gate signal from internal SCK, where the internal SCK is present all the time.

The SCK must stay low when CSN is high. CSN falls one SCK cycle before the first valid rising SCK edge, and rises on SCK cycle after the final valid rising SCK edge. Refer to ***Figure 45-7 CSN and SCK behavior***.

Figure 45-7 CSN and SCK behavior



45.3.3. HyperBusTM mode

OSPI can work in HyperBus™ mode by configuring the DTYSEL[2:0] bit field in the OSPI_DCFG0 register. In this mode, OSPI can communicate with external flash memory through the HyperBus™ protocol. The HyperBus™ frame consists of two phases: command/address phase and data phase. Pins definitions used in HyperBus™ mode refer to *Table 45-2 Signal line description in HyperBusTM mode*.

Table 45-2 Signal line description in HyperBus™ mode

Pin name	Direction	Description
CSN	O	chip select output(active low)
SCK, nSCK	O	differential clock, the nSCK is used in 1.8V HyperBus™ mode
RWDS	I	read and write data strobe and latency insertion
IO[7:0]	I/O	data input/output

Instruction/address phase

In instruction/address phase, OSPI sends 48 bits data to specify a series of operations such as address space, burst type, read or write, refer to *Table 45-3 Instruction/address phase description in HyperBusTM mode*.

Table 45-3 Instruction/address phase description in HyperBus™ mode

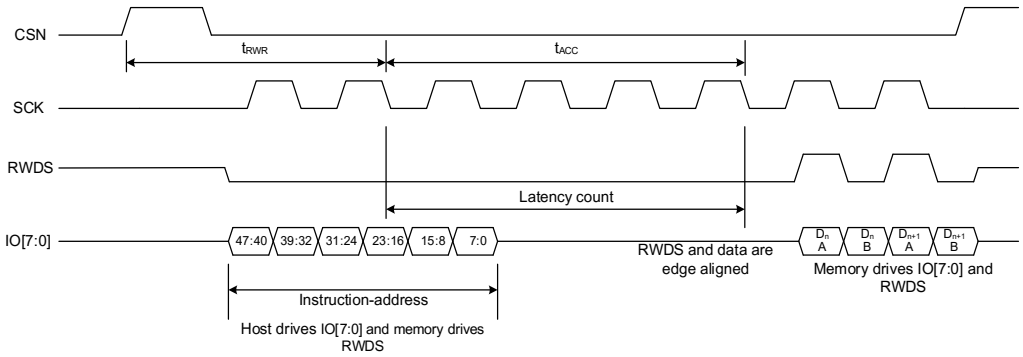
CA bit	Bit name	Description
47	R/W#	Indicates the access as a read or write operation, R/W# = 1 indicates read operation, R/W# = 0 indicates write operation.
46	Address space(AS)	Indicates whether the read or write transaction accesses the memory or register space. AS=0 indicates memory space, AS=1 indicates the register space.
45	Burst type	Indicates where the burst while be linear or wrapped. Burst type=0 indicates wrapped burst, Burst type=1 indicates linear burst.
44:16	Row and upper column address	Select the row and the upper column address
15:3	Reserved	-
2:0	Lower column address	Indicates lower column component of the target address: the starting word within a half-page.

The type of target space is configured by DTYSEL[2:0] bit field in the OSPI_DCFG0 register, and the total size of the device is configured in the MESZ[4:0] bit field in the OSPI_DCFG0 register.

Read operation

When OSPI read data from HyperBus™ device, OSPI send 48 bits CA to device firstly, and then start reading data after several cycles latency. When the data starts to be transmitted, the host reads the data and samples it on the edge of the RWDS signal. Refer to ***Figure 45-8 HyperBusTM read operation with initial latency***.

Figure 45-8 HyperBus™ read operation with initial latency



The t_{RWR} is the minimal recovery time for device, which is defined by the RWRTM[7:0] bit field of the OSPI_HBLCFG register. The t_{ACC} is the access time for the device, which is defined by ACCTM[7:0] bit field of the OSPI_HBLCFG register.

If external HyperBus™ device requires an additional latency, OSPI will add an additional t_{ACC} time, refer to ***Figure 45-9 HyperBusTM read operation with additional latency***. For accesses that cross page, an additional latency needs to be added when the page is crossed, refer to ***Figure 45-10 Read operation page crossing with latency***.

Figure 45-9 HyperBus™ read operation with additional latency

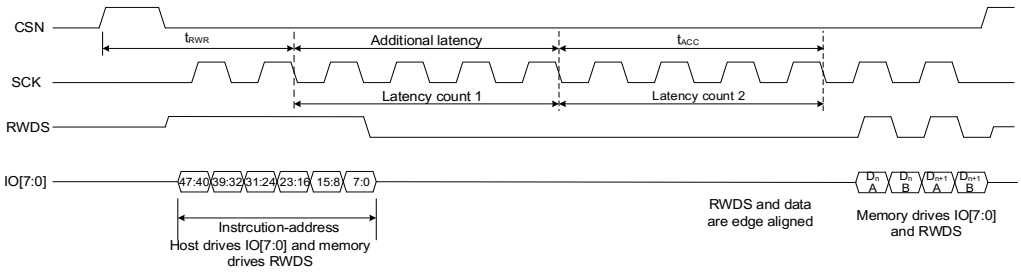
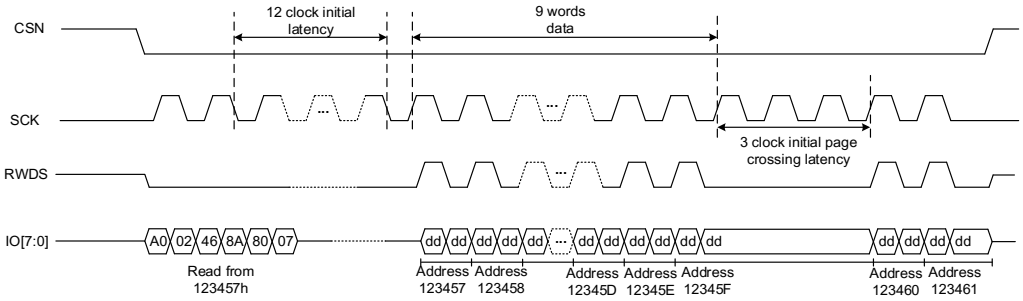


Figure 45-10 Read operation page crossing with latency



Write operation

When the host write data to HyperBus™ device, RWDS request an additional latency during the instruction/address phase and for write masking during the data phase, write operation with initial latency refer to ***Figure 45-11 HyperBusTM write operation with initial*** **latency**.

If external HyperBus™ device requires an additional latency, OSPI will add an additional t_{ACC}, refer to ***Figure 45-12 HyperBusTM write operation with additional latency***. For some devices that require zero write latency, the latency can be set to zero by configuring the WZLAT bit in the OSPI_HBLCFG register, refer to ***Figure 45-13 HyperBusTM write operation with no latency***.

For some devices that require fixed latency, the latency can be set to $2xt_{ACC}$ by configuring the LMOD bit in the OSPI_HBLCFG register. In fixed latency mode, the RWDS is not taken into account for OSPI.

Note: When there is additional delay in Hyper, to avoid interference caused by routing delay during data reception, the SSAMPLE in OSPI_TIMCFG needs to be configured.

Figure 45-11 HyperBus™ write operation with initial latency

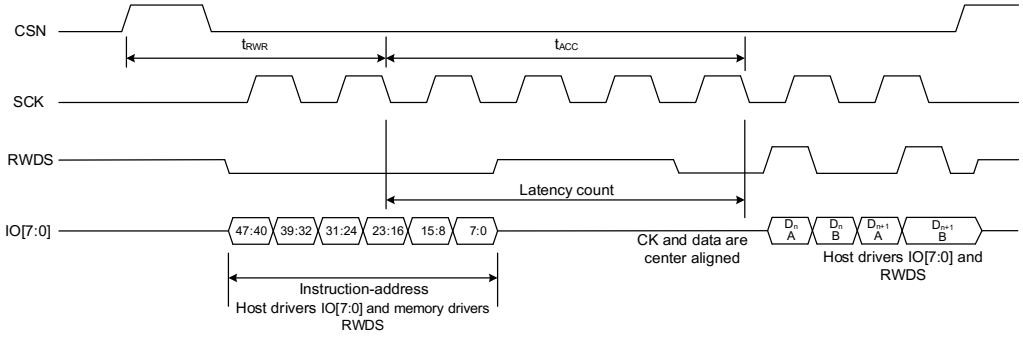


Figure 45-12 HyperBus™ write operation with additional latency

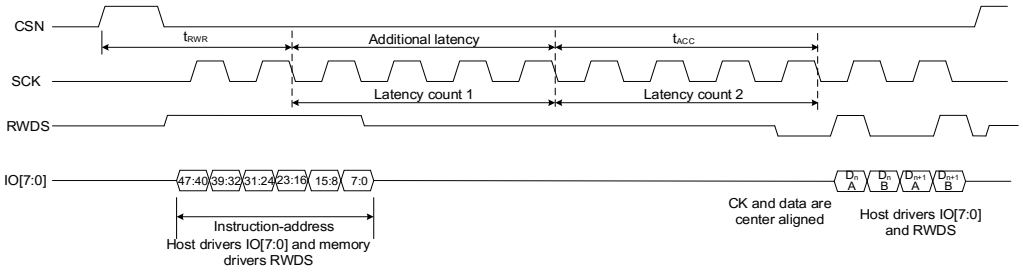
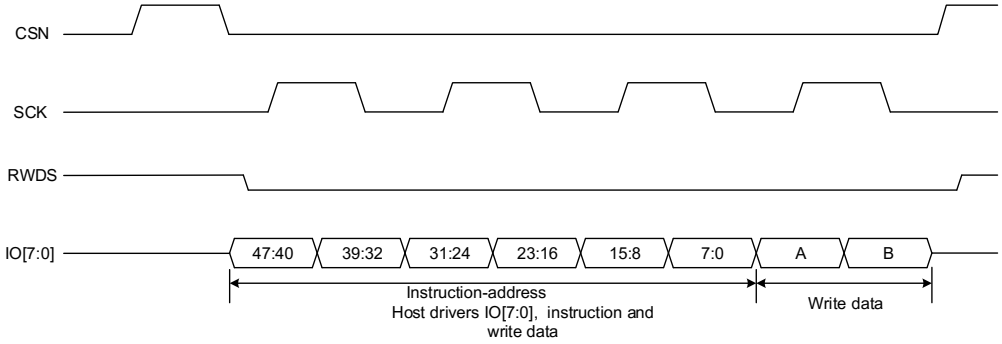


Figure 45-13 HyperBus™ write operation with no latency



45.3.4. Wrap support

Wrap is supported only in read mode when cache is enabled and when data size is 4 bytes for AHB or 8 bytes for AXI. To enable wrapped-read access, the OSPI_WPTCFG, OSPI_WPTIMCFG, OSPI_WPINS and OSPI_WPALTE must be programmed according to the wrapped read access characteristics.

45.3.5. Refresh support

The refresh feature is available only for 8 lines read in memory-mapped mode, which can limit the duration of transactions to the value defined in the REFRATE[31:0] field of the OSPI_DCFG3 register. The duration is expressed in the number of cycles, allowing external RAM devices to perform internal refresh operations regularly. The refresh value must be greater than the minimal transaction size in terms of number of cycles including the instruction/address/alternate/dummy phases.

45.4. Operating modes

45.4.1. Indirect mode

In indirect write mode, data to be transmitted are written in OSPI_DATA register. While in indirect read mode, data to be received are read from OSPI_DATA register.

OSPI_DTLEN register defines the number of byte to be transferred. Transfer complete flag TC is set when the number of byte programed in DTLEN is reached, in case of undefined transfer length, TC is set when the transmit/received byte number equals to external memory size. An interrupt is generated if TCIE and TC are both set, and it is cleared by setting TCC to 1.

Trigger a command sequence in regular command mode

The command sequence starts immediately after the last information is provided by software according to communication requirement. After the command starts, the BUSY bit is set to 1.

When neither address nor data are required, the sequence start immediately after OSPI_INS has be accessed.

When address is required and no data is required, the sequence starts after OSPI_ADDR has been accessed.

When both address and data are required in indirect write mode, the command sequence starts after OSPI_DATA has been accessed.

Trigger a command sequence in HyperBus™ mode

The command sequence starts immediately after the last information is provided by software according to communication requirement. After the command starts, the BUSY bit is set to 1.

In indirect read mode, the command sequence starts immediately after OSPI_ADDR has been accessed.

In indirect write mode, the command sequence starts immediately after OSPI_DATA has been accessed.

FIFO and flag control

A 32-byte FIFO is used for data transmission. In indirect write mode, 32-bit write access add 4-bytes to FIFO, 16-bit write access add 2-bytes to FIFO, and 8-bit add 1-byte.

FIFO threshold is defined by FTL, in indirect read mode, when the amount of bytes in the FIFO is equal or above the defined threshold, FIFO threshold flag FT is set. FT is also set after data phase is complete if FIFO is not empty. In indirect write mode, when the amount of the empty bytes in the FIFO is above the threshold, FT is set.

An interrupt is generated if both FTIE and FT is set. If MDMA is enabled, a MDMA request is generated by FT, until this flag is cleared.

In indirect read mode, when the FIFO becomes full, the OSPI temporarily stop SCK clock to avoid overrun. The reading sequence is not resumed until more than 4 byte are available in FIFO.

When the threshold exceeds 20 bytes, it is necessary to read at least 16 bytes; otherwise, the

OSPI will not continue fetching data from the external device, and the FT flag will not be set.

45.4.2. Status polling mode

In status polling mode, the OSPI periodically starts a read command with up to 4 bytes of data per frame. If the data length defined in the OSPI_DTLEN register is greater than 4, ignore the extra length and only read 4 bytes. The received data can be bit-wise masked and compared with a defined data content, if a match happens, then an interrupt is generated when SMIE is set.

Status polling access starts the same as indirect read sequence. BUSY stays high even between periodic intervals.

Polling match mode SPMOD controls the comparison match mode, if SPMOD = 0, the AND mode is selected. In this mode, status match flag SM is set only when there is a match on all the unmasked bits. While if SPMOD = 1, the OR mode is selected. In this mode, SM is set if there is a match on any of the unmasked bit.

If status-polling-mode-stop SPS is set, status polling sequence stop when a match is detected, and the BUSY flag is cleared at the end of data phase. Otherwise the periodic read continues until aborted or OSPI disabled

In status polling mode, FIFO is bypassed, the read status bytes are stored in OSPI_DATA register, and the stored status bytes are not affected by the MASK control field. OSPI_DATA contents is renewed at the beginning of data phase.

If the FT bit is set at the end of data phase, where the external flash memory status bytes are considered read, and it is cleared when OSPI_DATA register is read.

In status polling mode, the external memory must be configured in fixed latency mode for HyperBus.

45.4.3. Memory-mapped mode

In memory-mapped mode, the external flash memory is considered as internal memory, no more than 256MB can be addressed even if the external memory is larger. The Memory map mode also doesn't allow an address outside what defined by MESZ[4:0] but still within 256MB range.

If any of above condition happens, AXI will generate an error .The effect of the error depends on the AXI master. If the master is CPU, a hardfault is generate, if it is MDMA, transfer error is generated, and the corresponding MDMA channel is disabled.

In this mode, read operations support byte, half-word, word, or burst access, while write operations support word, double-word, or burst access.

Execute in place (XIP) is also supported, where OSPI continue to load bytes to the address after completing the most recent access. If the subsequent access is indeed made at a

continuous address, the access with be completed faster since the value is already prefetched. Otherwise, the read sequence is restarted, polling CSN low before the read sequence starts.

After the FIFO is empty, the OSPI enters hold state, in which no SCK is sent, CSN is maintained low during this period. If timeout counter is enabled, CSN will be pulled high when hold state contains number of SCK clock cycles equals to TMOUT control field.

At the beginning of a transfer, BUSY goes high before CSN falls, and is cleared when a timeout occurred or abort/disable is issued.

45.5. OSPI configuration

45.5.1. OSPI system configuration

The details of the OSPI system configuration as follows:

1. Configure the functional mode of OSPI with FMOD[1:0] bit field.
2. If OSPI is in status polling mode, then need to configure the SPMOD and SPS bits to select the polling matching mode and the stop mode of the automatic polling mode.
3. Configure the threshold of FIFO with FTL[4:0] bit field.
4. If need to use MDMA, set the MDMAEN bit to 1. Must be disable MDMA during OSPI configuration, otherwise unexpected requests may be generated.
5. If OSPI is in memory-mapped mode, configure the TMOUTEN bit to select whether to enable the timeout counter, and set the TIMEOUT[15:0] bit field of the OSPI_LPOUT register to configure the timeout value.
6. Set DQMOD bit to 1 if need to use dual-quad mode.
7. If need to use interrupt, the respective interrupt enable bit can be set.

45.5.2. OSPI device configuration

Configure parameters of OSPI and external device by OSPI_DCFG0 and OSPI_DCFG1 register, the details as follows:

1. Configure the external flash memory type by set the value of DTYSEL[2:0] bit field.
2. Configure the size of external flash memory by set the value of MESZ[4:0].
3. Configure chip-select minimum high time by set the value of CSHC[5:0].
4. Configure prescaler by set the value of PSC[7:0].

45.5.3. OSPI regular command configuration

Indirect mode

The details of the OSPI system configuration as follows:

1. Configure the data length by setting OSPI_DTLEN register.
2. Configure the frame timing by setting OSPI_TIMCFG register.

3. Configure the frame format by setting OSPI_TCFG register.
4. Specify the instruction to be sent to the external flash memory by setting OSPI_INS register.
5. Specify the alternate bytes to be sent to the external flash memory immediately after the address is sent by setting OSPI_ALTE register.
6. Specify the address to be sent to the external flash memory by setting OSPI_ADDR register.
7. Read/write the data through OSPI_DATA register.

Status polling mode

When OSPI is in status polling mode, the details as follows:

1. Specify to mask the received status byte by setting OSPI_STATMK register.
2. Specify the value to be compared with the OSPI_STATMK register by setting OSPI_STATMATCH.
3. Specify the number of clock cycles between read operations by setting OSPI_INTERVAL register.
4. Configure the data length by setting OSPI_DTLEN register.
5. Configure the frame timing by setting OSPI_TIMCFG register.
6. Configure the frame format by setting OSPI_TCFG register.
7. Specify the instruction to be sent to the external flash memory by set OSPI_INS register.
8. Specify the alternate bytes to be sent to the external flash memory immediately after the address is sent by setting OSPI_ALTE register.
9. Specify the address to be sent to the external flash memory by set OSPI_ADDR register.

Memory map mode

In memory map mode, external flash memory is accessed as internal memory. In this mode, the configuration of OSPI needs to be completed before accessing the memory for the first time. The specific configuration is as follows:

1. Configure the read operation frame timing by setting OSPI_TIMCFG register.
2. Configure the read frame format by setting OSPI_TCFG register.
3. Specify the instruction to be sent to the external flash memory by set OSPI_INS register.
4. Specify the alternate bytes to be sent to the external flash memory immediately after the address is sent by setting OSPI_ALTE register.
5. Configure the write operation frame timing by setting OSPI_WTIMCFG register.
6. Configure the write frame format by setting OSPI_WTCFG register.
7. Specify the instruction to be sent to the external flash memory by set OSPI_WINS register.
8. Specify the alternate bytes to be sent to the external flash memory immediately after the address is sent by setting OSPI_WALTE register.

45.5.4. OSPI HyperBusTM mode configuration

Indirect mode

When OSPI is in indirect mode, The specific configuration is as follows:

1. Configure the data length by setting OSPI_DTLEN register.
2. Specify the address to be sent to the external flash memory by set OSPI_ADDR register.
3. Read/write the data through OSPI_DATA register.

Status polling mode

When OSPI is in status polling mode, the specific configuration is as follows:

1. Specify to mask the received status byte by setting OSPI_STATMK register.
2. Specify the value to be compared with the OSPI_STATMK register by setting OSPI_STATMATCH.
3. Specify the number of clock cycles between read operations by setting OSPI_INTERVAL register.
4. Configure the data length by setting OSPI_DTLEN register.
5. Specify the address to be sent to the external flash memory by set OSPI_ADDR register.

Memory map mode

In memory mapped mode, external flash memory is accessed as internal memory. In this mode, the configuration of OSPI needs to be completed before accessing the memory for the first time. The specific configuration is as follows:

1. Configure the read operation frame timing by setting OSPI_TIMCFG register.
2. Configure the read frame format by setting OSPI_TCFG register.
3. Specify the instruction to be sent to the external flash memory by set OSPI_INS register.
4. Specify the alternate bytes to be sent to the external flash memory immediately after the address is sent by setting OSPI_ALTE register.
5. Configure the write operation frame timing by setting OSPI_WTIMCFG register.
6. Configure the write frame format by setting OSPI_WTCFG register.
7. Specify the instruction to be sent to the external flash memory by set OSPI_WINS register.
8. Specify the alternate bytes to be sent to the external flash memory immediately after the address is sent by setting OSPI_WALTE register.

45.6. Data sampling shift

The OSPI samples data after a reset, driven by the external memory for 1/2 clock cycles. When sampling with the internal clock instead of the external device's DQS, the SSAMPLE bit in the OSPI_TIMCFG register can be configured to shift the data sampling by 1/2 clock cycle. Alternatively, combined with RXSFT[2:0] in OSPI_DCFG0 register, the sampling can be

shifted by up to 7.5 cycles. When DQSEN = 1, the DQS signal can be offset to compensate for data propagation delays. This functionality can be executed and controlled either by an external delay module (CPDM) or internally.

45.7. OSPI send instruction only once

Sending instruction only once is set by SIOO, this function is valid for memory-mapped mode only, if SIOO bit is set, the instruction is sent only once after OSPI_TCFG has be accessed. Subsequent command sequence skip instruction phase, until OSPI_TCFG is accessed again. SIOO has no affect when IMOD = 00.

45.8. Busy

BUSY bit is set once the OSPI start to operate the external flash memory.

In indirect mode, BUSY is reset if the command phase is end and the FIFO is empty. In status polling mode, when a match occurs or due to an abort, the BUSY bit will be reset. After the first access in memory-mapped mode, the BUSY bit will be reset only an abort occurs.

Setting the ABORT bit to 1 allows any operation to be aborted. Once the abort action is completed, both BUSY and ABORT are reset, and the FIFO is flushed. Before setting ABORT to 1, the software must ensure that all current transactions are completed using synchronization barriers. If a write operation to a status register is aborted, some devices may exhibit abnormal behavior.

45.9. Error management

An error can be generated in the following case.

- In indirect or status polling, TERR is generated immediately when a wrong address has been programed in OSPI_ADDR register according to MESZ[4:0].
- In indirect mode, if the address (ADDR) plus data length (DTLEN) is greater than external memory size, TERR will be set once the OSPI is triggered.
- In memory-mapped mode, when an out of range access is done by AXI master will generate an AXI error.
- When an AXI master is accessing the memory-mapped space while the memory mapped mode is not enabled, will generate an AXI error.

The OSPI generate an AXI BUS error in following case:

- When an AXI read request while the memory-mapped mode is not enabled, will generate an AXI error.
- Read or write address exceeds the size of external memory.
- During a read or write, an abort occurs.

- During a read or write, OSPI is disabled.
- Write wrap burst received.

45.10. OSPI interrupt

Table 45-4 OSPI interrupt requests

Flag	Description	Clear method	Interrupt enable bit
FT	FIFO threshold	By hardware	FTIE
TC	Transfer complete	Set TCC bit in OSPI_STATC register	TCIE
TERR	Transfer error	Set TERRC bit in OSPI_STATC register	TERRIE
TMOUT	Timeout	Set TMOUTC bit in OSPI_STATC register	TMOUTIE
SM	Status match	Set SMC bit in OSPI_STATC register	SMIE

45.11. Register definition

OSPI0 base address: 0x5200 5000
OSPI1 base address: 0x5200 A000

45.11.1. Control register (OSPI_CTL)

Address offset: 0x00
Reset value: 0x0000 0000

This register can be accessed by word(32-bit)
This register cannot be modified when the BUSY bit is 1.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		FMOD[1:0]		RCKDV				SPMOD	SPS	RCKDVMUX	TMOUTIE	SMIE	FTIE	TCIE	TERRIE
		rw		rw				rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				FTL[4:0]				DEVS	DQMOD	DQSOEN	Reserved	TMOUTEN	MDMAEN	ABORT	OSPIEN
				rw				rw	rw	rw		rw	rw	rw	rw

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29:28	FMOD[1:0]	Functional mode 00: Indirect write mode 01: Indirect read mode 10: Status polling mode 11: Memory mapped mode If MDMAEN = 1, then the MDMA controller for the corresponding channel must be disabled before changing the FMOD value
27:24	RCKDV[3:0]	Receive SCK delay value(RCKDVMUX must set 1) 0000: Delay 1 cell 0001: Delay 2 cell …… 1111: Delay 16 cell
23	SPMOD	Status polling match mode 0: AND match mode. SM is set if all the unmasked bits received from the flash memory match the corresponding bits in the match register. 1: OR match mode. SM is set if any one of the unmasked bits received from the flash memory matches its corresponding bit in the match register.
22	SPS	Status polling mode stop

		This bit determines if status polling is stopped after a match. 0: Status polling mode stops by abort or OSPI is disabled. 1: Status polling mode stops as soon as there is a match.
21	RCKDVMUX	Receive SCK delay chain select 0: Enable CPDM delay 1: Enable receive SCK delay chain
20	TMOUTIE	Timeout interrupt enable 0: Disable timeout interrupt. 1: Enable timeout interrupt.
19	SMIE	Status match interrupt enable 0: Disable status match interrupt. 1: Enable status match interrupt.
18	FTIE	FIFO threshold interrupt enable 0: Disable FIFO threshold interrupt. 1: Enable FIFO threshold interrupt.
17	TCIE	Transfer complete interrupt enable 0: Disable transfer complete interrupt. 1: Enable transfer complete interrupt.
16	TERRIE	Transfer error interrupt enable 0: Disable transfer error interrupt. 1: Enable transfer error interrupt.
15:13	Reserved	Must be kept at reset value.
12:8	FTL[4:0]	FIFO threshold level This bit are useful in indirect mode, the threshold number of bytes in the FIFO that will cause the FIFO threshold flag to be set. In indirect write mode (FMOD = 00): 0: FT is set if there are 1 or more free bytes available to be written to the FIFO. 1: FT is set if there are 2 or more free bytes available to be written to the FIFO. ... 31: FT is set if there are 32 free bytes available to be written to the FIFO. In indirect read mode (FMOD = 01): 0: FT is set if there are 1 or more free bytes available to be read from the FIFO. 1: FT is set if there are 2 or more free bytes available to be read from the FIFO. ... 31: FT is set if there are 32 free bytes available to be read from the FIFO. If MDMAEN = 1, then the MDMA controller for the corresponding channel must be disabled before changing the FTL value.
7	DEVS	Device select This bit selects the device to be addressed in single/dual/quad mode in single

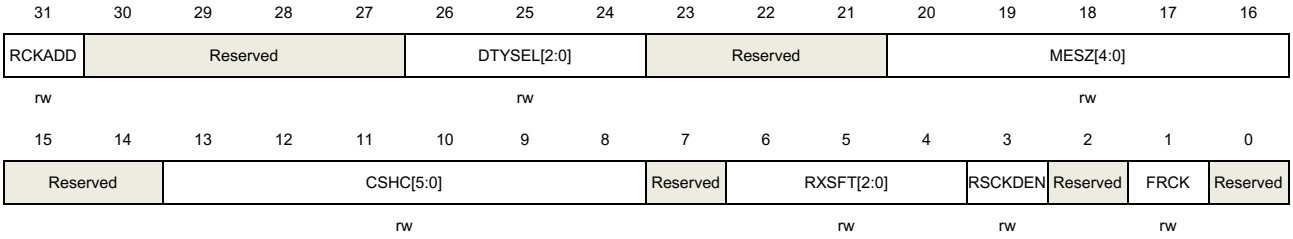
		memory mode (DQMOD=0) 0: Select device 0, transfer data through IO[3:0]. 1: Select device 1, transfer data through IO[7:4]. If OSPI is in octal/dual-quad mode, this bit is ignored.
6	DQMOD	Dual-quad mode 0: Disable dual-quad mode. 1: Enable dual-quad mode.
5	DQSOEN	DQS output enable Setting this bit to 1 will output the DQS signal during write operations. 0: Disable DQS output 1: Enable DQS output
4	Reserved	Must be kept at reset value.
3	TMOUTEN	Timeout counter enable This bit is valid only in memory-mapped mode. If there is no access to external flash, activating this bit will release CSN after the time defined by TIMEOUT[15:0]. 0: Timeout counter is disabled, and thus the chip select (CSN) remains active indefinitely after an access in memory-mapped mode. 1: Timeout counter is enabled, and thus the chip select is released in memory - mapped mode after TIMEOUT[15:0] cycles of flash memory inactivity.
2	MDMAEN	MDMA enable In indirect mode, MDMA can be used to transfer data through OSPI_DATA register. MDMA transfers are initiated when FT is set. 0: Disable MDMA. 1: Enable MDMA.
1	ABORT	Abort request This bit stops the current command and will be automatically cleared after the abort request is completed. In status polling mode or memory mapping mode, this bit will reset the SPS bit or MDMAEN bit. 0: No abort request. 1: Abort request.
0	OSPIEN	Enable OSPI 0: Disable OSPI. 1: Enable OSPI.

45.11.2. Device configuration register 0 (OSPI_DCFG0)

Address offset: 0x08
Reset value: 0x0000 0000

This register can be accessed by word(32-bit).

This register cannot be modified when the BUSY bit is 1.



Bits	Fields	Descriptions
31	RCKADD	An additional clock is sent during reception, but the received data remains unchanged. To prevent DQS from missing an edge, configuration is required when choosing DQS for reception. 0: No additional clock 1: One additional clock
30:27	Reserved	Must be kept at reset value.
26:24	DTYSEL[2:0]	Select device type 000: Micron mode, D0/D1 ordering in DTR 8 bit data mode. Regular SPI protocol in single/dual/quad/octal line mode. 001: Macronix mode, D1/D0 ordering in DTR 8 bit data mode. Regular SPI protocol in single/dual/quad/octal line mode. 010: Standard mode. 011: Macronix RAM mode, D1/D0 ordering in DTR 8 bit data mode. Regular SPI protocol in single/dual/quad/octal line mode with dedicated address mapping. 100: HyperBus memory mode, the protocol follows the HyperBus specification. 8-data-bit DTR mode must be selected. 101: HyperBus register mode, addressing register space. The memory-mapped accesses in this mode must be non-cacheable, or Indirect read/write modes must be used. Others: Reserved
23:21	Reserved	Must be kept at reset value.
20:16	MESZ[4:0]	Memory size This field defines the size of external memory using the following formula: Number of bytes in memory = $2^{[MESZ[4:0]+1]}$ MESZ[4:0]+1 is effectively the number of address bits in the memory. The memory capacity can be up to 4GB in indirect mode, while it is limited to 256MB in memory mapped mode.
15:14	Reserved	Must be kept at reset value.
13:8	CSHC[5:0]	Chip select high cycle CSHC+1 defines the minimum number of SCK cycle which the CSN must stay high between two command sequences.

		0: CSN stays high for at least 1 cycle between memory commands. 1: CSN stays high for at least 2 cycles between memory commands. ... 63: CSN stays high for at least 64 cycles between memory commands.
7	Reserved	Must be kept at reset value.
6:4	RXSFT[2:0]	Shift receive step if receive data delay more than 1/2 cycle. To configure this bit fields, ensure that the CSHC configuration time is at least 3 cycles longer than the bit fields. 000: No shift 001: 1 cycle 010: 2 cycles 011: 3 cycles 100: 4 cycles 101: 5 cycles 110: 6 cycles 111: 7 cycles
3	RSCKDEN	Sampling delay enable 0: Disable sampling delay 1: Enable sampling delay
2	Reserved	Must be kept at reset value.
1	FRCK	This bit indicates the SCK is free running or not when OSPI is free. 0: Disable SCK free running. 1: Enable SCK free running.
0	Reserved	Must be kept at reset value.

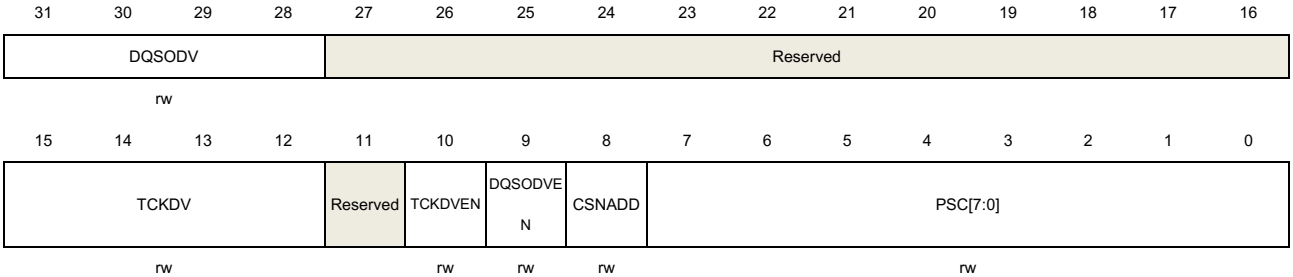
45.11.3. Device configuration register 1 (OSPI_DCFG1)

Address offset: 0x0C

Reset value: 0x0000 0000

This register can be accessed by word(32-bit)

This register cannot be modified when the BUSY bit is 1.



Bits	Fields	Descriptions

31:28	DQSODV	DQS out transfer SCK delay chain value 0000: Enable 1 delay cell 0001: Enable 2 delay cell …… 1111: Enable 16 delay cell
27:16	Reserved	Must be kept at reset value.
15:12	TCKDV	SCK transfer delay chain value 0000: Enable 1 delay cell 0001: Enable 2 delay cell …… 1111: Enable 16 delay cell
11	Reserved	Must be kept at reset value.
10	TCKDVEN	SCK transfer delay enable 0: Enable CPDM delay 1: Enable sck transfer delay
9	DQSODVEN	DQS out transfer SCK delay enable 0: Enable CPDM delay 1: Enable DQS transfer delay
8	CSNADD	CSN add 1/2 cycle 0: No add 1: Add 1/2 cycle
7:0	PSC[7:0]	This field defines the scaler factor for generating SCK based on the kernel clock (value+1) 0: $F_{CLK} = F_{KERNEL}$. In this case, DTR mode is not supported. 1: $F_{CLK} = F_{KERNEL} / 2$ 2: $F_{CLK} = F_{KERNEL} / 3$ … 255: Reserved For odd clock division factors, SCK's duty cycle is not 50%. The clock signal remains high one cycle longer than it stays low.

45.11.4. Device configuration register 3 (OSPI_DCFG3)

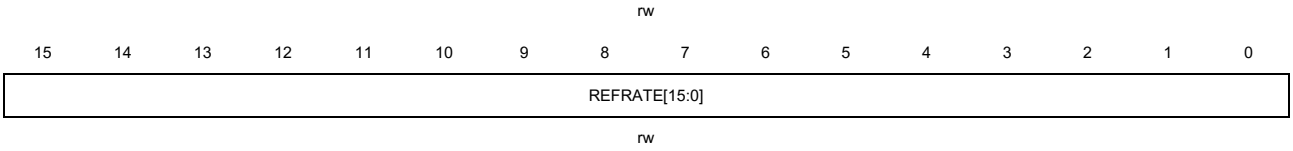
Address offset: 0x14

Reset value: 0x0000 0000

This register can be accessed by word(32-bit).

This register cannot be modified when the BUSY bit is 1.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REFRATE[31:16]															



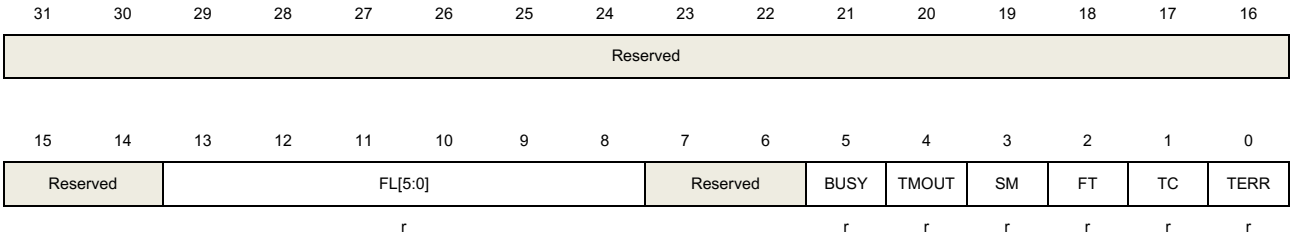
Bits	Fields	Descriptions
31:0	REFRATE	Refresh rate Only for 8 lines read in memory-mapped mode 0: Disable refresh others: The maximum communication length is REFRATE+1 clock cycles.

45.11.5. Status register (OSPI_STAT)

Address offset: 0x20

Reset value: 0x0000 0000

This register can be accessed by word(32-bit).



Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value.
13:8	FL[5:0]	FIFO level This field gives the number of valid bytes which are being held in the FIFO. In status polling mode, FL is zero.
7:6	Reserved	Must be kept at reset value.
5	BUSY	Busy This bit is set when a command is transferring. This bit is cleared once the operation with the flash memory is finished and the FIFO is empty.
4	TMOUT	Timeout flag This bit is set when timeout occurs. It is cleared by writing 1 to TMOUTC. When this bit is set, read operations in memory-mapped mode no longer guarantee data correctness. This bit must be cleared before the next transfer.
3	SM	Status match flag This bit is set in status polling mode when the unmasked received data matches the expected value. It is cleared by writing 1 to SMC.
2	FT	FIFO threshold flag

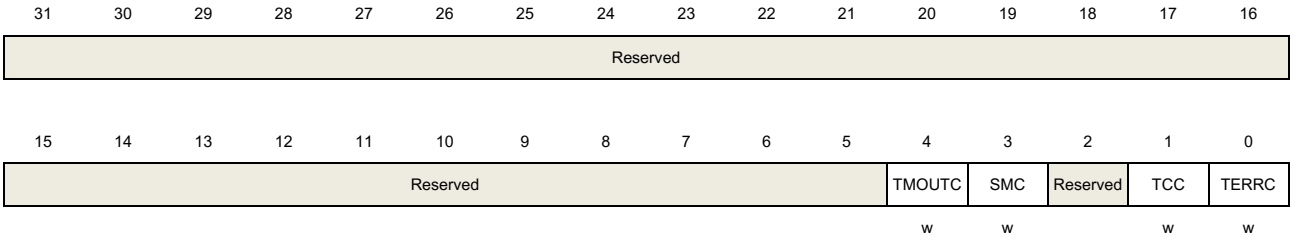
		In indirect mode, this bit is set when the FIFO threshold has been reached, or if the FIFO is not empty after the last read operation from the flash memory. In status polling mode, this bit is set when the status register is read from the external flash, and it is cleared when OSPI_DATA is read.
1	TC	Transfer complete flag This bit is set in indirect mode when the programmed number of data has been transmitted or in any mode when the transfer has been aborted. It is cleared by writing 1 TCC.
0	TERR	Transfer error flag This bit is set when an invalid address is being accessed in indirect mode. It is cleared by writing 1 to TERRC.

45.11.6. Status clear register (OSPI_STATC)

Address offset: 0x24

Reset value: 0x0000 0000

This register can be accessed by word(32-bit).



Bits	Fields	Descriptions
31:5	Reserved	Must be kept at reset value.
4	TMOUTC	Clear timeout flag Writing 1 clears the TMOUT flag in OSPI_STAT register.
3	SMC	Clear status match flag Writing 1 clears the SM flag in the OSPI_STAT register.
2	Reserved	Must be kept at reset value.
1	TCC	Clear transfer complete flag Writing 1 clears the TC flag in the OSPI_STAT register.
0	TERRC	Clear transfer error flag Writing 1 clears the TERR flag in the OSPI_STAT register.

45.11.7. Data length register (OSPI_DTLEN)

Address offset: 0x40

Reset value: 0x0000 0000

This register can be accessed by word(32-bit).

This register cannot be modified when the BUSY bit is 1.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DTLEN[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DTLEN[15:0]															
rw															

Bits	Fields	Descriptions
31:0	DTLEN[31:0]	Data length The data length is DTLEN+1 in indirect and status poling mode. The value of DTLEN no greater than 3 should be used for status polling mode. All 1's in indirect mode means undefined length, where OSPI will continue until the end of memory, as defined by MESZ[4:0]. 0x0000 0000: 1 byte will be transferred 0x0000 0001: 2 bytes will be transferred 0x0000 0002: 3 bytes will be transferred 0x0000 0003: 4 bytes will be transferred ... 0xFFFF FFFD: 4,294,967,294 (4G-2) bytes will be transferred 0xFFFF FFFE: 4,294,967,295 (4G-1) bytes will be transferred 0xFFFF FFFF: Reserved. Configuring DTLEN[31:0] to 0xFFFF FFFF may cause transmission issues. This field has no effect when in memory-mapped mode. **Note:** DTR mode does not support odd byte numbers, configuring odd bytes may cause unexpected errors.

45.11.8. Address register(OSPI_ADDR)

Address offset: 0x48

Reset value: 0x0000 0000

This register can be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR[15:0]															
rw															

Bits	Fields	Descriptions

31:0	ADDR[31:0]	Address Address to be send to the external flash memory. In HyperBus™ mode, the value written to this register must be an even number. This bit field can only be written when the BUSY bit is 0 and the memory-mapped mode is not configured.

45.11.9. Data register (OSPI_DATA)

Address offset: 0x50

Reset value: 0x0000 0000

This register can be accessed by word/half word/byte(32-bit/16-bit/8-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA[15:0]															
rw															

Bits	Fields	Descriptions
31:0	DATA[31:0]	Data Data to be transferred through the flash memory In indirect write mode, data written to this register is stored on the FIFO before sent to the flash memory. If the FIFO is full, a write operation is stalled until the FIFO has enough space. In indirect read mode, reading this register gives the data received from the flash memory. If the FIFO does not have as many bytes as requested by the read command and if BUSY=1, the read operation is stalled until enough data is present or until the transfer is complete. In status polling mode, this register contains the last data read from the flash memory.

45.11.10. Status mask register (OSPI_STATMK)

Address offset: 0x80

Reset value: 0x0000 0000

This register can be accessed by word(32-bit).

This register cannot be modified when the BUSY bit is 1.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MASK[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MASK [15:0]															

rw

Bits	Fields	Descriptions
31:0	MASK[31:0]	Status mask Mask to be applied to the status bytes received from the flash memory. For bit n: 0: Bit n of the data received is masked and its value is not considered in the matching logic. 1: Bit n of the data received is unmasked and its considered in the matching logic.

45.11.11. Status match register(OSPI_STATMATCH)

Address offset: 0x88
Reset value: 0x0000 0000

This register can be accessed by word(32-bit).
This register cannot be modified when the BUSY bit is 1.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MATCH[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MATCH[15:0]															

Bits	Fields	Descriptions
31:0	MATCH[31:0]	Status match Expected value to be compared with the masked status register to get a match.

45.11.12. Interval register (OSPI_INTERVAL)

Address offset: 0x90
Reset value: 0x0000 0000

This register can be accessed by word(32-bit).
This register cannot be modified when the BUSY bit is 1.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTERVAL[15:0]															
rw															

Bits	Fields	Descriptions

31:16	Reserved	Must be kept at reset value.
15:0	INTERVAL[15:0]	Interval cycle Number of SCK cycles between two read commands in status polling mode, INTERVAL[15:0] value must be greater than RXSFT[2:0] value. **Note:** INTERVAL[15:0] cannot be configured as 0xFFFF.

45.11.13. Transfer configuration register (OSPI_TCFG)

Address offset: 0x100

Reset value: 0x0000 0000

This register can be accessed by word(32-bit).

This register cannot be modified when the BUSY bit is 1.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SIOO	Reserved	DQSEN	Reserved	DADTR	DATAMOD[2:0]			Reserved		ALTESZ[1:0]		ABDTR	ALTEMOD[2:0]		
rw		rw		rw	rw					rw		rw	rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		ADDRSZ[1:0]		ADDRDTR	ADDRMOD[2:0]			Reserved		INSSZ[1:0]		INSDTR	IMOD[2:0]		
		rw		rw	rw					rw		rw	rw		

Bits	Fields	Descriptions
31	SIOO	Send instruction only once mode This bit has no effect when IMOD = 00. 0: Send instruction on every command sequence. 1: Send instruction only for the first command sequence.
30	Reserved	Must be kept at reset value.
29	DQSEN	DQS enable 0: Disable DQS. 1: Enable DQS.
28	Reserved	Must be kept at reset value.
27	DADTR	Data double transfer rate 0: Disable DTR mode in data phase 1: Enable DTR mode in data phase
26:24	DATAMOD[2:0]	Data mode This bit defines the operation mode of the data phase. 000: No data. 001: Data on a single line. 010: Data on two lines. 011: Data on four lines. 100: Data on eight lines. 101: Reserved.

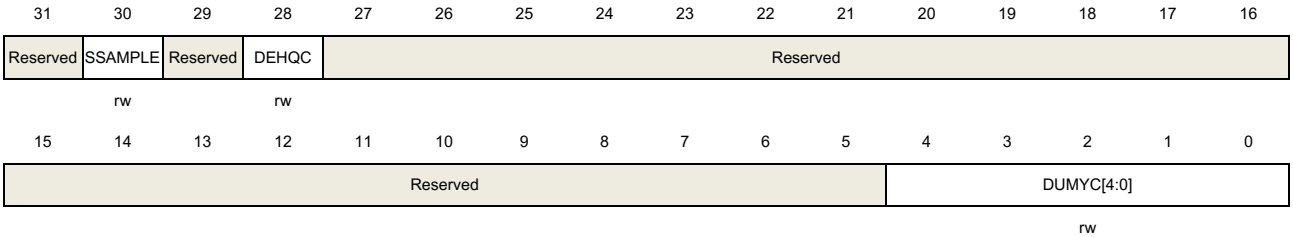
		110: Reserved. 111: Reserved.
23:22	Reserved	Must be kept at reset value.
21:20	ALTESZ[1:0]	Alternate bytes size This bit defines alternate bytes size. 00: 8-bit alternate byte 01: 16-bit alternate bytes 10: 24-bit alternate bytes 11: 32-bit alternate bytes
19	ABDTR	Alternate bytes double transfer rate 0: Disable DTR mode in alternate bytes phase 1: Enable DTR mode in alternate bytes phase
18:16	ALTEMOD[2:0]	Alternate bytes mode This field defines the alternate bytes phase mode of operation. 000: No alternate bytes. 001: Alternate bytes on a single line. 010: Alternate bytes on two lines. 011: Alternate bytes on four lines. 100: Alternate bytes on eight lines. 101: Reserved. 110: Reserved. 111: Reserved.
15:14	Reserved	Must be kept at reset value.
13:12	ADDRSZ[1:0]	Address size This bit defines address size. 00: 8-bit address. 01: 16-bit address. 10: 24-bit address. 11: 32-bit address.
11	ADDRDTR	Address double transfer rate 0: Disable DTR mode in address phase 1: Enable DTR mode in address phase
10:8	ADDRMOD[2:0]	Address mode This field defines the address phase mode of operation. 000: No address. 001: Address on a single line. 010: Address on two lines. 011: Address on four lines. 100: Address on eight lines. 101: Reserved.

		110: Reserved. 111: Reserved.
7:6	Reserved	Must be kept at reset value.
5:4	INSSZ[1:0]	Instruction size This field defines instruction size. 00: 8-bit instruction. 01: 16-bit instruction. 10: 24-bit instruction. 11: 32-bit instruction.
3	INSDTR	Instruction double transfer rate 0: Disable DTR mode in instruction phase. 1: Enable DTR mode in instruction phase.
2:0	IMOD[2:0]	Instruction mode This field defines the instruction phase mode of operation. 000: No instruction. 001: Instruction on a single line. 010: Instruction on two lines. 011: Instruction on four lines. 100: Instruction on eight lines. 101: Reserved. 110: Reserved. 111: Reserved.

45.11.14. Timing configuration register (OSPI_TIMCFG)

Address offset: 0x108
Reset value: 0x0000 0000

This register can be accessed by word(32-bit).
This register cannot be modified when the BUSY bit is 1.



Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30	SSAMPLE	Sample shift By default, the OSPI samples data 1/2 of a SCK cycle after the data is driven by

		the flash memory. This bit allows the data is to be sampled later in order to account for external signal delays. 0: No shift. 1: 1/2 cycle shift.
29	Reserved	Must be kept at reset value.
28	DEHQC	Delay hold 1/4 cycle 0: No delay hold. 1: delay hold 1/4 cycle. **Note:** to ensure correct data sampling, this bit must be set to 1.
27:5	Reserved	Must be kept at reset value.
4:0	DUMYC[4:0]	Number of dummy cycles This bit field defines the duration of the dummy instruction phase, at least 6 dummy cycles are required when DQS is used.

45.11.15. Instruction register (OSPI_INS)

Address offset: 0x110
Reset value: 0x0000 0000

This register can be accessed by word(32-bit).
This register cannot be modified when the BUSY bit is 1.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INSTRUCTION[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INSTRUCTION[15:0]															
rw															

Bits	Fields	Descriptions
31:0	INSTRUCTION[31:0]	Instruction Instruction to be send to the external flash memory.

45.11.16. Alternate bytes register (OSPI_ALTE)

Address offset: 0x120
Reset value: 0x0000 0000

This register can be accessed by word(32-bit).
This register cannot be modified when the BUSY bit is 1.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ALTE[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ALTE[15:0]															
rw															

Bits	Fields	Descriptions
31:0	ALTE[31:0]	Alternate bytes Alternate bytes to be send to the external flash memory.

45.11.17. Low power timeout register (OSPI_LPOUT)

Address offset: 0x130
Reset value: 0x0000 0000

This register can be accessed by word(32-bit).
This register cannot be modified when the BUSY bit is 1.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMEOUT[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	TIMEOUT[15:0]	Timeout cycle After access when OSPI works in memory mapped mode, this field indicates how many SCK cycles the OSPI waits after the clock becomes inactive and until raises the CSN, and the external memory is set to a low power consumption state.

45.11.18. Wrap transfer configuration register (OSPI_WPTCFG)

Address offset: 0x140
Reset value: 0x0000 0000

This register can be accessed by word(32-bit).
This register cannot be modified when the BUSY bit is 1.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		DQSEN	Reserved	DADTR	DATAMOD[2:0]			Reserved		ALTESZ[1:0]		ABDTR	ALTEMOD[2:0]		
		rw		rw	rw					rw		rw	rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		ADDRSZ[1:0]		ADDRDTR	ADDRMOD[2:0]			Reserved		INSSZ[1:0]		INSDTR	IMOD[2:0]		
		rw		rw	rw					rw		rw	rw		

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29	DQSEN	DQS enable 0: Disable DQS 1: Enable DQS
28	Reserved	Must be kept at reset value.
27	DADTR	Data double transfer rate 0: Disable DTR mode in data phase. 1: Enable DTR mode in data phase..
26:24	DATAMOD[2:0]	Data mode This bit defines the operation mode of the data phase. 000: No data. 001: Data on a single line. 010: Data on two lines. 011: Data on four lines. 100: Data on eight lines. 101: Reserved. 110: Reserved. 111: Reserved.
23:22	Reserved	Must be kept at reset value.
21:20	ALTESZ[1:0]	Alternate bytes size This bit defines alternate bytes size. 00: 8-bit alternate byte 01: 16-bit alternate bytes 10: 24-bit alternate bytes 11: 32-bit alternate bytes
19	ABDTR	Alternate bytes double transfer rate 0: Disable DTR mode in alternate bytes phase 1: Enable DTR mode in alternate bytes phase
18:16	ALTEMOD[2:0]	Alternate bytes mode This field defines the alternate bytes phase mode of operation. 000: No alternate bytes. 001: Alternate bytes on a single line. 010: Alternate bytes on two lines. 011: Alternate bytes on four lines. 100: Alternate bytes on eight lines. 101: Reserved. 110: Reserved. 111: Reserved.

15:14	Reserved	Must be kept at reset value.
13:12	ADDRSZ[1:0]	Address size This bit defines address size. 00: 8-bit address. 01: 16-bit address. 10: 24-bit address. 11: 32-bit address.
11	ADDRDTR	Address double transfer rate 0: Disable DTR mode in address phase 1: Enable DTR mode in address phase
10:8	ADDRMOD[2:0]	Address mode This field defines the address phase mode of operation. 000: No address. 001: Address on a single line. 010: Address on two lines. 011: Address on four lines. 100: Address on eight lines. 101: Reserved. 110: Reserved. 111: Reserved.
7:6	Reserved	Must be kept at reset value.
5:4	INSSZ[1:0]	Instruction size This field defines instruction size. 00: 8-bit instruction. 01: 16-bit instruction. 10: 24-bit instruction. 11: 32-bit instruction.
3	INSDTR	Instruction double transfer rate 0: Disable DTR mode in instruction phase. 1: Enable DTR mode in instruction phase.
2:0	IMOD[2:0]	Instruction mode This field defines the instruction phase mode of operation. 000: No instruction. 001: Instruction on a single line. 010: Instruction on two lines. 011: Instruction on four lines. 100: Instruction on eight lines. 101: Reserved. 110: Reserved. 111: Reserved.

45.11.19. Wrap timing configuration register (OSPI_WPTIMCFG)

Address offset: 0x148
Reset value: 0x0000 0000

This register can be accessed by word(32-bit).
This register cannot be modified when the BUSY bit is 1.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											DUMYC[4:0]				
											rw				

Bits	Fields	Descriptions
31:5	Reserved	Must be kept at reset value.
4:0	DUMYC[4:0]	Number of dummy cycles This bit field defines the duration of the dummy instruction phase, at least 5 dummy cycles are required when DQS is used.

45.11.20. Wrap instruction register (OSPI_WPINS)

Address offset: 0x150
Reset value: 0x0000 0000

This register can be accessed by word(32-bit).
This register cannot be modified when the BUSY bit is 1.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INSTRUCTION[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INSTRUCTION[15:0]															
rw															

Bits	Fields	Descriptions
31:0	INSTRUCTION[31:0]	Instruction Instruction to be send to the external flash memory.

45.11.21. Wrap alternate byte register (OSPI_WPALTE)

Address offset: 0x160
Reset value: 0x0000 0000

This register can be accessed by word(32-bit).

This register cannot be modified when the BUSY bit is 1.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ALTE[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ALTE[15:0]															
rw															

Bits	Fields	Descriptions
31:0	ALTE[31:0]	Alternate bytes Alternate bytes to be send to the external flash memory.

45.11.22. Write transfer configuration register (OSPI_WTCFG)

Address offset: 0x180

Reset value: 0x0000 0000

This register can be accessed by word(32-bit).

This register cannot be modified when the BUSY bit is 1.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		DQSEN	Reserved	DADTR	DATAMOD[2:0]			Reserved		ALTESZ[1:0]		ABDTR	ALTEMOD[2:0]		
		rw		rw	rw					rw		rw	rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		ADDRSZ[1:0]		ADDRDTR	ADDRMOD[2:0]			Reserved		INSSZ[1:0]		INSDTR	IMOD[2:0]		
		rw		rw	rw					rw		rw	rw		

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29	DQSEN	DQS enable 0: Disable DQS 1: Enable DQS
28	Reserved	Must be kept at reset value.
27	DADTR	Data double transfer rate 0: Disable DTR mode in data phase. 1: Enable DTR mode in data phase.
26:24	DATAMOD[2:0]	Data mode This bit defines the operation mode of the data phase. 000: No data. 001: Data on a single line. 010: Data on two lines. 011: Data on four lines. 100: Data on eight lines.

		101: Reserved. 110: Reserved. 111: Reserved.
23:22	Reserved	Must be kept at reset value.
21:20	ALTESZ[1:0]	Alternate bytes size This bit defines alternate bytes size. 00: 8-bit alternate byte 01: 16-bit alternate bytes 10: 24-bit alternate bytes 11: 32-bit alternate bytes
19	ABDTR	Alternate bytes double transfer rate 0: Disable DTR mode in alternate bytes phase 1: Enable DTR mode in alternate bytes phase
18:16	ALTEMOD[2:0]	Alternate bytes mode This field defines the alternate bytes phase mode of operation. 000: No alternate bytes. 001: Alternate bytes on a single line. 010: Alternate bytes on two lines. 011: Alternate bytes on four lines. 100: Alternate bytes on eight lines. 101: Reserved. 110: Reserved. 111: Reserved.
15:14	Reserved	Must be kept at reset value.
13:12	ADDRSZ[1:0]	Address size This bit defines address size. 00: 8-bit address. 01: 16-bit address. 10: 24-bit address. 11: 32-bit address.
11	ADDRDTR	Address double transfer rate 0: Disable DTR mode in address phase. 1: Enable DTR mode in address phase.
10:8	ADDRMOD[2:0]	Address mode This field defines the address phase mode of operation. 000: No address. 001: Address on a single line. 010: Address on two lines. 011: Address on four lines. 100: Address on eight lines.

		101: Reserved. 110: Reserved. 111: Reserved.
7:6	Reserved	Must be kept at reset value.
5:4	INSSZ[1:0]	Instruction size This field defines instruction size. 00: 8-bit instruction. 01: 16-bit instruction. 10: 24-bit instruction. 11: 32-bit instruction.
3	INSDTR	Instruction double transfer rate 0: Disable DTR mode in instruction phase. 1: Enable DTR mode in instruction phase.
2:0	IMOD[2:0]	Instruction mode This field defines the instruction phase mode of operation. 000: No instruction. 001: Instruction on a single line. 010: Instruction on two lines. 011: Instruction on four lines. 100: Instruction on eight lines. 101: Reserved. 110: Reserved. 111: Reserved.

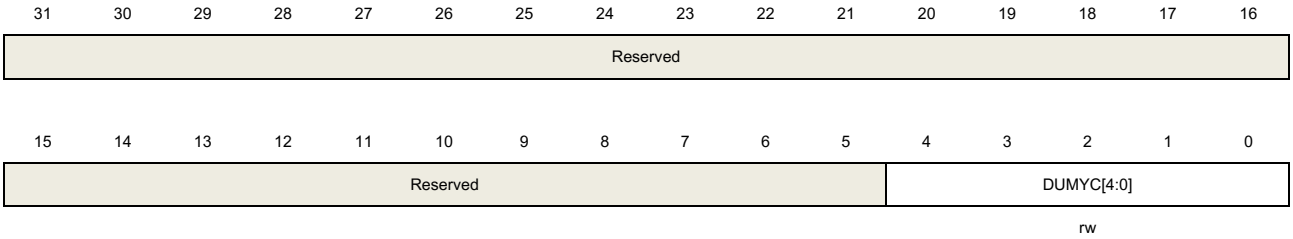
45.11.23. Write timing configuration register (OSPI_WTIMCFG)

Address offset: 0x188

Reset value: 0x0000 0000

This register can be accessed by word(32-bit).

This register cannot be modified when the BUSY bit is 1.



Bits	Fields	Descriptions
31:5	Reserved	Must be kept at reset value.
4:0	DUMYC[4:0]	Number of dummy cycles

This bit field defines the duration of the dummy instruction phase, at least 5 dummy cycles are required when DQS is used.

45.11.24. Write instruction register (OSPI_WINS)

Address offset: 0x190
Reset value: 0x0000 0000

This register can be accessed by word(32-bit).
This register cannot be modified when the BUSY bit is 1.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INSTRUCTION[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INSTRUCTION[15:0]															
rw															

Bits	Fields	Descriptions
31:0	INSTRUCTION[31:0]	Instruction Instruction to be send to the external flash memory.

45.11.25. Write alternate byte register (OSPI_WALTE)

Address offset: 0x1A0
Reset value: 0x0000 0000

This register can be accessed by word(32-bit).
This register cannot be modified when the BUSY bit is 1.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ALTE[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ALTE[15:0]															
rw															

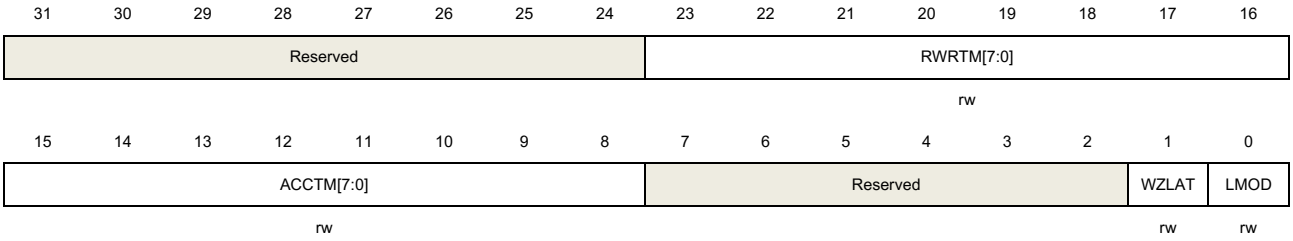
Bits	Fields	Descriptions
31:0	ALTE[31:0]	Alternate bytes Alternate bytes to be send to the external flash memory.

45.11.26. HyperBus™ latency configuration register (OSPI_HBLCFG)

Address offset: 0x200
Reset value: 0x0000 0000

This register can be accessed by word(32-bit).

This register cannot be modified when the BUSY bit is 1.



Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:16	RWRTM[7:0]	Read write recovery time The number of SCK cycles indicates the read and write recovery time.
15:8	ACCTM[7:0]	Access time The number of SCK cycles indicates the access time.
7:2	Reserved	Must be kept at reset value.
1	WZLAT	Write zero latency 0: Disable write operation zero latency. 1: Enable write operation zero latency.
0	LMOD	Latency mode 0: Variable initial latency. 1: Fixed latency.

46. Secure digital input/output interface (SDIO)

46.1. Overview

The secure digital input/output interface (SDIO) defines the SD, SD I/O and embedded MultiMediaCard (*e*•MMC) host interface, which provides command/data transfer between the AHB system bus and SD memory cards, SD I/O cards and *e*•MMC.

The supported SD memory card and SD I/O card system specifications are defined in the SD card Association website at ***www.sdcard.org***.

The supported embedded Multimedia Card system specifications are defined through the Multimedia Card Association website at ***www.jedec.org***, published by the JEDEC SOLID STATE TECHNOLOGY ASSOCIATION.

46.2. Characteristics

The SDIO features include the following:

- ***e*•MMC:** Support for embedded Multimedia Card System Specification Version 4.51 (and previous versions) Card and five different data bus modes: 1-bit (default), 4-bit (SDR/DDR) and 8-bit(SDR/DDR).
- **SD Card:** Full support for *SD Memory Card Specifications Version 3.0.*
- **SD I/O:** Full support for *SD I/O Card Specification Version 3.0* card and three different data bus modes: 1-bit (default) and 4-bit (SDR/DDR).
- 208MHz data transfer frequency and 8-bit data transfer mode.
- Interrupt and DMA request to processor.
- Support DDR double data rate signaling.

Note: SDIO supports only one SD card, SD I/O card or *e*•MMC at any one time and a stack of e•MMC V4.51 or previous.

There are two SDIO interfaces, and the implementation is shown in ***Table 46-1. 2 SDIOs***.

Table 46-1. 2 SDIOs

SDIO features	SDIO0	SDIO1
Delay module (SDR104, HS200)	+	+
SDIO_CLKIN	+	-
SDIO_CMDDIR, SDIO_DAT0DIR, SDIO_DAT123DIR	+	-
MDMA command end, MDMA data transfer end, MDMA buffer end	+	-

Note: "-" means no support; "+" means support.

46.3. SDIO function overview

46.3.1. SDIO bus topology

After a power-on reset, the host must initialize the card by a special message-based bus protocol.

Each message is represented by one of the following tokens:

Command: a command is a token which starts an operation. A command is sent from the host to a card. A command is transferred serially on the CMD line.

Response: a response is a token which is sent from the card to the host as an answer to a previously received command. A response is transferred serially on the CMD line.

Data: data can be transferred from the card to the host or vice versa. Data is transferred via the data lines. The number of data lines used for the data transfer can be 1(DAT0), 4(DAT0-DAT3) or 8(DAT0-DAT7).

The structure of commands, responses and data blocks is described in ***SDIO function overview***. One data transfer is a bus operation.

There are different types of operations. Operations always contain a command and a response token. In addition, some operations have a data token; the others transfer their information directly within the command or response structure. In this case no data token is present in an operation. The bits on the DAT0-DAT7 and CMD lines are transferred synchronous to the host clock.

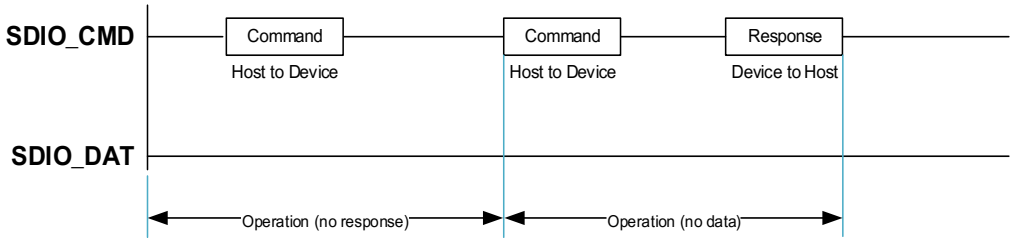
Three types of data transfer commands are defined:

- Stream commands: These commands initiate a continuous data stream; they are terminated only when a stop command follows on the CMD line. This mode reduces the command overhead to an absolute minimum. (only *e*•MMC supports)
- Block-oriented commands: These commands send a data block successfully by CRC bits. Both read and write operations allow either single or multiple block transmission. A multiple block transmission is terminated when a stop command follows on the CMD line similarly to the sequential read.
- Multibyte mode: Single data block with block size range 1-512 bytes. (only SD/SD I/O supports)

The basic transaction on the bus is the command/response transaction (refer to ***Figure 46-1. SDIO "no response" and "no data" operations***). This type of bus transaction transfers their information directly within the command or response structure. In addition, some operations

have a data token. Data transfers to/from the Card are done in blocks.

Figure 46-1. SDIO "no response" and "no data" operations



Note that the Multiple Block operation mode is faster than Single Block operation. A multiple block transmission is terminated when a stop command follows on the CMD line. Data transfer can be configured by the host to use single or multiple data lines. ***Figure 46-2. SDIO multiple blocks read operation*** is the multiple blocks read operation and ***Figure 46-3. SDIO multiple blocks write operation*** is the multiple block write operation. The block write operation uses a simple busy signal of the write operation duration on the data (DAT0) line.

Figure 46-2. SDIO multiple blocks read operation

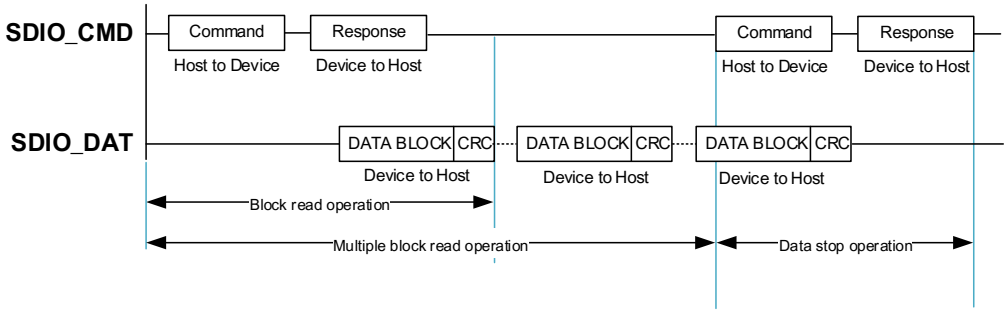
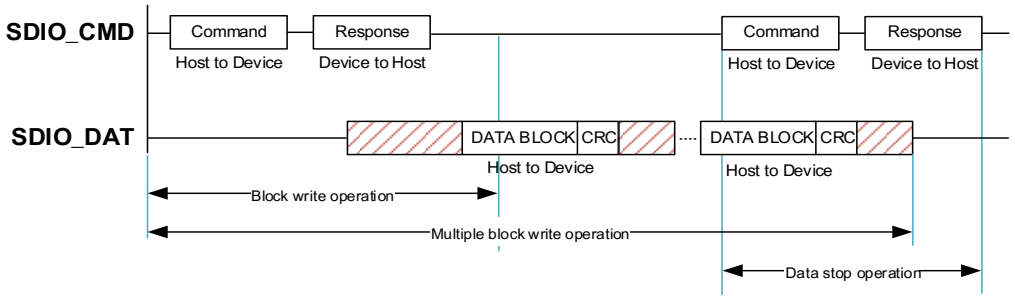


Figure 46-3. SDIO multiple blocks write operation



Data transfers to/from SD memory cards and SD I/O cards (both IO only card and combo card) are done in data blocks. Data transfers to/from e•MMC are done in data blocks or streams. ***Figure 46-4. SDIO sequential read operation*** and ***Figure 46-5. SDIO sequential***

write operation are the stream read and write operation.

Figure 46-4. SDIO sequential read operation

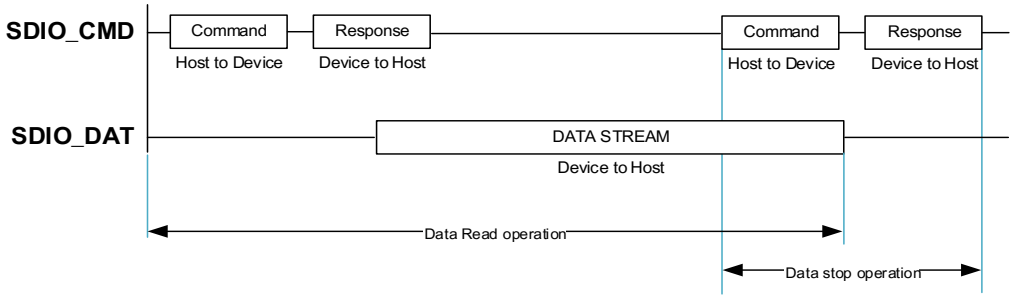
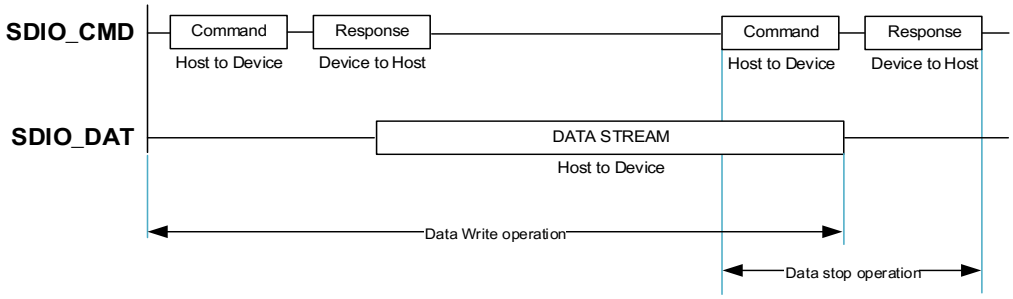


Figure 46-5. SDIO sequential write operation



46.3.2. SDIO operation modes

Table 46-2. SDIO operation modes for SD & SD I/O card

Bus Speed modes	Max Bus Speed [Mbyte/s]	Max Clock frequency [MHz]	Signal Voltage [V]
DS	12.5	25	3.3
HS	25	50	3.3
SDR12	12.5	25	1.8
SDR25	25	50	1.8
DDR50	50	50	1.8
SDR50	50	100	1.8
SDR104	104	208	1.8

Table 46-3. SDIO operation modes for *e*•MMC card

Bus Speed modes	Max Bus Speed [Mbyte/s]	Max Clock frequency [MHz]	Signal Voltage [V]
Backward compatible	26	26	3/1.8
HS SDR	52	52	3/1.8
HS DDR	104	52	3/1.8
HS200	200	200	1.8

Note: 1. DS (Default Speed), HS (High Speed). 2. SDR (single data rate signaling), DDR (double data rate signaling). 3. The SD/SD I/O card max bus speed with 4-bit bus width, and

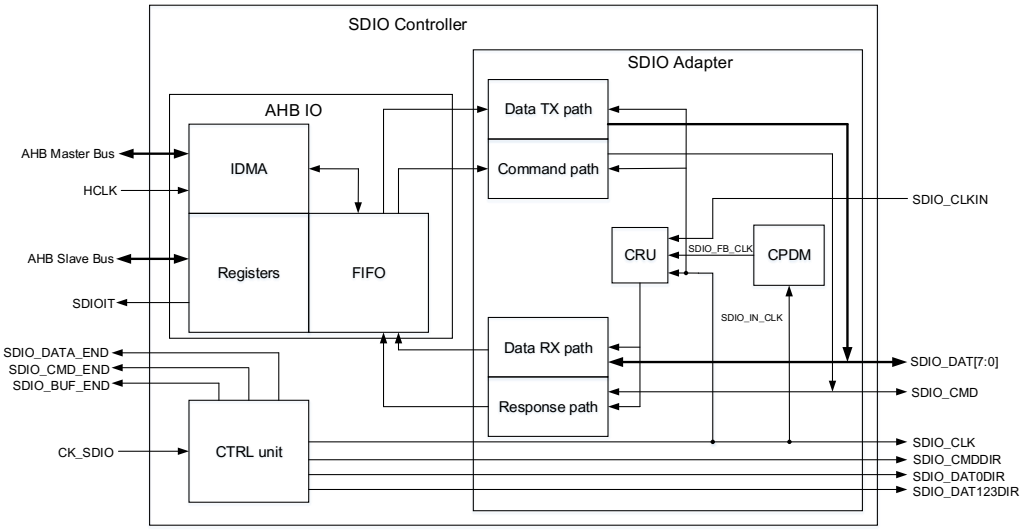
the e•MMC max bus speed with 8-bit bus width. 4. Maximum frequency depending on maximum allowed input/output speed. 5. The SDR104 and HS200 mode requires Delay module (CPDM) support using sampling point tuning. The use of Delay module for SDR50 mode is optional.

46.3.3. Block diagram

The following figure *Figure 46-6. SDIO block diagram* shows the SDIO structure. There have three main parts:

- The SDIO adapter block consists of control unit which manage clock, command unit which manage command transfer, data unit which manage data transfer.
- The AHB slave interface block contains access registers by AHB bus, contains FIFO unit which is data FIFO used for data transfer, and generates interrupt and DMA request signals.
- The internal DMA (IDMA) block with its AHB master interface.

Figure 46-6. SDIO block diagram



46.3.4. SDIO pins and internal signals

Table 46-4. SDIO internal input/output signals

Signal name	Signal type	Description
CK_SDIO	I	SDIO kernel clock
HCLK	I	AHB clock
SDIOIT	O	SDIO global interrupt
SDIO_CMD_END	O	SDIO command end trigger for MDMA
SDIO_DATA_END	O	SDIO data end trigger for MDMA
SDIO_BUF_END	O	SDIO internal DMA buffer end trigger for MDMA
SDIO_IN_CLK	I	The card feedback clock signal is connected to the SDIO_CLK

Signal name	Signal type	Description
		pin (for DS/HS modes).
SDIO_FB_CLK	I	The card tuned feedback clock signal after CPDM (for DDR50, SDR50, SDR104, HS200).

Table 46-5. SDIO pins functions

Pin name	Signal name	Signal type	Description
SDIO_CK	SDIO_CLK	O	Clock to SD/SD I/O/*e*•MMC card
SDIO_CKIN	SDIO_CLKIN	I	Clock feedback from an external driver for card. (for SDR12, SDR25, SDR50, DDR50)
SDIO_CMD	SDIO_CMD	I/O	Bidirectional command/response signal
SDIO_CMDDIR	SDIO_CMDDIR	O	Direction indication for the SDIO_CMD signal
SDIO_D[7:0]	SDIO_DAT[7:0]	I/O	Data input/output for data lines DAT[7:0]
SDIO_DAT0DIR	SDIO_DAT0DIR	O	Direction indication for the SDIO_DAT0 data line
SDIO_DAT123DIR	SDIO_DAT123DIR	O	Direction indication for the data lines SDIO_DAT[3:1]

Note: I: input; O: output.

46.3.5. General description

The SDIO_DAT[7:0] lines have different operating modes:

- When the card is powered on, it uses 1-bit (SDIO_DAT0) data bus by default, and can be modified by register later.
- For an SD or an SD I/O card, 1-bit (SDIO_DAT0) or 4-bit (SDIO_DAT [3:0]) can be used. All data lines operate in push-pull mode.
- For an *e*•MMC, 1-bit (SDIO_DAT0), 4-bit (SDIO_DAT[3:0]) or 8-bit (SDIO_DAT[7:0]) data bus widths can be used.

To allow the connection of a voltage switch transceiver, the I/O direction signals are used to indicates the direction of data flow on the data lines. The SDIO_DAT0DIR signal indicates the I/O direction for the SDIO_DAT0 data line, the SDIO_DAT123DIR for the SDIO_DAT[3:1] data lines, and the direction of data flow on the SDIO_CMD line is indicated with the I/O direction signal SDIO_CMDDIR. SDIO_CMD only operates in push-pull mode.

SDIO_CLK clock to the card originates from CK_SDIO:

- If the CK_SDIO clock has 50% duty cycle, it can be used even DIV = 0.
- If the CK_SDIO duty cycle is not 50%, the DIV > 0 must satisfy.
- The phase relation between the SDIO_CMD/SDIO_DAT[7:0] outputs and the SDIO_CLK can be selected through the CLKEDGE bit. The phase relation depends on the DIV, CLKEDGE, and DRSEL settings. Refer to ***Table 46-6. SDIO CMD and data phase***

selection.

Table 46-6. SDIO CMD and data phase selection

DIV	DRSEL	CLKEDGE	SDIO_CLK	CMD out	Data out
= 0	-	0	generated on CK_SDIO rising edge	generated on SDIO_CLK rising edge	
= 0		1		generated on SDIO_CLK falling edge	
> 0	0	0	generated on CK_SDIO rising edge	generated on CK_SDIO falling edge succeeding the SDIO_CLK rising edge.	
> 0	0	1		generated on SDIO_CLK falling edge (aligned with the rising edge of CK_SDIO).	
> 0	1	0	generated on CK_SDIO rising edge	generated on CK_SDIO falling edge succeeding the SDIO_CLK rising edge.	Generated on CK_SDIO falling edge succeeding the SDIO_CLK rising/ falling edge.
> 0	1	1		generated on SDIO_CLK falling edge (aligned with the rising edge of CK_SDIO).	1/4 of SDIO_CLK

By default, SDIO select SDIO_IN_CLK feedback clock input samples the incoming data in the SDIO receiving path. The SDIO_IN_CLK is derived from the SDIO_CLK pin. For tuning the phase of the sampling clock to adapt to the timing of the received data, the CPDM on the device can be connected to the SDIO_FB_CLK clock signal and the SDIO_IN_CLK. Then select the input of SDIO_FB_CLK clock for the receive path to use the phase adjusted sampling clock for the incoming data. This is what SDR104/HS200 working mode must meet, and SDR50/DDR50 is optional.

The SDIO_CLKIN feedback clock input signal can be selected to sample the received data, when using an external driver (voltage switching transceiver).

For an SD /SD I/O/ *e*•MMC card, the clock frequency can vary from 0 to 208 MHz (limited by maximum input/output speed).

Depending on the selected bus mode (SDR or DDR), one or two bits are transferred on SDIO_DAT[7:0] lines per clock cycle. The SDIO_CMD line transfers only one bit per clock cycle.

46.3.6. SDIO adapter

The SDIO adapter contains control unit, command unit and data unit, and generates signals to cards. The signals is descript bellow:

SDIO_CLK: The SDIO_CLK is the clock provided to the card. Each cycle of this signal directs a one bit transfer on the command line (SDIO_CMD). The SDIO_CLK frequency can vary between 0 MHz and 48 MHz for a MMC Card V4.2, between 0 and 200 MHz for a *e*•MMC V4.51, or between 0 and 208 MHz for an SD/SD I/O card.

SDIO uses two clock signals: the SDIO adapter clock and the AHB bus clock (HCLK).

SDIO_CMD: This signal is a bidirectional command channel used for card initialization and transfer of commands. Commands are sent from the SDIO controller to the card and responses are sent from the card to the host. The CMD signal has two operation modes: open-drain for initialization(used only for e•MMC card initialization), and push-pull for command transfer (SD/SD I/O card and data transfer for *e*•MMC card).

SDIO_DAT[7:0]: These are bidirectional data channels. The DAT signals operate in push-pull mode. Only the card or the host is driving these signals at a time. By default, after power up or reset, only DAT0 is used for data transfer. A wider data bus can be configured for data transfer, using either DAT0-DAT3 or DAT0-DAT7 (just for *e*•MMC 4.51), by the SDIO controller. The SDIO includes internal pull-ups for data lines DAT1-DAT7. Right after entering to the 4-bit mode the card disconnects the internal pull-ups of lines DAT1-DAT3. Correspondingly right after entering to the 8-bit mode the card disconnects the internal pull-ups of lines DAT1-DAT7.

SDIO_CLKIN: This signal is an digital input signal, and used for the clock feedback from an external driver of SD/SD I/O/*e*•MMC card. (for SDR12/SDR25/SDR50/DDR50)

SDIO_CMDDIR: This signal is an digital output signal. As card I/O direction indication for the SDIO_CMD signal.

SDIO_DAT0DIR: This signal is an digital output signal. As card I/O direction indication for the SDIO_DAT0 data line.

SDIO_DAT123DIR: This signal is an digital output signal. SD/SDIO/*e*•MMC card I/O direction indication for the data lines SDIO_DAT[3:1].

The SDIO adapter is an interface to SD, SD I/O and *e*•MMC. It consists of several subunits:

Control unit

The control unit contains the power management functions, the clock management functions for the memory card clock and I/Os direction management.

The power management is controlled by SDIO_PWRCTL register which implements power off or power on. The power management subunit will disable the card bus output signal during the reset, power-off, and power-up phases The power saving mode configured by setting CLKPWRSAV bit in SDIO_CLKCTL register, which implements close the SDIO_CLK when the bus is idle.

Clock management uses CK_SDIO to generate the SDIO_CLK clock signal to the card and provide divider control. Adapter register and FIFO use AHB clock domain (HCLK). The control unit, command send path, and data send path use the SDIO adapter clock domain (CK_SDIO). The command response path and data receive path use the SDIO adapter feedback clock domain from SDIO_IN_CLK, SDIO_CLKIN, or SDIO_FB_CLK (generated by CPDM).

The I/O direction management controls the external voltage transceiver and controls signals

for SDIO_CMDDIR, SDIO_D0DIR, and SDIO_D123DIR.

Command unit

The command unit contains a command send path and a response receive path, which are used to send commands and receive responses to cards on the SDIO_CMD line. The command send path is clocked by SDIO_CLK, and the response receive channel has a dedicated SDIO internal receive clock. The data transfer flow is controlled by Command State Machine (CSM). After a write operation to SDIO_CMDCTL register and CSMEN in SDIO_CMDCTL register is 1, the command transfer starts. It firstly sends a command to the card. The command contains 48 bits send by SDIO_CMD signal which sends 1 bit to card at one SDIO_CLK. The 48 bits command contains 1 bit Start bit, 1 bit Transmission bit, 6 bits command index defined by CMDIDX bits in SDIO_CMDCTL register, 32 bits argument defined in SDIO_CMDAGMT register, 7 bits CRC, and 1 bit end bit. Then receive response from the card if CMDRESP in SDIO_CMDCTL register is not 0b00. There are short response which have 48 bits or long response which have 136 bits. The response stores in SDIO_RESP0 - SDIO_RESP3 registers. The command unit also generates the command status flags defined in SDIO_STAT register.

Command state machine

CS_Idle	After reset, ready to send command.	
1.CSM enabled and WAITDEND enabled	->	CS_Pend
2.CSM enabled, WAITDEND disabled and BOOTMODEN disabled	->	CS_Send
3.BOOTMODEN enabled	->	CS_Boot
Note: The state machine remains in the Idle state for at least eight SDIO_CLK periods to meet the N_{CC} and N_{RC} timing constraints. N_{CC} is the minimum delay between two host commands, and N_{RC} is the minimum delay between the host command and the response.		

CS_Pend	Waits for the end of data transfer.	
1.The data transfer complete or WAITDEND disabled	->	CS_Send
Note: If DATALEN =< 6, CSM directly moves to the CS_Send state; If DATALEN > 6, CSM will wait for the signal from DSM before moves to the CS_Send state.		

CS_Send	Sending the command.	
1.The command transmitted has response	->	CS_Wait
2.The command transmitted doesn't have response	->	CS_Idle

CS_Wait	Wait for the start bit of the response.	
1.Receive the response(detected the start bit)	->	CS_Receive
2.When INTWAIT is disabled, a response timeout	->	CS_Idle

occurs.		
3. When INTWAIT transitions from disabled to enabled, a response is received (start bit detected).	->	CS_Receive
Note: The command timeout has a fixed value of 64 SDIO_CLK clock periods.		

CS_Receive	Receive the response and check the CRC.	
1.CMDRECV is set when the CRC check is successful or there is no CRC.	->	CS_Idle
2.Response received	->	CS_Idle
3.CCRCERR is set when the command CRC check fails.	->	CS_Idle

CS_Boot	Read boot data from device.	
1.Select the normal boot mode and enable TREN and DATADIR.	->	CS_Boot
2.Normal boot operation completed.	->	CS_Idle
3.BOOTMOD = 1	->	CS_Send

Data unit

The data unit performs data transfers to and from cards. The data transfer uses SDIO_DAT[7:0] signals when 8-bit data width (BUSMODE bits in SDIO_CLKCTL register is 0b10), use SDIO_DAT[3:0] signals when 4-bit data width (BUSMODE bits in SDIO_CLKCTL register is 0b01), or SDIO_DAT[0] signal when 1-bit data width (BUSMODE bits in SDIO_CLKCTL register is 0b00). The data transfer flow is controlled by Date State Machine (DSM). After a write operation to SDIO_DATACTL register and DATAEN in SDIO_DATACTL register is 1, the data transfer starts. It sends data to card when DATADIR in SDIO_DATACTL register is 0, or receive data from card when DATADIR in SDIO_DATACTL register is 1. The data unit also generates the data status flags defined in SDIO_STAT register. When data is received and the boot acknowledge is enabled, DSM move to the DS_WaitACK state and waits for the boot acknowledge, and then move to the DS_WaitR state.

Data state machine

DS_Idle	The data unit is inactive, waiting for send and receive.	
1.(DSM enabled or DATAEN = 1) and (Not busy and data transfer direction is from host to card)	->	DS_WaitS
2.DSM enabled and ACKEN disabled and data transfer direction is from card to host	->	DS_WaitR
3.DSM enabled and ACKEN enabled and data transfer direction is from card to host	->	DS_WaitACK

DS_WaitS	Wait until the data FIFO empty flag is deasserted or data

	transfer ended.		
	1.Data transfer ended	->	DS_Idle
	2.CSM abort and data FIFO empty	->	DS_Idle
	3.DATHOLD flag is set	->	DS_Idle
	4.Data FIFO empty flag is deasserted and DATHOLD = 0	->	DS_Send

DS_Send	Transmit data to the card.	
1.Data block transmitted	->	DS_Busy
2.CSM abort	->	DS_Busy
3.Stream transmission mode, data transfer is complete.	->	DS_Busy
4.FIFO underrun error occurs	->	DS_Send

DS_Busy	Waits for the CRC status flag.	
1.Correct CRC status received and not busy	->	DS_WaitS
2.CRC check failed and data transfer is not fully completed	->	DS_Busy
3.Receive a negative CRC status and not busy	->	DS_Idle
4.CSM abort and not busy	->	DS_Idle
Note: The command timeout programmed in the data timer register (SDIO_DATATO).		

DS_WaitR	Wait for the start bit of the receive data.	
1.Data receive ended and data FIFO empty	->	DS_Idle
2. CSM abort	->	DS_Idle
3. DATHOLD = 1 and FIFO empty	->	DS_Idle
4.Receives a start bit before timeout	->	DS_Receive
Note: The command timeout programmed in the data timer register (SDIO_DATATO).		

DS_Receive	Receive data from the card and write it to the data FIFO.	
1.Data block received, CRC verification OK, and RWEN = 0	->	DS_WaitR
2.Data FIFO overrun error occurs	->	DS_Receive
3.CSM abort and FIFO empty	->	DS_Idle
4.CRC check failed, FIFO empty and data receive completed	->	DS_Idle
5.In SD I/O mode, the data has been received, the CRC check is OK, not all data has been received, and RWEN = 1	->	DS_Readwait
6.CRC check failed, not all data received, and RWEN = 0		DS_Receive
7.Stream transmission mode, data transfer is complete.	->	DS_WaitR

DS_WaitACK	Wait for the boot acknowledgment token.	
1.Receive the boot acknowledge in time and check OK	->	DS_WaitR
2.Acknowledgement timed out or incorrect acknowledgement received	->	DS_WaitACK
3.CSM abort	->	DS_Idle

DS_Readwait	Wait for the read wait stop command.	
1.ReadWait stop enabled	->	DS_WaitR
3. CSM abort and FIFO empty	->	DS_Idle

CRU unit

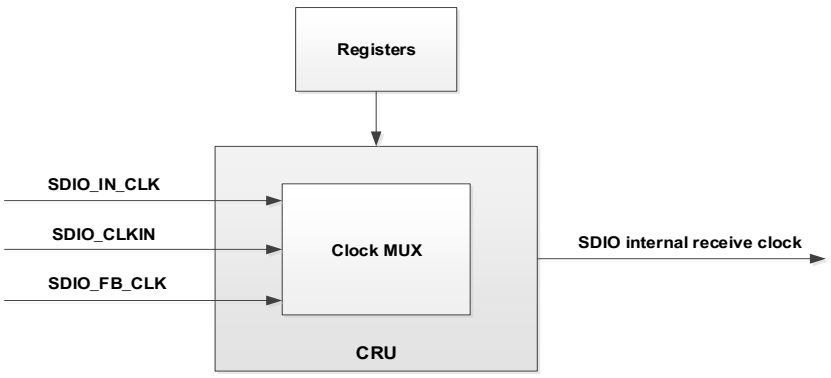
The CRU unit selects the source for SDIO internal receive clock to be used with the received command response and data.

The RCLK[1:0] register can be set to select the clock source for receiving data, the three options are as follows:

- SDIO_IN_CLK bus master main feedback clock.
- SDIO_CLKIN external bus feedback clock.
- SDIO_FB_CLK bus tuned feedback clock.

Note: 1. When there is no external drive and DS/HS are used, select SDIO_IN_CLK. 2. Select SDIO_CLKIN when there are external drives and SDR12, SDR25, SDR50 and DDR50 are used. 3. When using CPDM, SDIO_FB_CLK must be selected input. If it is SDR104 or HS200, CPDM must be selected. If it is SDR50 or DDR50, CPDM is optional. 4. If CPM and DSM are in the idle state, the SDIO internal receive clock source must be changed.

Figure 46-7. CRU unit



46.3.7. AHB slave interface

The AHB slave interface implements access to SDIO registers, data FIFO and generates interrupt. It includes a data FIFO unit, registers unit, and the interrupt.

It consists the following subunits:

Interrupts

The interrupt logic generates interrupt when at least one of the selected status flags is high. An interrupt enable register is provided to allow the logic to generate a corresponding interrupt. The status flag generates an interrupt request if the corresponding interrupt enable flag is set. Some status flags require an implicit clear in the clear register.

Register unit

The register unit which contains all system registers generates the signals to control the communication between the controller and card.

Data FIFO

The data FIFO unit has a data buffer, uses as transmit and receive FIFO. The FIFO contains a 32-bit wide, 16-word deep data buffer. When accessing the FIFO with half word or byte accesses an AHB bus fault is generated.

The transmit FIFO is used when write data to card. The data to be transferred is written to transmit FIFO by AHB bus, the data unit in SDIO adapter read data from transmit FIFO, and then send the data to card. When the DATALEN is not an integer multiple of 4, the last data less than 4 bytes (1/2/3 bytes) will be written in word transfer.

The receive FIFO is used when read data from card. The data to be transferred is read from the card and then write to receive FIFO. The data in receive FIFO is read to AHB bus when needed. When the DATALEN is not an integer multiple of 4, the last data less than 4 bytes (1/2/3 bytes) are read with a word transfer padded with 0 value bytes.

46.3.8. AHB master interface

AHB main interface uses SDIO internal DMA (IDMA) to transfer data between FIFO and memory.

IDMA

IDMA provides a bidirectional high-speed data transmission channel between memory and FIFO.

The IDMA is enabled by setting the IDMAEN bit to 1, and supports 8-beat burst transmission.

Burst mode includes transmit mode and receive mode:

- Transmit of burst mode
- As long as the FIFO is empty for the number of burst transfers, the data will be obtained from memory in the form of burst until all data indicated by DATALEN have been transmitted. If DATALEN is not a multiple of the burst size, the remaining data with less

than the burst size will be transmitted in single transmission mode. When DATALEN is not a multiple of 4, the last data less than 4 bytes (1/2/3 bytes) is obtained by word transmission.

- Receive of burst mode
- As long as the FIFO is empty for the number of burst transfers, the data will be stored in the memory in the form of burst until all the data indicated by DATALEN have been transmitted. If DATALEN is not a multiple of burst transmission, the remaining data with less than burst size will be transmitted in single transmission mode. When DATALEN is not a multiple of 4, the last data less than 4 bytes (1/2/3 bytes) are stored through half word or byte transmission.

In addition, IDMA also provides two channel configurations (selected by bit BUFMOD).

- Single buffered channel
- In single buffer configuration, data on the memory side is accessed linearly from the base address IDMAADDR0. When IDMA completes the transmission of all data and DSM also completes the transmission, the DTEND flag will be set to 1.
- Double buffered channel
- In the double buffer configuration, the data on the memory side is accessed from two buffers, one from IDMAADDR0 and the other from IDMAADDR1. In this way, when IDMA accesses one of the memory buffers, the firmware can process the other. The size of the memory buffer is defined by IDMASIZE. The buffer size should be an multiple of the burst size. When the channel is enabled, the base address of the buffer can be updated immediately.

In the double buffered channel mode, the address of access memory is configured by the BUFSEL bit.

- When the BUFSEL bit is "0", IDMA hardware uses IDMAADDR0 to access the memory. When trying to write the register through firmware, the write operation will be discarded and IDMAADDR0 data will not change. Allow firmware write IDMAADDR1.
- When the BUFSEL bit is "1", IDMA hardware uses IDMAADDR1 to access memory. When an attempt is made to write the register through firmware, the write operation will be discarded and the IDMAADDR1 data will not change. Allow firmware to write IDMAADDR0.

When IDMA completes data transmission in one of the buffers, the buffer transmission completion flag (IDMAEND) will be set to 1 and the BUFSEL bit will flip, and then IDMA will continue to transmit data from the other buffer. When IDMA completes the transmission of all data and DSM also completes the transmission, the DTEND flag will be set to 1.

IDMAADDR0 and IDMAADDR1 addresses should be word aligned.

46.3.9. MDMA request

The internal trigger line (SDIO_DATA_END、SDIO_CMD_END and SDIO_BUF_END) from SDIO can send direct request to MDMA controller to realize continuous transmission from/to

different internal RAM addresses without using CPU.

The request signal of MDMA is transmitted through SDIO_DATA_END pin input. At the same time, the signal input triggers the clearing of DTEND and CMDRECV flag, and finally starts a new transmission by directly accessing the SDIO control and configuration registers with MDMA, without CPU intervention.

When the response to the command is successfully received, the CMDRECV flag will be set. When the busy state after R1b response ends, DAT0BSY flag of status register is cleared and DAT0BSYEND flag is set. When the sequence command response associated with an eventual busy signal ends, set the SDIO_CMD_END output connected to the MDMA. So, the MDMA can manage CMD12 (STOP_TRANSMISSION) command (needed to support open mode transfers) by clearing CMDRECV and DAT0BSYEND status flags.

When using Linux operating system, the data to be transmitted through SDIO bus is contained in the 1-4kbyte blocks of the discontinuous address in the internal memory of the device. The double buffer mode allows the destination address of IDMA to be changed in internal memory. The IDMAEND flag of the status register is set each time the buffer transfer is completed. By signaling this event to MDMA through the SDIO_BUF_END output connected to an MDMA request input, the new buffer address base can alternatively fill the IDMAADDR0/IDMAADDR1 fields without CPU intervention.

The *Table 46-7. SDIO connections to MDMA* shows the actions of programming in MDMA according to SDIO requests:

Table 46-7. SDIO connections to MDMA

Trigger signal	Event signaled	Event occurrence	MDMA transfer config	MDMA action
SDIO_DATA_END	End of successful data transfer	DTEND = 1	single	Set DTENDC
SDIO_CMD_END	End of command sequence	CMDSEND = 1, or (CMDRECV = 1 and DAT0BSY = 0)	single	Set CMDSENDC Set CMDRECVC Set DAT0BSYENDC
SDIO_BUF_END	End of buffer reached	IDMAENDC = 1	link list	Set IDMAENDC Update IDMAADDR0/1

46.3.10. AHB and SDIO_CLK clock relation

The AHB bus clock (HCLK) must be greater than CK_SDIO.

46.3.11. Hardware flow control

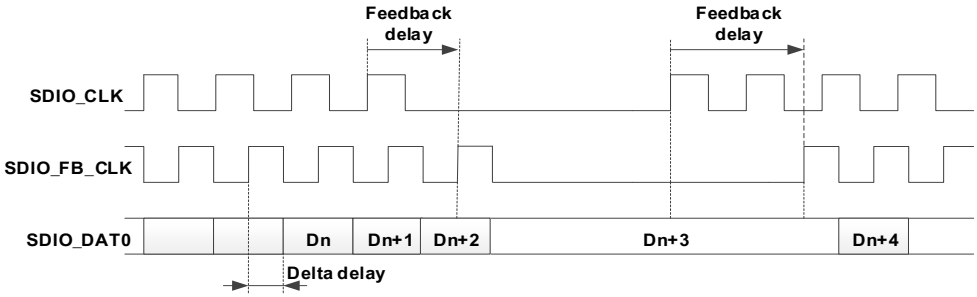
The hardware flow control function freezes the state machine by stopping the SDIO_CLK during data transmission, thereby preventing FIFO underrun (transmit data) and overrun (receive data) errors.

Stoppage of the FIFO causes a pause in data transmission until the transmit FIFO is half full or the DATALEN of the data is stored, or the receive FIFO is half empty. When the state machine freezes, the AHB interface may remain active. Therefore, the FIFO can operate even if hardware flow control is performed.

Hardware flow control is enabled by setting the HWEN register.

Hardware flow control must only be used when the SDIO_DAT data is cycle-aligned with the SDIO_CLK. SDR104 mode using CPDM cannot use hardware control.

Figure 46-8. Hardware flow timing



46.4. Card function overview

46.4.1. Card registers

Within the card interface registers are defined: OCR, CID, CSD, EXT_CSD, RCA, DSR and SCR. These can be accessed only by corresponding commands. The OCR, CID, CSD and SCR registers carry the card/content specific information, while the RCA and DSR registers are configuration registers storing actual configuration parameters. The EXT_CSD register carries both, card specific information and actual configuration parameters. For specific information, please refer to the relevant specifications.

OCR register: The 32-bit operation conditions register (OCR) stores the V_{DD} voltage profile of the card and the access mode indication (*e*•MMC). In addition, this register includes a status information bit. This status bit is set if the card power up procedure has been finished. The register is a little different between *e*•MMC and SD card. The host can use CMD1 (e•MMC), ACMD41 (SD memory), CMD5 (SD I/O) to get the content of this register.

CID register: The Card Identification (CID) register is 128 bits wide. It contains the card identification information used during the card identification phase. Every individual Read/Write (RW) card shall have a unique identification number. The host can use CMD2 and CMD10 to get the content of this register.

CSD register: The Card-Specific Data register provides information regarding access to the card contents. The CSD defines the data format, error correction type, maximum data access time, data transfer speed, whether the DSR register can be used, etc. The programmable part of the register can be changed by CMD27. The host can use CMD9 to get the content

of this register.

Extended CSD Register: Just *e*•MMC4.51 has this register. The Extended CSD register defines the card properties and selected modes. It is 512 bytes long. The most significant 320 bytes are the Properties segment, which defines the card capabilities and cannot be modified by the host. The lower 192 bytes are the Modes segment, which defines the configuration the card is working in. These modes can be changed by the host by means of the SWITCH command. The host can use CMD8 (just *e*•MMC supports this command) to get the content of this register.

RCA register: The writable 16-bit relative card address register carries the card address that is published by the card during the card identification. This address is used for the addressed host-card communication after the card identification procedure. The host can use CMD3 to ask the card to publish a new relative address (RCA).

Note: The default value of the RCA register is 0x0001(*e*•MMC) or 0x0000(SD/SD I/O). The default value is reserved to set all cards into the Stand-by State with CMD7.

DSR register (Optional): The 16-bit driver stage register can be optionally used to improve the bus performance for extended operating conditions (depending on parameters like bus length, transfer rate or number of cards). The CSD register carries the information about the DSR register usage. The default value of the DSR register is 0x404. The host can use CMD4 to get the content of this register.

SCR register: Just SD/SD I/O (if has memory port) have this register. In addition to the CSD register, there is another configuration register named SD CARD Configuration Register (SCR), which is only for SD card. SCR provides information on the SD Memory Card's special features that were configured into the given card. The size of SCR register is 64 bits. This register shall be set in the factory by the SD Memory Card manufacturer. The host can use ACMD51 to get the content of this register.

46.4.2. Commands

Commands types

There are four kinds of commands defined to control the Card:

- Broadcast commands (bc), no response
- Broadcast commands with response (bcr) response from all cards simultaneously
- Addressed (point-to-point) commands (ac) no data transfer on DAT
- Addressed (point-to-point) data transfer commands (adtc) data transfer on DAT

Command format

All commands have a fixed code length of 48 bits, as show in ***Figure 46-9. Command Token Format***, needing a transmission time of 1.92μs (25 MHz) 0.96μs (50 MHz) and 0.92us (52 MHz).

Figure 46-9. Command Token Format

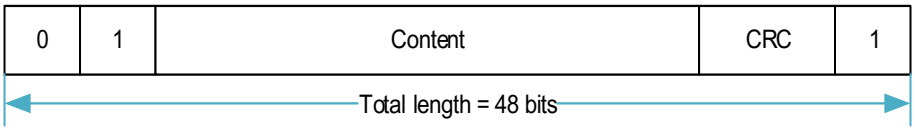


Table 46-8. Command format

Bit position	47	46	[45:40]	[39:8]	[7:1]	0
Width	1	1	6	32	7	1
Value	'0'	'1'	x	x	x	'1'
Description	start bit	transmission bit	command index	argument	CRC7	end bit

A command always starts with a start bit (always 0), followed by the bit indicating the direction of transmission (host = 1). The next 6 bits indicate the index of the command, this value being interpreted as a binary coded number (between 0 and 63). Some commands need an argument (e.g. an address), which is coded by 32 bits. A value denoted by 'x' in the table above indicates this variable is dependent on the command. All commands are protected by a CRC7. Every command code word is terminated by the end bit (always 1).

Command classes

The command set of the Card system is divided into several classes (See ***Table 46-9. Card command classes (CCCs)***. Each class supports a set of card functionalities. ***Table 46-9. Card command classes (CCCs)*** determines the setting of CCC from the card supported commands.

For SD cards, Class 0, 2, 4, 5 and 8 are mandatory and shall be supported. Class 7 except CMD40 is mandatory for SDHC. The other classes are optional. The supported Card Command Classes (CCC) are coded as a parameter in the card specific data (CSD) register of each card, providing the host with information on how to access the card.

For *e*•MMC cards, Class 0 is mandatory and shall be supported. The other classes are either mandatory only for specific card types or optional. By using different classes, several configurations can be chosen (e.g. a block writable card or a stream readable card). The supported Card Command Classes (CCC) are coded as a parameter in the card specific data (CSD) register of each card, providing the host with information on how to access the card.

Table 46-9. Card command classes (CCCs)

	Card command class(CCC)	0	1	2	3	4	5	6	7	8	9	10	11

Supported command	Class description	basic	Stream read	Block read	Stream write	Block write	erase	write protection	Lock card	application specific	I/O mode	Security Protocols	reserved
CMD0	M	+											
CMD1	M	+											
CMD2	M	+											
CMD3	M	+											
CMD4	M	+											
CMD5	M	+									+		
CMD6	M	+											
CMD7	M	+											
CMD8	M	+											
CMD9	M	+											
CMD10	M	+											
CMD11	O	+	+										
CMD12	M	+											
CMD13	M	+											
CMD14	M	+											
CMD15	M	+											
CMD16	M			+		+			+				
CMD17	M			+									
CMD18	M			+									
CMD19	M	+											
CMD20	M				+								
CMD21	M			+									
CMD23	M			+		+							
CMD24	M					+							
CMD25	M					+							
CMD26	M					+							
CMD27	M					+							
CMD28	M							+					
CMD29	M							+					
CMD30	M							+					
CMD32	M						+						
CMD33	M						+						
CMD34	M												+
CMD35	O						+						
CMD36	O						+						
CMD37	O												+

CMD38	M						+						
CMD39											+		
CMD40											+		
CMD42									+				
CMD49						+							
CMD52	O										+		
CMD53	O										+	+	
CMD54	O											+	
CMD55	M									+			
CMD56	M									+			
CMD57	O												+
ACMD6	M									+			
ACMD13	M									+			
ACMD22	M									+			
ACMD23	M									+			
ACMD41	M									+			
ACMD42	M									+			
ACMD51	M									+			

Note: 1. M: mandatory, O: optional.

2. CMD5 is Class 0 command for SD card and Class 9 command in *e*•MMC.

3. In SD cards that support UHS-I, CMD11 is a type 0 command and mandatory, and it is optional in other SD cards; in e•MMC, CMD11 is a type 1 command.

4. CMD14, CMD20, CMD21, CMD26, CMD39, CMD40, CMD49 and CMD54 are only available for *e*•MMC. CMD32, CMD33, CMD52, CMD57 and ACMDx are only available for SD card.

5. All the ACMDx shall be preceded with APP_CMD command (CMD55).

6. CMD5, CMD8 , CMD11 and CMD53 has different meaning for *e*•MMC and SD memory.

7. Class 1 and Class 3 are obsolete.

Detailed command description

The following tables describe in detail all bus commands. The responses R1-R7 are defined in ***Responses.*** The registers CID, CSD and DSR are described in ***Card registers***. The card shall ignore stuff bits and reserved bits in an argument.

Table 46-10. Basic commands (class 0)

Cmd index	type	argument	Response format	Abbreviation	Description
CMD0	bc	[31:0] 00000000	-	GO_IDLE_STATE	Resets all cards to idle state
	bc	[31:0] F0F0F0F0	-	GO_PRE_IDLE_STATE	Resets all cards to pre-idle state

Cmd index	type	argument	Response format	Abbreviation	Description
	-	[31:0] FFFFFFFA	-	BOOT_INITIATION	Initiate alternative boot operation
CMD1	bc	[31:0] OCR without busy	R3	SEND_OP_COND	Asks the card, in idle state, to send its Operating Conditions Register contents in the response on the CMD line.
CMD2	bcr	[31:0] stuff bits	R2	ALL_SEND_CID	Asks any card to send the CID numbers on the CMD line (any card that is connected to the host will respond)
CMD3	bcr	[31:0] stuff bits	R6	SEND_RELATIVE_ADDR	Ask the card to publish a new relative address (RCA)
CMD4	bc	[31:16] DSR [15:0] stuff bits	-	SET_DSR	Programs the DSR of all cards
CMD5	bcr	[31:25]reserved bits [24]S18R [23:0] I/O OCR	R4	IO_SEND_OP_COND	Only for I/O cards. It is similar to the operation of ACMD41 for SD memory cards, used to inquire about the voltage range needed by the I/O card.
CMD5	ac	[31:16] RCA [15] Sleep/Awake [14:0] stuff bits	R1b	SLEEP_AWAKE	Only for *e*•MMC. Toggles the Device between Sleep state and Standby state.
CMD6	adtc	[31] Mode 0:Check function 1:Switch function [30:24] reserved [23:20] reserved for function group 6 (0h or Fh) [19:16] reserved for function group 5 (0h or Fh) [15:12] reserved for function group 4 (0h or Fh) [11:8] reserved for function group 3 (0h or Fh) [7:4] function group 2 for	R1	SWITCH_FUNC	Only for SD memory and SD I/O. Checks switchable function (mode 0) and switch card function (mode 1).

Cmd index	type	argument	Response format	Abbreviation	Description
		command system [3:0] function group 1 for access mode			
CMD6	ac	[31:26] Set to 0 [25:24] Access [23:16] Index [15:8] Value [7:3] Set to 0 [2:0] Cmd Set	R1b	SWITCH	Only for e•MMC. Switches the mode of operation of the selected card or modifies the EXT_CSD registers.
CMD7	ac	[31:16] RCA [15:0] stuff bits	R1b	SELECT/DESELECT_CARD	Command toggles a card between the stand-by and transfer states or between the programming and disconnects states. In both cases the card is selected by its own relative address and gets deselected by any other address; address 0 deselects the card.
CMD8	bcr	[31:12]reserved bits [11:8]supply voltage(VHS) [7:0]check pattern	R7	SEND_IF_COND	Sends SD Memory Card interface condition, which includes host supply voltage information and asks the card whether card supports voltage. Reserved bits shall be set to '0'.
CMD8	adtc	[31:0] stuff bits	R1	SEND_EXT_CSD	For e•MMC only. The card sends its EXT_CSD register as a block of data.
CMD9	ac	[31:16] RCA [15:0] stuff bits	R2	SEND_CSD	Addressed card sends its card-specific data (CSD) on the CMD line.
CMD10	ac	[31:16] RCA [15:0] stuff bits	R2	SEND_CID	Addressed card sends its card identification (CID) on CMD the line.
CMD11	ac	[31:0] 00000000	R1	VOLTAGE_SWITCH	Switch to 1.8V bus signaling level.
CMD12	ac	[31:0] stuff bits	R1b	STOP TRANSMISSION	Forces the card to stop transmission
CMD13	ac	[31:16] RCA [15:0] stuff bits	R1	SEND_STATUS	Addressed card sends its status register.
CMD14	adtc	[31:0] stuff bits	R1	BUSTEST_R	A host reads the reversed bus testing data pattern from a card.

Cmd index	type	argument	Response format	Abbreviation	Description
CMD15	ac	[31:16] RCA [15:0] reserved bits	-	GO_INACTIVE_STATE	Sends an addressed card into the Inactive State. This command is used when the host explicitly wants to deactivate a card.
CMD19	adtc	[31:0] stuff bits	R1	BUSTEST_W	A host sends the bus test data pattern to a card.

Table 46-11. Block-Oriented read commands (class 2)

Cmd index	type	argument	Response format	Abbreviation	Description
CMD16	ac	[31:0] block length	R1	SET_BLOCKLEN	In the case of a Standard Capacity SD and e•MMC, this command sets the block length (in bytes) for all following block commands (read, write, lock). Default is 512 Bytes. Set length is valid for memory access commands only if partial block read operation are allowed in CSD. In the case of a High Capacity SD Memory Card, block length set by CMD16 command does not affect the memory read and write commands. Always 512 Bytes fixed block length is used. In both cases, if block length is set larger than 512Bytes, the card sets the BLOCK_LEN_ERROR bit.
CMD17	adtc	[31:0] data address	R1	READ_SINGLE_BLOCK	In the case of a Standard Capacity SD and e•MMC, this command reads a block of the size selected by the SET_BLOCKLEN command. In the case of a High Capacity Card, block length is fixed 512 Bytes regardless of the SET_BLOCKLEN command.
CMD18	adtc	[31:0] data address	R1	READ_MULTIPLE_BLOCK	Continuously transfers data blocks from card to host until interrupted by a

Cmd index	type	argument	Response format	Abbreviation	Description
					STOP_TRANSMISSION command. Block length is specified the same as READ_SINGLE_BLOCK command.
CMD21	adtc	[31:0] stuff bits	R1	SEND_TUNNING_BLOCK	128 clocks of tuning pattern (64byte in 4 bit mode or 128byte in 8 bit mode) is sent for HS200 optimal sampling point detection.
Note: The transferred data must not cross a physical block boundary, unless READ_BLK_MISALIGN is set in the CSD register					

Table 46-12. Stream read commands (class 1) and stream write commands (class 3)

Cmd index	type	argument	Response format	Abbreviation	Description
CMD11	adtc	[31:0] data address	R1	READ_DAT_UNTIL_STOP	Reads data stream from the card, starting at the given address, until a STOP_TRANSMISSION follows.
CMD20	adtc	[31:0] data address	R1	WRITE_DAT_UNTIL_STOP	Writes data stream from the host, starting at the given address, until a STOP_TRANSMISSION follows.
Note: The transferred data must not cross a physical block boundary, unless READ_BLK_MISALIGN is set in the CSD register					

Table 46-13. Block-Oriented write commands (class 4)

Cmd index	type	argument	Response format	Abbreviation	Description
CMD16	ac	[31:0] block length	R1	SET_BLOCKLEN	See description in ***Table 46-11. Block-Oriented read commands (class 2)***.
CMD23	ac	[31:16] set to 0 [15:0] number of blocks	R1	SET_BLOCK_COUNT	Defines the number of blocks which are going to be transferred in the immediately succeeding multiple block read or write command. If the argument is all 0s, the subsequent read/write operation will be open-ended.

Cmd index	type	argument	Response format	Abbreviation	Description
CMD24	adtc	[31:0] data address	R1	WRITE_BLOCK	In the case of a Standard Capacity SD, this command writes a block of the size selected by the SET_BLOCKLEN command. In the case of a SDHC, block length is fixed 512 Bytes regardless of the SET_BLOCKLEN command.
CMD25	adtc	[31:0] data address	R1	WRITE_MULTIPLE _BLOCK	Continuously writes blocks of data until a STOP_TRANSMISSION follows. Block length is specified the same as WRITE_BLOCK command.
CMD26	adtc	[31:0] stuff bits	R1	PROGRAM_CID	Programming of the card identification register. This command shall be issued only once. The card contains hardware to prevent this operation after the first programming. Normally this command is reserved for the manufacturer.
CMD27	adtc	[31:0] stuff bits	R1	PROGRAM_CSD	Programming of the programmable bits of the CSD.
CMD49	adtc	[31:0] stuff bits	R1	SET_TIME	Sets the real time clock according to the RTC information in the 512B data block.
Note: 1. The data transferred shall not cross a physical block boundary unless WRITE_BLK_MISALIGN is set in the CSD. In the case that write partial blocks is not supported, then the block length=default block length (given in CSD). 2. Data address is in byte units in a Standard Capacity SD Memory Card and in block (512 Byte) units in a High Capacity SD Memory Card.					

Table 46-14. Erase commands (class 5)

Cmd index	type	argument	Response format	Abbreviation	Description
CMD32	ac	[31:0] data address	R1	ERASE_WR_BLK _START	Sets the address of the first write block to be erased.(SD)
CMD33	ac	[31:0] data address	R1	ERASE_WR_BLK _END	Sets the address of the last write block of the continuous range to be erased.(SD)

CMD35	ac	[31:0]data address	R1	ERASE_GROUP_START	Sets the address of the first erase group within a range to be selected for erase.(e•MMC)
CMD36	ac	[31:0]data address	R1	ERASE_GROUP_END	Sets the address of the last erase group within a continuous range to be selected for erase. (e•MMC)
CMD38	ac	[31:0] stuff bits	R1b	ERASE	Erases all previously selected write blocks.
Note: 1.CMD34 and CMD37 are reserved in order to maintain backwards compatibility with older versions of the e•MMC. 2. Data address is in byte units in a Standard Capacity SD Memory Card and in block (512 Byte) units in a High Capacity SD Memory Card.					

Table 46-15. Block oriented write protection commands (class 6)

Cmd index	type	argument	Response format	Abbreviation	Description
CMD28	ac	[31:0] data address	R1b	SET_WRITE_PROT	If the card has write protection features, this command sets the write protection bit of the addressed group. The properties of write protection are coded in the card specific data (WP_GRP_SIZE). A High Capacity SD Memory Card does not support this command.
CMD29	ac	[31:0] data address	R1b	CLR_WRITE_PROT	If the card provides write protection features, this command clears the write protection bit of the addressed group.
CMD30	adtc	[31:0] write protect data address	R1	SEND_WRITE_PROT	If the card provides write protection features, this command asks the card to send the status of the write protection bits.
Note: 1. High Capacity SD Memory Card does not support these three commands.					

Table 46-16. Lock card (class 7)

Cmd index	type	argument	Response format	Abbreviation	Description
CMD16	ac	[31:0] block length	R1	SET_BLOCK_LEN	See description in *Table 46-11. Block-Oriented read*

					commands (class 2).
CMD42	adtc	[31:0] Reserved bits (Set all 0)	R1	LOCK_UNLOCK	Used to set/reset the password or lock/unlock the card. The size of the data block is set by the SET_BLOCK_LEN command. Reserved bits in the argument and in Lock Card Data Structure shall be set to 0.

Table 46-17. Application-specific commands (class 8)

Cmd index	type	argument	Response format	Abbreviation	Description
ACMD41	bcr	[31]reserved bit [30]HCS [29:24]reserved bits [23:0]V_{DD} Voltage Window(OCR[23:0])	R3	SD_SEND_OP_COND	Sends host capacity support information (HCS) and asks the accessed card to send its operating condition register(OCR) content in the response. HCS is effective when card receives SEND_IF_COND command. CCS bit is assigned to OCR[30].
ACMD42	ac	[31:1] stuff bits [0]set_cd	R1	SET_CLR_CARD_DETECT	Connect[1]/Disconnect[0] the 50K pull-up resistor on CD/DAT3 (pin 1) of the card.
ACMD51	adtc	[31:0] stuff bits	R1	SEND_SCR	Reads the SD Configuration Register (SCR).
CMD55	ac	[31:16] RCA [15:0] stuff bits	R1	APP_CMD	Indicates to the card that the next command is an application specific command rather than a standard command.
CMD56	adtc	[31:1] stuff bits. [0] RD/WR	R1	GEN_CMD	Used either to transfer a data block to the card or to get a data block from the card for general purpose/application specific command. The host sets RD/WR=1 for reading data from the card and sets to 0 for writing data to the card.
Note: 1. ACMDx is Application-specific Commands for SD memory.					

Table 46-18. I/O mode commands (class 9)

Cmd index	type	argument	Response format	Abbreviation	Description
CMD39	ac	[31:16] RCA [15] register write flag [14:8] register address [7:0] register data	R4	FAST_IO	Used to write and read 8 bit (register) data fields. The command addresses a card and a register and provides the data for writing if the write flag is set. The R4 response contains data read from the addressed register if the write flag is cleared to 0. This command accesses application dependent registers which are not defined in the *e*•MMC standard.
CMD40	bcr	[31:0] stuff bits	R5	GO_IRQ_STATE	Sets the system into interrupt mode
CMD52	adtc	[31] R/W Flag [30:28] Function Number [27] RAW Flag [26] stuff bits [25:9] Register Address [8] stuff bits [7:0] Write Data/Stuff bits	R5	IO_RW_DIRECT	The IO_RW_DIRECT is the simplest means to access a single register within the total 128K of register space in any I/O function, including the common I/O area (CIA). This command reads or writes 1 byte using only 1 command/response pair. A common use is to initialize registers or monitor status values for I/O functions. This command is the fastest means to read or write single I/O registers, as it requires only a single command/response pair.
CMD53	adtc	[31] R/W Flag [30:28] Function Number [27] Block Mode [26] OP code [25:9] Register Address [8:0] Byte/Block Count		IO_RW_EXTENDED	This command allows the reading or writing of a large number of I/O registers with a single command.

Cmd index	type	argument	Response format	Abbreviation	Description
Note: 1.CMD39, CMD40 are only for e•MMC. 2. CMD52, CMD53 are only for SD I/O card.					

Table 46-19. Switch function commands (class 10)

Cmd index	type	argument	Response format	Abbreviation	Description
CMD53	adtc	[31:16] Security Protocol Specific [15:8] Security Protocol [7:0] reserved	R1	PROTOCOL_RD	Only for SD memory and SD I/O. Continuously transfers data blocks from device to host. Number of data blocks shall be defined by a preceding CMD23. Data Transfer may be interrupted by a STOP_TRANSMISSION command, This command is not supported if sent as a packed command. Block size is always 512bytes.
CMD54	adtc	[31:16] Security Protocol Specific [15:8] Security Protocol [7:0] reserved	R1	PROTOCOL_WR	Only for SD memory and SD I/O. Continuously transfers data blocks from host to device. Number of data blocks shall be defined by a preceding CMD23. Data Transfer may be interrupted by a STOP_TRANSMISSION command, This command is not supported if sent as a packed command. Block size is always 512bytes.

46.4.3. Responses

All responses are sent on the CMD line. The response transmission always starts with the left

bit of the bit string corresponding to the response code word. The code length depends on the response type.

Responses types

There are 7 types of responses show as follows.

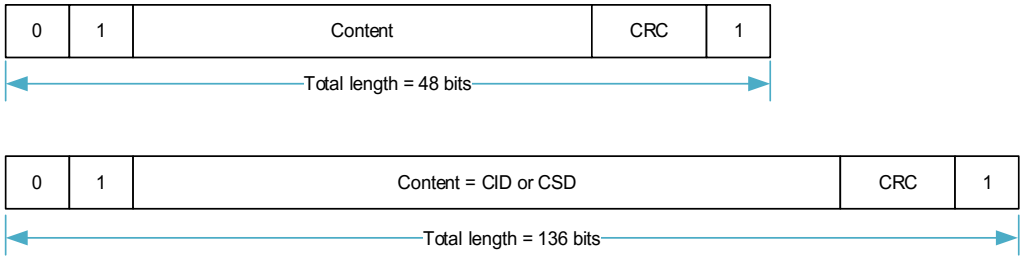
- **R1 / R1b :** normal response command.
- **R2 :** CID, CSD register.
- **R3 :** OCR register.
- **R4 :** Fast I/O.
- **R5 :** Interrupt request.
- **R6 :** Published RCA response.
- **R7 :** Card interface condition.

The SD Memory Card support five types of them, R1 / R1b, R2, R3, R6, R7. And the SD I/O Card and *e*•MMC supports additional response types named R4 and R5, but they are not exactly the same for SD I/O Card and *e*•MMC.

Responses format

Responses have two formats, as show in ***Figure 46-10. Response Token Format***, all responses are sent on the CMD line. The code length depends on the response type. Except R2 is 136 bits length, others are all 48 bits length.

Figure 46-10. Response Token Format



A response always starts with a start bit (always 0), followed by the bit indicating the direction of transmission (card = 0). A value 'x' in the tables below indicates a variable entry. All responses except for the type R3 are protected by a CRC. Every command code word is terminated by the end bit (always 1).

R1 (normal response command)

Code length is 48 bits. The bits 45:40 indicate the index of the command to be responded to, this value being interpreted as a binary coded number (between 0 and 63). The status of the card is coded in 32 bits. Note that if a data transfer to the card is involved, then a busy signal may appear on the data line after the transmission of each block of data. The host shall check for busy after data block transmission. The card status is described in ***Two status fields of the card***.

Table 46-20. Response R1

Bit position	47	46	[45:40]	[39:8]	[7:1]	0
Width	1	1	6	32	7	1
Value	'0'	'0'	x	x	x	'1'
description	start bit	transmission bit	command index	card status	CRC7	end bit

R1b

R1b is identical to R1 with an optional busy signal transmitted on the data line DAT0. The card may become busy after receiving these commands based on its state prior to the command reception. The Host shall check for busy at the response.

R2 (CID, CSD register)

Code length is 136 bits. The contents of the CID register are sent as a response to the commands CMD2 and CMD10. The contents of the CSD register are sent as a response to CMD9. Only the bits [127..1] of the CID and CSD are transferred, the reserved bit [0] of these registers is replaced by the end bit of the response.

Table 46-21. Response R2

Bit position	135	134	[133:128]	[127:1]	0
Width	1	1	6	127	1
Value	'0'	'0'	'111111'	x	'1'
description	start bit	transmission bit	reserved	CID or CSD register and internal CRC7	end bit

R3 (OCR register)

Code length is 48 bits. The contents of the OCR register are sent as a response to ACMD41 (SD memory), CMD1 (*e*•MMC). The response of different cards may have a little different.

Table 46-22. Response R3

Bit position	47	46	[45:40]	[39:8]	[7:1]	0
Width	1	1	6	32	7	1
Value	'0'	'0'	'111111'	x	'1111111'	'1'
description	start bit	transmission bit	reserved	OCR register	reserved	end bit

R4 (Fast I/O)

For *e*•MMC only. Code length 48 is bits. The argument field contains the RCA of the addressed card, the register address to be read out or written to, and its contents. The status bit in the argument is set if the operation was successful.

Table 46-23. Response R4 for e•MMC

Bit position	47	46	[45:40]	[39:8] Argument field				[7:1]	0
Width	1	1	6	16	1	7	8	7	1
Value	'0'	'0'	'100111'	x	x	x	x	x	'1'
description	start bit	transmission bit	CMD39	RCA [31:16]	status [15]	register address [14:8]	read register contents [7:0]	CRC7	end bit

R4b

For SD I/O only. Code length is 48 bits. The SDIO card receive the CMD5 will respond with a unique SD I/O response R4.

Table 46-24. Response R4 for SD I/O

Bit position	47	46	[45:40]	39	[38:36]	35	[34:32]	31	[30:8]	[7:1]	0
Width	1	1	6	1	3	1	3	1	23	7	1
Value	'0'	'0'	'111111'	x	x	x	'000'	x	x	'1111111'	1
description	start bit	transmission bit	Reserved	C	Number of I/O functions	Memory Present	Stuff Bits	S18A	I/O OCR	Reserved	end bit

R5 (Interrupt request)

For *e*•MMC only. Code length is 48 bits. If the response is generated by the host, the RCA field in the argument will be 0x0.

Table 46-25. Response R5 for *e*•MMC

Bit position	47	46	[45:40]	[39:8] Argument field		[7:1]	0
Width	1	1	6	16	16	7	1
Value	'0'	'0'	'101000'	x	x	x	'1'
description	start bit	transmission bit	CMD40	RCA [31:16] of winning card or of the host	[15:0] Not defined. May be used for IRQ data	CRC7	end bit

R5b

For SD I/O only. The SDIO card's response to CMD52 and CMD53 is R5. If the communication between the card and host is in the 1-bit or 4-bit SD mode, the response shall be in a 48-bit response (R5).

Table 46-26. Response R5 for SD I/O

Bit position	47	46	[45:40]	[39:24]	[23:16]	[15:8]	[7:1]	0
Width	1	1	6	16	8	8	7	1

Value	'0'	'0'	'11010x'	'0'	x	x	x	'1'
description	start bit	transmission bit	CMD52/53	Stuff Bits	Response Flags	Read or Write Data	CRC7	end bit

R6 (Published RCA response)

Code length is 48 bit. The bits [45:40] indicate the index of the command to be responded to (CMD3). The 16 MSB bits of the argument field are used for the Published RCA number.

Table 46-27. Response R6

Bit position	47	46	[45:40]	[39:8] Argument field		[7:1]	0
Width	1	1	6	16	16	7	1
Value	'0'	'0'	'000011'	x	x	x	'1'
description	start bit	transmission bit	CMD3	New published RCA of the card	card status bits:23,22,19,12:0	CRC7	end bit

R7 (Card interface condition)

For SD memory only. Code length is 48 bits. The card support voltage information is sent by the response of CMD8. Bits 19-16 indicate the voltage range that the card supports. The card that accepted the supplied voltage returns R7 response. In the response, the card echoes back both the voltage range and check pattern set in the argument.

Table 46-28. Response R7

Bit position	47	46	[45:40]	[39:20]	[19:16]	[15:8]	[7:1]	0
Width	1	1	6	20	4	8	7	1
Value	'0'	'0'	'001000'	'00000h'	x	x	x	'1'
description	start bit	transmission bit	CMD8	Reserved bits	Voltage accepted	echo-back of check pattern	CRC7	end bit

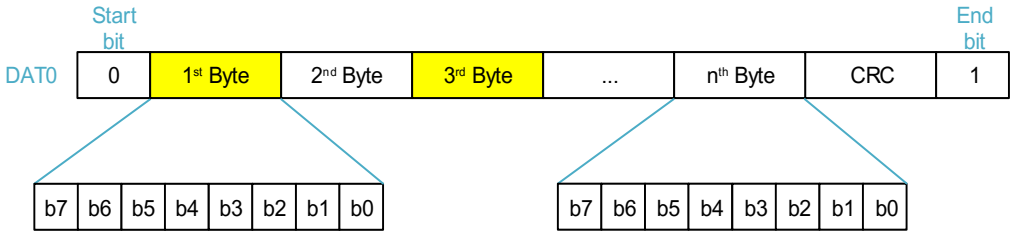
46.4.4. Data packets format

There are 5 data bus mode, 1-bit, 4-bit SDR/DDR, 8-bit SDR/DDR. 1-bit mode is mandatory, 4-bit and 8-bit mode is optional.

10.1.11 bit data packet format

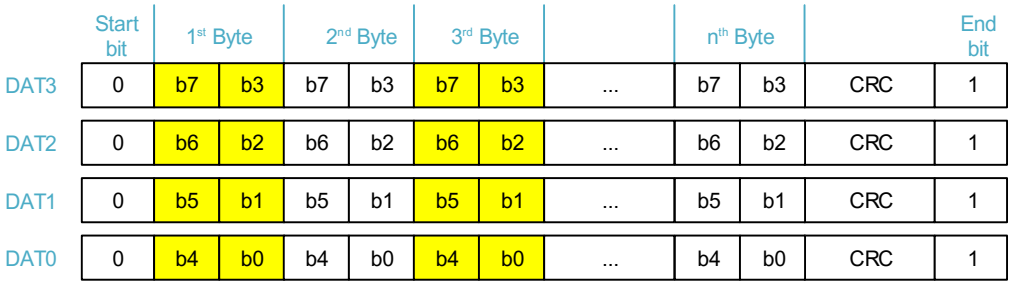
After card reset and initialize, only DAT0 pin is used to transfer data. And other pin can be used freely. *Figure 46-11. 1-bit data bus width*, *Figure 46-12. 4-bit data bus width* and *Figure 46-13. 8-bit data bus width* show the data packet format when data bus wide is 1-bit, 4-bit and 8-bit.

Figure 46-11. 1-bit data bus width



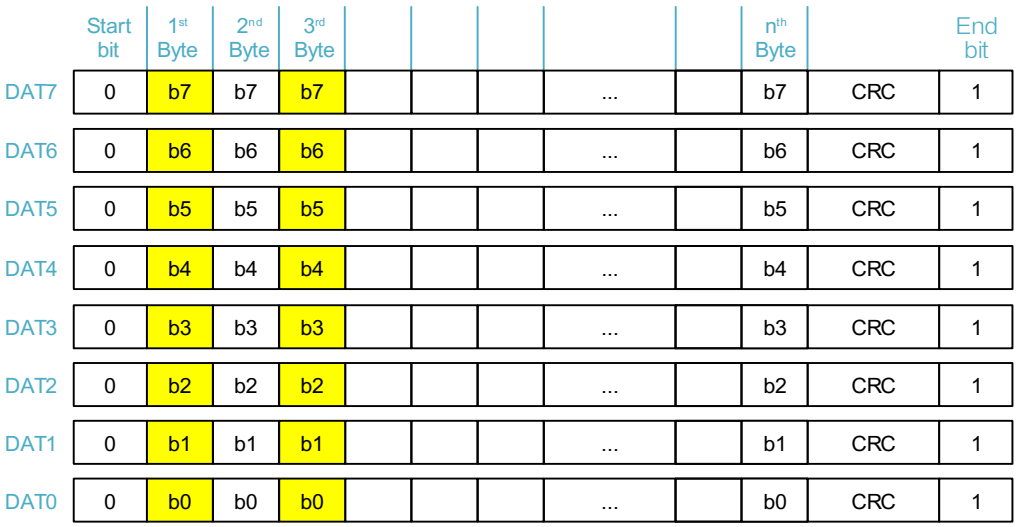
4-bit data packet format

Figure 46-12. 4-bit data bus width



8-bit data packet format

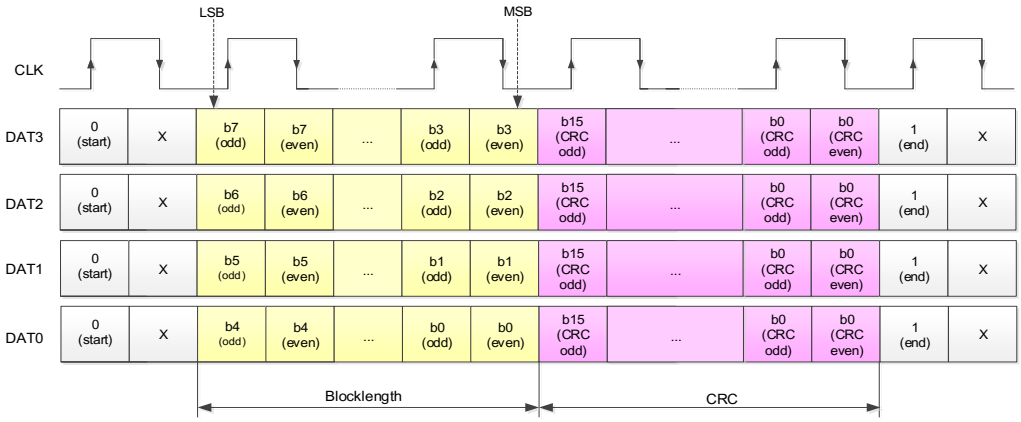
Figure 46-13. 8-bit data bus width



4-bit bus data packet format for DDR (DAT3-DAT0 used)

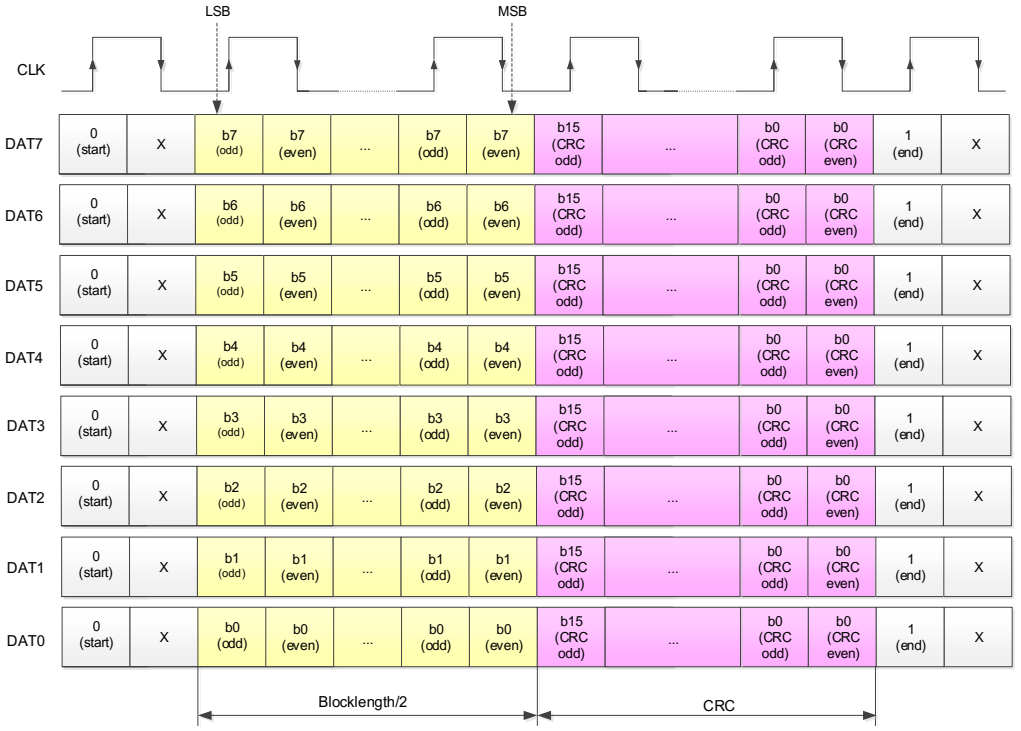
For each data lines, data can be transferred at the rate of one bit (single data rate) or two bits (double data rate) per clock cycle. DDR (double data rate) signaling, data is sampled on both SDIO_CLK clock edges. ***Figure 46-14. 4-bit data bus DDR*** and ***Figure 46-15. 8-bit data bus DDR*** show the data packet format when data bus wide is 4-bit DDR and 8-bit DDR.

Figure 46-14. 4-bit data bus DDR



8-bit bus data packet format for DDR (DAT7-DAT0 used)

Figure 46-15. 8-bit data bus DDR



Note: For DDR data bus: 1.That bytes data are not interleaved but CRC are interleaved. 2. Start bits are valid on the rising and falling edge. 3. End bits are only valid on the rising edge ("x" undefined).

46.4.5. Two status fields of the card

The SD Memory supports two status fields and others just support the first one:

Card Status: Error and state information of a executed command, indicated in the response

SD Status: Extended status field of 512 bits that supports special features of the SD Memory Card and future Application-Specific features.

Card status

The response format R1 contains a 32-bit field named card status. This field is intended to transmit the card's status information (which may be stored in a local status register) to the host. If not specified otherwise, the status entries are always related to the previous issued command.

The type and clear condition fields in the table are abbreviated as follows:

Type

•E: Error bit. Send an error condition to the host. These bits are cleared as soon as the response (reporting the error) is sent out.

•S: Status bit. These bits serve as information fields only, and do not alter the execution of the command being responded to. These bits are persistent, they are set and cleared in accordance with the card status.

•R: Exceptions are detected by the card during the command interpretation and validation phase (Response Mode).

•X: Exceptions are detected by the card during command execution phase (Execution Mode).

Clear condition

•A: According to current state of the card.

•B: Always related to the previous command. Reception of a valid command will clear it (with a delay of one command).

•C: Cleared by read

Table 46-29. Card status

Bits	Identifier	Type	Value	Description	Clear Condition
31	OUT_OF_RANGE	ERX	'0'= no error '1'= error	The command's argument was out of the allowed range for this card.	C
30	ADDRESS_ERROR	ERX	'0'= no error '1'= error	A misaligned address which did not match the block length was used in the command.	C
29	BLOCK_LEN_ERROR	ERX	'0'= no error '1'= error	The transferred block length is not allowed for this card, or the number of transferred bytes does not match the block length.	C

Bits	Identifier	Type	Value	Description	Clear Condition
28	ERASE_SEQ_ERROR	ER	'0'= no error '1'= error	An error in the sequence of erase commands occurred.	C
27	ERASE_PARAM	ERX	'0'= no error '1'= error	An invalid selection of write-blocks for erase occurred.	C
26	WP_VIOLATION	ERX	'0'= not protected '1'= protected	Set when the host attempts to write to a protected block or to the temporary or permanent write protected card.	C
25	CARD_IS_LOCKED	SX	'0' = card unlocked '1' = card locked	When set, signals that the card is locked by the host	A
24	LOCK_UNLOCK_FAILED	ERX	'0'= no error '1'= error	Set when a sequence or password error has been detected in lock/unlock card command.	C
23	COM_CRC_ERROR	ER	'0'= no error '1'= error	The CRC check of the previous command failed.	B
22	ILLEGAL_COMMAND	ER	'0'= no error '1'= error	Command not legal for the card state.	B
21	CARD_ECC_FAILED	ERX	'0'= success '1'= failure	Card internal ECC was applied but failed to correct the data.	C
20	CC_ERROR	ERX	'0'= no error '1'= error	Internal card controller error.	C
19	ERROR	ERX	'0'= no error '1'= error	A general or an unknown error occurred during the operation.	C
18	UNDERRUN	ERX	'0'= no error '1'= error	Only for *e*•MMC. The card could not sustain data transfer in stream read mode.	C
17	OVERRUN	ERX	'0'= no error '1'= error	Only for *e*•MMC. The card could not sustain data programming in stream write mode.	C
16	CID/ CSD_OVERWRITE	ERX	'0'= no error '1'= error	Can be either one of the following errors: - The read only section of the CSD does not match the card content. - An attempt to reverse the copy (set as original) or permanent WP(unprotected) bits was made.	C
15	WP_ERASE_SKIP	ERX	'0'= not protected	Set when only partial address	C

Bits	Identifier	Type	Value	Description	Clear Condition
			'1'= protected	space was erased due to existing write protected blocks or the temporary or permanent write protected card was erased.	
14	CARD_ECC_DISABLED	SX	'0'= enabled '1'= disabled	The command has been executed without using the internal ECC.	A
13	ERASE_RESET	SR	'0'= cleared '1'= set	An erase sequence was cleared before executing because an out of erase sequence command was received.	C
[12:9]	CURRENT_STATE	SX	0 = idle 1 = ready 2 = identification 3 = stand by 4 = transfer 5 = send data 6 = receive data 7 = programming 8 = disconnect 9 = Bus-Test 10 = sleep 11-14 = reserved 15 = reserved for I/O mode	The state of the card when receiving the command. If the command execution causes a state change, it will be visible to the host in the response to the next command. The four bits are interpreted as a binary coded number between 0 and 15. Sleep state for *e*•MMC only.	B
8	READY_FOR_DATA	SX	'0'= not ready '1'= ready	Corresponds to buffer empty signaling on the bus.	A
7	SWITCH_ERROR	EX	'0'= no error '1'= switch error	If set, the card don't switch to the expected mode as requested by the SWITCH command.	B
6	Reserved				
5	APP_CMD	SR	'0'= enabled '1'= disabled	The card will expect ACMD, or an indication that the command has been interpreted as ACMD.	C
4	Reserved				
3	AKE_SEQ_ERROR	ER	'0'= no error '1'= error	Only for SD memory. Error in the sequence of the authentication process.	C

Bits	Identifier	Type	Value	Description	Clear Condition
2	Reserved for application specific commands.				
[1:0]	Reserved for manufacturer test mode.				

Note: 18, 17, 7 bits are only for *e*•MMC. 14, 3 bits are only for SD memory.

SD status register

The SD Status contains status bits that are related to the SD Memory Card proprietary features and may be used for future application-specific usage. The size of the SD Status is one data block of 512 bits. The content of this register is transmitted to the Host over the DAT bus along with a 16-bit CRC. The SD Status is sent to the host over the DAT bus as a response to ACMD13 (CMD55 followed with CMD13). ACMD13 can be sent to a card only in 'transfer state' (card is selected). The SD Status structure is described below.

The same abbreviation for 'type' and 'clear condition' were used as for the Card Status above.

Table 46-30. SD status

Bits	Identifier	Type	Value	Description	Clear Condition
[511:510]	DAT_BUS_WIDTH	SR	'00'= 1 (default) '01'= reserved '10'= 4 bit width '11'= reserved	Shows the currently defined data bus width that was defined by SET_BUS_WIDTH command	A
509	SECURED_MODE	SR	'0'= Not in the mode '1'= In Secured Mode	Card is in Secured Mode of operation (refer to the "SD Security Specification").	A
[508:496]	reserved				
[495:480]	SD_CARD_TYPE	SR	The following cards are currently defined: '0000'= Regular SD RD/WR Card. '0001'= SD ROM Card '0002'= OTP	In the future, the 8 LSBs will be used to define different variations of an SD Memory Card (Each bit will define different SD Types). The 8 MSBs will be used to define SD Cards that do not comply with current SD Physical Layer Specification.	A
[479:448]	SIZE_OF_PROTECTED_AREA	SR	Size of protected area	(See below)	A
[447:440]	SPEED_CLASS	SR	Speed class of the card	(See below)	A

Bits	Identifier	Type	Value	Description	Clear Condition
[439:432]	PERFORMANCE_MOVE	SR	Performance of move indicated by 1 [MB/s] step.	(See below)	A
[431:428]	AU_SIZE	SR	Size of AU	(See below)	A
[427:424]	reserved				
[423:408]	ERASE_SIZE	SR	Number of Aus to be erased at a time	(See below)	A
[407:402]	ERASE_TIMEOUT	SR	Timeout value for erasing areas specified by UNIT_OF_ERASE_AU	(See below)	A
[401:400]	ERASE_OFFSET	SR	Fixed offset value added to erase time.	(See below)	A
[399:312]	reserved				
[311:0]	reserved for manufacturer				

SIZE_OF_PROTECTED_AREA

Setting this field differs between SDSC and SDHC/SDXC.

In case of SDSC Card, the capacity of protected area is calculated as follows:

Protected Area = SIZE_OF_PROTECTED_AREA* MULT * BLOCK_LEN.

SIZE_OF_PROTECTED_AREA is specified by the unit in MULT*BLOCK_LEN.

In case of SDHC and SDXC Cards, the capacity of protected area is calculated as follows:

Protected Area = SIZE_OF_PROTECTED_AREA

SIZE_OF_PROTECTED_AREA is specified by the unit in byte.

SPEED_CLASS

This 8-bit field indicates the Speed Class.

00h: Class 0

01h: Class 2

02h: Class 4

03h: Class 6

04h: Class 10

05h-FFh: Reserved

PERFORMANCE_MOVE

This 8-bit field indicates Pm and the value can be set by 1 [MB/sec] step. If the card does not move using Rus, Pm should be considered as infinity. Setting to FFh means infinity. The minimum value of Pm is defined in *Table 46-31. Performance move field*.

Table 46-31. Performance move field

PERFORMANCE_MOVE	Value Definition
00h	Sequential Write
01h	1 [MB/sec]
02h	2 [MB/sec]
.....	.....
Feh	254 [MB/sec]
FFh	Infinity

AU_SIZE

This 4-bit field indicates AU Size and the value can be selected from 16 KB.

Table 46-32. AU_SIZE field

AU_SIZE	Value Definition
0h	Not Defined
1h	16 KB
2h	32 KB
3h	64 KB
4h	128 KB
5h	256 KB
6h	512 KB
7h	1 MB
8h	2 MB
9h	4 MB
Ah	8 MB
Bh	12 MB
Ch	16 MB
Dh	24 MB
Eh	32 MB
Fh	64 MB

The maximum AU size, depends on the card capacity, is defined in *Table 46-32. AU_SIZE field*. The card can set any AU size specified in *Table 46-33. Maximum AU size* that is less

than or equal to the maximum AU size. The card should set smaller AU size as possible.

Table 46-33. Maximum AU size

Card Capacity	up to 64MB	up to 256MB	up to 512MB	up to 32GB	up to 2TB
Maximum AU Size	512 KB	1 MB	2 MB	4 MB1	64MB

ERASE_SIZE

This 16-bit field indicates N_{ERASE}. When N_{ERASE} of Aus are erased, the timeout value is specified by ERASE_TIMEOUT (Refer to ERASE_TIMEOUT). The host should determine proper number of Aus to be erased in one operation so that the host can indicate progress of erase operation. If this field is set to 0, the erase timeout calculation is not supported.

Table 46-34. Erase size field

ERASE_SIZE	Value Definition
0000h	Erase Time-out Calculation is not supported.
0001h	1 AU
0002h	2 AU
0003h	3 AU
.....	.....
FFFFh	65535 AU

ERASE_TIMEOUT

This 6-bit field indicates the T_{ERASE} and the value indicates erase timeout from offset when multiple Aus are erased as specified by ERASE_SIZE. The range of ERASE_TIMEOUT can be defined as up to 63 seconds and the card manufacturer can choose any combination of ERASE_SIZE and ERASE_TIMEOUT depending on the implementation. Once ERASE_TIMEOUT is determined, it determines the ERASE_SIZE. The host can determine timeout for any number of AU erase by the equation below.

$$\text{Erase timeout of X AU} = \frac{T_{ERASE}}{N_{ERASE}} * X + T_{OFFSET} \quad (46\text{-}1)$$

Table 46-35. Erase timeout field

ERASE_TIMEOUT	Value Definition
00	Erase Time-out Calculation is not supported.
01	1 [sec]
02	2 [sec]
03	3 [sec]
.....	.....
63	63 [sec]

If ERASE_SIZE field is set to 0, this field shall be set to 0.

ERASE_OFFSET

This 2-bit field indicates the T_{OFFSET} and one of four values can be selected. This field is

meaningless if ERASE_SIZE and ERASE_TIMEOUT fields are set to 0.

Table 46-36. Erase offset field

ERASE_OFFSET	Value Definition
0h	0 [sec]
1h	1 [sec]
2h	2 [sec]
3h	3 [sec]

46.5. Programming sequence

46.5.1. Card identification

The host will be in card identification mode after reset and while it is looking for new cards on the bus. While in card identification mode the host resets all the cards, validates operation voltage range, identifies cards and asks them to publish Relative Card Address (RCA). This operation is done to each card separately on its own CMD line. All data communication in the Card Identification Mode uses the command line (CMD) only.

During the card identification process, the card shall operate in the clock frequency of the identification clock rate F_{OD} (400 kHz).

Card reset

The command GO_IDLE_STATE (CMD0) is the software reset command and sets *e*•MMC and SD memory card into Idle State regardless of the current card state. The reset command (CMD0) is only used for memory or the memory portion of Combo cards. In order to reset an I/O only card or the I/O portion of a combo card, use CMD52 to write 1 to the RES bit in the CCCR. Cards in Inactive State are not affected by this command.

After power-on by the host, all cards are in Idle State, including the cards that have been in Inactive State before. After power-on or CMD0, all cards' CMD lines are in input mode, waiting for start bit of the next command. The cards are initialized with a default relative card address (RCA) and with a default driver strength with 400 KHz clock frequency.

Operating voltage range validation

At the start of communication between the host and the card, the host may not know the card supported voltage and the card may not know whether it supports the current supplied voltage. To verify the voltage, the following commands are defined in the related specification.

The SEND_OP_COND (CMD1 for *e*•MMC), SD_SEND_OP_COND (ACMD41 for SD memory), IO_SEND_OP_COND (CMD5 for SD I/O) command is designed to provide hosts with a mechanism to identify and reject cards which do not match the V_{DD} range desired by the host. This is accomplished by the host sending the required V_{DD} voltage window as the

operand of this command. If the card cannot perform data transfer in the specified range it must discard itself from further bus operations and go into Inactive State. Otherwise, the card shall respond sending back its V_{DD} range.

If the card can operate on the supplied voltage, the response echoes back the supply voltage and the check pattern that were set in the command argument.

If the card cannot operate on the supplied voltage, it returns no response and stays in idle state. It is mandatory to issue CMD8 prior to ACMD41 to initialize SDHC Card. Receipt of CMD8 makes the cards realize that the host supports the Physical Layer Version 2.00 and the card can enable new functions.

Card identification process

The card identification process differs in different cards. The card can be of the type *e*•MMC, SD, or SD I/O. All types of SD I/O cards are supported, that is, SDIO_IO_ONLY, SDIO_MEM_ONLY, and SDIO COMBO cards. The identification process sequence includes the following steps:

1. Check if the card is connected.

2. Identify the card type; SD, *e*•MMC, or SD I/O.
- Send CMD5 first. If a response is received, then the card is SD I/O
- If not, send ACMD41; if a response is received, then the card is SD.
- Otherwise, the card is an *e*•MMC.

10.1.12 Initialization the card according to the card type.

Use a clock source with a frequency = F_{OD} (that is, 400 KHz) and use the following command sequence:
- SD card - Send CMD0, ACMD41, CMD2, CMD3.
- SDHC card - send CMD0, CMD8, ACMD41, CMD2, CMD3.
- SD I/O - Send CMD52, CMD0, CMD5, if the card doesn't have memory port, send CMD3; otherwise send ACMD41, CMD11 (optional), CMD2, and CMD3.
- *e*•MMC - Send CMD0, CMD1, CMD2, CMD3.

46.5.2. Boot operation

If, after power-on or reset (either hardware reset or CMD0 with the argument 0xF0F0F0F0), the CMD line is held low for more than 74 clock cycles before sending the first command (normal boot mode), or CMD0 with the argument 0xFFFFFFFA is sent (alternative boot mode), the card recognizes that the boot mode has been initiated and starts internal boot data preparation.

The partition from which the host will read the boot data can be selected in advance using EXT_CSD byte [179], bits [5:3]. The data size that the host can read during boot operation can be calculated as 128KB × BOOT_SIZE_MULT (EXT_CSD byte [226]).

The host can choose to use single data rate mode with backward-compatible interface timing,

SDR with high-speed interface timing or DDR timing (if it supported) by setting a proper value in EXT_CSD register byte [177] bits [4:3]. EXT_CSD register byte [228], bit 2 tells the host if the high-speed timing during boot is supported by the device. EXT_CSD register byte [228], bit 1 tells the host if the dual data rate mode during boot is supported by the device.

HS200 mode is not supported during boot operation.

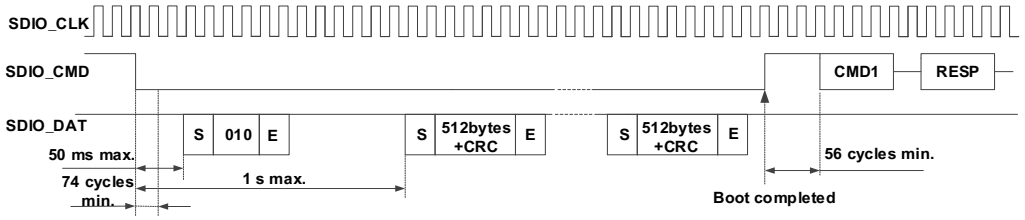
The host can choose to receive boot acknowledge from the card by setting "1" in EXT_CSD register, byte [179], bit 6, so that the host can recognize that the card is operating in boot mode.

If boot acknowledge is enabled, the card has to send acknowledge pattern "010" to the host within 50ms after the CMD line goes low. If boot acknowledge is disabled, the card will not send out acknowledge pattern "010".

Normal boot operation

If the CMD line is held low for at least 74 clock cycles before the first command is issued, the card recognizes that boot mode is being initiated. Within one second after the CMD line goes low, the card starts sending the first boot code data on the SDIO_DAT line(s). The host must keep the SDIO_CMD line low until all boot data has been read. The host can end boot mode by pulling the SDIO_CMD line high.

Figure 46-16. Boot operation timing



To perform the normal boot process the following steps needed:

1. Reset or power-up the card.

2. If a boot enable ACK, need to enable the ACKEN, configuration SDIO_ACKTO register and enable the ACKFAIL and ACKTO flag.

3. Enable the data reception by setting DATADIR to 1 and the number of data to be received in DATALEN. Enable the DTTMOUT, DTEND and CMDSEND interrupts for end of command confirmation.

4. Set BOOTMOD to 0, select the normal boot operation mode, and enable boot in BOOTMODEN. Boot acknowledge timeout is enabled and CMD line remains low.

5. The ACKTO or ACKFAIL flag can be used to detect whether the boot acknowledgment is received.
- If the boot acknowledgment is not received in time, the ACKTO flag will appears.
- If an incorrect boot acknowledgment is received, the ACKFAIL flag appears.

10.1.13 When all boot data has been received the DTEND flag appears.

- If data CRC fails the DTCRCERR flag will be set.
- If data timeout the DTTMOUT flag will be set.

7. When last boot data has been received, read boot data from the FIFO until FIFO is empty (RFE = 1) after which DTEND flag is then set.

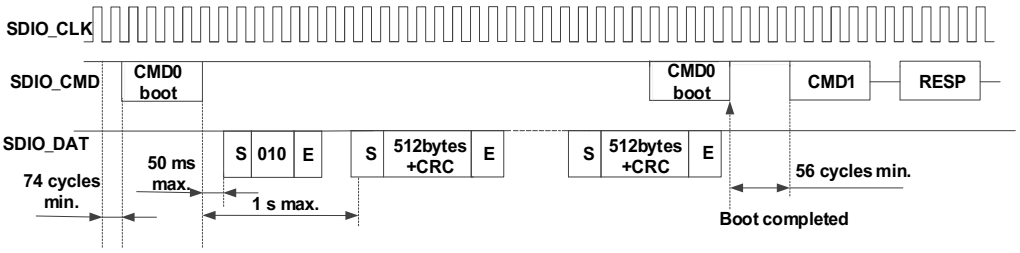
8. The boot process can be terminated by clearing BOOTMODEN bit, and the SDIO_CMD line becomes high. After 56 cycles, a CMDSEND flag is generated to indicate that the boot procedure is over and the card is ready to receive a new command.

- If the boot process is aborted by the firmware before all data has been received, the CSM abort signal stops data transfer and disables DSM which triggers DATABOR flag when enabled.

Alternative boot operation

After card reset or power-up, if the master send CMD0 (with argument of 0xFFFFFFFA) after 74 clock cycles, before CMD1 is issued, the card recognizes that alternative boot mode is being initiated and starts preparing boot data internally. Within 1 second after the CMD0 (with argument of 0xFFFFFFFA) has been sent, the slave card starts to send the first boot data on the SDIO_DAT line(s). The host can terminate boot mode by issuing CMD0 (with argument of 0xF0F0F0F0).

Figure 46-17. Alternative Boot operation timing



The following steps are perform the alternative boot sequence:

1. Reset or power-up the card.

2. If a boot enable ACK, need to enable the ACKEN, set the SDIO_ACKTO register and enable the ACKFAIL and ACKTO flag.

3. Enable the data reception by setting DATADIR to 1 and the number of data to be received in DATALEN. Enable the DTTMOUT and DTEND flags.

4. Set BOOTMOD to 1, select the alternative boot operation mode, load the CMD0 (with argument of 0xFFFFFFFA) in the command registers. Enable CMDSEND flag for end of command confirmation, enable boot in BOOTMODEN, and set CSMEN to 1. When the command has been sent the CMDSEND flag is generated, and the BOOTMODEN bit must be cleared.

5. The ACKTO or ACKFAIL flag can be used to detect whether the boot acknowledgment is

received.

- If the boot acknowledgment is not received in time, the ACKTO flag will appears.
- If an incorrect boot acknowledgment is received, the ACKFAIL flag appears.

10.1.14 When all boot data has been received, the DTEND flag appears.

- If data CRC fails the DCRCFAIL flag will be set.
- If data timeout the DTIMEOUT flag will be set.

7. When last boot data has been received, read boot data from the FIFO until FIFO is empty (RFE = 1) after which DTEND flag is then set.

8. The BOOTMODEN bit must be cleared, before terminating the boot procedure by sending CMD0 (Reset) with alternative boot. This causes the CMDSEND flag to occur 56 cycles after the command. The CMDSEND flag indicates that the boot procedure is over and the card is ready to receive a new command. When the RESET command has been sent successfully, the BOOTMOD has to be cleared to terminate the boot operation.

- If the boot process is aborted by the firmware before all data has been received, the CSM abort signal stops data transfer and disables DSM which triggers DATABOR flag when enabled.

46.5.3. No data commands

To send any non-data command, the software needs to program the SDIO_CMDCTL register and the SDIO_CMDAGMT register with appropriate parameters. Using these two registers, the host forms the command and sends it to the command bus. The host reflects the errors in the command response through the error bits of the SDIO_STAT register.

When a response is received the host sets the CMDRECV (CRC check passed) or CCRCERR (CRC check error) bit in the SDIO_STAT register. A short response is copied in SDIO_RESP0, while a long response is copied to all four response registers. The SDIO_RESP3 bit 31 represents the MSB, and the SDIO_RESP0 bit 0 represents the LSB of a long response.

46.5.4. Single block or multiple block write

During block write (CMD24 - 27) one or more blocks of data are transferred from the host to the card. The block consists of start bits (1 or 4 bits LOW), data block, CRC and end bits(1 or 4 bits HIGH). If the CRC fails, the card indicates the failure on the SDIO_DAT line and the transferred data are discarded and not written, and all further transmitted blocks are ignored.

If the host uses partial blocks whose accumulated length is not block aligned, block misalignment is not allowed (CSD parameter WRITE_BLK_MISALIGN is not set), the card will detect the block misalignment error before the beginning of the first misaligned block. The card shall set the ADDRESS_ERROR error bit in the status register, and while ignoring all further data transfer. The write operation will also be aborted if the host tries to write data on a write protected area. In this case, however, the card will set the WP_VIOLATION bit (in the status register).

Programming of the CID and CSD registers does not require a previous block length setting. The transferred data is also CRC protected. If a part of the CSD or CID register is stored in ROM, then this unchangeable part must match the corresponding part of the receive buffer. If this match fails, then the card reports an error and does not change any register contents.

Some cards may require long and unpredictable time to write a block of data. After receiving a block of data and completing the CRC check, the card will begin writing and hold the DAT0 line low if its write buffer is full and unable to accept new data from a new WRITE_BLOCK command. The host may poll the status of the card with a SEND_STATUS command (CMD13) at any time, and the card will respond with its status. The status bit READY_FOR_DATA indicates whether the card can accept new data or whether the write process is still in progress). The host may deselect the card by issuing CMD7 (to select a different card) which will displace the card into the Disconnect State and release the DAT line without interrupting the write operation. When reselecting the card, it will reactivate busy indication by pulling DAT to low if programming is still in progress and the write buffer is unavailable.

For SD card. Setting a number of write blocks to be pre-erased (ACMD23) will make a following Multiple Block Write operation faster compared to the same operation without preceding ACMD23. The host will use this command to define how many number of write blocks are going to be send in the next write operation.

Steps involved in a single-block or multiple-block write are:

1. Write the data size in bytes in the SDIO_DATALEN register.

2. Write the block size in bytes (BLKSZ) in the SDIO_DATACTL register; the host sends data in blocks of size BLKSZ.

3. Program SDIO_CMDAGMT register with the data address to which data should be written.

4. Program the SDIO_CMDCTL register. For SD memory and *e*•MMC cards, use CMD24 for a single-block write and CMD25 for a multiple-block write. For SD I/O cards, use CMD53 for both single-block and multiple-block transfers.

5. Write data to SDIO_FIFO.

6. Software should look for data error interrupts. If required, software can terminate the data transfer by sending the STOP command (CMD12).

7. When a DTEND interrupt is received, the data transfer is over. For an open-ended block transfer, if the byte count is 0, the software must send the STOP command. If the byte count is not 0, then upon completion of a transfer of a given number of bytes, the host should send the STOP command.

46.5.5. Single block or multiple block read

Block read is block oriented data transfer. The basic unit of data transfer is a block whose maximum size is defined in the CSD (READ_BL_LEN), it is always 512 bytes. If READ_BL_PARTIAL(in the CSD) is set, smaller blocks whose starting and ending address

are entirely contained within 512 bytes boundary may be transmitted.

CMD17 (READ_SINGLE_BLOCK) initiates a block read and after completing the transfer, the card returns to the Transfer state. CMD18 (READ_MULTIPLE_BLOCK) starts a transfer of several consecutive blocks. CRC is appended to the end of each block, ensuring data transfer integrity.

Block Length set by CMD16 can be set up to 512 bytes regardless of READ_BL_LEN.

Blocks will be continuously transferred until a STOP_TRANSMISSION command (CMD12) is issued. The stop command has an execution delay due to the serial command transmission. The data transfer stops after the end bit of the stop command.

When the last block of user area is read using CMD18, the host should ignore OUT_OF_RANGE error that may occur even the sequence is correct.

If the host uses partial blocks whose accumulated length is not block aligned and block misalignment is not allowed, the card shall detect a block misalignment at the beginning of the first misaligned block, set the ADDRESS_ERROR error bit in the status register, abort transmission and wait in the Data State for a stop command.

Steps involved in a single block or multiple block read are:

1. Write the data size in bytes in the SDIO_DATALEN register.

2. Write the block size in bytes (BLKSZ) in the SDIO_DATACTL register. The host expects data from the card in blocks of size BLKSZ each.

3. Program the SDIO_CMDAGMT register with the data address of the beginning of a data read.

4. Program the SDIO_CMDCTL. For SD and e•MMC cards, using CMD17 for a single-block read and CMD18 for a multiple-block read. For SD I/O cards, using CMD53 for both single-block and multiple-block transfers. After writing to the CMD register, the host starts executing the command, when the command is sent to the bus, the CMDRECV flag is set.

5. Software should look for data error interrupts. If required, software can terminate the data transfer by sending a STOP command.

6. The software should read data from the FIFO and make space in the FIFO for receiving more data.

7. When a DTEND interrupt is received, the software should read the remaining data in the FIFO.

46.5.6. Stream write and stream read (e•MMC only)

Stream write

Stream write (CMD20) starts the data transfer from the host to the card beginning from the

starting address until the host issues a stop command. If partial blocks are allowed (if CSD parameter WRITE_BL_PARTIAL is set) the data stream can start and stop at any address within the card address space, otherwise it shall start and stop only at block boundaries. Since the amount of data to be transferred is not determined in advance, CRC cannot be used.

If the host provides an out of range address as an argument to CMD20, the card will reject the command, remain in Tran state and respond with the ADDRESS_OUT_OF_RANGE bit set.

Note that the stream write command works only on a 1 bit bus configuration (on DAT0). If CMD20 is issued in other bus configurations, it is regarded as an illegal command.

In order to sustain data transfer in stream mode of the card, the time it takes to receive the data (defined by the bus clock rate) must be less than the time it takes to program it into the main memory field (defined by the card in the CSD register). Therefore, the maximum clock frequency for the stream write operation is given by the following formula:

$$\text{max write frequency} = \min\left(\text{TRAN_SPEED}, \frac{8*2^{\text{WRITE_BL_LEN}}-100*\text{NSAC}}{\text{TAAC}*\text{R2W_FACTOR}}\right) \quad (46\text{-}2)$$

TRAN_SPEED: Max bus clock frequency.

WRITE_BL_LEN: Max write data block length.

NSAC: Data read access-time 2 in CLK cycles.

TAAC: Data read access-time 1.

R2W_FACTOR: Write speed factor.

All the parameters are defined in CSD register. If the host attempts to use a higher frequency, the card may not be able to process the data and will stop programming, and while ignoring all further data transfer, wait (in the Receive-data-State) for a stop command. As the host sends CMD12, the card will respond with the TXURE bit set and return to Transfer state

Stream read

There is a stream oriented data transfer controlled by READ_DAT_UNTIL_STOP (CMD11). This command instructs the card to send its data, starting at a specified address, until the host sends a STOP_TRANSMISSION command (CMD12). The stop command has an execution delay due to the serial command transmission. The data transfer stops after the end bit of the stop command.

If the host provides an out of range address as an argument to CMD11, the card will reject the command, remain in Transfer state and respond with the ADDRESS_OUT_OF_RANGE bit set.

Note that the stream read command works only on a 1 bit bus configuration (on DAT0). If CMD11 is issued in other bus configurations, it is regarded as an illegal command.

If the end of the memory range is reached while sending data, and no stop command has

been sent yet by the host, the contents of the further transferred payload is undefined. As the host sends CMD12 the card will respond with the ADDRESS_OUT_OF_RANGE bit set and return to Tran state.

In order to sustain data transfer in stream mode of the card, the time it takes to transmit the data (defined by the bus clock rate) must be less than the time it takes to read it out of the main memory field (defined by the card in the CSD register). Therefore, the maximum clock frequency for stream read operation is given by the following formula:

$$\text{max read frequency} = \min\left(\text{TRAN_SPEED}, \frac{8*2^{\text{READ_BL_LEN}}-100*\text{NSAC}}{\text{TAAC}*\text{R2W_FACTOR}}\right) \quad (46\text{-}3)$$

TRAN_SPEED: Max bus clock frequency.

READ_BL_LEN: Max read data block length.

NSAC: Data read access-time 2 in CLK cycles.

TAAC: Data read access-time 1.

R2W_FACTOR: Write speed factor.

All the parameters are defined in CSD register. If the host attempts to use a higher frequency, the card may not be able to process the data and will stop programming, and while ignoring all further data transfer, wait (in the Receive-data-State) for a stop command. As the host sends CMD12, the card will respond with the RXORE bit set and return to Transfer state

46.5.7. Erase

The erasable unit of the *e*•MMC/SD memory is the "Erase Group"; Erase group is measured in write blocks which are the basic writable units of the card. The size of the Erase Group is a card specific parameter and defined in the CSD.

The host can erase a contiguous range of Erase Groups. Starting the erase process is a three steps sequence. First the host defines the start address of the range using the ERASE_GROUP_START (CMD35)/ERASE_WR_BLK_START(CMD32) command, next it defines the last address of the range using the ERASE_GROUP_END (CMD36)/ERASE_WR_BLK_END(CMD33) command and finally it starts the erase process by issuing the ERASE (CMD38) command. The address field in the erase commands is an Erase Group address in byte units. The card will ignore all LSB's below the Erase Group size, effectively rounding the address down to the Erase Group boundary.

If an erase command (CMD35, CMD36, and CMD38) is received out of the defined erase sequence, the card shall set the ERASE_SEQ_ERROR bit in the status register and reset the whole sequence.

If the host provides an out of range address as an argument to CMD35 or CMD36, the card will reject the command, respond with the ADDRESS_OUT_OF_RANGE bit set and reset the whole erase sequence.

If an 'non erase' command (neither of CMD35, CMD36, CMD38 or CMD13) is received, the card shall respond with the ERASE_RESET bit set, reset the erase sequence and execute the last command.

If the erase range includes write protected blocks, they shall be left intact and only the non-protected blocks shall be erased. The WP_ERASE_SKIP status bit in the status register shall be set.

As described above for block write, the card will indicate that an erase is in progress by holding DAT0 low. The actual erase time may be quite long, and the host may issue CMD7 to deselect the card.

46.5.8. Bus width selection

After the host has verified the functional pins on the bus it should change the bus width configuration.

For *e*•MMC, using the SWITCH command (CMD6).The bus width configuration is changed by writing to the BUS_WIDTH byte in the Modes Segment of the EXT_CSD register (using the SWITCH command to do so). After power-on or software reset, the contents of the BUS_WIDTH byte is 0x00. If the host tries to write an invalid value, the BUS_WIDTH byte is not changed and the SWITCH_ERROR bit is set. This register is write only.

For SD memory, using SET_BUS_WIDTH command (ACMD6) to change the bus width. The default bus width after power up or GO_IDLE_STATE command (CMD0) is 1 bit. SET_BUS_WIDTH (ACMD6) is only valid in a transfer state, which means that the bus width can be changed only after a card is selected by SELECT/DESELECT_CARD (CMD7).

46.5.9. Protection management

In order to allow the host to protect data against erase or write, three methods for the cards are supported in the card:

CSD register for card protection (optional)

The entire card may be write protected by setting the permanent or temporary write protect bits in the CSD. Some cards support write protection of groups of sectors by setting the WP_GRP_ENABLE bit in the CSD. It is defined in units of WP_GRP_SIZE erase groups as specified in the CSD. The SET_WRITE_PROT command sets the write protection of the addressed write protected group, and the CLR_WRITE_PROT command clears the write protection of the addressed write protected group.

The High Capacity SD Memory Card does not support Write Protection and does not respond to write protection commands (CMD28, CMD29 and CMD30).

Write protect switch on the card (SD memory and SD I/O card)

A mechanical sliding tablet on the side of the card will be used by the user to indicate that a

given card is write protected or not. If the sliding tablet is positioned in such a way that the window is open it means that the card is write protected. If the window is closed the card is not write protected.

Password card Lock/Unlock Operation

The Password Card Lock/Unlock protection is described in ***Card Lock/Unlock operation***.

46.5.10. Card Lock/Unlock operation

The password protection feature enables the host to lock a card while providing a password, which later will be used for unlocking the card. The password and its size are kept in a 128-bit PWD and 8-bit PWD_LEN registers, respectively. These registers are non-volatile so that a power cycle will not erase them.

Locked cards respond to (and execute) all commands in the basic command class (class 0), ACMD41, CMD16 and lock card command class (class 7). Thus, the host is allowed to reset, initialize, select, query for status, but not to access data on the card. If the password was previously set (the value of PWD_LEN is not 0), the card will be locked automatically after power on.

Similar to the existing CSD register write commands, the lock/unlock command is available in "transfer state" only. This means that it does not include an address argument and the card shall be selected before using it.

The card lock/unlock command has the structure and bus transaction type of a regular single block write command. The transferred data block includes all the required information of the command (password setting mode, PWD itself, card lock/unlock etc.). ***Table 46-37. Lock card data structure*** describes the structure of the command data block.

Table 46-37. Lock card data structure

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Reserved(all set to 0)				ERASE	LOCK_UNLOCK	CLR_PWD	SET_PWD
1	PWDS_LEN							
2	Password data(PWD)							
……								
PWDS_LEN+1								

ERASE: 1 Defines Forced Erase Operation. In byte 0, bit 3 will be set to 1 (all other bits shall be 0). All other bytes of this command will be ignored by the card.

LOCK/UNLOCK: 1 = Locks the card. 0 = Unlock the card (note that it is valid to set this bit together with SET_PWD but it is not allowed to set it together with CLR_PWD).

CLR_PWD: 1 = Clears PWD.

SET_PWD: 1 = Set new password to PWD.

PWDS_LEN: Defines the following password(s) length (in bytes). In case of a password

change, this field includes the total password length of old and new passwords. The password length is up to 16 bytes. In case of a password change, the total length of the old password and the new password can be up to 32 bytes.

Password data: In case of setting a new password, it contains the new password. In case of a password change, it contains the old password followed by the new password.

Setting the password

- Select a card (CMD7), if not previously selected.
- Define the block length (CMD16), given by the 8-bit card lock/unlock mode, the 8-bit password size (in bytes), and the number of bytes of the new password. In the case that a password replacement is done, then the block size shall consider that both passwords (the old and the new one) are sent with the command.
- Send the Card Lock/Unlock command with the appropriate data block size on the data line including the 16-bit CRC. The data block shall indicate the mode (SET_PWD), the length (PWDS_LEN) and the password itself. In the case that a password replacement is done, then the length value (PWDS_LEN) shall include both passwords (the old and the new one) and the password data field shall include the old password (currently used) followed by the new password. Note that the card shall handle the calculation of the new password length internally by subtracting the old password length from PWDS_LEN field.
- In the case that the sent old password is not correct (not equal in size and content), then the LOCK_UNLOCK_FAILED error bit will be set in the status register and the old password does not change. In the case that the sent old password is correct (equal in size and content), then the given new password and its size will be saved in the PWD and PWD_LEN registers, respectively.

Reset the password

- Select a card (CMD7), if not previously selected.
- Define the block length (CMD16), given by the 8-bit card lock/unlock mode, the 8-bit password size (in bytes), and the number of bytes of the currently used password.
- Send the card lock/unlock command with the appropriate data block size on the data line including the 16-bit CRC. The data block shall indicate the mode CLR_PWD, the length (PWDS_LEN), and the password itself. If the PWD and PWD_LEN content match the sent password and its size, then the content of the PWD register is cleared and PWD_LEN is set to 0. If the password is not correct, then the LOCK_UNLOCK_FAILED error bit will be set in the status register.

Locking a card

- Select a card (CMD7), if not previously selected.
- Define the block length (CMD16), given by the 8-bit card lock/unlock mode, the 8-bit password size (in bytes), and the number of bytes of the currently used password.
- Send the card lock/unlock command with the appropriate data block size on the data line including the 16-bit CRC. The data block shall indicate the mode LOCK, the length (PWDS_LEN) and the password itself.

If the PWD content is equal to the sent password, then the card will be locked and the card-locked status bit will be set in the status register. If the password is not correct, then the LOCK_UNLOCK_FAILED error bit will be set in the status register.

Unlocking the card

- Select a card (CMD7), if not previously selected.
- Define the block length (CMD16), given by the 8-bit card lock/unlock mode, the 8-bit password size (in bytes), and the number of bytes of the currently used password.
- Send the card lock/unlock command with the appropriate data block size on the data line including the 16-bit CRC. The data block shall indicate the mode UNLOCK, the length (PWDS_LEN) and the password itself.

If the PWD content is equal to the sent password, then the card will be unlocked and the card-locked status bit will be cleared in the status register. If the password is not correct, then the LOCK_UNLOCK_FAILED error bit will be set in the status register.

46.5.11. Sleep

The *e*•MMC card can switch between Sleep state and Standby state through CMD5. In the sleep state, the power consumption of the card is minimized, and the Vcc power can be turned off at this time.

- Sleep command: The bit 15 as set to 1 in CMD5 argument.
- Awake command: The bit 15 as set to 0 in CMD5 argument.

Use Sleep command to start the state transition from Standby state to Sleep state. The card indicates the transition phase busy by pulling down the SDIO_DAT0 line. No further commands should be sent during the busy. When the device stops pulling down the SDIO_DAT0 line, indicates reach Sleep state.

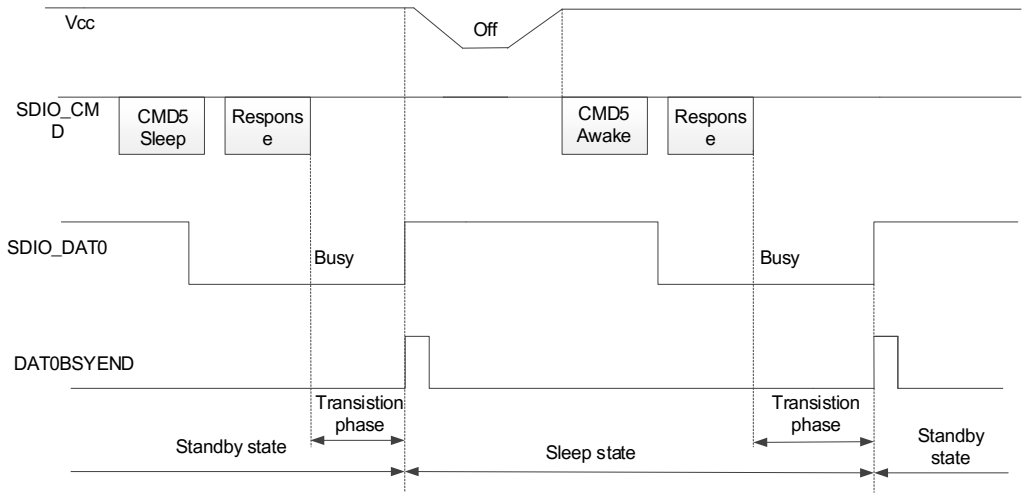
Use Awake command to start the state transition from Sleep state to Standby state. The card indicates the transition phase busy by pulling down the SDIO_DAT0 line. No further commands should be sent during the busy. When the device stops pulling down the SDIO_DAT0 line, indicates reach Standby state.

To set the card to sleep, follow these steps:
1. Enable DAT0BSYEND interrupt.
2. Send CMD5 (Sleep command).
3. When DAT0BSYEND is interrupted, the card is in Sleep state.
4. It is allowed to turn off the Vcc power.

To set the card to Standby, follow these steps:
1. Turn on the Vcc power supply and wait for it to reach the minimum working voltage.
2. Enable DAT0BSYEND interrupt.
3. Send CMD5 (Awake command).
4. When DAT0BSYEND interrupt occurs, the card is in Standby mode.

During the Sleep state the Vcc power supply may be switched off. This is to enable even further system power consumption saving. The Vcc supply is allowed to be switched off only after the Sleep state has been reached (the card has stopped to pull down the SDIO_DAT0 line). The Vcc supply have to be ramped back up at least to the minimum working voltage level before the state transition from Sleep state to Standby state is allowed to be initiated (Awake command).

Figure 46-18. CMD5 timing



46.5.12. CMD12 send timing

CMD12 (stop transmission command) is used to stop or abort the data transmission. The card data transmission is terminated after 2 clock cycles after the end bit of a CMD12.

All data read and write commands can be terminated at any time by a CMD12. If data transmission is in progress, the CMD12 send sequence includes the following steps:

1. Configure the CMD12 command in registers and set the TRSTOP register. When DSM receives the command, it will generate CSM stop signal.

2. Clear WAITDEND register.

3. When IDMAEN = 0, set FIFOREST bit to 1.
- Host sends data, when the CMDRECV flag appears, the firmware will stop transfer data to the FIFO. Subsequently set FIFOREST bit to 1 and flushes the FIFO.
- Host receives data, when the CMDRECV flag appears, the firmware will read the remaining data from the FIFO. Subsequently set FIFOREST bit to 1.

10.1.15 When IDMAEN = 1, hardware operates the FIFO.
- Host sends data, when the Abort flag appears, the hardware will stop IDMA. Subsequently flushes the FIFO.
- Host receives data, when the Abort flag appears, the hardware uses IDMA to transfer the remaining FIFO data to RAM.

10.1.16 When FIFO is empty/reset, the DATABOR flag is generated

Table 46-38. CMD12 instructions

Data operation type	CMD12 Description
Multiple block write with pre-defined block count	The device will receives the requested number of data blocks, terminate the transaction and return to TRANSFER state. Stop command is not required at the end of this type of multiple block write, unless terminated with an error.
Multiple block read with pre-defined block count	The card will transfer the requested number of data blocks, terminate the transaction and return to TRANSFER state. Stop command is not required at the end of this type of multiple block read, unless terminated with an error.
Open-ended Multiple block write	The number of blocks for the write multiple block operation is not defined. The Device will continuously receive and program data blocks until a stop transmission command is received.
Open-ended Multiple block read	The number of blocks for the read multiple block operation is not defined. The Device will continuously transfer data blocks until a stop transmission command is received
stream write	The data flow is terminated by CMD12.
Stream read	The data flow is terminated by CMD12.

CMD12 affect block operation

To stop block transmission at the end of data, the end bit of CMD12 should be sent after the end bit of last block.

When data is written to the card, the CMD12 end bit is sent after the CRC token end bit of the data block is written. This requires the CMD12 transmission process to follow the block transmission timing.

To stop the Open-ended Multiple block write operation, follow these steps:

1. Before starting data transmission, set TRANSMOD[1:0] to "11".

2. Wait for DTEND to be set, the data sent by DSM will not exceed SDIO_DATALEN.

3. CSM sents CMD12. The card is set to idle state.

When reading data from the card, the end bit of CMD12 should be sent as early as possible. That is, when the card reads the last bit of the data block.

To stop the Open-ended Multiple block read operation, follow these steps:

1. Before starting data transmission, set TRANSMOD[1:0] to "11".

2. Wait for DTEND to be set, even if the card is sending more data, the data received by DSM will not be greater than SDIO_DATALEN.

3. CSM sents CMD12. The card sending data is stopped and the card is set to idle state.

CMD12 affect stream operation

To stop stream transmission after the last byte to be transmitted, the CMD12 end bit timing should be sent at the end of the last byte of the data stream. Follow these steps to write stream data:

1. Initialize DSM, set TRANSMOD[1:0] to "10".

2. Set TREN to 1 and sent stream write command.

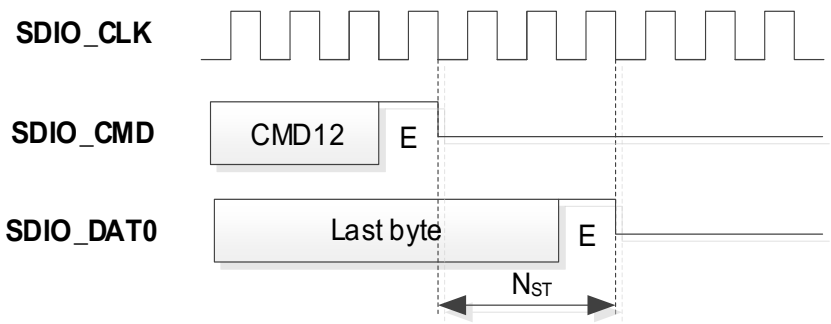
3. Preload CMD12 in command registers with the TRSTOP bit set.

4. Configure the CSM to send a command only after WAITDEND = 1 end of last data of DATALEN.

5. CSM sents CMD12. The stream data end bit and command end bit are aligned.
- If DATALEN > 5, CMD12 is waited in the CSM to be aligned with the data transfer end bit.
- If DATALEN < 5, CMD12 will start early and DSM will keep WaitS state, to align the data transfer end bit with the CMD12 end bit.

10.1.17 The stream data writing process can be stopped at any time by clearing the WAITDEND bit. This will cause the preloaded CMD12 to be sent immediately and stop the stream writing process.

Figure 46-19. CMD12 stream timing



To stop the stream transfer read process after the last byte, the CMD12 end bit timing should be sent after the last byte of the data stream.

Note: 1. Wait for DTEND = 1 (data receiving completed) before sending the CMD12. Even if the card is sending more data. The data received by DSM will not be greater than SDIO_DATALEN. 2. The SDIO does not receive any more data when DATACNT = 0, even when the card continues sending data.

46.6. Specific operations

46.6.1. SD I/O specific operations

The SD I/O only card and SD I/O combo card support these specific operations:

Read Wait operation

Suspend/resume operation

Interrupts

The SD I/O supports these operations only if the SDIO_DATACTL[11] bit is set, except for read suspend that does not need specific hardware implementation.

SD I/O read wait operation

The optional Read Wait (RW) operation is defined only for the SD 1-bit and 4-bit modes. The Read Wait operation allows a host to signal a card that is executing a read multiple (CMD53) operation to temporarily stall the data transfer while allowing the host to send commands to any function within the SD I/O card. To determine if a card supports the Read Wait protocol, the host shall test SRW capability bit in the Card Capability byte of the CCCR. The timing for Read Wait is based on the Interrupt Period. If a card does not support the Read Wait protocol, the only means a host has to stall (not abort) data in the middle of a read multiple command is to control the SDIO_CLK. The limitation of this method is that with the clock stopped, the host cannot issue any commands, and so cannot perform other operations during the delay time. Read Wait support is mandatory for the card to support suspend/resume. ***Figure 46-20. Read wait control by stopping SDIO_CLK*** and ***Figure 46-21. Read wait operation using SDIO_DAT[2]*** show the Read Wait mode about stop the SDIO_CLK and use SDIO_DAT[2].

Figure 46-20. Read wait control by stopping SDIO_CLK

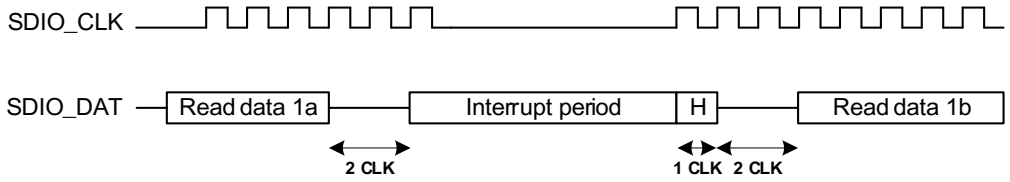
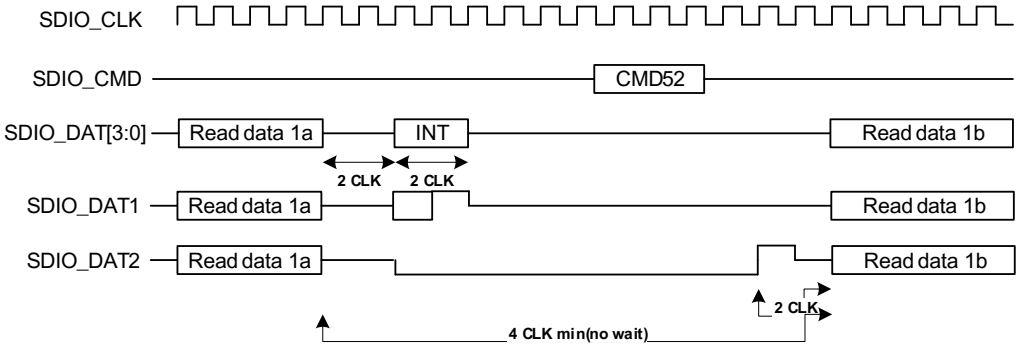


Figure 46-21. Read wait operation using SDIO_DAT[2]



We can start the Read Wait interval before the data block is received: when the data unit is enabled (SDIO_DATACTL[0] bit set), the SD I/O specific operation is enabled (SDIO_DATACTL[11] bit set), Read Wait starts (SDIO_DATACTL[10] = 0 and SDIO_DATACTL[8] = 1) and data direction is from card to SD I/O (SDIO_DATACTL[1] = 1), the DSM directly moves from Idle to Read Wait. In Read Wait the DSM drives SDIO_DAT[2] to 0 after 2 SDIO_CLK clock cycles. In this state, when you set the RWSTOP bit

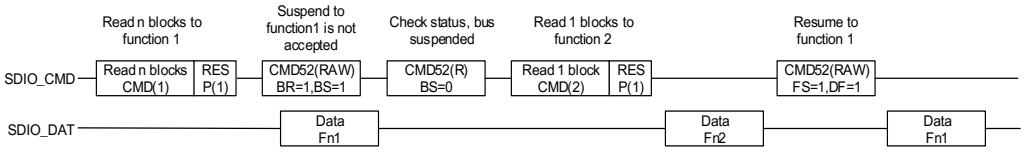
(SDIO_DATACTL[9]), the DSM remains in Wait for two more SDIO_CLK clock cycles to drive SDIO_DAT[2] to 1 for one clock cycle. The DSM then starts waiting again until it receives data from the card. The DSM will not start a Read Wait interval while receiving a block even if Read Wait start is set: the Read Wait interval will start after the CRC is received. The RWSTOP bit has to be cleared to start a new Read Wait operation. During the Read Wait interval, the SDIO can detect SD I/O interrupts on SDIO_DAT[1].

SD I/O suspend/resume operation

Within a multi-function SD I/O or a Combo card, there are multiple devices (I/O and memory) that share access to the SD bus. In order to allow the sharing of access to the host among multiple devices, SD I/O and combo cards can implement the optional concept of suspend/resume. If a card supports suspend/resume, the host may temporarily halt a data transfer operation to one function or memory (suspend) in order to free the bus for a higher priority transfer to a different function or memory. Once this higher-priority transfer is completed, the original transfer is re-started where it left off (resume).

Figure 46-22. Function2 read cycle inserted during function1 multiple read cycle shows a condition where the first suspend request is not immediately accepted. The host then checks the status of the request with a read and determines that the bus has now been released (BS = 0). At this time, a read to function 2 is started. Once that single block read is completed, the resume is issued to function, causing the data transfer to resume (DF = 1).

Figure 46-22. Function2 read cycle inserted during function1 multiple read cycle



When the host sends data to the card, the host can suspend the write operation. The SDIO_CMDCTL[16] bit is set (and TREN = 0), and indicates to the CSM that the current command is a suspend command. The CSM analyzes the response and when the response is received from the card (suspend accepted), it acknowledges the DSM that goes Idle after receiving the CRC token of the current block.

To suspend a read operation, the DSM waits in the WaitR state, when the function to be suspended sends a complete packet just before stopping the data transaction. The application should continue reading receive FIFO until the FIFO is empty, and the DSM goes Idle state automatically.

If suspend function will be used, the BLKSZ must not be 0000 or 0001 in multi block transfer mode.

Interrupts

In order to allow the SD I/O card to interrupt the host, an interrupt function is added to a pin on the SD interface. Pin number 8, which is used as SDIO_DAT[1] when operating in the 4-bit SD mode, is used to signal the card's interrupt to the host. The use of interrupt is optional

for each card or function within a card. The SD I/O interrupt is "level sensitive", that is, the interrupt line shall be held active (low) until it is either recognized and acted upon by the host or de-asserted due to the end of the Interrupt Period. Once the host has serviced the interrupt, it is cleared via function unique I/O operation.

When setting the SDIO_DATACTL[11] bit SD I/O interrupts can detect on the SDIO_DAT[1] line.

Figure 46-23. Read Interrupt cycle timing shows the timing of the interrupt period for single data transaction read cycles.

Figure 46-23. Read Interrupt cycle timing

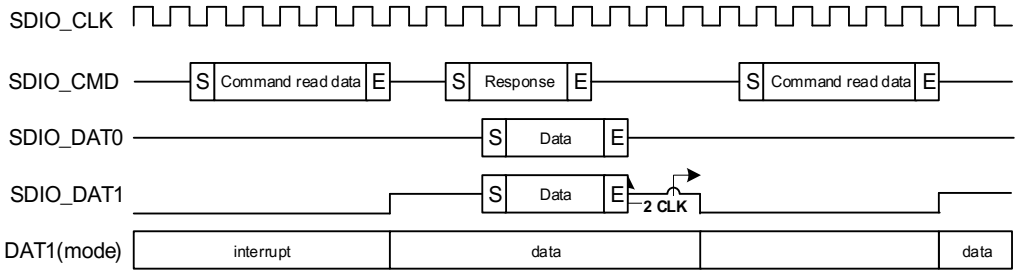
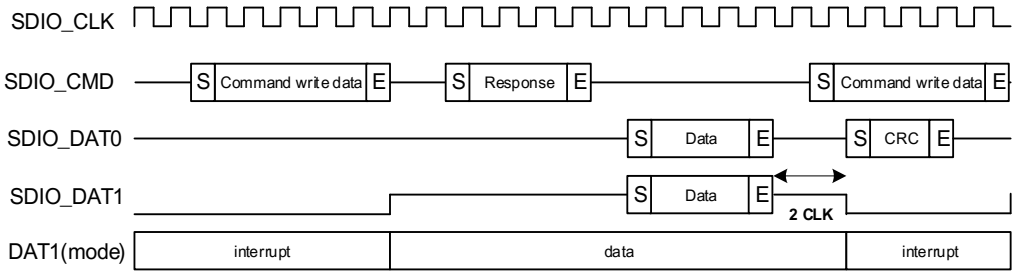


Figure 46-24. Write interrupt cycle timing



When transferring multiple blocks of data in the 4-bit SD mode, a special definition of the interrupt period is required. In order to allow the highest speed of communication, the interrupt period is limited to a 2-clock interrupt period. Card that wants to send an interrupt signal to the host shall assert DAT1 low for the first clock and high for the second clock. The card shall then release DAT1 into the hi-Z State. *Figure 46-25. Multiple block 4-Bit read interrupt cycle timing* shows the operation for an interrupt during a 4-bit multi-block read and *Figure 46-26. Multiple block 4-Bit write interrupt cycle timing* shows the operation

for an interrupt during a 4-bit multi-block write.

Figure 46-25. Multiple block 4-Bit read interrupt cycle timing

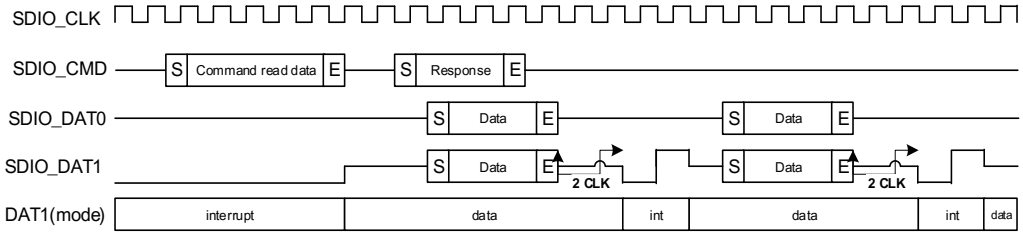
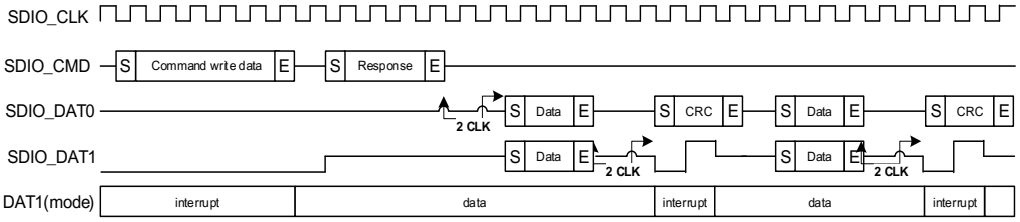


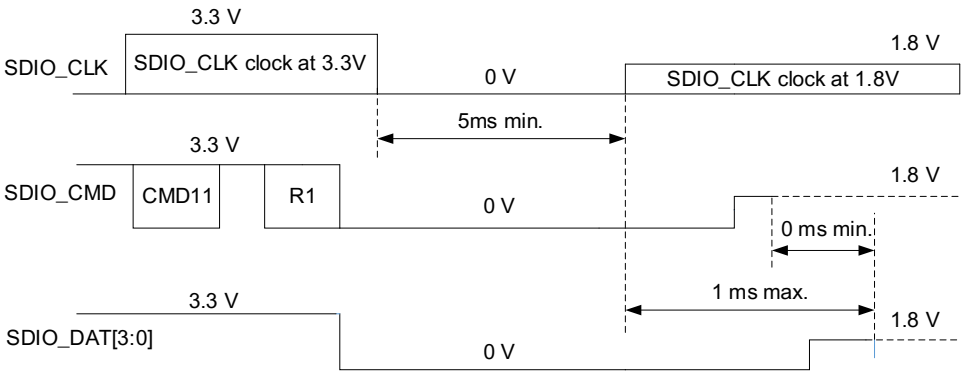
Figure 46-26. Multiple block 4-Bit write interrupt cycle timing



46.6.2. UHS-I

Ultra High Speed Bus Speed Mode (SDR12, SDR25, SDR50, DDR50 and SDR104) phase I Voltage Switch (UHS-1) mode requires the support voltage switching from 3.3V to 1.8V. The default voltage is 3.3V when the card is powered on. A signal voltage switch sequence is started by this CMD11. When the voltage sequence is completed successfully, the card will enter the UHS-I mode with the default SDR12, and the input and output timings of the card will change.

Figure 46-27. CMD11 voltage switch sequence



The following steps are perform the signal Voltage Switch sequence:

1. The SDIO_CLK clock frequency must be set in the range 100 kHz - 400 kHz, before starting of Voltage Switch operation.

2. The host sets VSEN = 1 for starts the Voltage Switch, and then sends the CMD11.

3. Host receiving card R1 response.

- If the response CRC passes, the Voltage Switch process will continue and the host will no

longer drive the CMD and SDIO_DAT[3:0] signals until the voltage switching procedure is completed. After several cycles of response, SDIO_CLK will stop and CLKSTOP flag will be set to 1.
- If the response CRC fails (CCRCERR = 1) or the response timeout (CMDTMOUT = 1), the Voltage Switch procedure is stopped.

4. At the next clock after the R1 response, the card drives CMD and SDIO_DAT[3:0] to low.

5. After receiving the R1 response, the host monitors the SDIO_DAT0 line by reading the DAT0BSY register. The SDIO_DAT0 line is sampled after 2 SDIO_CLK clock cycles following the response. The Firmware may read the DAT0BSY register bit following the CLKSTOP flag.
- When DAT0BSY is detected as low level, the firmware switches the voltage regulator to 1.8V. After that, it instructs the SDIO to start timing the critical section of the Voltage Switch sequence by setting register bit VSSTART to 1. The hardware continues to stop the SDIO_CLK by holding it low for at least 5 ms.
- When DAT0BSY is detected as high level, the firmware will abort the Voltage Switch sequence and cycle power the device card.

6. When SDIO_CLK is low, the card will start to switch the signal voltage to 1.8V.

7. The SDIO hardware will restart the SDIO_CLK at least 5 ms later.

8. Within 1 ms after SDIO_CLK switch is detected, the card will drive the CMD and DAT[3:0] to high level for at least 1 clock cycle, and then stop driving the CMD and DAT[3:0].

9. After 1 ms after the SDIO_CLK is restarted, the SDIO hardware will sample SDIO_DAT0 into DAT0BSY and generate VSEND flag.

10. If VSEND flag is set, the host needs to read DAT0BSY register to judge SDIO_DAT0 line, to confirm whether the voltage switching procedure is completed:
- If DAT0BSY is high, Voltage Switch has been completed successfully.
- If DAT0BSY is low, Voltage Switch has failed, the host cycles the card power.

46.7. Register definition

SDIO0 base address: 0x5200 7000

SDIO1 base address: 0x4802 2400

46.7.1. Power control register (SDIO_PWRCTL)

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											DIRPS	VSEN	VSSTART	PWRCTL[1:0]	
											rw	rw	rw	rw	

Bits	Fields	Descriptions
31:5	Reserved	Must be kept at reset value.
4	DIRPS	Command and data direction polarity selection bit This bit can only be configured when the PWRCTL[1:0] bits are cleared. 0: The voltage transceiver IOs are driven as outputs when the direction signal is low. 1: The voltage transceiver IOs are driven as outputs when the direction signal is high.
3	VSEN	Voltage switch enable bit This bit is used to stop the SDIO_CLK after the voltage switch command response. This bit can only be configured by firmware when CSMEN bit is cleared. 0: SDIO_CLK clock kept unchanged, after successfully received command response. 1: SDIO_CLK clock stopped, after successfully received command response.
2	VSSTART	Voltage switch start bit This bit is used to start the timing of the voltage switch sequence. 0: Voltage switch sequence not active and not started 1: Voltage switch sequence active or started
1:0	PWRCTL[1:0]	SDIO power control bits These bits control the SDIO state, card input or output. These bits can only be configured when the SDIO is in the power down state. 00: State after reset (reset: SDIO is disabled, clock is stopped, SDIO_CMD and SDIO_DAT are HiZ state, and SDIO_CLK is driven low), if written 00, SDIO power

off (power off: SDIO is disabled, clock is stopped, SDIO_CLK, SDIO_CMD, SDIO_DAT are driven high).

01: Reserved (When written 01, PWRCTL value does not change)

10: Power-cycle, the SDIO is disabled and the clock to the card is stopped, SDIO_DAT, SDIO_CMD and SDIO_CLK are driven low.

11: SDIO Power on. After power on, SDIO enables have 74 SDIO_CLK cycles delay, any further write is ignored, PWRCTL[1:0] value keeps 11.

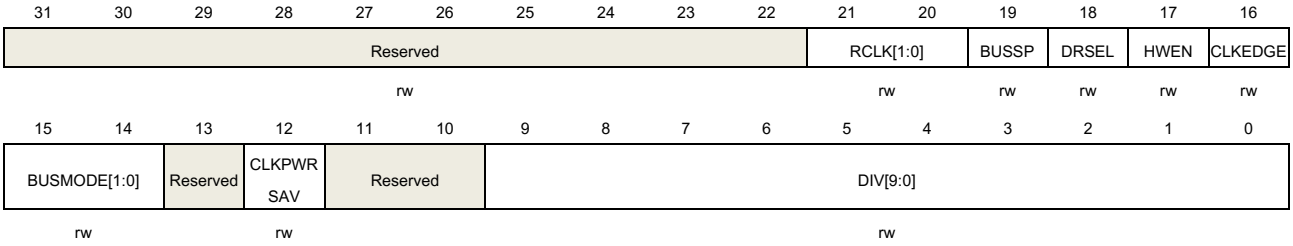
46.7.2. Clock control register (SDIO_CLKCTL)

Address offset: 0x04

Reset value: 0x0000 0000

This register controls the output clock SDIO_CLK.

This register has to be accessed by word(32-bit).



Bits	Fields	Descriptions
31:22	Reserved	Must be kept at reset value.
21:20	RCLK[1:0]	Receive clock selection bits These bits can only be configured when the (DATSTA = 0) and (CMDSTA = 0). 00: select SDIO_IN_CLK clock 01: select SDIO_CLKIN clock 10: select SDIO_FB_CLK clock 11: Reserved (select SDIO_IN_CLK clock)
19	BUSSP	Bus speed mode selection bit This bit can only be configured when the (DATSTA = 0) and (CMDSTA = 0). 0: DS, HS, SDR12, SDR25 bus speed. 1: SDR50, DDR50, SDR104 bus speed.
18	DRSEL	Data rate selection bit This bit can only be configured when the (DATSTA = 0) and (CMDSTA = 0). 0: SDR selected 1: DDR selected
17	HWEN	Hardware flow control enable bit If this bit is set, the meaning of the TFF and RFF flags will change. This bit can only be configured when the (DATSTA = 0) and (CMDSTA = 0).

		0: HW flow control is disabled 1: HW flow control is enabled
16	CLKEDGE	Command and data SDIO_CLK dephasing selection bit These bits can only be written when the (DATSTA = 0) and (CMDSTA = 0). 0: If DIV = 0, data and command change on SDIO_CLK rising edge. SDIO_CLK edge occurs on CK_SDIO rising edge. If DIV > 0 and DRSEL = 0, command and data changed on the CK_SDIO falling edge succeeding the rising edge of SDIO_CLK. SDIO_CLK edge occurs on CK_SDIO rising edge. If DIV > 0 and DRSEL = 1, command changed on the CK_SDIO falling edge succeeding the rising edge of SDIO_CLK. Data changed on the CK_SDIO falling edge succeeding the rising/falling edge of SDIO_CLK. SDIO_CLK edge occurs on CK_SDIO rising edge. 1: if DIV = 0, command and data changed on the falling edge of SDIO_CLK. SDIO_CLK edge occurs on SDIO_CLK rising edge. If DIV > 0 and DRSEL = 0, command and data changed on the SDIO_CLK falling edge (aligned with the rising edge of CK_SDIO). SDIO_CLK edge occurs on SDIO_CLK rising edge. If DIV > 0 and DRSEL= 1, command changed on the SDIO_CLK falling edge (aligned with the rising edge of CK_SDIO). Data changed on the SDIO_CLK 1/4 SDIO_CLK. SDIO_CLK edge occurs on CK_SDIO rising edge.
15:14	BUSMODE[1:0]	SDIO card bus mode control bit These bits can only be configured when the (DATSTA = 0) and (CMDSTA = 0). 00: 1-bit SDIO card bus mode selected (default and SDR only), SDIO_DAT0. 01: 4-bit SDIO card bus mode selected, SDIO_DAT [3:0]. 10: 8-bit SDIO card bus mode selected, SDIO_DAT[7:0]. 11: No effect
13	Reserved	Must be kept at reset value.
12	CLKPWRSAV	SDIO_CLK clock dynamic switch on/off for power saving. This bit can only be configured when the (DATSTA = 0) and (CMDSTA = 0). This bit controls SDIO_CLK clock dynamic switch on/off when the bus is idle for power saving. 0: SDIO_CLK clock is always on 1: SDIO_CLK closed when bus idle
11:10	Reserved	Must be kept at reset value.
9:0	DIV[9:0]	Clock division These bits can only be configured when the (DATSTA = 0) and (CMDSTA = 0). This field defines the division factor to generator SDIO_CLK clock to card. The SDIO_CLK is divider from CK_SDIO, and the SDIO_CLK frequency = CK_SDIO / (DIV[9:0] * 2). 0x000: SDIO_CLK = CK_SDIO / 1 (Does not support DDR)

0x001: SDIO_CLK = CK_SDIO / 2

0x002: SDIO_CLK = CK_SDIO / 4

……

0x3FF: SDIO_CLK = CK_SDIO / 2046

Note: When card is in identification mode, the frequency of SDIO_CLK must be less than 400 kHz.

If the RCA are assigned to all cards, you can change the clock frequency to the maximum card bus frequency.

Between Two write accesses to this register, it needs at least 7 HCLK periods which used to sync the registers to SDIO_CLK clock domain.

46.7.3. Command argument register (SDIO_CMDAGMT)

Address offset: 0x08

Reset value: 0x0000 0000

This register defines 32bit command argument, which will be used as part of the command (bit 39 to bit 8).

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CMDAGMT[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMDAGMT[15:0]															
rw															

Bits	Fields	Descriptions
31:0	CMDAGMT[31:0]	SDIO card command argument This field defines the card command argument which sent to card. This field is the bits [39:8] of command message. If the command message contains an argument, this field must update before writing SDIO_CMDCTL register when sending a command. These bits can only be configured by firmware when CSMEN = 0.

46.7.4. Command control register (SDIO_CMDCTL)

Address offset: 0x0C

Reset value: 0x0000 0000

The SDIO_CMDCTL register contains the command index and other command control bits to control command state machine.

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Reserved	CMDSR
	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BOOTMODEN	BOOTMOD	HOLD	CSMEN	WAITDEND	INTWAIT	CMDRESP[1:0]		TRSTOP	TREN	CMDIDX[5:0]					
rw	rw	rw	rw	rw	rw	rw		rw	rw	rw					

Bits	Fields	Descriptions
31:17	Reserved	Must be kept at reset value.
16	CMDSR	Suspend or Resume command and signals interrupt period start/end bit. This bit can only be configured by firmware when the CSMEN = 0. 0: No effect. 1: If TREN = 0, the CSM treats the command as a Suspend command. If TREN = 1, the CSM treats the command as a Resume command.
15	BOOTMODEN	Boot mode procedure enable bit 0: Boot disable 1: Boot enable
14	BOOTMOD	Boot mode selection bit This bit can only be configured by firmware when the CSMEN = 0. 0: Normal boot mode 1: Alternative boot mode
13	HOLD	Hold DSM transmission and reception of new data block If HOLD bit is set, the DSM does not move from the DS_WaitS state to the DS_Send state or from the DS_WaitR state to the DS_Receive state. 0: No effect 1: Hold transmission and reception data
12	CSMEN	Command state machine (CSM) enable bit This bit can only be configured by firmware. It is cleared by hardware. 0: Command state machine disable (stay on CS_Idle) 1: Command state machine enable
11	WAITDEND	Waits for ends of data transfer If this bit is set, the command state machine starts to send a command must wait the end of data transfer. 0: No effect 1: Wait the end of data transfer
10	INTWAIT	Interrupt wait request If this bit is cleared in Wait state, the interrupt mode is aborted. 0: No effect. 1: No command timeout and wait interrupt while the command response.
9:8	CMDRESP[1:0]	Waits command response type bits

		These bits define the response type after sending a command message. These bits can only be written by firmware when the CSMEN = 0. 00: No response 01: Short response 10: Short response (No CRC) 11: Long response
7	TRSTOP	Data transfer stop command mode enable bit Transfer stop command mode (CSM treats the command as a data stop transfer command). This bit can only be configured by firmware when the CSMEN = 0. 0: No effect 1: If command is sent, enable transfer stop command mode and stops DSM data transfer.
6	TREN	Data transfer command mode enable bit Transfer command mode (CSM treats the command as a data transfer command). This bit can only be configured by firmware when the CSMEN = 0. 0: No effect 1: If command is sent, enable transfer command mode and stops the interrupt period.
5:0	CMDIDX[5:0]	Command index These bits can only be configured by firmware when the CSMEN = 0. This field defines the command index to be sent to card.

Note: Between Two write accesses to this register, it needs at least seven HCLK clock periods which used to sync the registers to SDIO_CLK clock domain.

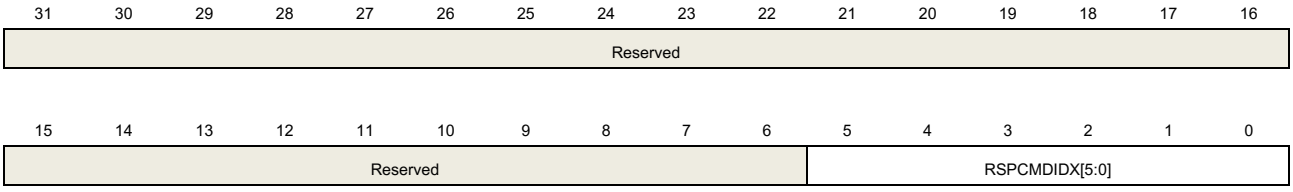
The e•MMC can send two kinds of response: short responses, 48 bits, or long responses,136 bits.

SD card and SD I/O card can send only short responses, the argument can vary according to the type of response: the software distinguishes the type of response according to the send command.

46.7.5. Command index response register (SDIO_RSPCMDIDX)

Address offset: 0x10
Reset value: 0x0000 0000

This register has to be accessed by word(32-bit).



r

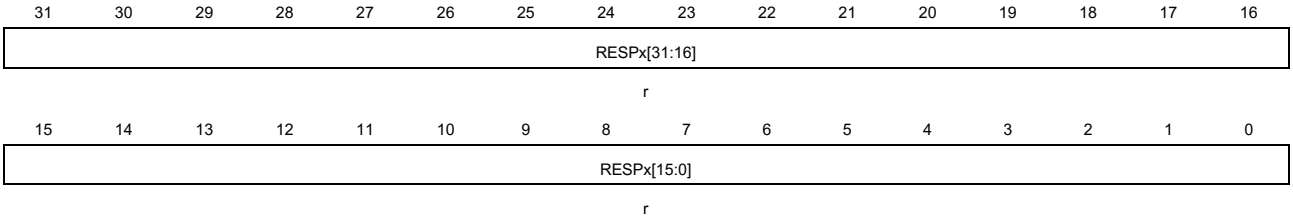
Bits	Fields	Descriptions
31:6	Reserved	Must be kept at reset value.
5:0	RSPCMDIDX[5:0]	Last response command index Read-only bits field. This field contains the command index of the last command response received. If the response doesn't have the command index (long response and short response of R3), the content of this register is undefined.

46.7.6. Response register (SDIO_RESPx x = 0..3)

Address offset: 0x14+(4*x), x = 0..3
Reset value: 0x0000 0000

These register contains the content of the last card response received.

This register has to be accessed by word(32-bit).



Bits	Fields	Descriptions
31:0	RESPx[31:0]	Card state. The content of the response, see ***Table 46-39. SDIO_RESPx register at different response type.***

The short response is 32 bits, the long response is 127 bits (bit 128 is the end bit 0).

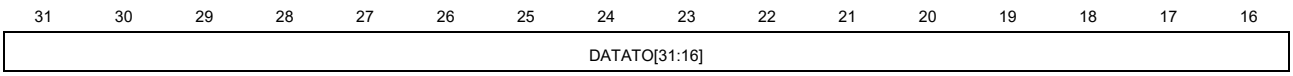
Table 46-39. SDIO_RESPx register at different response type

Register	Short response	Long response
SDIO_RESP0	Card response[31:0]	Card response[127:96]
SDIO_RESP1	reserved	Card response [95:64]
SDIO_RESP2	reserved	Card response [63:32]
SDIO_RESP3	reserved	Card response [31:1],plus bit 0

46.7.7. Data timeout register (SDIO_DATATO)

Address offset: 0x24
Reset value: 0x0000 0000

This register has to be accessed by word(32-bit).



rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATATO[15:0]															

rw

Bits	Fields	Descriptions
31:0	DATATO[31:0]	Data timeout period These bits define the data timeout period count by SDIO_CLK. When the DSM enter the state WaitR or Busy, the internal counter which loads from this register starts decrement. The DSM timeout and enter the state Idle and set the DTTMOUT flag when the counter decreases to 0. These bits can only be written when the (DATSTA = 0) and (CMDSTA = 0).

Note: The data timeout register and the data length register must be updated before being written to the data control register when need a data transfer.

46.7.8. Data length register (SDIO_DATALEN)

Address offset: 0x28

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved							DATALEN[24:16]								

rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATALEN[15:0]															

rw

Bits	Fields	Descriptions
31:25	Reserved	Must be kept at reset value.
24:0	DATALEN[24:0]	Data transfer length This register defined the number of bytes to be transferred. When the data transfer starts, the data counter loads this register and starts decrement. These bits can only be written when the DATSTA = 0. If DRSEL = 1, DATALEN is truncated to a multiple of 2. Indicates that the last data odd byte is not transferred. If DATALEN = 0, no data byte are transferred, when requested by a CSMEN and TREN = 1 also no command is transferred. DATAEN and CSMEN are cleared.

Note: If block data transfer selected, the content of this register must be a multiple of the block size (refer to SDIO_DATACTL). The data timeout register and the data length register must be updated before being written to the data control register when need a data transfer.

For multibyte transfer the value in the data length register must be between 1 and 512.

46.7.9. Data control register (SDIO_DATACTL)

Address offset: 0x2C

Reset value: 0x0000 0000

This register controls the DSM.

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		FIFOREST	ACKEN	IOEN	RWTYPE	RWSTOP	RWEN	BLKSZ[3:0]				TRANSMOD[1:0]		DATADIR	DATAEN
		rw	rw	rw	rw	rw	rw	rw				rw		rw	rw

Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value.
13	FIFOREST	FIFO buffer reset, flush all data This bit can only be configured by firmware when IDMAEN = 0 and DATSTA = 1. And automatically cleared by hardware when DATSTA = 0. 0: No effect 1: Flush any remaining data and reset the FIFO pointers
12	ACKEN	Boot acknowledgment enable bit This bit can only be configured by firmware when DATSTA = 0. 0: Boot acknowledgment disabled 1: Boot acknowledgment enabled
11	IOEN	SD I/O specific function enable(SD I/O only) This bit can only be configured by firmware when DATSTA = 0. 0: Not SD I/O specific function 1: SD I/O specific function
10	RWTYPE	Read wait type(SD I/O only) This bit can only be configured by firmware when DATSTA = 0. 0: Read Wait control using SDIO_DAT[2] 1: Read Wait control by stopping SDIO_CLK
9	RWSTOP	Read wait stop(SD I/O only) This bit can only be configured by firmware. And automatically cleared by hardware when the DSM switches from the DS_Readwait state to the DS_WaitR or DS_Idle state. 0: No effect 1: Stop the read wait process if RWEN bit is set
8	RWEN	Read wait mode enabled(SD I/O only)

0: Read wait mode is disabled
1: Read wait mode is enabled

7:4 BLKSZ[3:0] Data block size
These bits defined the block size when data transfer is block transfer.
These bits can only be written by firmware when DATSTA = 0.
0000: block size = 2^0 = 1 byte
0001: block size = 2^1 = 2 bytes
0010: block size = 2^2 = 4 bytes
0011: block size = 2^3 = 8 bytes
0100: block size = 2^4 = 16 bytes
0101: block size = 2^5 = 32 bytes
0110: block size = 2^6 = 64 bytes
0111: block size = 2^7 = 128 bytes
1000: block size = 2^8 = 256 bytes
1001: block size = 2^9 = 512 bytes
1010: block size = 2^{10} = 1024 bytes
1011: block size = 2^{11} = 2048 bytes
1100: block size = 2^{12} = 4096 bytes
1101: block size = 2^{13} = 8192 bytes
1110: block size = 2^{14} = 16384 bytes
1111: reserved
When DATALEN is not a multiple of BLKSZ, the transferred data is truncated at a multiple of BLKSZ.
If DRSEL = 1, BLKSZ = 0000 must no data are transferred.
If it is in multi block mode and suspend is to be used, the BLKSZ must be set to a size in units of word.

3:2 TRANSMOD[1:0] Data transfer mode
These bits can only be written by firmware when DATSTA = 0.
00: Block count data transfer
01: Multibyte data transfer (only SD/SD I/O)
10: Stream transfer (only *e*•MMC)
11: Transfer ends with CMD12

1 DATADIR Data transfer direction
This bit can only be configured by firmware when DATSTA = 0.
0: Write data to card.
1: Read data from card.

0 DATAEN Data transfer enable bit
This bit is only used to transfer data when the associated data transfer command is not used.
This bit is cleared by hardware when data transfer completes.
0: No effect
1: Start data transfer without CSM

46.7.10. Data counter register (SDIO_DATACNT)

Address offset: 0x30
Reset value: 0x0000 0000

This register is read only. When the DSM from Idle to WaitR or WaitS, it loads value from data length register (SDIO_DATALEN). It decrements with the data transferred, when it reaches 0, the flag DTEND is set.

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved							DATACNT[24:16]								
							r								

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATACNT[15:0]															
r															

Bits	Fields	Descriptions
31:25	Reserved	Must be kept at reset value.
24:0	DATACNT[24:0]	Data count value Read-only bits field. When these bits are read, the number of remaining data bytes to be transferred is returned.

Note: This register should be read only after transmission is completed or held. When reading data after an error event, the read count value may be different from the actual number of data bytes transferred.

46.7.11. Status register (SDIO_STAT)

Address offset: 0x34
Reset value: 0x0000 0000

This register is read only. The following describes the types of flags:

The flags of bit [28:21, 11:0] can only be cleared by writing 1 to the corresponding bit in interrupt clear register (SDIO_INTC).

The flags of bit [20:12] are changing depend on the hardware logic.

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			IDMAEND	IDMAERR	CLKSTOP	VSEND	ACKTO	ACKFAIL	SDIOINT	DAT0BSYEND	DAT0BSY	RFE	TFE	RFF	TFF
			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
RFH	TFH	CMDSTA	DATSTA	DATABOR	DTBLKEND	DATHOLD	DTEND	CMDSEND	CMDRECV	RXORE	TXURE	DTTMOUT	CMDTMOUT	DTCRCERR	CCRCERR
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

Bits	Fields	Descriptions
31:29	Reserved	Must be kept at reset value.
28	IDMAEND	IDMA transfer end
27	IDMAERR	IDMA transfer error
26	CLKSTOP	SDIO_CLK stopped in Voltage Switch procedure
25	VSEND	Voltage Switch critical timing section end
24	ACKTO	Boot acknowledgment timeout
23	ACKFAIL	Boot acknowledgment received and check fail
22	SDIOINT	SD I/O interrupt received
21	DAT0BSYEND	End of CMD response, card DAT0 line signal changed from busy to ready.
20	DAT0BSY	End of CMD response and a second time 2 clock cycles after the CMD response, card signals busy on DAT0 line. This is a hardware status flag only, does not generate an interrupt.
19	RFE	Receive FIFO is empty This is a hardware status flag only, does not generate an interrupt.
18	TFE	Transmit FIFO is empty This bit is cleared when one FIFO location becomes full.
17	RFF	Receive FIFO is full This bit is cleared when one FIFO location becomes empty.
16	TFF	Transmit FIFO is full This is a hardware status flag only, does not generate an interrupt. This bit is cleared when one FIFO location becomes empty.
15	RFH	Receive FIFO is half full: at least half the number of words can be read in the FIFO This bit is cleared when the FIFO becomes (half + 1) empty.
14	TFH	Transmit FIFO is half empty: at least half the number of words can be written into the FIFO. This bit is cleared when the FIFO becomes (half + 1) full.
13	CMDSTA	Command path active state This is a hardware status flag only, does not generate an interrupt.
12	DATSTA	Data path active state This is a hardware status flag only, does not generate an interrupt.
11	DATABOR	Data transfer aborted by CMD12
10	DTBLKEND	Data block sent/received (CRC check passed)

9	DATHOLD	Data transfer hold
8	DTEND	Data end (data counter, SDIO_DATACNT is zero)
7	CMDSEND	Command sent (no response required)
6	CMDRECV	Command response received (CRC check passed)
5	RXORE	Received FIFO overrun error occurs
4	TXURE	Transmit FIFO underrun error occurs
3	DTTMOUT	Data timeout The data timeout period depends on the SDIO_DATATO register.
2	CMDTMOUT	Command response timeout The command timeout period has a fixed value of 64 SDIO_CLK clock periods.
1	DTCRCERR	Data block sent/received (CRC check failed)
0	CCRCERR	Command response received (CRC check failed)

Note: If using IDMA mode, the FIFO Interrupt cannot be enabled.

46.7.12. Interrupt clear register (SDIO_INTC)

Address offset: 0x38
Reset value: 0x0000 0000

This register is write only. Writing 1 to the bit can clear the corresponding bit in the SDIO_STAT register.

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			IDMAENDC	IDMAERRC	CLKSTOPC	VSENDC	ACKTOC	ACKFAILC	SDIOINTC	DAT0BSYENDC	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				DATABORC	DTBLKENDC	DATHOLDC	DTENDC	CMDSENDC	CMDRECVC	RXORECC	TXURECC	DTTMOUTC	CMDTMOUTC	DTCRCERRC	CCRCERRC
				w	w	w	w	w	w	w	w	w	w	w	w

Bits	Fields	Descriptions
31:29	Reserved	Must be kept at reset value.
28	IDMAENDC	IDMAEND flag clear bit Write 1 to this bit to clear the flag.
27	IDMAERRC	IDMAERR flag clear bit Write 1 to this bit to clear the flag.
26	CLKSTOPC	CLKSTOP flag clear bit

		Write 1 to this bit to clear the flag.
25	VSENDC	VSEND flag clear bit Write 1 to this bit to clear the flag.
24	ACKTOC	ACKTO flag clear bit Write 1 to this bit to clear the flag.
23	ACKFAILC	ACKFAIL flag clear bit Write 1 to this bit to clear the flag.
22	SDIOINTC	SDIOINT flag clear bit Write 1 to this bit to clear the flag.
21	DAT0BSYENDC	DAT0BSYEND flag clear bit Write 1 to this bit to clear the flag.
20:12	Reserved	Must be kept at reset value.
11	DATABORC	DATABOR flag clear bit Write 1 to this bit to clear the flag.
10	DTBLKENDC	DTBLKEND flag clear bit Write 1 to this bit to clear the flag.
9	DATHOLDC	DATHOLD flag clear bit Write 1 to this bit to clear the flag.
8	DTENDC	DTEND flag clear bit Write 1 to this bit to clear the flag.
7	CMDSENDC	CMDSEND flag clear bit Write 1 to this bit to clear the flag.
6	CMDRECVC	CMDRECV flag clear bit Write 1 to this bit to clear the flag.
5	RXOREC	RXORE flag clear bit Write 1 to this bit to clear the flag.
4	TXUREC	TXURE flag clear bit Write 1 to this bit to clear the flag.
3	DTTMOUTC	DTTMOUT flag clear bit Write 1 to this bit to clear the flag.
2	CMDTMOUTC	CMDTMOUT flag clear bit Write 1 to this bit to clear the flag.
1	DTCRCERRC	DTCRCERR flag clear bit Write 1 to this bit to clear the flag.
0	CCRCERRC	CCRCERR flag clear bit

Write 1 to this bit to clear the flag.

46.7.13. Interrupt enable register (SDIO_INTEN)

Address offset: 0x3C

Reset value: 0x0000 0000

This register enables the corresponding interrupt in the SDIO_STAT register.

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			IDMAENDIE	IDMAERRIE	CLKSTOPIE	VSENDIE	ACKTOIE	ACKFAILIE	SDIOINTIE	DAT0BSYENDIE	Reserved		TFEIE	RFFIE	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
RFHIE	TFHIE	Reserved		DATABORIE	DTBLKENDIE	DATHOLDIE	DTENDIE	CMDSENDIE	CMDRECVIE	RXOREIE	TXUREIE	DTTMOUTIE	CMDTMOUTIE	DTCRCERRIE	CCRCERRIE
rw	rw			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	IDMAENDIE	IDMA transfer end interrupt enable Write 1 to this bit to enable the interrupt.
27	IDMAERRIE	IDMA transfer error interrupt enable Write 1 to this bit to enable the interrupt.
26	CLKSTOPIE	Voltage Switch clock stopped interrupt enable Write 1 to this bit to enable the interrupt.
25	VSENDIE	Voltage Switch critical timing section end interrupt enable Write 1 to this bit to enable the interrupt.
24	ACKTOIE	Boot acknowledgment timeout enable Write 1 to this bit to enable the interrupt.
23	ACKFAILIE	Boot acknowledgment received and check fail interrupt enable Write 1 to this bit to enable the interrupt.
22	SDIOINTIE	SD I/O interrupt received interrupt enable Write 1 to this bit to enable the interrupt.
21	DAT0BSYENDIE	DAT0 line signal changed from busy to ready interrupt enable. Write 1 to this bit to enable the interrupt.
20:19	Reserved	Must be kept at reset value.
18	TFEIE	Transmit FIFO empty interrupt enable Write 1 to this bit to enable the interrupt.

17	RFFIE	Receive FIFO full interrupt enable Write 1 to this bit to enable the interrupt.
16	Reserved	Must be kept at reset value.
15	RFHIE	Receive FIFO half full interrupt enable Write 1 to this bit to enable the interrupt.
14	TFHIE	Transmit FIFO half empty interrupt enable Write 1 to this bit to enable the interrupt.
13:12	Reserved	Must be kept at reset value.
11	DATABORIE	Data transfer abort interrupt enable Write 1 to this bit to enable the interrupt.
10	DTBLKENDIE	Data block end interrupt enable Write 1 to this bit to enable the interrupt.
9	DATHOLDIE	Data hold interrupt enable Write 1 to this bit to enable the interrupt.
8	DTENDIE	Data end interrupt enable Write 1 to this bit to enable the interrupt.
7	CMDSENDIE	Command sent interrupt enable Write 1 to this bit to enable the interrupt.
6	CMDRECVIE	Command response received interrupt enable Write 1 to this bit to enable the interrupt.
5	RXOREIE	Received FIFO overrun error interrupt enable Write 1 to this bit to enable the interrupt.
4	TXUREIE	Transmit FIFO underrun error interrupt enable Write 1 to this bit to enable the interrupt.
3	DTTMOUTIE	Data timeout interrupt enable Write 1 to this bit to enable the interrupt.
2	CMDTMOUTIE	Command response timeout interrupt enable Write 1 to this bit to enable the interrupt.
1	DTCRCERRIE	Data CRC fail interrupt enable Write 1 to this bit to enable the interrupt.
0	CCRCERRIE	Command response CRC fail interrupt enable Write 1 to this bit to enable the interrupt.

46.7.14. ACK timeout register (SDIO_ACKTO)

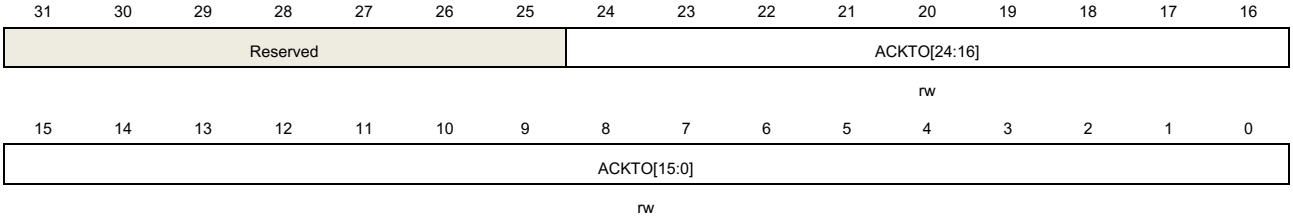
Address offset: 0x40

Reset value: 0x0000 0000

When DSM enters the WaitACK state, the SDIO_ACKTO register value will loaded into a counter. It is used to answer whether the acknowledgment timeout occurs. If the counter reaches 0 and the DSM is in this state, the ACK timeout status flag is set to 1.

The data transfer must be written to the SDIO_ACKTO register before being written to the SDIO_DATACTL register.

This register has to be accessed by word(32-bit).



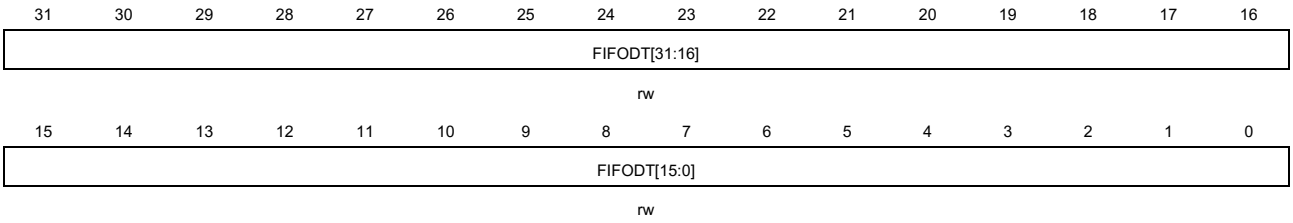
Bits	Fields	Descriptions
31:25	Reserved	Must be kept at reset value.
24:0	ACKTO[24:0]	Boot ACK timeout period. These bits can only be configured by firmware when CSMEN = 0.

46.7.15. FIFO data register (SDIO_FIFO)

Address offset: 0x80
Reset value: 0x0000 0000

This register occupies 16 entries of 32-bit words, the address offset is from 0x80 to 0xBC.

This register has to be accessed by word(32-bit).



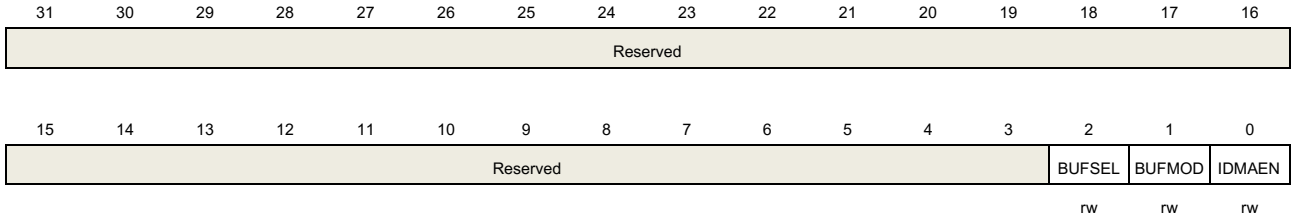
Bits	Fields	Descriptions
31:0	FIFODT[31:0]	Receive FIFO data or transmit FIFO data These bits are the data of receive FIFO or transmit FIFO. Write to or read from this register is write data to FIFO or read data from FIFO. This register can only be read or written by firmware when DATSTA = 1.

46.7.16. IDMA control register (SDIO_IDMACTL)

Address offset: 0x50

Reset value: 0x0000 0000

The transmit and receive FIFO can be written or read as a 32-bit wide register. The FIFO have 32 consecutive addresses as 32 entries. This allows the CPU to use its store/load multiple operator commands to write/read FIFO.



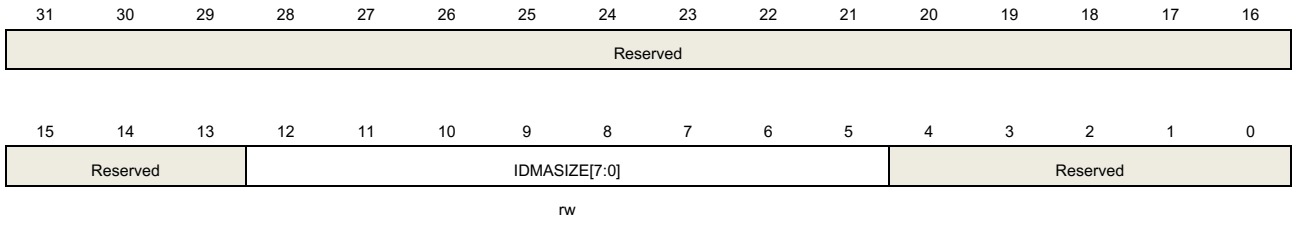
Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value.
2	BUFSEL	IDMA double buffer selection bit This bit can only be configured by firmware when DATSTA = 0. When IDMA is enabled, this bit is toggled by hardware. 0: If IDMA is enabled, uses buffer0, and firmware write access to IDMAADDR0 is not allow. 1: If IDMA is enabled, uses buffer1, and firmware write access to IDMAADDR1 is not allow.
1	BUFMOD	Double buffer enable bit This bit can only be configured by firmware when DATSTA = 0. 0: Single buffer mode. 1: Double buffer mode.
0	IDMAEN	FIFO internal DMA enable bit This bit can only be configured by firmware when DATSTA = 0. 0: IDMA disable. 1: IDMA enable.

46.7.17. IDMA buffer size register (SDIO_IDMASIZE)

Address offset: 0x54

Reset value: 0x0000 0000

This register contains the buffers size when in double buffer configuration.



Bits	Fields	Descriptions

31:13	Reserved	Must be kept at reset value.
12:5	IDMASIZE[7:0]	Number of bytes per buffer. IDMA buffer size = IDMASIZE[7:0] * 8 words. 0x00: buffer size = 0 words 0x01: buffer size = 8 words 0x02: buffer size = 16 words …… 0xFF: buffer size = 2040 words
4:0	Reserved	Must be kept at reset value.

46.7.18. IDMA buffer 0 base address register (SDIO_IDMAADDR0)

Address offset: 0x58
Reset value: 0x0000 0000

This register contains the base address of memory buffer in single buffer configuration and the base address of buffer 0 in double buffer configuration.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IDMAADDR0[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDMAADDR0[15:0]															
rw															

Bits	Fields	Descriptions
31:0	IDMAADDR0[31:0]	The address is a multiple of 4. The IDMAADDR0[1:0] are always 0 and read only. This register can be written by firmware when DATSTA = 0, and can dynamically be written by firmware when DATSTA and BUFSEL all set to 1 .

46.7.19. IDMA buffer 1 base address register (SDIO_IDMAADDR1)

Address offset: 0x5C
Reset value: 0x0000 0000

This register contains the base address of buffer 1 in double buffer configuration.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IDMAADDR1[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDMAADDR1[15:0]															
rw															

Bits	Fields	Descriptions
31:0	IDMAADDR1[31:0]	The address is a multiple of 4. The IDMAADDR1[1:0] are always 0 and read only.

This register can be written by firmware when DATSTA = 0, and can dynamically be written by firmware when DATSTA = 1 and BUFSEL = 0.

47. Management data input / output (MDIO)

47.1. Overview

The MDIO interface can receive complete MDIO frames. As long as the data is written to the register before receiving the turnaround bits (TA) of the read or post read increment address frame, the MDIO interface can transmit complete MDIO frames. Interrupts are generated at the end of every complete frame, which can be used or provided at correct time. Interrupts can also be generated after every valid PHYADR and DEVADD, which allows more complex controls within frames.

47.2. Characteristics

- Support slave mode with a frequency up to 4MHz.
- Support CFP / CFP2 MSA Management Interface Specification.
- Support various kinds of interrupts.
- Physical address can be setted:
  - By software;
  - By hardware pin.

47.3. Pins and internal signals

Figure 47-1. CFP Management Interface Architecture shows the MDIO block diagram. *Table 47-1. MDIO pins definition* gives the MDIO internal signals and pins description.

Table 47-1. MDIO pins definition

Name	Signal type
MDIO_MDIO	Input / Output, Digital signal
MDIO_MDC	Input, Digital clock signal
MDIO_Ax(x=0…4)	Input, Digital address signals

47.4. Function overview

Figure 47-1. CFP Management Interface Architecture

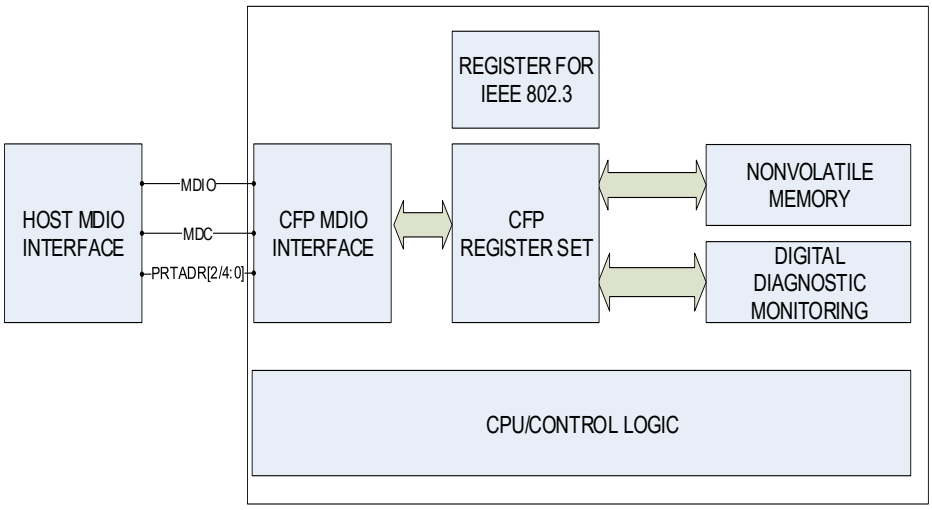
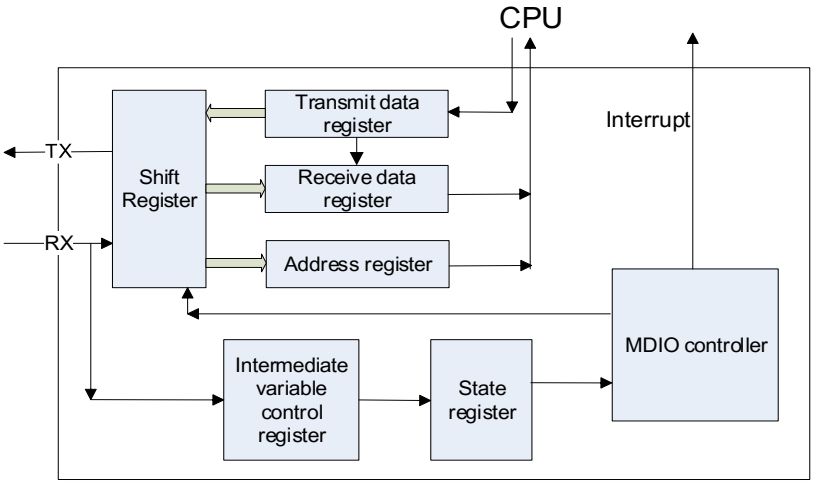


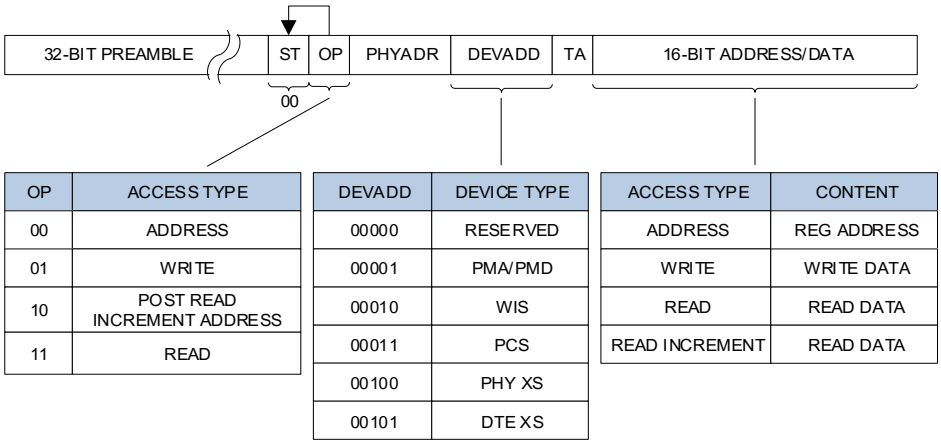
Figure 47-2. MDIO Architecture



47.4.1. Frame structure

CFP MDIO interface uses the communication data frame structure defined in IEEE 802.3 Clause 45. Each frame can be either an address frame or a data frame. The total bit length of each frame is 64, consisting of 32 bits preamble, and the frame command body.The command body consists of 6 parts illustrated in ***Figure 47-3. MDIO Frame Structure***.

Figure 47-3. MDIO Frame Structure



OP	ACCESS TYPE
00	ADDRESS
01	WRITE
10	POST READ INCREMENT ADDRESS
11	READ

DEVADD	DEVICE TYPE
00000	RESERVED
00001	PMA/PMD
00010	WIS
00011	PCS
00100	PHY XS
00101	DTE XS

ACCESS TYPE	CONTENT
ADDRESS	REG ADDRESS
WRITE	WRITE DATA
READ	READ DATA
READ INCREMENT	READ DATA

Note:

1. ST = Start bits (2 bits)
2. OP = Operation code (2 bits)
3. PHYADR = Physical port Address (5 bits)
4. DEVADD = MDIO Device Address (or called Device Type, 5 bits)
5. TA = Turnaround bits (2 bits)
6. 16-BIT ADDRESS/DATA is the payload

Table 47-2. Frame Details for Different Frame Types (1)

Frame	Idle	Management Frame Fields							Idle
		PRE	ST	OP	PHYADR	DEVADD	TA	Address/Data	
Write Address	Z	1…1	00	00	aaaaa	aaaaa	10	aaaaaaaaaaaaaaaa	Z
Write Data	Z	1…1	00	01	aaaaa	aaaaa	10	dddddddddddddddd	Z
Read Data	Z	1…1	00	11	aaaaa	aaaaa	z0	dddddddddddddddd	Z
Post Read Increment Address Frame	Z	1…1	00	10	aaaaa	aaaaa	z0	dddddddddddddddd	Z

(1). During the idle condition, the MDIO clock (MDC) and MDIO are not actively driven. For a read or a post read increment address frame, the TA bits is 1.5 bits, and the ahead 0.5 bit cycle of TA is driven by the host, the other one bit cycle of TA and during the 16-bit data time, MDIO is driven by the MDIO manageable device (slave), then another 0.5 bit cycle of TA is followed, MDIO is driven by the host. At all other times, the station management entity (host) drives the MDIO.

Idle Condition (Idle)

The idle condition for the MDIO is a high impedance state.

Preamble (PRE)

At the beginning of each transaction, the station management entity (host) sends a sequence of at least 32 contiguous bits of bit '1' at a time to the MDIO, with 32 corresponding clock cycles on the MDIO clock (MDC), to establish the start of a frame.

Start of Frame (ST)

After PRE, the ST (consisting of two zero bits) indicates the start of the frame information.

Operation Code (OP)

The OP specifies the action to take, as described in *Table 47-3. Operation Code*.

Table 47-3. Operation Code

OP	Descriptions
00	Set the address for a subsequent write or read frame.
01	Write to the previously set address.
11	Read from the previously set address.
10	Read from the previously set address, and then increment the address. User code must increment the address.

Physical Address (PHYADR)

The physical address is five bits, allowing 32 unique addresses. PHYADR is set either by five pins or by the software.

Device Address (DEVADD)

DEVADD is five bits and selects the device type. In the CFP standard, only MDIO Device Address 1 is supported.

Turnaround (TA)

The MDIO changes from the station management entity (host) to MDIO manageable device (slave) during the TA time.

Address / Data

The address / data field is 16 bits.

47.4.2. Typical Usage Sequence

Most of the MDIO interface is implemented in the hardware, requiring correct software operation. The following is the usage flow:

1. Reset the MDIO and configure the GPIO module to make selected PADs to alternate function.
2. Set the frame parameters by writing the MDIO_CTL, MDIO_CFG and MDIO_TO registers as you need.
3. Set the interrupts by writing the MDIO_INTEN register and the required system interrupt settings.
4. For a write frame, the write address and write data frames can be received in the MDIO_RADDR and MDIO_RDATA registers after receiving complete frame. For a read data frame or post read increment address frame, data must be placed in the MDIO_TDATA register in advance of the frame so that the data can be automatically

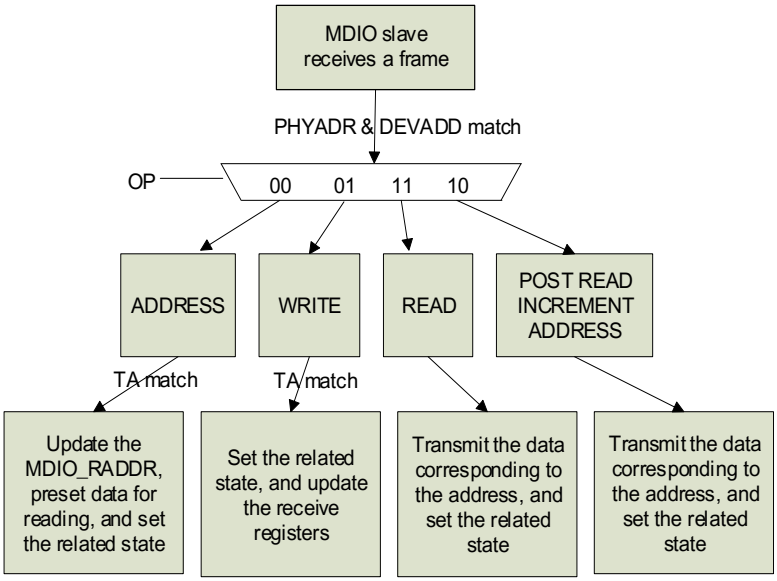
inserted for the frame.

In this process, no software intervention is required. The frame progress can be monitored with the MDIO_RFRM register when frame field match or unmatch event occur during each frame or at the completion of each frame. Due to bit [9:0] of MDIO_STAT register is automatically cleared, do not use MDIO_STAT register to monitor frame progress, and the MDIO_STAT register can be lost if read at an inappropriate time. Read the MDIO_STAT register only once per frame. To monitor frame progress, read the status in MDIO_STAT register according to interrupts or by polling the related bit 8 (interrupt num 40) in SETPEND1 register (in NVIC registers). MDIO interrupts must have the highest priority of all peripherals, otherwise they are easy to lose.

Notes:

1. The system clock frequency should be more than 3 times that of the MDC clock.
2. Be sure to send data to MDIO_TDATA register before TA.
3. Please configure the software reset before use if timeout or other error occurs.

Figure 47-4. MDIO slave communication process



After receiving the address frame sent by the host, no matter whether the next frame is write or read, the data can be preset to the transfer data register in advance, thus if the next frame is read, MDIO will send the preset data. If the address frame is received but no read operation is performed following, the new data can be overwritten to the original transfer data register when the next address frame is accepted.

47.5. Register definition

MDIO base address: 0x4000 9400

47.5.1. Control register (MDIO_CTL)

Address offset: 0x00
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
Reserved													ROVEN	PHYB	SWRST
													rw	rw	w

Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value.
2	ROVEN	Receive overrun enable 0: enable overwriting when receive data is full. When receiving data, if an overflow occurs, the data in MDIO_RDATA is overwritten by the new incoming data. 1: disable overwriting when receive data is full. When receiving data, if an overflow occurs, the data in MDIO_RDATA is discarded, and the data in MDIO_RDATA remains unchanged and is not overwritten.
1	PHYB	MDIO PHY bit length 0: MDIO PHY uses 5 bits 1: MDIO PHY uses 3 bits. Unused PHY bits are ignored
0	SWRST	Write 1 to reset MDIO block. Registers will not be reset. Hardware immediately clears this bit.

47.5.2. Received frame information register (MDIO_RFRM)

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		RTA[1:0]		RDEV[4:0]					RPHY[4:0]					ROP[1:0]	

r r r r

Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value.
13:12	RTA[1:0]	Received frame field TA (only accept TA bits of a write data frame or a write address frame)
11:7	RDEV[4:0]	Received frame field DEVADD
6:2	RPHY[4:0]	Received frame field PHYADR
1:0	ROP[1:0]	Received frame field OP 00: write address frame 01: write data frame 10: post read increment address frame 11: read data frame

47.5.3. Received data register (MDIO_RDATA)

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
RDATA[15:0]															

r

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	RDATA[15:0]	Received frame field DATA

47.5.4. Received address register (MDIO_RADDR)

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
RADDR[15:0]															

r

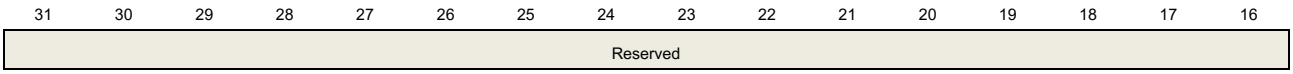
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	RADDR[15:0]	Received frame field ADDRESS

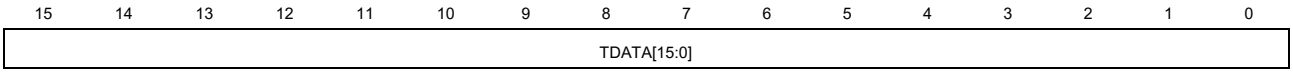
47.5.5. Transfer data register (MDIO_TDATA)

Address offset: 0x10

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).





rw

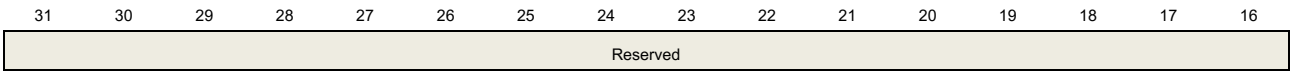
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	TDATA[15:0]	Data that is transmitted by the next read or post read increment address frame. Before a read data frame, the master sends a write address frame to specify which data is to be read. After this write address frame, the user software must place this requested data into MDIO_TDATA before it is required by the read frame. The time available is at latest 3 MDIO clock cycles before TA.

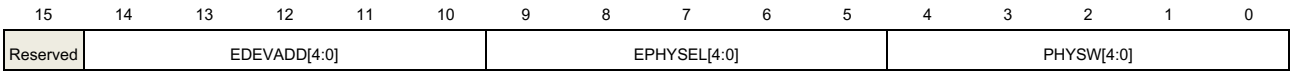
47.5.6. Configuration register (MDIO_CFG)

Address offset: 0x14

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).





rw rw rw

Bits	Fields	Descriptions
31:15	Reserved	Must be kept at reset value.
14:10	EDEVADD[4:0]	Expected DEVADD. Normally 01.

9:5	EPHYSEL[4:0]	Selects expected PHYADR. For each bit x in the 5-bit EPHYSEL: 0: sets expected PHYADR.x = PHYPIN[4:0].x 1: sets expected PHYADR.x = PHYSW[4:0].x
4:0	PHYSW[4:0]	Software provided PHYADR. Chosen according to corresponding EPHYSEL[4:0] bits.

47.5.7. Status register (MDIO_STAT)

Address offset: 0x18

Reset value: 0x0000 1000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		RBNE	Reserved	OVR	UDR	TO	TANM	PHYNM	PHYM	DEVNM	DEVM	RDFRM	RDINCFRM	ADDRFRM	WRFRM
		r		r	r	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc

Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value.
13	RBNE	Read data buffer not empty flag. Data is received and ready to be read. Cleared by reading the MDIO_RDATA register.
12	Reserved	Must be kept at reset value.
11	OVR	Receive overrun flag. Set by hardware when the data in the receive shift register is ready to be transferred into the MDIO_RDATA register while the RBNE bit is set. Cleared by reading the MDIO_RDATA register.
10	UDR	Transmit underrun flag. Set by hardware if the data has not been written to MDIO_TDATA register 3 MDIO clock (APB clock) cycles before TA of a read / post read increment address frame when the MDIO_TDATA is empty. Cleared by a reading to the MDIO_STAT register.
9	TO	Timeout flag. If a new bit of a frame is not received or a bit of the prepared data is not transmitted before reaching the configured timeout value, this bit will be set by hardware. Cleared by reading the MDIO_STAT register.
8	TANM	Set at end bit of TA of a write data frame or a write address frame if the received TA nonmatches expected '10'. Cleared by reading the MDIO_STAT register.
7	PHYNM	Set at end bit of PHYADR if PHYADR nonmatches. Cleared by reading the MDIO_STAT register.

6	PHYM	Set at end bit of PHYADR if PHYADR matches. Cleared by reading the MDIO_STAT register.
5	DEVNM	Set at end bit of DEVADD if DEVADD nonmatches. Cleared by reading the MDIO_STAT register.
4	DEVM	Set at end bit of DEVADD if DEVADD matches. Cleared by reading the MDIO_STAT register.
3	RDFRM	Set at end bit (after frame field DATA transmission) of a read data frame if DEVADD and PHYADR both match. Cleared by reading the MDIO_STAT register.
2	RDINCFRM	Set at end bit (after frame field DATA transmission) of a post read increment address frame if DEVADD and PHYADR both match. Cleared by reading MDIO_STAT register.
1	ADDRFRM	Set at end bit of a write address frame if DEVADD and PHYADR both match. Cleared by reading the MDIO_STAT register.
0	WRFRM	Set at end bit of a write data frame if DEVADD and PHYADR both match. Cleared by reading the MDIO_STAT register.

47.5.8. Interrupt enable register (MDIO_INTEN)

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	120	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		RBNEIE	Reserved	OVRIE	UDRIE	TOIE	TANMIE	PHYNMIE	PHYMIE	DEVNMIE	DEVMIE	RDFRMIE	RDINCFRMIE	ADDRFRMIE	WRFRMIE
		rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

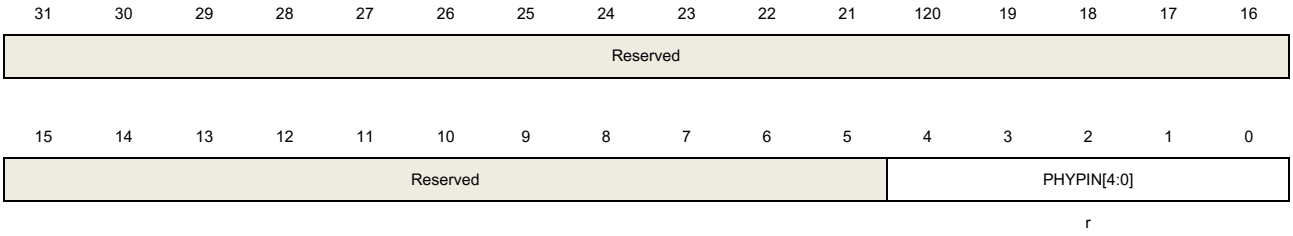
Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value.
13	RBNEIE	If set, interrupt is requested when RBDE bit in MDIO_STAT register becomes active.
12	Reserved	Must be kept at reset value.
11	OVRIE	If set, interrupt is requested when OVR bit in MDIO_STAT register becomes active.
10	UDRIE	If set, interrupt is requested when UDR bit in MDIO_STAT register becomes active.
9	TOIE	If set, interrupt is requested when TO bit in MDIO_STAT register becomes active.
8	TANMIE	If set, interrupt is requested when TANM bit in MDIO_STAT register becomes

		active.
7	PHYNMIE	If set, interrupt is requested when PHYNM bit in MDIO_STAT register becomes active.
6	PHYMIE	If set, interrupt is requested when PHYM bit in MDIO_STAT register becomes active.
5	DEVNMIE	If set, interrupt is requested when DEVNM bit in MDIO_STAT register becomes active.
4	DEVMIE	If set, interrupt is requested when DEVM bit in MDIO_STAT register becomes active.
3	RDFRMIE	If set, interrupt is requested when RDFRM bit in MDIO_STAT register becomes active.
2	RDINCFRMIE	If set, interrupt is requested when RDINCFRM bit in MDIO_STAT register becomes active.
1	ADDRFRMIE	If set, interrupt is requested when ADDRFRM bit in MDIO_STAT register becomes active.
0	WRFRMIE	If set, interrupt is requested when WRFRM bit in MDIO_STAT register becomes active.

47.5.9. Pin value register (MDIO_PIN)

Address offset: 0x20
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:5	Reserved	Must be kept at reset value.
4:0	PHYPIN[4:0]	Pin value read from hardware MDIO_Ax(x=0…3) pins

47.5.10. Timeout register (MDIO_TO)

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	120	19	18	17	16
Reserved															TOCNT[15]
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOCNT[14:0]															TOEN
rw															rw

Bits	Fields	Descriptions
31:17	Reserved	Must be kept at reset value.
16:1	TOCNT[15:0]	MDIO time out = TOCNT[15:0]*PCLK1
0	TOEN	Timeout enable 0: Timeout is disabled 1: Timeout is enabled

48. External memory controller (EXMC)

48.1. Overview

The external memory controller EXMC, is used as a translator for CPU to access a variety of external memory, it automatically converts AXI memory access protocol into a specific memory access protocol defined in the configuration register, such as SRAM, ROM, NOR Flash, PSRAM, NAND Flash and SDRAM. Users could also tweak with the timing parameters in the configuration register to boost up 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 controlling register.

48.2. Characteristics

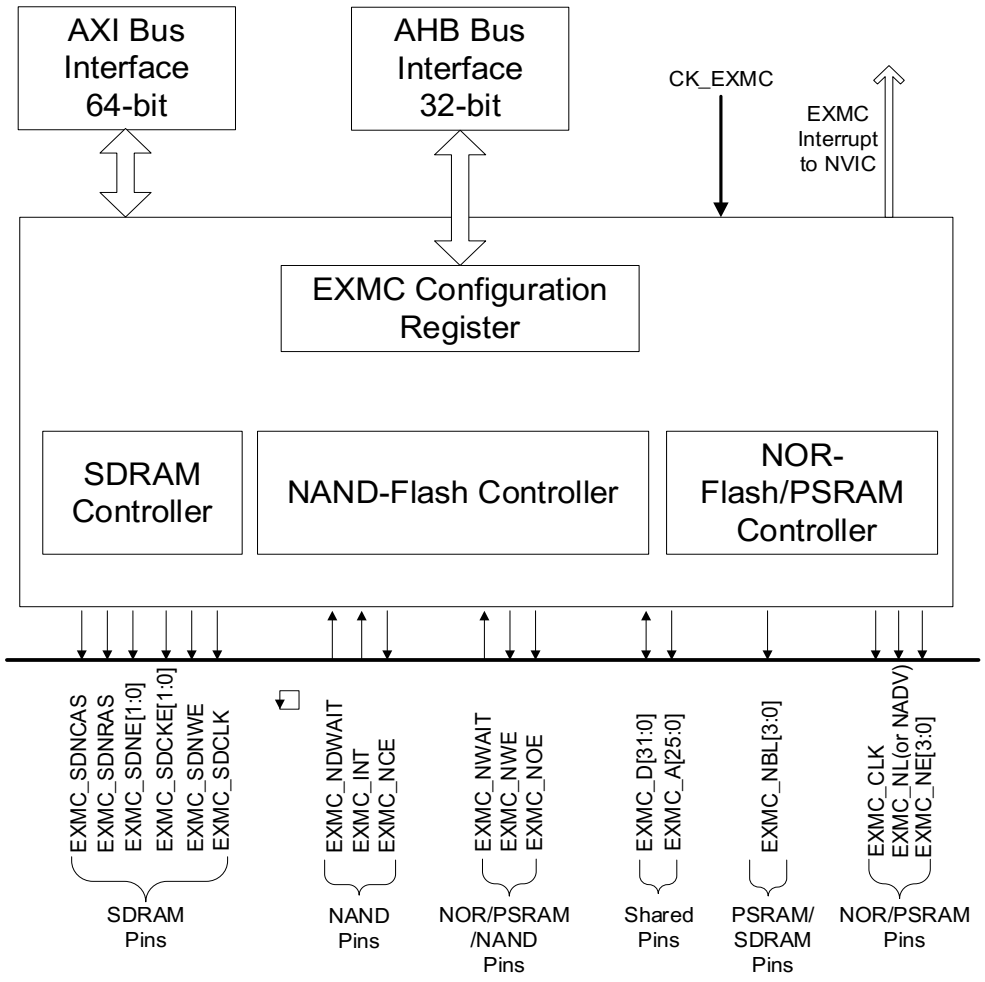
- Supported external memory:
  - SRAM
  - PSRAM
  - ROM
  - NOR Flash
  - 8-bit or 16-bit NAND Flash
  - Synchronous DRAM(SDRAM)
- Protocol translation between the AXI and the multitude of external memory protocol.
- Offering a variety of programmable timing parameters to meet user's specific needs.
- Each bank has its own chip-select signal which can be configured independently.
- Independent read / write timing configuration to a sub-set memory type.
- Embedded ECC hardware for NAND Flash access.
- 8,16, or 32 bits bus width.
- Address and data bus multiplexing mechanism for NOR Flash and PSRAM.
- Write enable and byte select are provided as needed.
- Automatic AXI transaction split when internal and external bus width is not compatible.

48.3. Function overview

48.3.1. Block diagram

EXMC is the combination of 7 modules: The AHB bus interface, AXI bus interface, EXMC configuration registers, NOR/PSRAM controller, NAND controller, SDRAM controller and external device interface. AHB clock (HCLK) is the reference clock, which is used to configure the EXMC registers.

Figure 48-1. The EXMC block diagram



48.3.2. Bus interface

AHB bus interface: The CPU configures the EXMC register through the AHB slave interface.

AXI bus interface: CPU and AXI bus master device access external memory through AXI bus slave interface.

The clock of NOR, NAND, SDRAM controller is the asynchronous CK_EXMC (refer to EXMCSEL bits in ***Clock configuration register 4 (RCU_CFG4)***)

48.3.3. AXI error

Accessing EXMC bank region x (x = 0,.., 3) which is not enabled will generate AXI slave error.

If NREN bit in EXMC_SNCTLx (x = 0,.., 3) register is set to 0, accessing EXMC NOR Flash memory area will generate AXI slave error.

For the write operation of SDRAM device (WPEN is set to 1) that has been write protected, will generate AXI slave error.

Access SDRAM when it is not initialized.

Accessing EXMC NAND reserved area will generate AXI slave error.

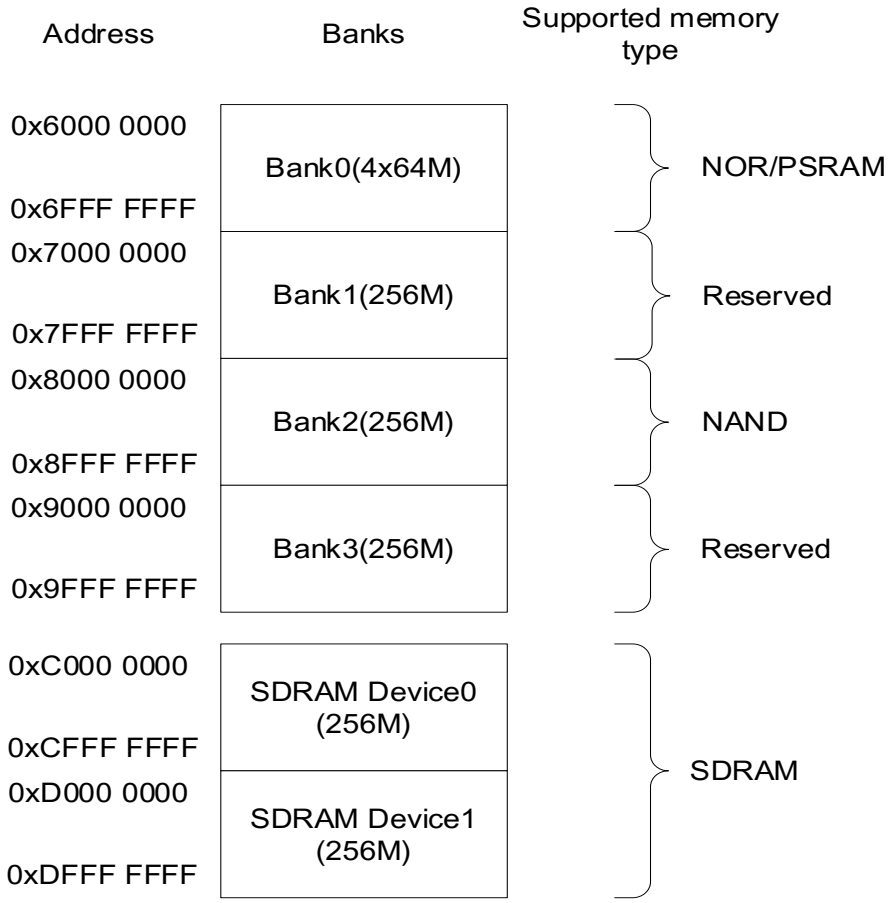
48.3.4. Basic regulation of EXMC access

EXMC is the conversion interface between AXI bus and external device protocol. Since the bit width of the AXI data bus is 64 bits, AXI transactions will split an access into several consecutive 8-bit, 16-bit or 32-bit accesses according to the size of the data. In the process of data transfer, AXI transaction data width and memory data width may not be the same. In order to ensure consistency of data transmission, EXMC's read / write accesses follow the following basic regulation.

- When the width of AXI transaction data width equals to the memory bus width, no conversion is applied.
- When the width of AXI transaction data width is greater than memory bus width, the AXI accesses will automatically split into several continuous memory accesses.
- When the width of AXI transaction data width is smaller than memory bus width, if the external memory devices have the byte selection function, such as SRAM, ROM, PSRAM, SDRAM, the application can access the corresponding byte through their byte lane EXMC_NBL[3:0]. Otherwise, write operation is prohibited, but read operation is allowed unconditionally. (See ***Table 48-17. Bank2 of EXMC support the memory and access mode***)

48.3.5. External device address mapping

Figure 48-2. EXMC memory banks



EXMC access space is divided into multiple banks. Each bank is 256 Mbytes. The first bank (Bank0) is further divided into four regions, and each region is 64 Mbytes. Bank2 is each divided into two spaces, the attribute memory space and the common memory space.

Each bank or region has a separate chip-select control signal, which can be configured independently.

Bank0 is used for NOR and PSRAM device access.

Bank2 are used to access NAND Flash exclusively.

SDRAM Device0 and Device1 are used for Synchronous DRAM (SDRAM) access.

The EXMC bank mapping can be modified through the BKREMAP[1:0] bis in the EXMC_SNCTL register. The EXMC bank mapping is shown in *Table 48-1. EXMC bank mapping*.

Table 48-1. EXMC bank mapping

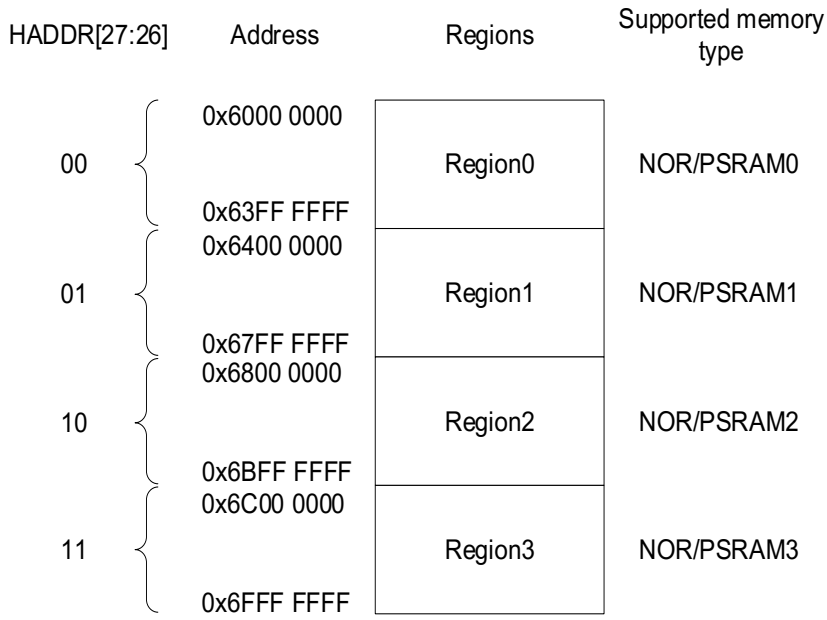
Address	BKREMAP[1:0]=00	BKREMAP[1:0]=01
0x6000 0000 - 0x6FFF FFFF	NOR/PSRAM bank	SDRAM Device 0

Address	BKREMAP[1:0]=00	BKREMAP[1:0]=01
0x7000 0000 - 0x7FFF FFFF	Reserved	
0x8000 0000 - 0x8FFF FFFF	NAND bank	
0x9000 0000 - 0x9FFF FFFF	Reserved	
0xC000 0000 - 0xCFFF FFFF	SDRAM Device 0	NOR/PSRAM bank
0xD000 0000 - 0xDFFF FFFF	SDRAM Device 1	

NOR/PSRAM address mapping

Figure 48-3. Four regions of bank0 address mapping reflects the address mapping of the four regions of bank0. Internal address lines HADDR[27:26] bit are used to select the four regions.

Figure 48-3. 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.

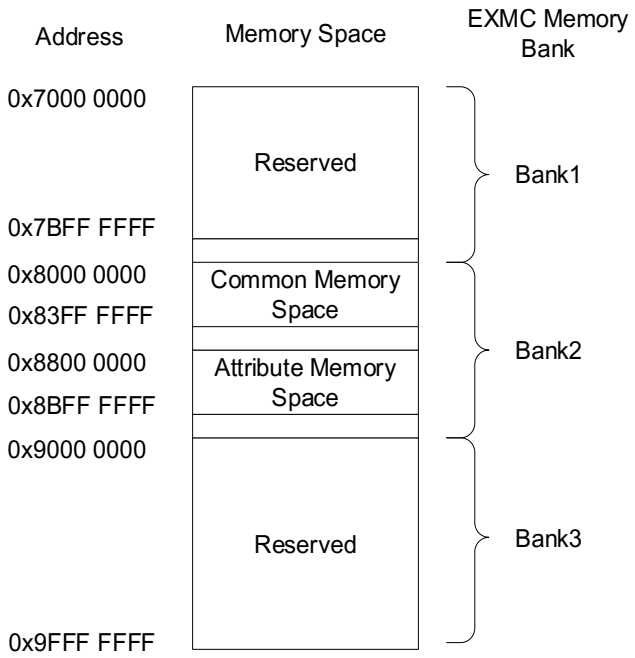
- When data bus width of the external memory is 8-bits. In this case the memory address is byte aligned. HADDR[25:0] is connected to EXMC_A[25:0] and then the EXMC_A[25:0] is connected to the external memory address lines.
- When data bus width of the external memory is 16-bits. In this case the memory 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.
- When data bus width of the external memory is 32-bits. In this case the memory address is word aligned. HADDR byte address must be converted into word aligned by

connecting HADDR[25:2] with EXMC_A[23:0]. The EXMC_A[23:0] is connected to the external memory address lines.

NAND address mapping

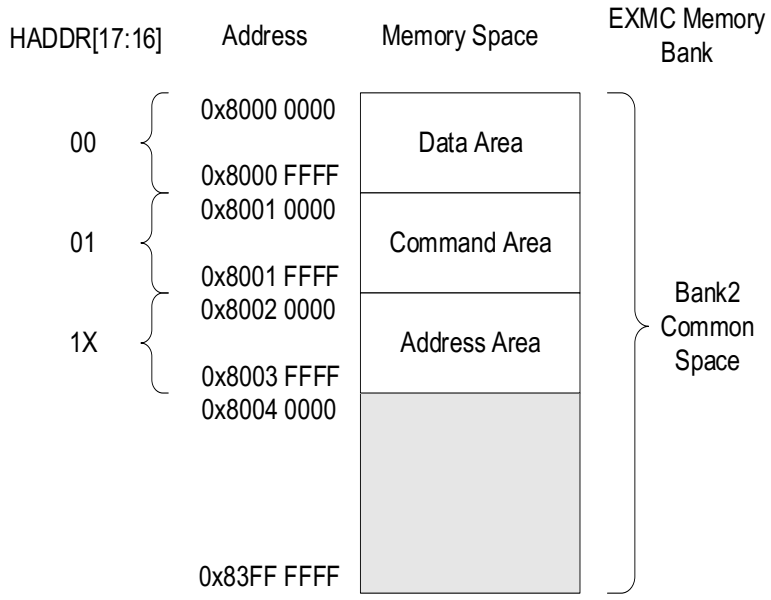
Bank2 is designed to access NAND Flash, bank1 and bank3 are reserved. Each bank is further divided into several memory spaces as shown in ***Figure 48-4. NAND address mapping***.

Figure 48-4. NAND address mapping



For NAND Flash, the common space and the attribute space are further-divided into three areas individually, the data area, the command area and the address area as shown in ***Figure 48-5. Diagram of bank2 common space***.

Figure 48-5. Diagram of bank2 common space



HADDR[17:16] bits are used to select one of the three areas.

- When HADDR[17:16] = 00, the data area is selected.
- When HADDR[17:16] = 01, the command area is selected.
- When HADDR[17:16] = 1X, the address area is selected.

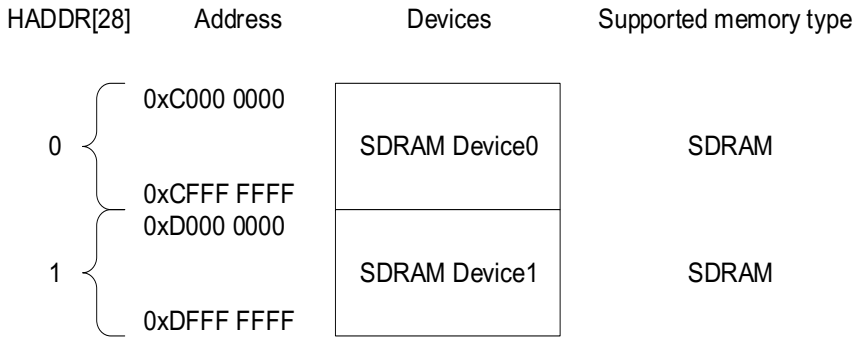
Application software uses these three areas to access NAND Flash, their definitions are as follows.

- Address area: This area is where the NAND Flash access address should be issued by software, the EXMC will pull the address latch enable (ALE) signal automatically in address transfer phase. ALE is mapped to EXMC_A[17].
- Command area: This area is where the NAND Flash access command should be issued by the software, the EXMC will pull the command latch enable (CLE) signal automatically in command transfer phase. CLE is mapped to EXMC_A[16].
- Data area: This area is where the NAND Flash read / write data should be accessed. When the EXMC is in data transfer mode, software should write the data to be transferred to the NAND Flash in this area. When the EXMC is in data reception mode, software should read the data from the NAND Flash by reading this area. Data access address is incremented automatically in consecutive mode, users do not need to be concerned with access address.

SDRAM address mapping

The HADDR[28] bit is used to choose one of the two memory banks as shown in ***Figure 48-6. SDRAM address mapping***.

Figure 48-6. SDRAM address mapping



The *Table 48-2. SDRAM mapping* shows SDRAM address mapping of a 13-bit row and an 11-bit column configuration.

Table 48-2. SDRAM mapping

Memory width	Internal bank	Row address	Column address	Maximum memory capacity
8-bit	HADDR[25:24]	HADDR[23:11]	HADDR[10:0]	64 Mbytes: 4 x 8K x 2K
16-bit	HADDR[26:25]	HADDR[24:12]	HADDR[11:1]	128 Mbytes: 4 x 8K x 2K x 2
32-bit	HADDR[27:26]	HADDR[25:13]	HADDR[12:2]	256 Mbytes: 4 x 8K x 2K x 4

48.3.6. 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, and owns its corresponding unique register.

Note:

In asynchronous mode, all output signals of controller will change on the rise edge of internal CK_EXMC.

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 48-3. NOR flash interface signals description

EXMC Pin	Direction	Mode	Functional description
EXMC_CLK	Output	Sync	Clock signal for sync

EXMC Pin	Direction	Mode	Functional description
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 48-4. 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 48-5. EXMC bank 0 supports all transactions shows an example of the supported devices type, access modes and transactions when the memory data bus is 16-bit for NOR, PSRAM and SRAM.

Table 48-5. EXMC bank 0 supports all transactions

Memory	Access Mode	R/W	AXI 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	

Memory	Access Mode	R/W	AXI Transaction Size	Memory Transaction Size	Comments
	Async	R	32	16	Split into 2 EXMC accesses
	Async	W	32	16	Split into 2 EXMC accesses
	Async	R	64	16	Split into 4 EXMC accesses
	Async	W	64	16	Split into 4 EXMC accesses
	Sync	R	8	16	Not allowed
	Sync	R	16	16	
	Sync	R	32	16	
	Sync	R	64	16	
PSRAM	Async	R	8	16	
	Async	W	8	16	Use of byte lanes EXMC_NBL[1:0]
	Async	R	16	16	
	Async	W	16	16	
	Async	R	32	16	Split into 2 EXMC accesses
	Async	W	32	16	Split into 2 EXMC accesses
	Async	R	64	16	Split into 4 EXMC accesses
	Async	W	64	16	Split into 4 EXMC accesses
	Sync	R	8	16	Not allowed
	Sync	R	16	16	
	Sync	R	32	16	
	Sync	R	64	16	
	Sync	W	8	16	Use of byte lanes EXMC_NBL[1:0]
	Sync	W	16	16	
	Sync	W	32	16	
	Sync	W	64	16	
SRAM and ROM	Async	R	8	16	
	Async	R	16	16	
	Async	R	32	16	Split into 2 EXMC accesses
	Async	R	64	16	Split into 4 EXMC accesses

Memory	Access Mode	R/W	AXI Transaction Size	Memory Transaction Size	Comments
	Async	W	8	16	Use of byte lanes EXMC_NBL[1:0]
	Async	W	16	16	
	Async	W	32	16	Use of byte lanes EXMC_NBL[1:0]
	Async	W	64	16	Use of byte lanes EXMC_NBL[1:0]

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 48-6. NOR / PSRAM controller timing parameters

Parameter	Function	Access mode	Unit	Min	Max
CKDIV	Sync Clock divide ratio	Sync	CK_EXMC	2	16
DLAT	Data latency	Sync	EXMC_CLK	2	17
BUSLAT	Bus latency	Async	CK_EXMC	0	15
DSET	Data setup time	Async	CK_EXMC	1	255
DHLD	Data hold time	Async	CK_EXMC	0	3
AHLD	Address hold time	Async(muxed)	CK_EXMC	1	15
ASET	Address setup time	Async	CK_EXMC	0	15

Table 48-7. EXMC timing models

Timing model		Extend mode	Mode description	Write timing parameter	Read timing parameter
Async	Mode 1	0	SRAM/PSRAM/CRAM	DSET DHLD ASET	DSET DHLD ASET
	Mode 2	0	NOR Flash	DSET DHLD ASET	DSET DHLD ASET
	Mode A	1	SRAM/PSRAM/CRAM with EXMC_OE toggling on data phase	WDSET WDHLD WASET	DSET DHLD ASET
	Mode B	1	NOR Flash	WDSET WDHLD WASET	DSET DHLD ASET
	Mode C	1	NOR Flash with EXMC_OE toggling on data phase	WDSET WDHLD WASET	DSET DHLD ASET

Timing model		Extend mode	Mode description	Write timing parameter	Read timing parameter
	Mode D	1	With address hold capability	WDSET WDHLD WAHLD WASET	DSET DHLD AHLD ASET
	Mode AM	0	NOR Flash address/data mux	DSET AHLD ASET BUSLAT	DSET AHLD ASET BUSLAT
Sync	Mode E	0	NOR/PSRAM/CRAM synchronous read PSRAM/CRAM synchronous write	DLAT CKDIV	DLAT CKDIV
	Mode SM	0	NOR Flash address/data mux	DLAT CKDIV	DLAT CKDIV

As shown in ***Table 48-7. EXMC timing models***, EXMC NOR Flash / PSRAM controller provides a variety of timing model, users can modify those parameters listed in ***Table 48-6. 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 48-7. Mode 1 read access

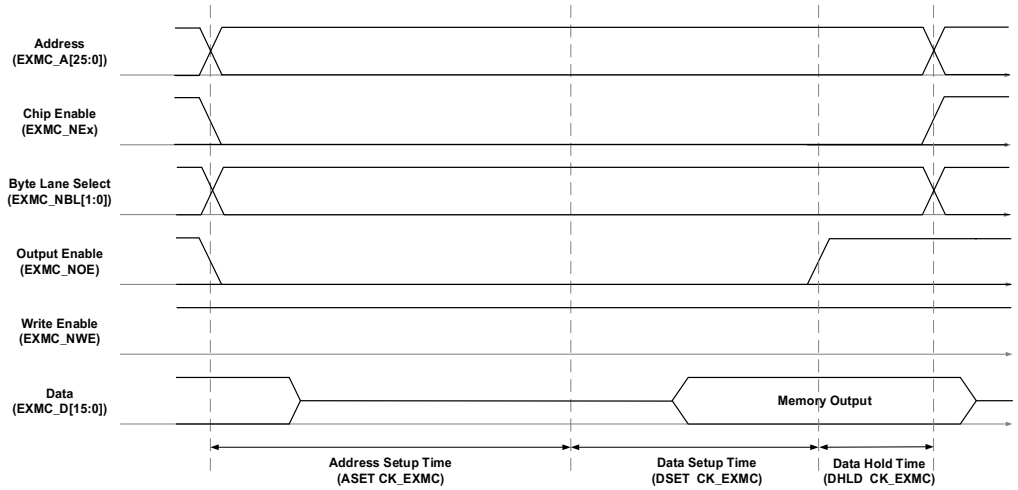


Figure 48-8. Mode 1 write access

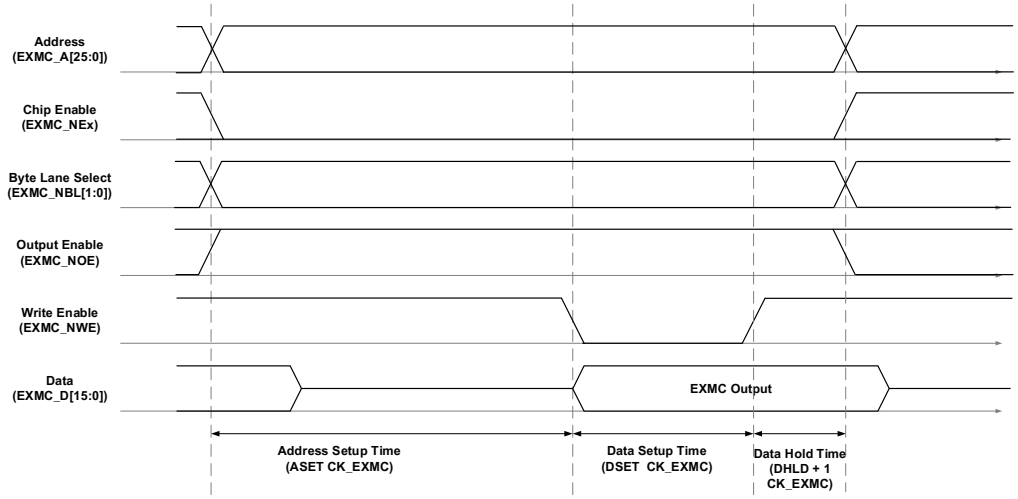


Table 48-8. Mode 1 related registers configuration

Bit Position	Bit Name	Reference Setting Value
EXMC_SNCTLx		
31-21	Reserved	0x000
20	CCK	Depends on memory and user
19	SYNCWR	0x0
18-16	CPS	0x0
15	ASYNCWTEN	Depends on memory
14	EXMODEN	0x0
13	NRWTEN	0x0
12	WEN	Depends on user
11	NRWTCFG	No effect
10	Reserved	0x0
9	NRWTPOL	Meaningful only when the bit 15 is set to 1

Bit Position	Bit Name	Reference Setting Value
EXMC_SNCTLx		
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	DHLD	Depends on memory and user
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
7-4	AHLD	No effect
3-0	ASET	Depends on memory and user

Mode A - SRAM/PSRAM(CRAM) OE toggling

Figure 48-9. Mode A read access

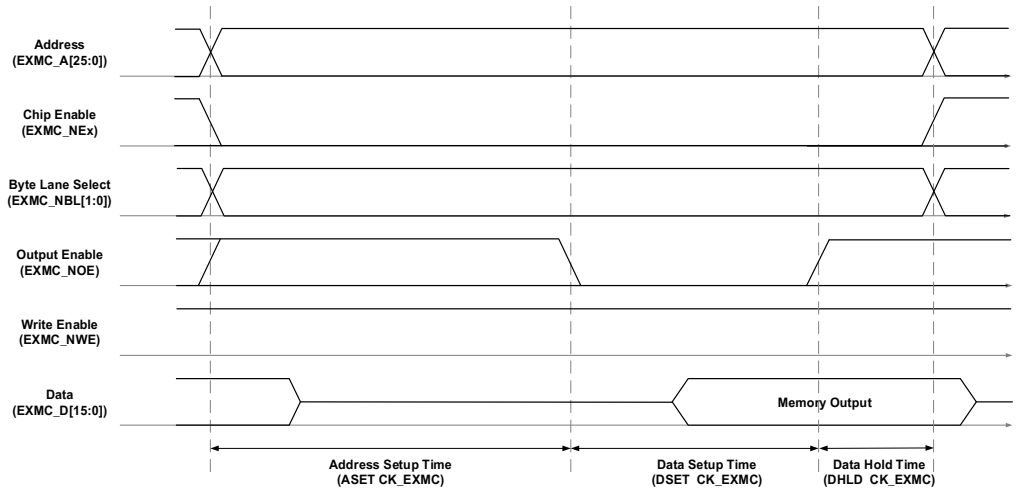
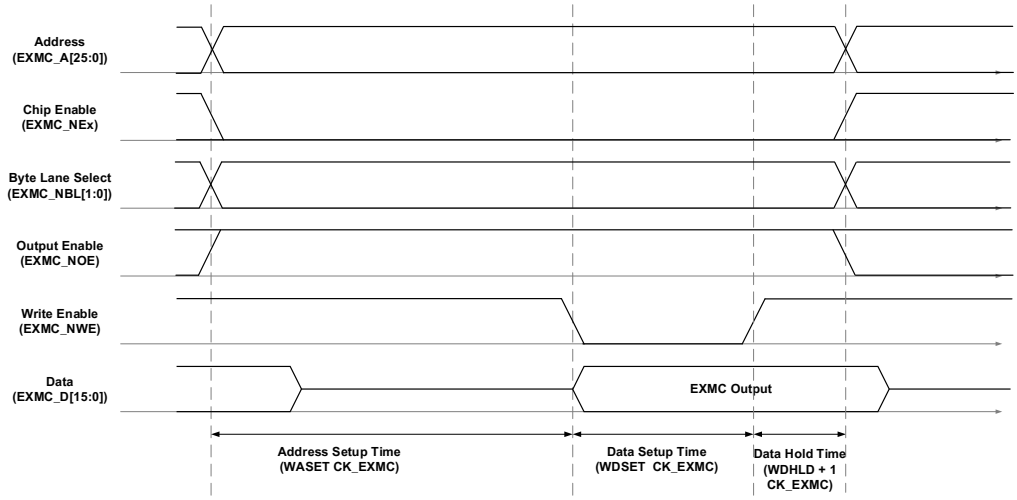


Figure 48-10. Mode A write access



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 48-9. Mode A related registers configuration

Bit Position	Bit Name	Reference Setting Value
EXMC_SNCTLx		
31-21	Reserved	0x000
20	CCK	Depends on memory and user
19	SYNCWR	0x0
18-16	CPS	0x0
15	ASYNCWTEN	Depends on memory
14	EXMODEN	0x1
13	NRWTEN	0x0
12	WEN	Depends on user
11	NRWTCFG	No effect
10	Reserved	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	DHLD	Depends on memory and user
29-28	ASYNCMOD	0x0
27-24	DLAT	No effect

Bit Position	Bit Name	Reference Setting Value
EXMC_SNCTLx		
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
7-4	AHLD	No effect
3-0	ASET	Depends on memory and user
EXMC_SNWTCFGx(Write)		
31-30	WDHLD	Depends on memory and user
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
7-4	WAHLD	0x0
3-0	WASET	Depends on memory and user

Mode 2/B - NOR Flash

Figure 48-11. Mode 2/B read access

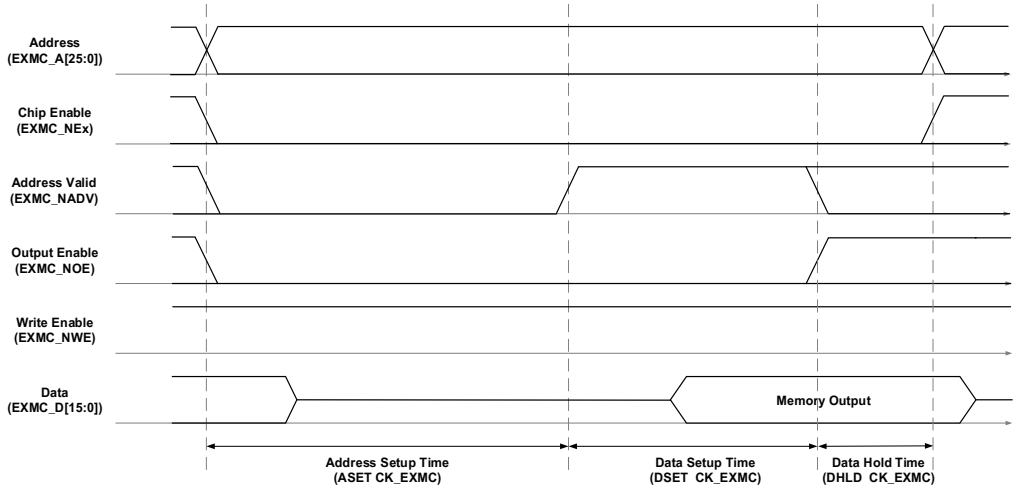


Figure 48-12. Mode 2 write access

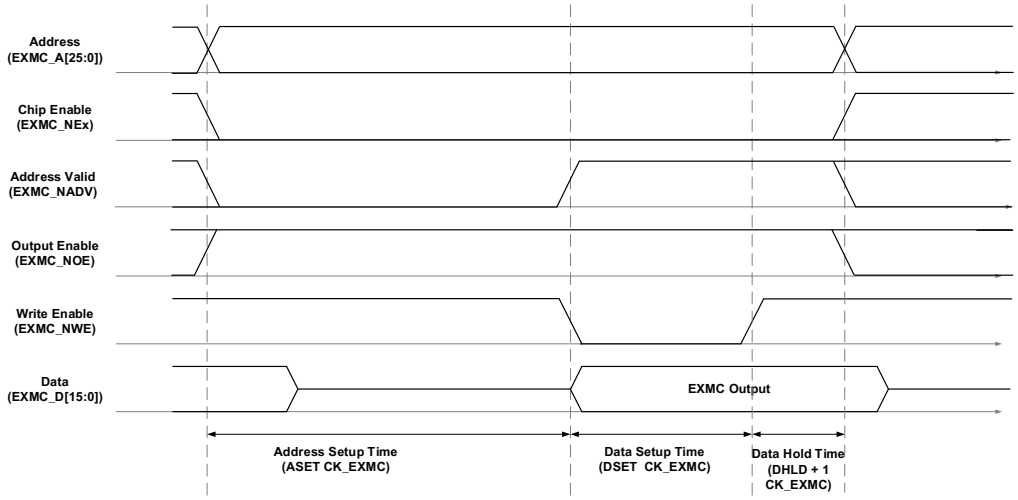


Figure 48-13. Mode B write access

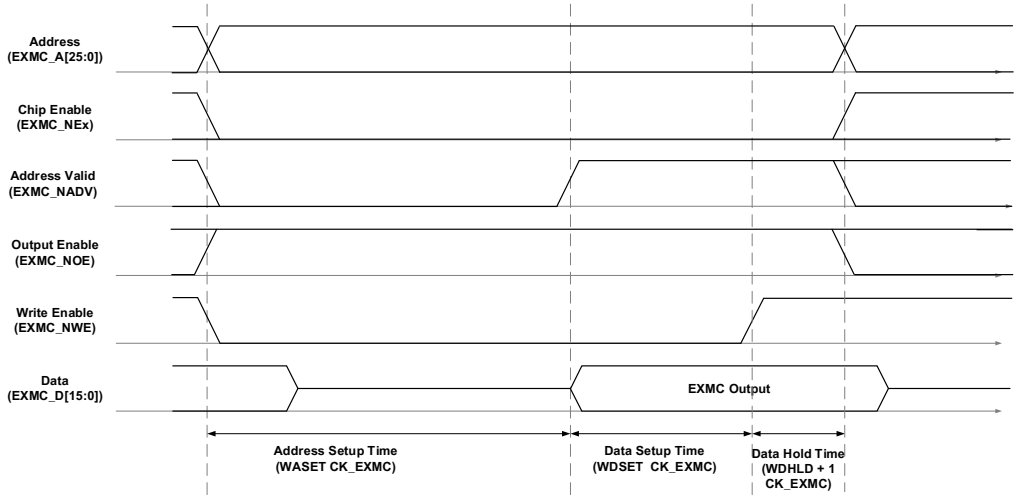


Table 48-10. Mode 2/B related registers configuration

Bit Position	Bit Name	Reference Setting Value
EXMC_SNCTLx(Mode 2, Mode B)		
31-21	Reserved	0x000
20	CCK	Depends on memory and user
19	SYNCWR	0x0
18-16	CPS	0x0
15	ASYNCWTEN	Depends on memory
14	EXMODEN	Mode 2:0x0, Mode B:0x1
13	NRWTEN	0x0
12	WEN	Depends on user
11	NRWTCFG	No effect
10	Reserved	0x0
9	NRWTPOL	Meaningful only when the bit 15 is set to 1

Bit Position	Bit Name	Reference Setting Value
EXMC_SNCTLx(Mode 2, Mode B)		
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	DHLD	Depends on memory and user
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
7-4	AHLD	0x0
3-0	ASET	Depends on memory and user
EXMC_SNWTCFGx(Write in mode B)		
31-30	WDHLD	Depends on memory and user
29-28	WASYNCMOD	Mode B:0x1
27-20	Reserved	0x000
19-16	WBUSLAT	Time between EXMC_NE[x] rising edge to EXMC_NE[x] falling edge
15-8	WDSET	Depends on memory and user
7-4	WAHLD	0x0
3-0	WASET	Depends on memory and user

Mode C - NOR Flash OE toggling

Figure 48-14. Mode C read access

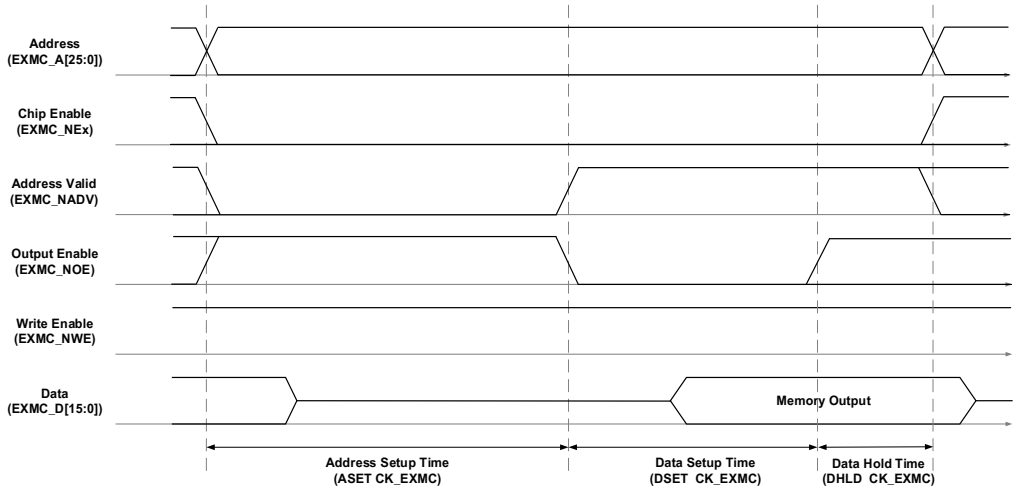
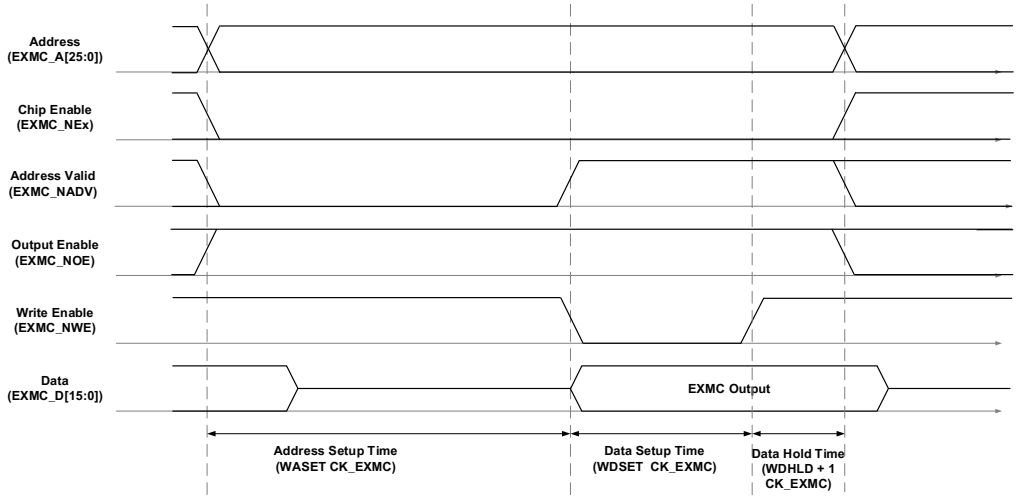


Figure 48-15. Mode C write access



The different between mode C and mode 1 write timing is 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.

Table 48-11. Mode C related registers configuration

Bit Position	Bit Name	Reference Setting Value
EXMC_SNCTLx		
31-21	Reserved	0x000
20	CCK	Depends on memory and user
19	SYNCWR	0x0
18-16	CPS	0x0
15	ASYNCWTEN	Depends on memory
14	EXMODEN	0x1
13	NRWTEN	0x0
12	WEN	Depends on user

Bit Position	Bit Name	Reference Setting Value
EXMC_SNCTLx		
11	NRWTCFG	No effect
10	Reserved	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		
31-30	DHLD	Depends on memory and user
29-28	ASYNCMOD	Mode C:0x2
27-24	DLAT	0x0
23-20	CKDIV	0x0
19-16	BUSLAT	Time between EXMC_NE[x] rising edge to EXMC_NE[x] falling edge
15-8	DSET	Depends on memory and user
7-4	AHLD	0x0
3-0	ASET	Depends on memory and user
EXMC_SNWTCFGx		
31-30	WDHLD	Depends on memory and user
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
7-4	WAHLD	0x0
3-0	WASET	Depends on memory and user

Mode D - Asynchronous access with extended address

Figure 48-16. Mode D read access

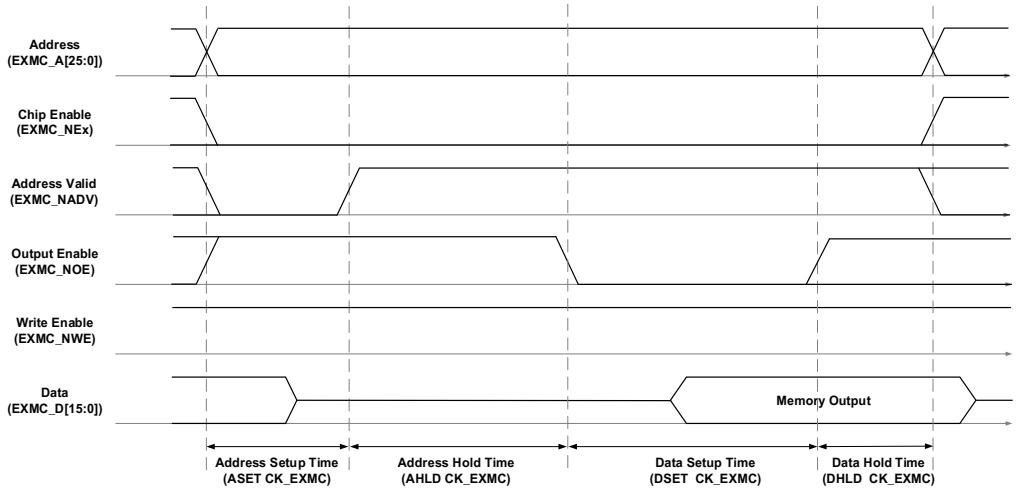


Figure 48-17. Mode D write access

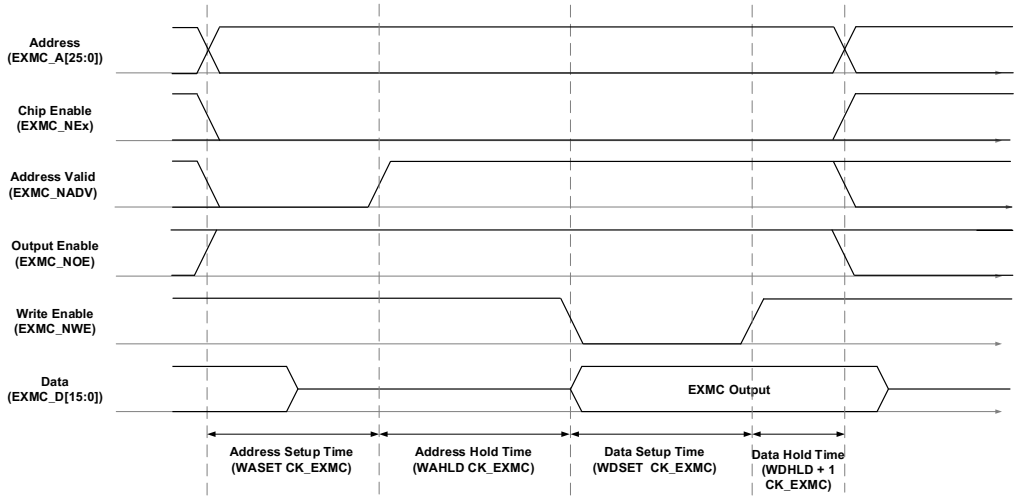


Table 48-12. Mode D related registers configuration

Bit Position	Bit Name	Reference Setting Value
EXMC_SNCTLx		
31-21	Reserved	0x000
20	CCK	Depends on memory and user
19	SYNCWR	0x0
18-16	CPS	0x0
15	ASYNCWTEN	Depends on memory
14	EXMODEN	0x1
13	NRWTEN	0x0
12	WEN	Depends on user
11	NRWTCFG	No effect
10	Reserved	0x0
9	NRWTPOL	Meaningful only when the bit 15 is set to 1

Bit Position	Bit Name	Reference Setting Value
EXMC_SNCTLx		
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		
31-30	DHLD	Depends on memory and user
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
7-4	AHLD	Depends on memory and user
3-0	ASET	Depends on memory and user
EXMC_SNWTCFGx		
31-30	WDHLD	Depends on memory and user
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
7-4	WAHLD	Depends on memory and user
3-0	WASET	Depends on memory and user

Mode AM - NOR Flash address / data bus multiplexing

Figure 48-18. Multiplex mode read access

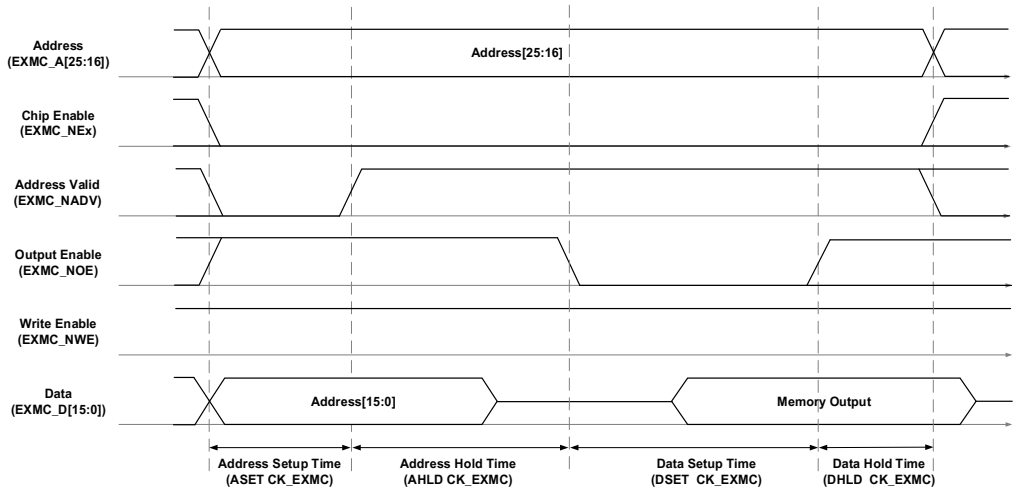


Figure 48-19. Multiplex mode write access

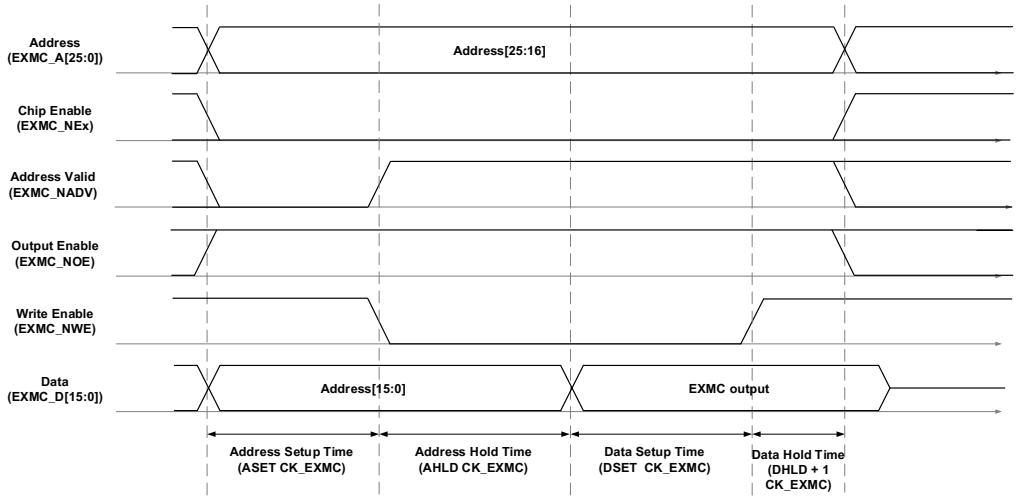


Table 48-13. Multiplex mode related registers configuration

Bit Position	Bit Name	Reference Setting Value
EXMC_SNCTLx		
31-21	Reserved	0x000
20	CCK	Depends on memory
19	SYNCWR	0x0
18-16	CPS	0x0
15	ASYNCWTEN	Depends on memory
14	EXMODEN	0x0
13	NRWTEN	0x0
12	WEN	Depends on memory
11	NRWTCFG	No effect
10	Reserved	0x0
9	NRWTPOL	Meaningful only when the bit 15 is set to 1

Bit Position	Bit Name	Reference Setting Value
EXMC_SNCTLx		
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	DHLD	Depends on memory and user
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
7-4	AHLD	Depends on memory and user
3-0	ASET	Depends on memory and user

Wait timing of asynchronous communication

Wait feature is controlled by the bit ASYNCWTEN in register EXMC_SNCTLx. During extern memory access, data setup phase will be automatically extended by the active EXMC_NWAIT signal if ASYNCWTEN bit is set. The extend time is calculated as follows:

If memory wait signal is aligned to EXMC_NOE / EXMC_NWE:

$$T_{DATA_SETUP} >= maxT_{WAIT_ASSERTION} + 4CK_EXMC \quad (48\text{-}1)$$

If memory wait signal is aligned to EXMC_NE:

If

$$maxT_{WAIT_ASSERTION} >= T_{ADDRES_PHASE} + T_{HOLD_PHASE} \quad (48\text{-}2)$$

$$T_{DATA_SETUP}>=(maxT_{WAIT_ASSERTION}-T_{ADDRES_PHASE}-T_{HOLD_PHASE})+4CK_EXMC \quad (48\text{-}3)$$

Otherwise

$$T_{DATA_SETUP} >= 4CK_EXMC \quad (48\text{-}4)$$

Figure 48-20. Read access timing diagram under async-wait signal assertion

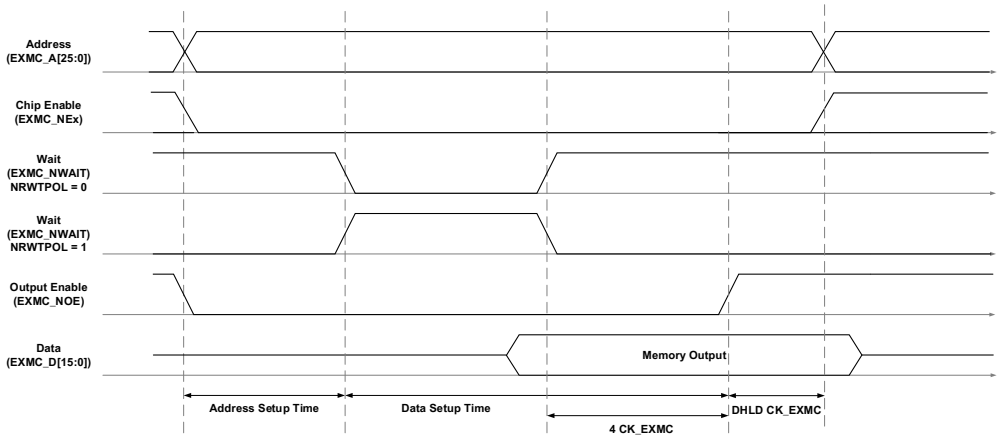
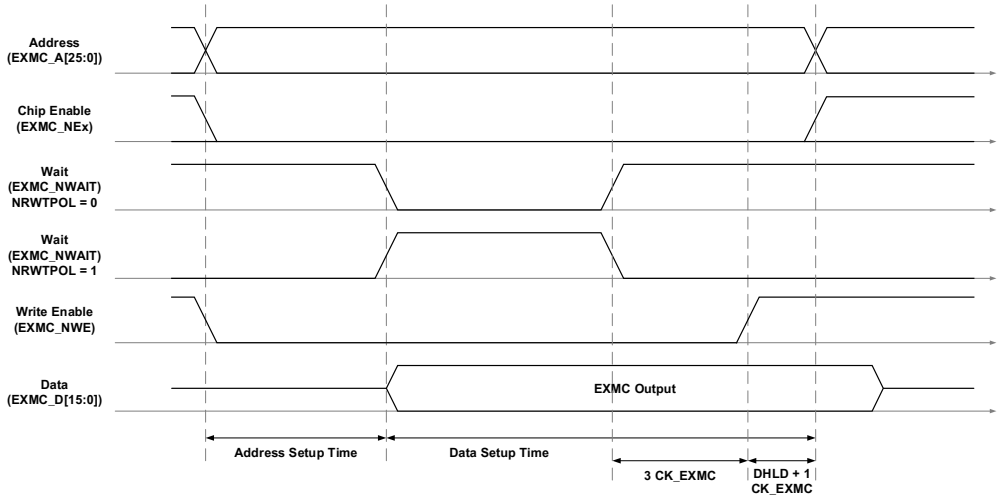


Figure 48-21. Write access timing diagram under async-wait signal assertion



Synchronous access timing diagram

The relation between memory clock (EXMC_CLK) and system clock (CK_EXMC) is as follows:

$$EXMC_CLK=\frac{CK_EXMC}{CKDIV+1} \quad (48\text{-}5)$$

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 (DLAT) 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:

$$NOR\ Flash\ latency=DLAT+2 \quad (48\text{-}6)$$

For NOR Flash's specification including the EXMC_NADV cycle, their relationship should be

NOR Flash latency=DLAT+3 (48-7)

2. Data wait

Users should guarantee that EXMC_NWAIT signal's behavior 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 as 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 AXI is 16-bit, EXMC will perform a burst transmission whose length is 1. If the needed data of AXI is 32-bit, EXMC will make the transmission divided into two 16-bit transmissions, that is, EXMC performs a burst transmission whose length is 2.

For other configurations please refers to ***Table 48-5. EXMC bank 0 supports all transactions***.

Synchronous mux burst read timing - NOR, PSRAM (CRAM)

Figure 48-22. Read timing of synchronous multiplexed burst mode

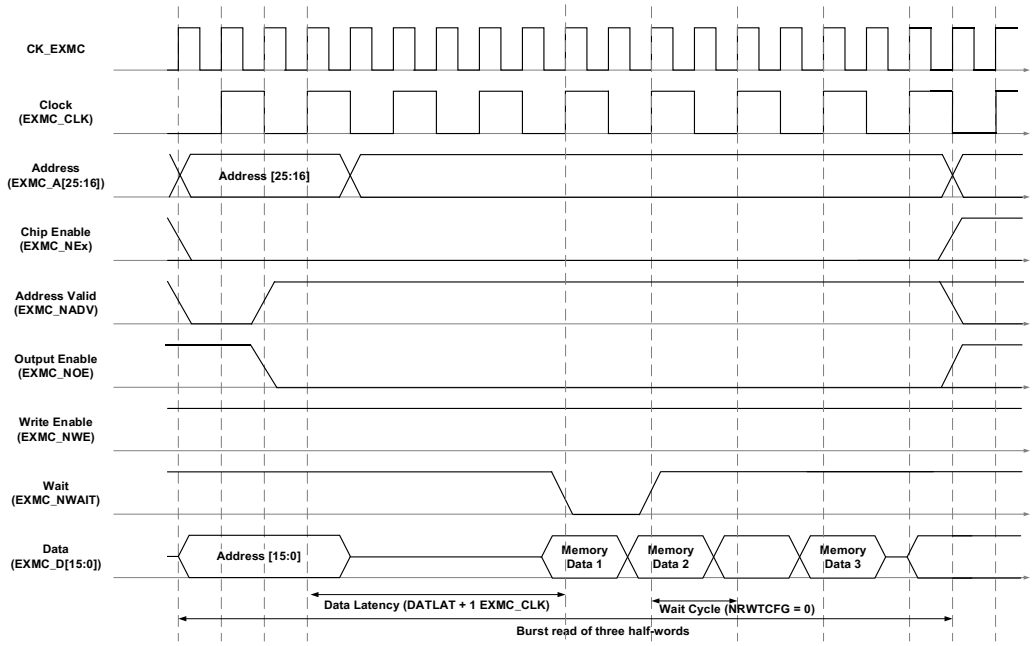


Table 48-14. Timing configurations of synchronous multiplexed read mode

Bit Position	Bit Name	Reference Setting Value
EXMC_SNCTLx		
31-21	Reserved	0x000
20	CCK	Depends on memory
19	SYNCWR	No effect
18-16	CPS	Depends on memory
15	ASYNCWTEN	0x0
14	EXMODEN	0x0
13	NRWTEN	Depends on memory
12	WEN	No effect
11	NRWTCFG	Depends on memory
10	Reserved	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	DHLD	No effect
29-28	ASYNCMOD	0x0

27-24	DLAT	Data latency
23-20	CKDIV	The figure above: 0x1,EXMC_CLK=2 CK_EXMC
19-16	BUSLAT	Time between EXMC_NE[x] rising edge to EXMC_NE[x] falling edge
15-8	DSET	No effect
7-4	AHLD	No effect
3-0	ASET	No effect

Mode SM -Synchronous mux burst write timing - NOR, PSRAM (CRAM)

Figure 48-23. Write timing of synchronous multiplexed burst mode

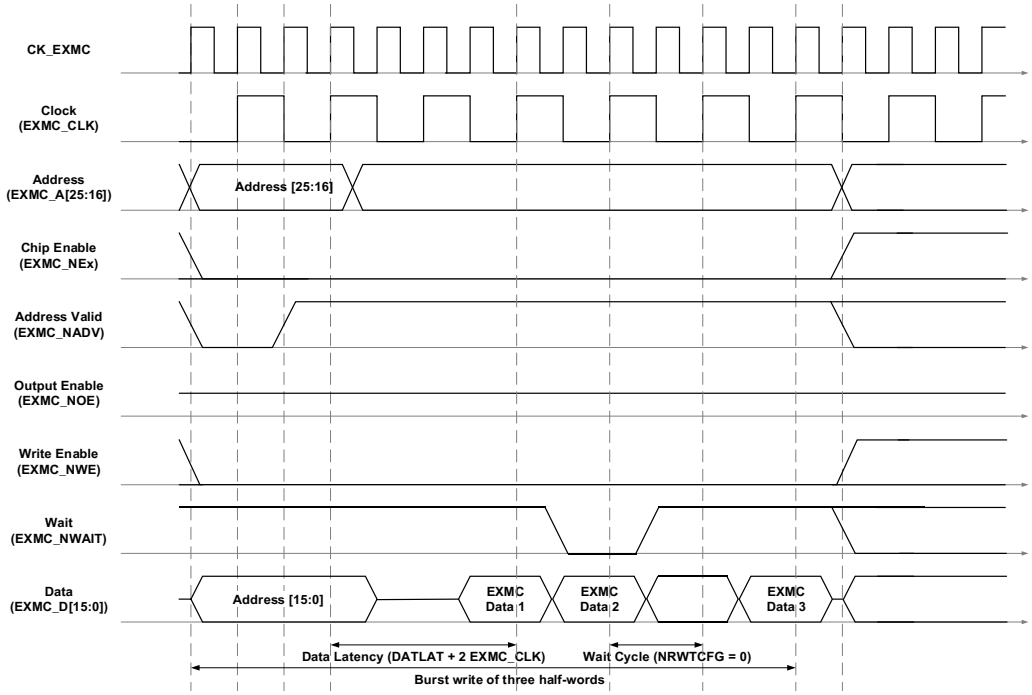


Table 48-15. Timing configurations of synchronous multiplexed write mode

Bit Position	Bit Name	Reference Setting Value
EXMC_SNCTLx		
31-21	Reserved	0x000
20	CCK	Depends on memory
19	SYNCWR	0x1, synchronous write enable
18-16	CPS	Depends on memory
15	ASYNCWAIT	0x0
14	EXMODEN	0x0
13	NRWTEN	Depends on memory
12	WEN	0x1
11	NRWTCFG	0x0(Here must be zero)
10	Reserved	0x0
9	NRWTPOL	Depends on memory

Bit Position	Bit Name	Reference Setting Value
EXMC_SNCTLx		
8	SBRSTEN	No effect
7	Reserved	0x1
6	NREN	Depends on memory
5-4	NRW	0x1
3-2	NRTP	0x1
1	NRMUX	0x1, Depends on users
0	NRBKEN	0x1
EXMC_SNTCFGx(Write)		
31-30	DHLD	No effect
29-28	ASYNCMOD	0x0
27-24	DLAT	Data latency
23-20	CKDIV	The figure above: 0x1, EXMC_CLK=2 CK_EXMC
19-16	BUSLAT	No effect
15-8	DSET	No effect
7-4	AHLD	No effect
3-0	ASET	No effect

48.3.7. NAND flash controller

EXMC has partitioned Bank2 as NAND Flash access field, Bank1 and Bank3 are reserved. Each bank has its own set of control register for access timing configuration. 8- and 16-bit NAND Flash are supported. An ECC hardware is provided for the NAND Flash controller to ensure the robustness of data transfer and storage.

NAND flash interface function

Table 48-16. 8-bit or 16-bit NAND interface signal

EXMC Pin	Direction	Functional description
EXMC_A[17]	Output	NAND Flash address latch (ALE)
EXMC_A[16]	Output	NAND Flash command latch (CLE)
EXMC_D[7:0]/ EXMC_D[15:0]	Input /Output	8-bit multiplexed, bidirectional address/data bus
		16-bit multiplexed, bidirectional address/data bus
EXMC_NCE	Output	Chip select
EXMC_NOE(NRE)	Output	Output enable
EXMC_NWE	Output	Write enable
EXMC_NWAIT(EXMC_NDWAIT)/ EXMC_INT	Input	NAND Flash ready/busy input signal to the EXMC

Supported memory access mode

Table 48-17. Bank2 of EXMC support the memory and access mode

Memory	Mode	R/W	AXI transaction size	Comments
8-bit NAND	Async	R	8	
	Async	W	8	
	Async	R	16	Automatically split into 2 EXMC accesses
	Async	W	16	
	Async	R	32	Automatically split into 4 EXMC accesses
	Async	W	32	
	Async	R	64	Automatically split into 8 EXMC accesses
	Async	W	64	
16-bit NAND	Async	R	8	Not support burst accesses
	Async	W	8	Not support this operation
	Async	R	16	
	Async	W	16	
	Async	R	32	Automatically split into 2 EXMC accesses
	Async	W	32	
	Async	R	64	Automatically split into 4 EXMC accesses
	Async	W	64	

NAND flash controller timing

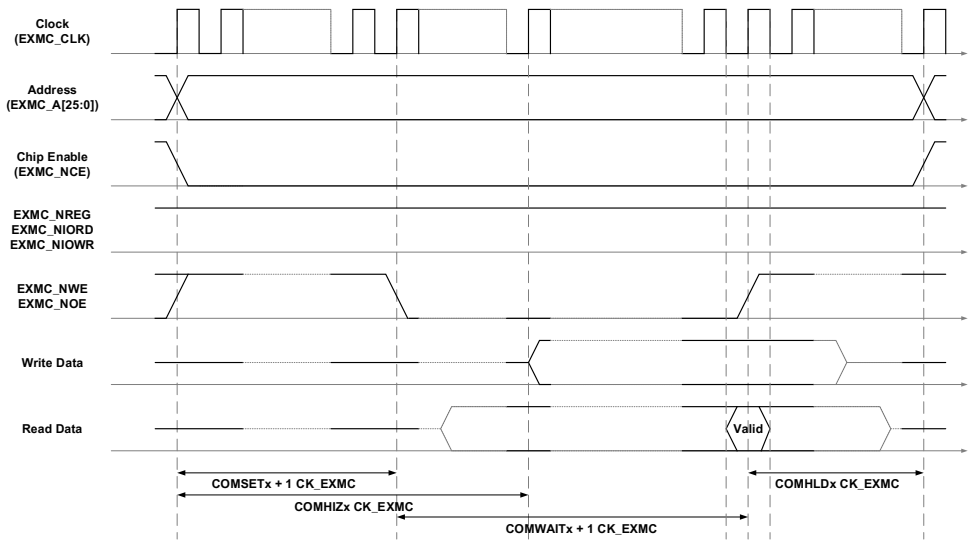
EXMC can generate the appropriate signal timing for NAND Flash and other devices. Each bank has a corresponding register to manage and control the external memory, such as EXMC_NCTL, EXMC_NINTEN, EXMC_NCTCFG, EXMC_NATCFG and EXMC_NECC. Among these registers, EXMC_NCTCFG, EXMC_NATCFG registers contain four timing parameters individually which are configured according to user specification and features of the external memory.

Table 48-18. NAND flash programmable parameters

Programmable parameter	W/R	Unit	Functional description	NAND Flash	
				Min	Max
High impedance time of the memory data bus (HIZ)	W/R	CK_EXMC	Time to keep the data bus high impedance after starting write operation	1	255
Memory hold time (HLD)	W/R	CK_EXMC	The number of CK_EXMC clock cycles to keep address valid after sending the command. In write mode, it is also data hold time.	1	254
Memory wait time (WAIT)	W/R	CK_EXMC	Minimum duration of sending command	2	255
Memory setup time (SET)	W/R	CK_EXMC	The number of CK_EXMC clock cycles to build address before sending command	1	256

The *Figure 48-24. Access timing of common memory space of NAND flash controller* shows the programmable parameters which are defined in the common memory space operations. The programmable parameters of Attribute memory space are defined as well.

Figure 48-24. Access timing of common memory space of NAND flash controller



NAND flash operation

When EXMC sends command or address to NAND Flash, it needs to use the command latch signal (A[16]) or address latch signal (EXMC_A[17]), namely, the CPU needs to perform write operation in particular address.

Example: NAND Flash read operation steps:

1. Configure EXMC_NCTL and EXMC_NCTCFG register. When pre-waiting is needed,

EXMC_NATCFG has to be configured.

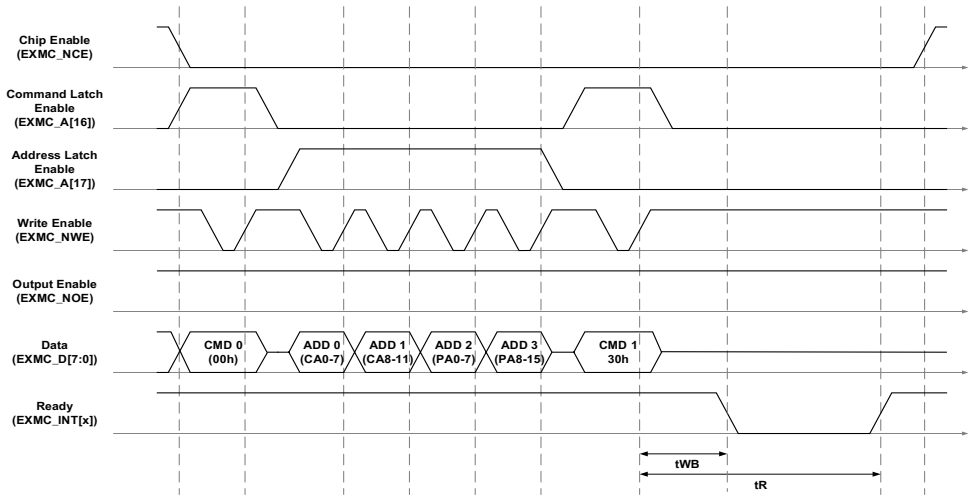
2. Send the command of NAND Flash read operation to the common space. Namely, during the valid period of EXMC_NCE and EXMC_NWE, when EXMC_CLE (EXMC_A[16]) becomes valid (high level), data on the I/O pins is regarded as a command by NAND Flash.
3. Send the start address of read operation to the common space. During the valid period of EXMC_NCE and EXMC_NWE, when EXMC_ALE (EXMC_A[17]) becomes valid (high level), the data on the I/O pins is regarded as an address by NAND Flash.
4. Waiting for NAND ready signal. In this period, NAND controller will maintain EXMC_NCE valid.
5. Read data byte by byte from the data area of the common space.
6. If new commands or address haven't been written, data of the next page can be read out automatically. You can also read the data of the next page by going to step 3 and then writing a new address or writing a new command and address in step 2.

NAND flash pre-wait functionality

Some NAND Flash requires that the controller should wait for NAND Flash to be ready, after the first command byte and following the address bytes are send, and some EXMC_NCE-sensitive NAND Flash also requires that the EXMC_NCE must remain valid before it is ready.

Taking TOSHIBA128 M x 8 bit NAND Flash as an example:

Figure 48-25. Access to none "NCE don't care" NAND Flash



1. Write CMD0 into NAND Flash bank common space command area.
2. Write ADD0 into NAND Flash bank common space address area.
3. Write ADD1 into NAND Flash bank common space address area.
4. Write ADD2 into NAND Flash bank common space address area.
5. Write ADD3 into NAND Flash bank common space address area.
6. Write CMD1 into NAND Flash bank attribute space command area.

In step 6, EXMC uses the operation timing defined in EXMC_NATCFG register. After a period

of ATTHLD, NAND Flash waits for EXMC_INT signal to be busy, and the time period of ATTHLD should be greater than t_{WB} (t_{WB} is defined as the time from EXMC_NWE high to EXMC_INT low). For NCE-sensitive NAND Flash, after the first command byte following address bytes has been entered, EXMC_NCE must remain low until EXMC_INT goes from low to high. The ATTHLD value of attribute space can be set in EXMC_NATCFG register to meet the timing requirements of t_{WB}. MCU can use the attribute space timing when writing the first command byte following address bytes to the NAND Flash device. In other times, the MCU must use the common space timing.

NAND flash ECC calculation module

An ECC calculation hardware is implemented in Bank2 respectively. Users can choose page size according to the ECCSZ control field in the EXMC_NCTL register. ECC offers one bit error correction and two bits errors detection.

When NAND memory block is enabled, ECC module will detect EXMC_D[15:0], EXMC_NCE and EXMC_NWE signals. When a data size of ECCSZ has been read or written, software must read the calculated ECC in the EXMC_NECC register. When a recalculation of ECC is needed, software must clear the EXMC_NECC register value by resetting ECCEN bit of EXMC_NCTL register to zero, and then restart ECC calculation by setting the ECCEN bit of EXMC_NCTL to 1.

48.3.8. SDRAM controller

Characteristics

- Two independent SDRAM devices.
- 8-,16- or 32-bit data bus width.
- Up to 13-bits Row Address, 11-bits Column Address and 2-bits internal banks address.
- Supported memory size: 4x16Mx32bit(256 MB), 4x16Mx16bit (128 MB) and 4x16Mx8bit (64 MB).
- AXI double word, word, half-word and byte access.
- Independent Chip Select control for each memory device.
- Independent configuration for each memory device.
- Write enable and byte lane select outputs.
- Automatic row and bank boundary management.
- Multi-device Ping-Pong access.
- SDRAM clock configured as $f_{CK_EXMC}/2$, $f_{CK_EXMC}/3$, $f_{CK_EXMC}/4$ or $f_{CK_EXMC}/5$.
- Programmable timing parameters.
- Automatic Refresh operation with programmable Refresh rate.
- SDRAM power-up initialization by software.
- CAS latency of 1,2,3.
- Write Data FIFO with 16 x35-bit depth.
- Write Address FIFO with 16x31-bit depth.

- Cacheable Read Data FIFO with 6 x32-bit depth.
- Cacheable Read address FIFO with 6 x14-bit depth.
- Adjustable read data sample clock.
- Self-refresh mode.
- Power-down mode.

Note: The frequency of CK_EXMC must not exceed 4 times the frequency of ACLK.

SDRAM overview

Synchronous dynamic random-access memory (SDRAM) is a dynamic random access memory (DRAM) whose external interface is coordinated by a synchronous external clock, this clock is provided by the EXMC through the SDRAM clock (EXMC_SDCLK) pin, and its frequency could be configured to be $f_{CK_EXMC}/2$, $f_{CK_EXMC}/3$, $f_{CK_EXMC}/4$ or $f_{CK_EXMC}/5$ according to the SDRAM clock configuration bit (SDCLK) in the EXMC_SDCTLx register. Commands and data are always latched by the SDRAM on the rising edge of EXMC_SDCLK and change on its falling edge.

SDRAM is divided into several independent sections of memory called banks, allowing the device to operate on several memory access commands in an interleaved fashion to achieve greater concurrency and higher data transfer rates. Each bank could be pictured as a matrix with each entry size equals to the memory data bus width, and the size of the matrix is the number of rows by the number of columns, thus each memory bank size could be calculated as entry_size*rows*columns. When interfacing with SDRAM, users should specify the memory dimension configurations to EXMC through NBK, SDW, RAW and CAW bits in the SDRAM control register EXMC_SDCTLx.

Due to the volatile nature of SDRAM, periodic refresh cycle is necessary to maintain the stored information. Two refresh mode could be selected, self-refresh and auto-refresh mode. Self-refresh mode is typically set in low power mode when EXMC is suspended, refresh is provided by the SDRAM and timed by its internal counter. In auto-refresh mode, refresh command is provided by the EXMC, this is necessary because SDRAM must maintain the stored information during an on-going transaction, refresh commands are issued periodically on the data bus timed by ARINTV bits in EXMC_SDARI register, the number of consecutive refresh needed is configured through NARF bits in EXMC_SDCMD register. Refresh command always take precedence over other command or read / write operation to guarantee correct data storage, when memory access occurs simultaneously with refresh command, memory access is buffered and processed when refresh command is completed. If a new refresh command occurs while the previous refresh command is buffered, a refresh error flag (REIF) is raised in EXMC_SDSTAT register, and interrupt is generated if REIE is set and cleared by setting REC bit in EXMC_SDARI register.

CAS latency defines the delay in clock cycles, between the issued read command and the availability of the first piece of data form SDRAM. CAS latency is configured by the CL bits in the EXMC_SDCTLx register.

Mode register is used to define the specific operating mode of SDRAM, including burst length,

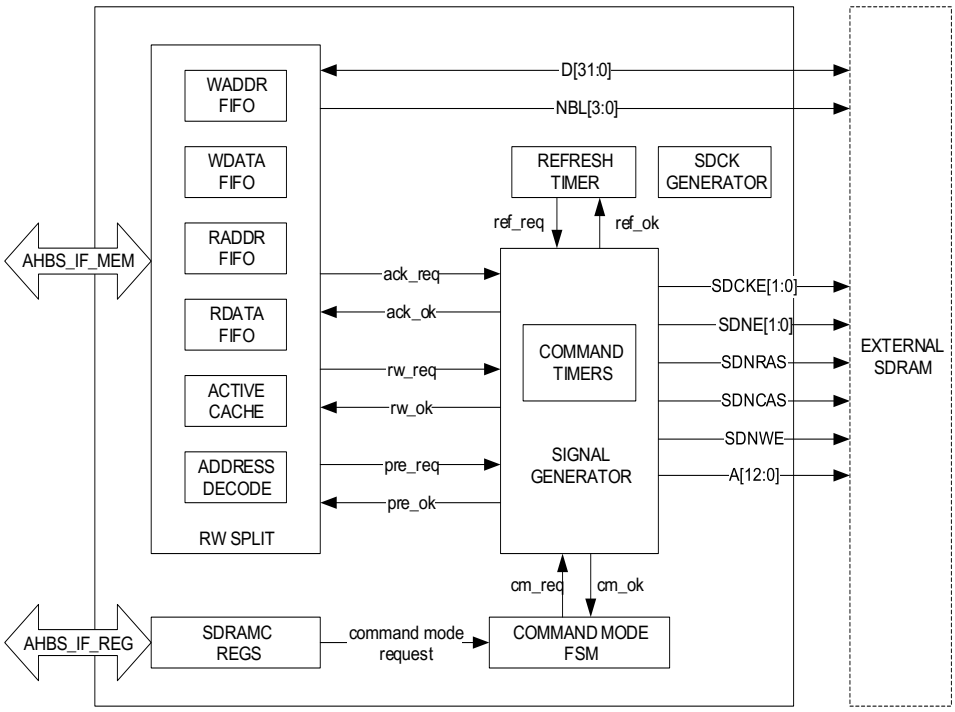
burst type, CAS latency, and write mode. Users should refer to the SDRAM's specification for correct configuration. Once the operating mode has been decided, users should write the mode register content to MRC bits and issuer load mode register command through CMD bits in EXMC_SDCMD register. Load mode register command should be performed before read or write access, otherwise SDRAM might not work as expected.

SDRAM controller overview

The synchronous dynamic random-access memory controller (SDRAMC) block acts as the interface between MCU and SDRAM memory. It translates AXI transactions into the appropriate SDRAM protocol, and meanwhile, makes sure the access timing requirements of the external SDRAM devices are satisfied by the configuration of EXMC_SDTCFG register.

SDRAMC could be divided into 4 sub-modules, the read / write split, control registers, finite state machine, and signal generator. Two pairs of FIFO is implemented to increase memory access efficiency, one pair for write address and data, the other pair for read address and data. SDRAMC's block diagram is shown in *Figure 48-26. SDRAM controller block diagram*.

Figure 48-26. SDRAM controller block diagram



The signal generator handles requests from command mode FSM, Refresh timer and the RW split module.

The command timers are composed by timing counters which take case the timing specification of the SDRAM protocol.

SDRAM commands are issued by the SDRAM controller interface in the following pattern.

Table 48-19. SDRAM command truth table

SDNE	NRAS	NCAS	SDNWE	A[n]	A[10]	A[m]	Command
H	X	X	X	X	X	X	Command inhibit (No operation)
L	H	H	H	X	X	X	No operation
L	H	H	L	X	X	X	Burst Terminate
L	H	L	H	Bank	L	Col	Burst read from current row
L	H	L	H	Bank	H	Col	Burst read from current row, precharge when done
L	H	L	L	Bank	L	Col	Burst write to current row
L	H	L	L	Bank	H	Col	Burst write to current row, precharge when done
L	L	H	H	Bank	Row	Row	Active, open row for read / write
L	L	H	L	Bank	L	X	Precharge, close current row of the selected bank
L	L	H	L	X	H	X	Precharge all, close current row of all banks
L	L	L	H	X	X	X	Auto-refresh when SDCKE = 1 Self-refresh when SDCKE = 0
L	L	L	L	L	Mode	Mode	Load mode register

SDRAM controller operation sequence

IO configuration

SDRAMC IO port must be configured first to interface with external SDRAM, otherwise it is left as general purpose Ios, and could be utilized by other modules. IO ports related to SDRAM operations are summarized in *Table 48-20. IO definition of SDRAM controller*.

Table 48-20. IO definition of SDRAM controller

Signal	Direction	Description
EXMC_SDCLK	O	SDRAM memory clock
EXMC_SDCKE[0]	O	Clock enable for SDRAM memory 0
EXMC_SDCKE[1]	O	Clock enable for SDRAM memory 1
EXMC_SDNE[0]	O	Chip select for SDRAM memory 0, active low
EXMC_SDNE[1]	O	Chip select for SDRAM memory 1, active low
EXMC_NRAS	O	Row address strobe, active low
EXMC_NCAS	O	Column address strobe, active low
EXMC_SDNWE	O	Write enable, active low
EXMC_A[12:0]	O	Address
EXMC_A[15:14]	O	Bank address
EXMC_D[31:0]	I/O	Read / write Data
EXMC_NBL[3:0]	O	Write data mask, the Low byte lane is accessed

Controller initialization

Users should follow procedure to initialize the SDRAM controller, the initialization sequence

could be applied to a single SDRAM, or two SDRAM simultaneously. This choice is made by the device selection bits DS0 and DS1 in EXMC_SDCMD register. Initialization sequence must be performed before any read / write memory access, otherwise, EXMC's behavior is not guaranteed.

1. Control parameter specification: SDRAM control register EXMC_SDCTLx should be programed first to specify the external memory dimension, clock configuration, and read / write strategy.
2. Timing parameter specification: SDRAM timing configuration register EXMC_SDTCFGx should be programed according to external SDRAM data sheet for SDRAM controller to keep pace with the operation of the external SDRAM. RPD and ARFD must be programed in EXMC_SDTCFG0, those corresponding bit position in EXMC_SDTCFG1 are reserved.
3. Enable SDCLK: SDCLK enable command should be issued to the corresponding SDRAM devices, this is done by writing 0b001 to the CMD bits in the EXMC_SDCMD register, DS0 and DS1 selected which device will accept the command and start receiving EXMC_SDCLK.
4. Power-up delay: typical delay is around 100us.
5. Precharge all: A precharge all command should be issued to reset all the SDRAM memory banks to their idle state, waiting for subsequent operation. This is done by writing 0b010 to the CMD bits in the EXMC_SDCMD register, DS0 and DS1 defines which SDRAM device will receive this command.
6. Set auto-refresh: Auto-refresh command is sent by writing 0b011 in the CMD bits in EXMC_SDCMD register. Users should also specify the number of consecutive refresh command to issue each time by configuring the NARF bits, this configuration is requested by SDRAM specification, it is also where users should refer to.DS0 and DS1 defines which SDRAM device will receive this command.
7. Mode register configuration: Mode register is programed by writing the mode register content in MRC bits in EXMC_SDCMD register, mode register specifies the operating mode of SDRAM, such modes include burst length, burst type, CAS latency, and write mode. Users should refer to the SDRAM's specification for correct configuration. CAS latency should be the same as the CL bits in EXMC_SDCTLx register, and burst length of 1 must be selected, otherwise SDRAMC's behavior is not guaranteed. If the mode register contents are different for both SDRAM devices, this step should be repeated, targeting one device a time by the DS0 and DS1 configuration.
8. Set auto-refresh rate: Auto-refresh rate corresponds to the time between refresh cycles, users must ensure that this time period match that of the SDRAM specification.

Now SDRAMC is ready to proceed with memory access. If system reset happens, the initialization sequence must be repeated. Initialization must be performed at least once before SDRAM read / write access.

Precharge

When the memory controller needs to access a different row, it must first return that bank's

sense amplifiers to an idle state, ready to sense the next row. This is known as a precharge operation, or deactivating the row. A precharge may be commanded explicitly by the precharge all command, or it may perform automatically at the conclusion of a read or write operation. There is a minimum time, the row precharge delay (RPD), which must elapse before that bank is fully idle and it may receive another activate command.

Activate

The activate command activates an idle bank. It presents a 2-bit bank address EXMC_A[15:14] and a 13-bit row address EXMC_A[12:0], and causes a read of that row into the bank's array of 16,384 column sense amplifiers. This also known as opening the row. This operation has the side effect of refreshing the dynamic memory storage cells of that row.

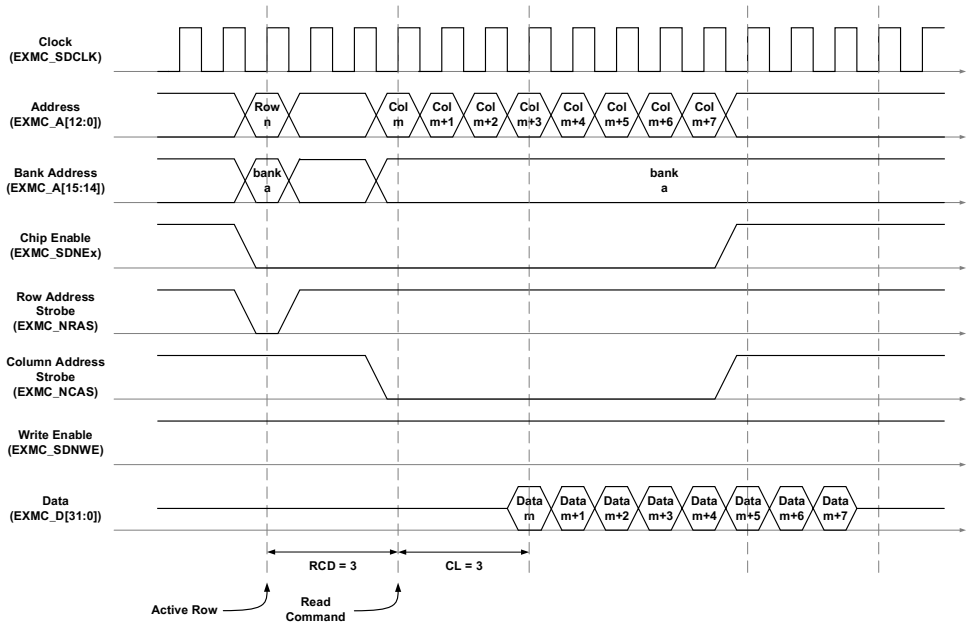
Once the row has been activated, read / write commands are possible to that row. Activation requires a minimum amount of time, called the row-to-column delay (RCD) before read / write to it may occur. This time, rounded up to the next multiple of the clock period, specifies the minimum number of wait cycles between an active command and a read / write command. During these wait cycles, additional commands may be sent to other banks, because each bank operates completely independently.

Read / write access

SDRAMC can translate AXI single and burst read operation into single memory access. SDRAMC always keeps track of the activated row number in order to perform consecutive read access. If the next read location is in the same row or another active row, read access is proceeded without interruption, else a precharge command is issued to deactivate the current row, followed by the activation of the row where the next read access is targeted, and then the read access is performed. A read FIFO is design to cache the read data during CAS latency and pipe line delay (PIPED), Burst read (BRSTRD) must be set in order to enable the FIFO.

The ***Figure 48-27. Burst read operation*** shows a burst read access to an in active row, a row activation command is issued before read access. If read operation were performed on an active row, row address strobe is not necessary, only column address strobe is needed.

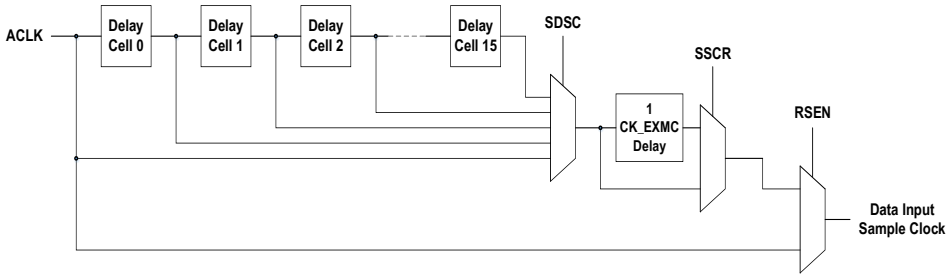
Figure 48-27. Burst read operation



An internal generated clock, which has an adjustable delay from the CK_EXMC can be used to sample read data from external memories. This clock can be helpful when the read data can't be sampled correctly by CK_EXMC. When this clock is enabled, the read data will be firstly stored in an asynchronous FIFO before returned to the AXI bus. Additional delays of about 2~3 CK_EXMC may be brought into the reading command process.

A clock delay chain module is added after the CK_EXMC input to the signal generator, this delayed clock is used as the sampling clock of the input data. The delay chain is controlled by the EXMC_SDRSCTL register, RSEN bit select whether the CK_EXMC output is delay at all, SSCR bit select whether 1 additional CK_EXMC cycle is added to the total delay, and SDSC select how many delay cells is add, the number of delay cell could be added is within 0 and 15. The following diagram shows how delay chain is added.

Figure 48-28. Data sampling clock delay chain

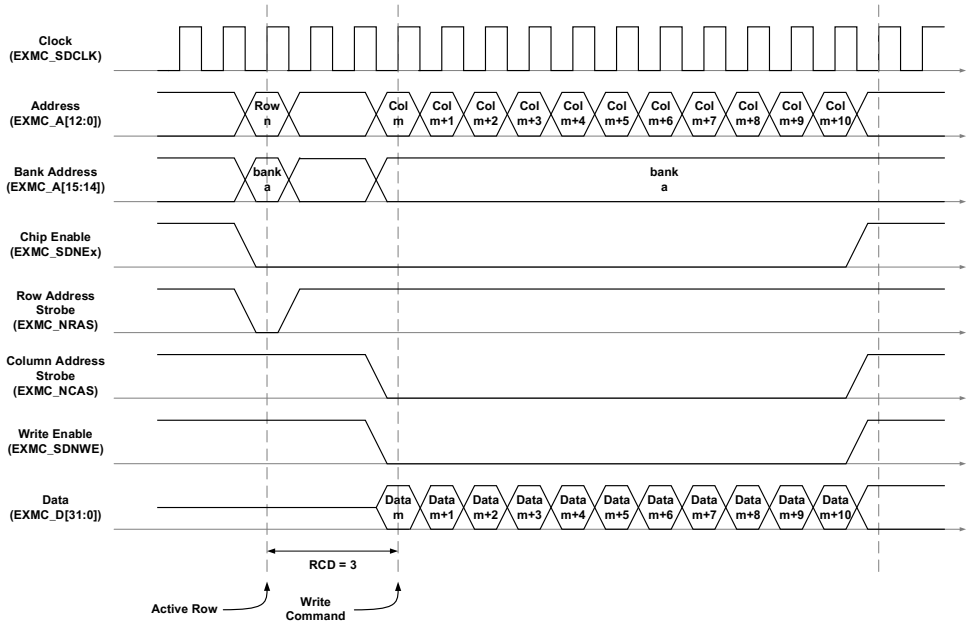


SDRAMC can translate AXI single and burst write operation into single memory access. Write protection must be disabled by resetting WPEN bit in EXMC_SDCTLx register. SDRAMC always keeps track of the activated row number in order to perform consecutive write access. If next write location is in the same row or another active row, write access is proceeded without interruption, else a precharge command is issued to deactivate the current row, followed by the activation of the row where the next write access is targeted, and then the

write access is performed.

The *Figure 48-29. Burst write operation* shows a write burst access to an inactive row, a row activation command is issued before write access. If write operations were performed on an active row, row address strobe is not necessary, only column address strobe is needed.

Figure 48-29. Burst write operation



The RW split module accepts AXI commands, and transfers them to single read / write accesses on the SDRAM memory according to the ratio of the data width between the AXI bus and the SDRAM memory interface.

Inside the RW split module, there are two write FIFOs, which buffers the data and address of the AXI write commands. When neither of the write FIFOs is empty, write access occurs.

When the BRSTRD bit of EXMC_SDCTL0 register is set, the RW split module can anticipate the next read access. The read FIFOs are used to store data read in advance during the CAS latency period (configured by the CL bits of EXMC_SDCTLx) and during the PIPED delay (configured by the PIPED bits of EXMC_SDCTL0).

The RDATA FIFO can buffers up to 6 32-bit read data words, while the RADDR FIFO carries 6 14-bit read address tags to identify each of them. Every address tag is comprised of 11 bits for the column address, 2 bits for the internal bank address and 1 bit to select the SDRAM memories.

When there is a read commands on the AXI bus, the RW split module will firstly checks whether the address matches one of the address tags, and data are directly read from the FIFO when it is true. Otherwise, a new read command is issued to the memory and the FIFO is updated with new data. If the FIFO is full, the older data are lost.

Figure 48-30. Read access when FIFO not hit (BRSTRD=1, CL=2, SDCLK=2, PIPED=2) and *Figure 48-31. Read access when FIFO hit (BRSTRD=1)* specify the Read FIFO

operation.

Figure 48-30. Read access when FIFO not hit (BRSTRD=1, CL=2, SDCLK=2, PIPED=2)

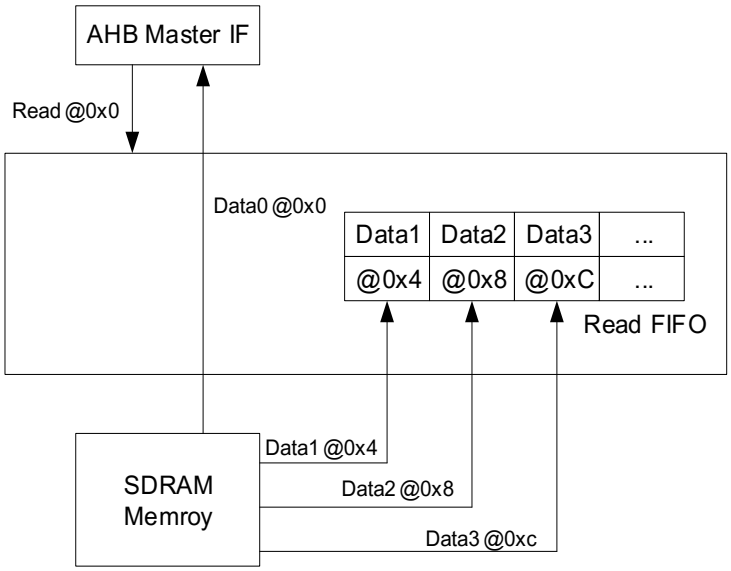
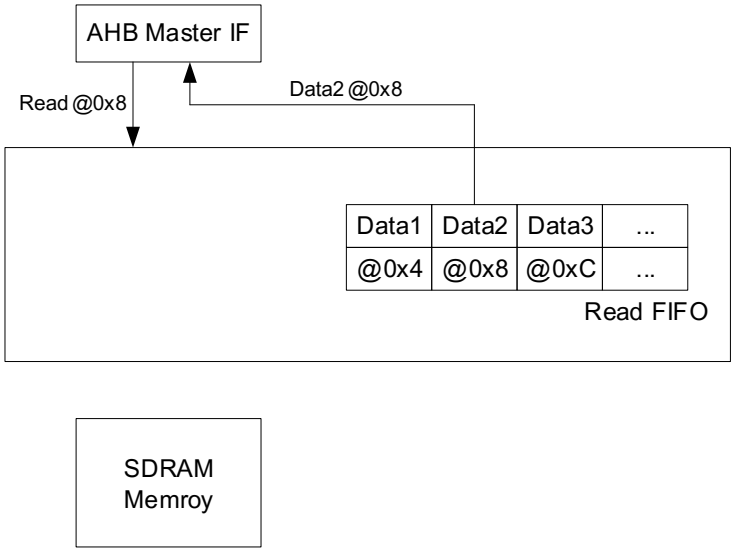


Figure 48-31. Read access when FIFO hit (BRSTRD=1)



The read FIFO will be flushed and ready to be filled with new data, when a write access or a precharge command occurs.

The address decoder sub-module translates the address of the AXI bus address to chip select, internal bank address, row address and column address according to the configuration of external memory device.

The active cache sub-module records whether the internal banks (up to 8) are in the active state. When an internal bank is in active state, the corresponding row address is also recorded. When an AXI access or an auto-refresh command is issued, the RW split module will look up this record and decide whether to generate the Active/Precharge commands or not.

Before read / write operation, the targeted row must be activated, the value of EXMC_A[15:14] selects the bank, and EXMC_A[12:0] select the row. The selected row remains active until a precharge command is issued. The precharge command is used to deactivate an active row in a particular bank or the active row in all banks. A precharge command must be issued before activating a different row in the same bank. Active and precharge are automatically issued by the EXMC, its correctness depends on memory dimension configurations discussed previously, read and write timing diagram concerning automatic row activation and precharge are depicted as follows.

Figure 48-32. Cross boundary read operation

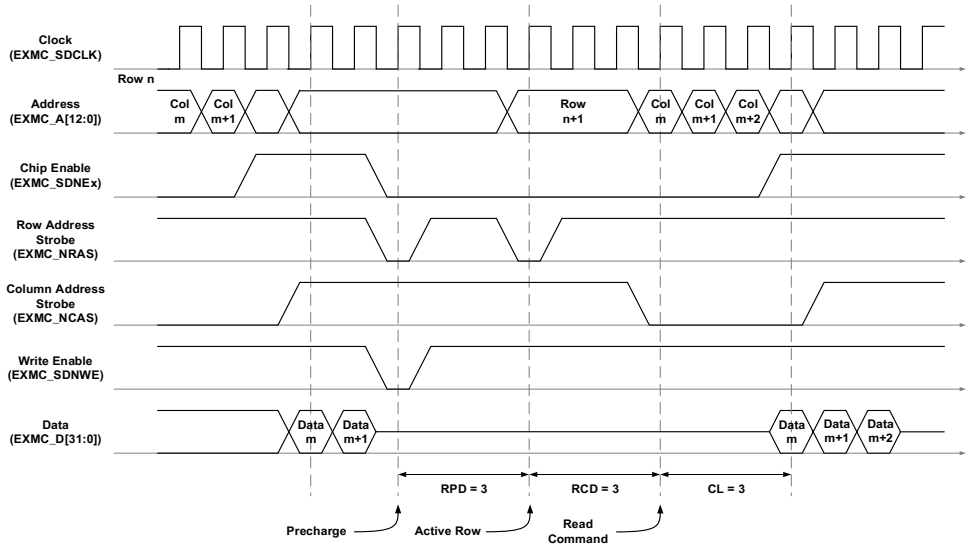
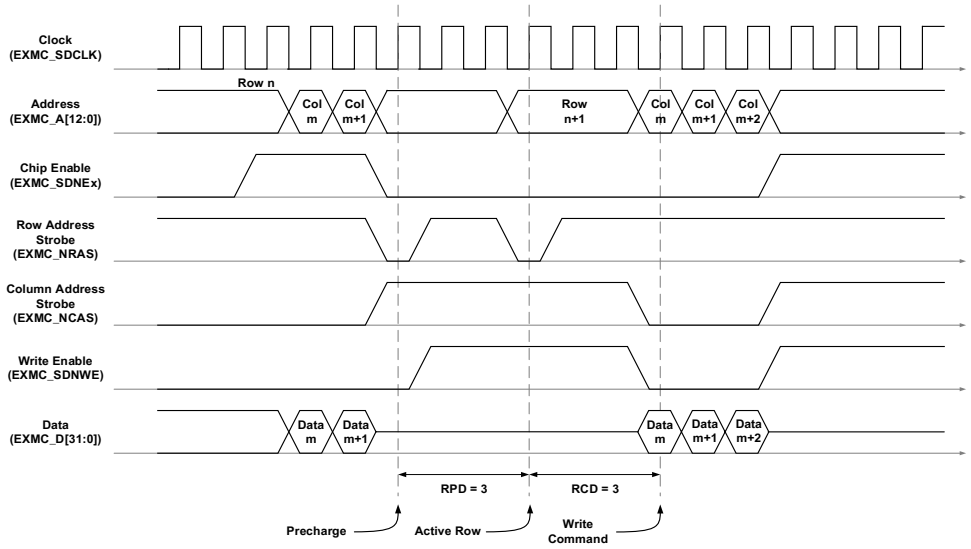


Figure 48-33. Cross boundary write operation



The above diagrams depict read and write timing waveform when memory access crosses row boundary, the following steps are preformed automatically:

1. Precharge the current active row.
2. Next row's activation.

3. Read / write access.

Precharge delay (RPD) and row to column delay (RCD) are added according to their configuration in EXMC_SDTCFGx register, other timing parameters should be configured as SDRAM specification requires.

When this boundary happens to be at the end of a bank, two cases are possible:

1. When the current bank is not the last bank, the activation of the first row of the next bank is performed, and this supports all row, column, and bus width configuration.
2. When the current bank is the last bank, and row, column, and bus width are configured as, 13-bit, 11-bit, and 32-bit respectively, EXMC continues to read / write from the second SDRAM device (SDRAM device 1), assuming that the current SDRAM is device 0.

Low power modes

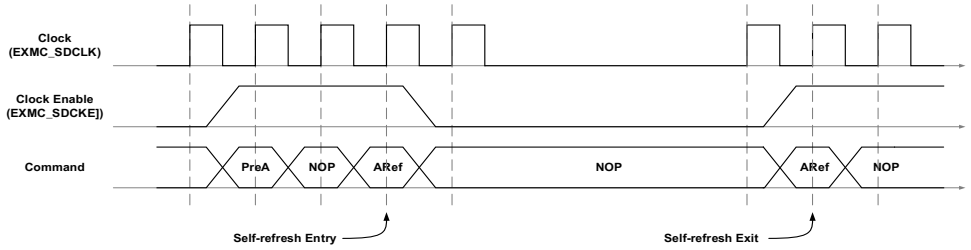
Two low power mode are supported:

1. Self-refresh mode: In self-refresh mode, refresh is provided by the SDRAM itself to maintain data integrity without external clock (EXMC_SDCLK). It is entered by writing 0b101 to CMD bits in EXMC_SDCMD register, DS0 andDS1 determines which SDRAM device will receive the command. EXMC_SDCLK stops running after a RASD delay if this command is issued to both SDRAM devices or one of the SDRAM device is not initialized.
2. Power-down mode: In power-down mode, refresh is provided by the SDRAM controller. It is entered by writing 0b110 to CMD bits in EXMC_SDCMD register, DS0 and DS1 determines which SDRAM device will receive the command. If the write data FIFO is not empty, all data are sent to the memory before activating power-down mode.

The Command Mode FSM also controls the switching process of between the normal mode and the low-power modes (self-refresh/power-down).

The SDRAM controller returns to normal mode from self-refresh mode when a read / write access occurs. If a read / write access occurs while the SDRAM controller is entering self-refresh mode, the self-refresh entry process will be interrupted, and the SDRAM controller remains in normal mode after the read / write access completed.

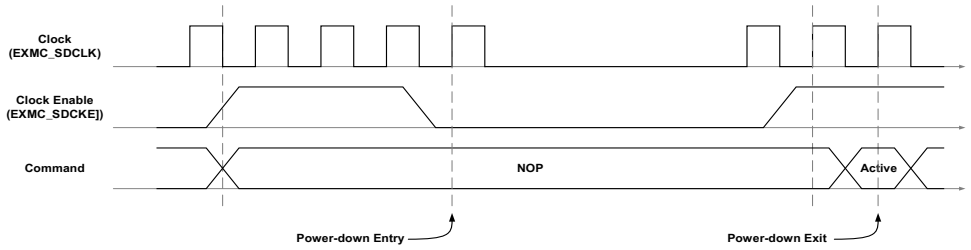
Figure 48-34. Process for self-refresh entry and exit



If an auto-refresh request occurs when the SDRAM controller is in power-down mode, the SDRAM controller returns to normal mode, issues the Precharge all and Auto-Refresh

command sequence, and enters power-down mode again automatically.

Figure 48-35. Process for power-down entry and exit



Status and interrupt

The not ready status NRDY bit in EXMC_SDSTAT register specifies whether the SDRAM controller is ready for a new command, this bit is cleared immediately after the command in the SDRAMC's internal register is sent.

Device0 and Device1 status bits STA0 and STA1 in EXMC_SDSTAT register defines the status of SDRAM deivce0 and device1 respectively, 0b00 represents normal mode, 0b01 indicates that the corresponding SDRAM devices is in self-refresh mode, and 0b10 signifies the power-down mode.

If a new refresh request occurs while the previous refresh command has not been served yet, a refresh error flag (REIF) is raised in EXMC_SDSTAT register, and interrupt is generated if REIE is set, refresh error flag is cleared by setting REC bit in EXMC_SDARI register.

48.4. Register definition

EXMC base address: 0x5200 4000

48.4.1. NOR/PSRAM controller registers

The peripheral registers have to be accessed by words (32-bit).

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

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						BKREMAP[1:0]		Reserved			CCK	SYNCWR	CPS[2:0]		
						rw					rw	rw	rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ASYNCWTEN	EXMODEN	NRWTEN	WEN	NRWTCFG	Reserved	NRWTPOL	SBRSTEN	Reserved	NREN	NRW[1:0]		NRTP[1:0]		NRMUX	NRBKEN
rw	rw	rw	rw	rw		rw	rw		rw	rw		rw		rw	rw

Bits	Fields	Descriptions
31:26	Reserved	Must be kept at reset value.
25:24	BKREMAP[1:0]	Bank remap 00: Default mapping 01: NOR/PSRAM bank and SDRAM device 0 swapped 10: Reserved 11: Reserved **Note:** The BKREMAP bits are only present in EXMC_SNCTL0 register, and theses bits position in EXMC_SNCTLx (x = 1, 2, 3) registers are meaningless.
23:21	Reserved	Must be kept at reset value.
20	CCK	Consecutive clock 0: EXMC_CLK is generated only during synchronous access. 1: EXMC_CLK is generated unconditionally. **Note**: Consecutive clock (CCK) bit is only present in EXMC_SNCTL0 register, and this bit position in EXMC_SNCTLx (x = 1, 2, 3) registers is meaningless. When this bit is set, only CKDIV[3:0] bits in EXMC_SNTCFG0 can effect EXMC_CLK output.
19	SYNCWR	Synchronous write 0: Asynchronous write

		1: Synchronous write
18:16	CPS[2:0]	CRAM page size 000: Automatic burst split on page boundary crossing 001: 128 bytes 010: 256 bytes 011: 512 bytes 100: 1024 bytes Others: Reserved
15	ASYNCWTEN	Asynchronous wait enable 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 to the NWAIT signal. 0: Disable NWAIT signal 1: Enable NWAIT signal
12	WEN	Write enable 0: Disable write in the bank by the EXMC, otherwise an AXI error is reported 1: Enable 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 **Note:** This bit need to deassert when PSRAM synchrous write.
10	Reserved	Must be kept at reset value.
9	NRWTPOL	NWAIT signal polarity 0: Low level of NWAIT is active 1: High level of NWAIT is active
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: 32 bits 11: Reserved
3:2	NRTP[1:0]	NOR region memory type 00: SRAM, ROM 01: PSRAM (CRAM) 10: NOR Flash 11: Reserved
1	NRMUX	NOR region memory address/data multiplexing 0: Disable address/data multiplexing function 1: Enable address/data multiplexing function
0	NRBKEN	NOR region enable 0: Disable the corresponding memory bank 1: Enable the corresponding memory bank

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)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DHLD[1:0]		ASYNCMOD[1:0]		DLAT[3:0]				CKDIV[3:0]				BUSLAT[3:0]			
rw		rw		rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSET[7:0]								AHLD[3:0]				ASET[3:0]			
rw								rw				rw			

Bits	Fields	Descriptions
31:30	DHLD[1:0]	Data hold time This field is meaningful only in asynchronous access. For read accesses: 00: Data hold time = 0 * CK_EXMC 01: Data hold time = 1 * CK_EXMC 10: Data hold time = 2 * CK_EXMC 11: Data hold time = 3 * CK_EXMC For write accesses: 00: Data hold time = 1 * CK_EXMC 01: Data hold time = 2 * CK_EXMC 10: Data hold time = 3 * CK_EXMC 11: Data hold time = 4 * CK_EXMC

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 CLK 0x1: Data latency of first burst access is 3 CLK …… 0xF: Data latency of first burst access is 17 CLK
23:20	CKDIV[3:0]	Synchronous clock divide ratio. This filed is only effect in synchronous mode. 0x0: No EXMC_CLK output 0x1: EXMC_CLK period = 2 * CK_EXMC period …… 0xF: EXMC_CLK period = 16 * CK_EXMC period
19:16	BUSLAT[3:0]	Bus latency The bits are defined in multiplexed read mode in order to avoid bus contention, and the bits represent the minimum time the data bus used to return to a high impedance state. 0x0: Bus latency = 0 * CK_EXMC period 0x1: Bus latency = 1 * CK_EXMC period …… 0xF: Bus latency = 15 * CK_EXMC period
15:8	DSET[7:0]	Data setup time This field is meaningful only in asynchronous access. 0x00: Reserved 0x01: Data setup time = 1 * CK_EXMC period …… 0xFF: Data setup time = 255 * CK_EXMC 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 * CK_EXMC …… 0xF: Address hold time = 15 * CK_EXMC
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 * CK_EXMC

……

0xF: Address setup time = 15 * CK_EXMC

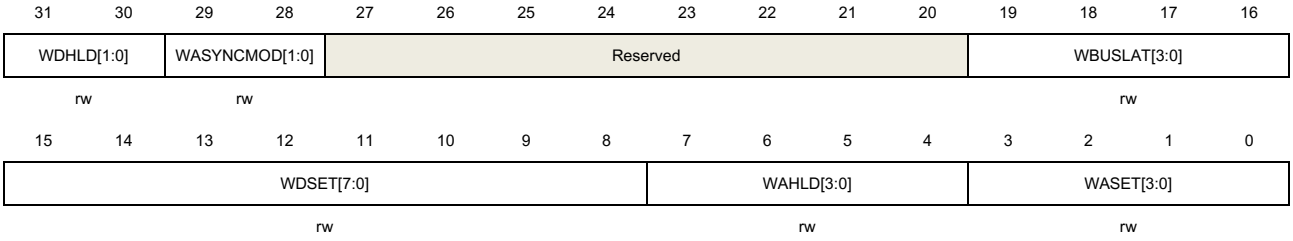
SRAM/NOR flash write timing configuration registers (EXMC_SNWTCFGx) (x=0, 1, 2, 3)

Address offset: 0x104 + 8 * x, (x = 0, 1, 2, and 3)

Reset value: 0x0FFF FFFF

This register is meaningful only when the EXMODEN bit in EXMC_SNCTLx is set to 1.

This register has to be accessed by word(32-bit)



Bits	Fields	Descriptions
31:30	WDHLD[1:0]	Data hold time This field is meaningful only in asynchronous access. 00: Data hold time = 1 * CK_EXMC 01: Data hold time = 2 * CK_EXMC 10: Data hold time = 3 * CK_EXMC 11: Data hold time = 4 * CK_EXMC
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 * CK_EXMC period 0x1: Bus latency = 1 * CK_EXMC period …… 0xF: Bus latency = 15 * CK_EXMC period

15:8	WDSET[7:0]	Data setup time This field is meaningful only in asynchronous access. 0x00: Reserved 0x01: Data setup time = 1 * CK_EXMC period …… 0xFF: Data setup time = 255 * CK_EXMC 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 * CK_EXMC …… 0xF: Address hold time = 15 * CK_EXMC
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 * CK_EXMC 0x1: Address setup time = 1 * CK_EXMC …… 0xF: Address setup time = 15 * CK_EXMC

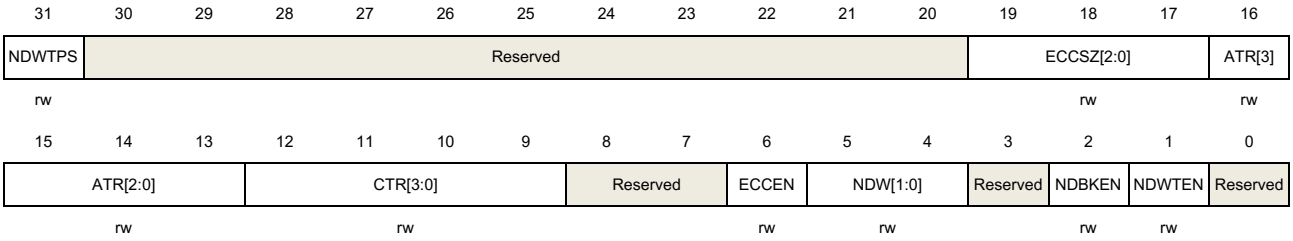
48.4.2. NAND flash controller registers

NAND flash control registers (EXMC_NCTL)

Address offset: 0x80

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)



Bits	Fields	Descriptions
31	NDWTPS	NAND NWAIT pin select **Note:** The configuration of GPIO specifically refers to the GPIO section and the datasheet. 0: Select shared EXMC_NWAIT(PC6/PD6) 1: Select EXMC_NDNWAIT(PA1)

30:20	Reserved	Must be kept at reset value.
19:17	ECCSZ[2:0]	ECC size 000: 256 bytes 001: 512 bytes 010: 1024 bytes 011: 2048 bytes 100: 4096 bytes 101: 8192 bytes
16:13	ATR[3:0]	ALE to RE delay 0x0: ALE to RE delay = 1 * CK_EXMC …… 0xF: ALE to RE delay = 16 * CK_EXMC
12:9	CTR[3:0]	CLE to RE delay 0x0: CLE to RE delay = 1 * CK_EXMC 0x1: CLE to RE delay = 2 * CK_EXMC …… 0xF: CLE to RE delay = 16 * CK_EXMC
8:7	Reserved	Must be kept at reset value.
6	ECCEN	ECC enable 0: Disable ECC, and reset EXMC_NECC 1: Enable ECC
5:4	NDW[1:0]	NAND bank memory data bus width 00: 8 bits 01: 16 bits Others: Reserved
3	Reserved	Must be kept at reset value.
2	NDBKEN	NAND bank enable 0: Disable corresponding memory bank 1: Enable corresponding memory bank
1	NDWTEN	Wait function enable 0: Disable wait function 1: Enable wait function
0	Reserved	Must be kept at reset value.

NAND flash interrupt enable registers (EXMC_NINTEN)

Address offset: 0x84

Reset value: 0x0000 0042

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									FFEPT	INTFEN	INTHEN	INTREN	INTFS	INTHS	INTRS
									r	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value.
6	FFEPT	FIFO empty flag It is cleared by hardware. 0: FIFO is not empty. 1: FIFO is empty.
5	INTFEN	Interrupt falling edge detection enable 0: Disable interrupt falling edge detection 1: Enable interrupt falling edge detection
4	INTHEN	Interrupt high-level detection enable 0: Disable interrupt high-level detection 1: Enable interrupt high-level detection
3	INTREN	Interrupt rising edge detection enable bit 0: Disable interrupt rising edge detection 1: Enable interrupt rising edge detection
2	INTFS	Interrupt falling edge status Write 1 to this bit, this bit will be set. 0: Not detect interrupt falling edge 1: Detect interrupt falling edge
1	INTHS	Interrupt high-level status This bit is set by hardware and cleared by software. 0: Not detect interrupt high-level 1: Detect interrupt high-level
0	INTRS	Interrupt rising edge status Write 1 to this bit, this bit will be set. 0: Not detect interrupt rising edge 1: Detect interrupt rising edge

NAND flash common space timing configuration registers (EXMC_NCTCFG)

Address offset: 0x88

Reset value: 0xFFFF FFFF

These operations applicable to common memory space for NAND Flash.

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
COMHIZ[7:0]								COMHLD[7:0]							
rw								rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COMWAIT[7:0]								COMSET[7:0]							
rw								rw							

Bits	Fields	Descriptions
31:24	COMHIZ[7:0]	Common memory data bus HIZ time The bits are defined as time of bus keep high impedance state after writing the data. 0x00: COMHIZ = 1 * CK_EXMC …… 0xFE: COMHIZ = 255 * CK_EXMC 0xFF: Reserved
23:16	COMHLD[7:0]	Common memory hold time After sending the address, the bits are defined as the address hold time. In write operation, they are also defined as the data signal hold time. 0x00: Reserved 0x01: COMHLD = 1 * CK_EXMC …… 0xFE: COMHLD = 254 * CK_EXMC 0xFF: Reserved
15:8	COMWAIT[7:0]	Common memory wait time Define the minimum time to maintain command 0x00: Reserved 0x01: COMWAIT = 2 * CK_EXMC (+NWAIT active cycles) …… 0xFE: COMWAIT = 255 * CK_EXMC (+NWAIT active cycles) 0xFF: Reserved
7:0	COMSET[7:0]	Common memory setup time Define the time to build address before sending command 0x00: COMSET = 1 * CK_EXMC …… 0xFE: COMSET = 255 * CK_EXMC 0xFF: Reserved

NAND flash attribute space timing configuration registers (EXMC_NATCFG)

Address offset: 0x8C

Reset value: 0xFFFF FFFF

It is used for 8-bit accesses to the attribute memory space of the NAND Flash for the last address write access if another timing must be applied.

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ATTHIZ[7:0]								ATTHLD[7:0]							
rw								rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ATTWAIT[7:0]								ATTSET[7:0]							
rw								rw							

Bits	Fields	Descriptions
31:24	ATTHIZ[7:0]	Attribute memory data bus HIZ time The bits are defined as time of bus keep high impedance state after writing the data. 0x00: ATTHIZ = 0 * CK_EXMC …... 0xFE: ATTHIZ = 254 * CK_EXMC 0xFF: Reserved
23:16	ATTHLD[7:0]	Attribute memory hold time After sending the address, the bits are defined as the address hold time. In write operation, they are also defined as the data signal hold time. 0x00: Reserved 0x01: ATTHLD = 1 * CK_EXMC …… 0xFE: ATTHLD = 254 * CK_EXMC 0xFF: Reserved
15:8	ATTWAIT[7:0]	Attribute memory wait time Define the minimum time to maintain command 0x00: Reserved 0x01: ATTWAIT = 2 * CK_EXMC (+NWAIT active cycles) …… 0xFE: ATTWAIT = 255 * CK_EXMC (+NWAIT active cycles) 0xFF: ATTWAIT = Reserved
7:0	ATTSET[7:0]	Attribute memory setup time Define the time to build address before sending command 0x00: ATTSET = 1 * CK_EXMC …… 0xFE: ATTSET = 255 * CK_EXMC 0xFF: Reserved

NAND flash ECC registers (EXMC_NECC)

Address offset: 0x94

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ECC[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECC[15:0]															
r															

Bits	Fields	Descriptions
31:0	ECC[31:0]	ECC result

ECCSZ[2:0]	NAND Flash page size	ECC bits
0b000	256	ECC[21:0]
0b001	512	ECC[23:0]
0b010	1024	ECC[25:0]
0b011	2048	ECC[27:0]
0b100	4096	ECC[29:0]
0b101	8192	ECC[31:0]

48.4.3. SDRAM controller registers

SDRAM control registers (EXMC_SDCTLx) (x=0, 1)

Address offset: 0x140+4*x, (x = 0, 1)

Reset value: 0x0000 02D0

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDCLK[2]	PIPED[1:0]		BRSTRD	SDCLK[1:0]		WPEN	CL[1:0]		NBK	SDW[1:0]		RAW[1:0]		CAW[1:0]	
rw	rw		rw	rw		rw	rw		rw	rw		rw		rw	

Bits	Fields	Descriptions
31:16	Reserved	Forced by hardware to 0.
15	SDCLK[2]	Refer to SDCLK[1:0] description.
14:13	PIPED[1:0]	Pipeline delay

		These bits specify the delay for reading data after CAS latency in CK_EXMC clock cycles. 00: 0 CK_EXMC clock cycle delay 01: 1 CK_EXMC clock cycle delay 10: 2 CK_EXMC clock cycle delay 11: reserved **Note:** The corresponding bits in the EXMC_SDCTL1 register are reserved.
12	BRSTRD	Burst read When this bit is set, The SDRAM controller anticipates the next read commands during the CAS latency and stores data in the Read FIFO. 0: burst read disabled 1: burst read enabled **Note:** The corresponding bits in the EXMC_SDCTL1 register are reserved.
11:10	SDCLK[1:0]	SDRAM clock configuration These bits specifies the SDRAM clock period for both SDRAM devices. The memory clock should be disabled before change, and the SDRAM memory must be re-initialized after this configuration is changed. 000: SDCLK memory clock disabled 001: Reserved 010: SDCLK memory period = 2 x CK_EXMC periods 011: SDCLK memory period = 3 x CK_EXMC periods 110: SDCLK memory period = 4 x CK_EXMC periods 111: SDCLK memory period = 5 x CK_EXMC periods Others: Reserved **Note:** The corresponding bits in the EXMC_SDCTL1 register are reserved. The SDCLK[2] bit is not contiguous, which is located in bit 15.
9	WPEN	Write protection enable This bit enables the write protection function. 0: Disable write protection, write accesses allowed 1: Enable write protection, write accesses ignored
8:7	CL[1:0]	CAS Latency This bits sets specifies SDRAM CAS latency in SDRAM memory clock cycle unit 00: reserved, do not use. 01: 1 cycle 10: 2 cycles 11: 3 cycles
6	NBK	Number of banks This bit specifies the number of internal banks. 0: 2 internal Banks 1: 4 internal Banks
5:4	SDW[1:0]	SDRAM data bus width.

		These bits specify the SDRAM memory data width. 00: 8 bits 01: 16 bits 10: 32 bits 11: reserved
3:2	RAW[1:0]	Row address bit width These bits specify the bit width of a row address. 00: 11 bit 01: 12 bits 10: 13 bits 11: reserved
1:0	CAW[1:0]	Column address bit width These bits specify the bit width of column address. 00: 8 bits 01: 9 bits 10: 10 bits 11: 11 bits

SDRAM timing configuration registers (EXMC_SDTCFGx) (x=0, 1)

Address offset: 0x148+4*x, (x = 0, 1)

Reset value: 0x0FFF FFFF

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				RCD[3:0]				RPD[3:0]				WRD[3:0]			
				rw				rw				rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARFD[3:0]				RASD[3:0]				XSRD[3:0]				LMRD[3:0]			
rw				rw				rw				rw			

Bits	Fields	Descriptions
31:28	Reserved	Forced by hardware to 0.
27:24	RCD[3:0]	Row to column delay These bits specify the delay between an Activate command and a Read / write command in SDRAM memory clock cycle unit. 0x0: 1 cycle. 0x1: 2 cycles .... 0xF: 16 cycles
23:20	RPD[3:0]	Row precharge delay

Bits	Fields	Descriptions
		These bits specify the delay between a Precharge command and the next command in SDRAM memory clock cycle unit. 0x0: 1 cycle 0x1: 2 cycles .... 0xF: 16 cycles **Note:** The corresponding bits in the EXMC_SDTCFG1 register are reserved. If two SDRAM memories are used, the RPD must be programmed with the timings of the slower one.
19:16	WRD[3:0]	Write recovery delay These bits specify the delay between a Write and a Precharge command in SDRAM memory clock cycle unit. 0x0: 1 cycle 0x1: 2 cycles ..... 0xF: 16 cycles **Note:** The corresponding bits in the EXMC_SDTCFG1 register are reserved. If two SDRAM memories are used, the WRD must be programmed with the timings of the slower one.
15:12	ARFD[3:0]	Auto refresh delay These bits specify the delay between two consecutive Refresh commands, the delay between two Activate commands, as well as the delay between the Refresh command and the Activate command in SDRAM memory clock cycle unit. 0x0: 1 cycle 0x1: 2 cycles .... 0xF: 16 cycles **Note:** The corresponding bits in the EXMC_SDTCFG1 register are reserved. If two SDRAM memories are used, the ARFD must be programmed with the timings of the slower one.
11:8	RASD[3:0]	Row address select delay These bits specify the delay between an Activate command and a Precharge command in SDRAM memory clock cycle unit. The minimum delay between two successive Self-refresh commands is also specified by these bits. 0x0: 1 cycle 0x1: 2 cycles ..... 0xF: 16 cycles
7:4	XSRD[3:0]	Exit Self-refresh delay These bits specify the delay from a Self-refresh command to an Activate command in SDRAM memory clock cycle unit.

Bits	Fields	Descriptions
		0x0: 1 cycle 0x1: 2 cycles …… 0xF: 16 cycles
3:0	LMRD[3:0]	Load Mode Register Delay These bits specify the delay between a Load Mode Register command and a Refresh or Active command in SDRAM memory clock cycle unit. 0x0: 1 cycle 0x1: 2 cycles ..... 0xF: 16 cycles

SDRAM command register (EXMC_SDCMD)

Address offset: 0x150

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										MRC[12:7]					
										rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MRC[6:0]							NARF[3:0]				DS0	DS1	CMD[2:0]		
rw							rw				rw	rw	rw		

Bits	Fields	Descriptions
31:22	Reserved	Forced by hardware to 0.
21:9	MRC[12:0]	Mode register content These bits specify the SDRAM Mode Register content which will be programmed when CMD = '100'.
8:5	NARF[3:0]	Number of successive Auto-refresh These bits specify how many successive Auto-refresh cycles will be send when CMD = '011'. 0x0: 1 Auto-refresh cycle 0x1: 2 Auto-refresh cycles .... 0xE: 15 Auto-refresh cycles 0xF: Reserved
4	DS0	Device select 0 This bit indicates whether the SDRAM Device0 is selected or not. This bit is read as 0.

		0: SDRAM Device0 is not selected 1: SDRAM Device0 is selected
3	DS1	Device select 1 This bit indicates whether the SDRAM Device1 is selected or not. This bit is read as 0. 0: SDRAM Device1 is not selected 1: SDRAM Device1 is selected
2:0	CMD[2:0]	Command These bits specify the commands, which are issued to the SDRAM device. These bits are read as 0. 000: Normal operation command 001: Clock enable command 010: Precharge All command 011: Auto-refresh command 100: Load Mode Register command 101: Self-refresh command 110: Power-down entry command 111: Reserved **Note:** At least one command device select bit (DS1 or DS0) must be set, when a command is issued. If both devices are used, the commands must be issued to the two devices by setting the DS1and DS0 bits at the same time.

SDRAM auto-refresh interval register (EXMC_SDARI)

Address offset: 0x154

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	REIE	ARINTV[12:0]													REC
	rw	rw													w

Bits	Fields	Descriptions
31:15	Reserved	Forced by hardware to 0.
14	REIE	Refresh error interrupt Enable 0: Interrupt is disabled 1: An Interrupt is generated if REIF bit of the Status Register is set
13:1	ARINTV[12:0]	Auto-Refresh Interval

		This bit field specifies the interval of two successive auto-refresh commands in memory clock cycle unit. ARINTV = (SDRAM refresh period / Number of rows) - 20
0	REC	Refresh error flag clear The Refresh Error Flag (REIF) in the Status Register will be cleared when this bit is set. 0: no effect 1: Clear the Refresh Error flag

SDRAM status register (EXMC_SDSTAT)

Address offset: 0x158

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										NRDY	STA1[1:0]		STA0[1:0]		REIF
										r	r		r		r

Bits	Fields	Descriptions
31:6	Reserved	Forced by hardware to 0.
5	NRDY	Not Ready status This bit specifies whether the SDRAM controller is ready for a new command 0: SDRAM Controller is ready for a new command 1: SDRAM Controller is not ready for a new command
4:3	STA1[1:0]	Device1 status This bit defines the Status of SDRAM Device1. 00: Normal status 01: Self-refresh status 10: Power-down status
2:1	STA0[1:0]	Device 0 status This bit defines the Status of SDRAM Device 0. 00: Normal status 01: Self-refresh status 10: Power-down status
0	REIF	Refresh error interrupt flag 0: No refresh error 1: A refresh error occurred. An interrupt is generated when REIE = 1.

SDRAM read sample control register (EXMC_SDRSCTL)

Address offset: 0x180

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								SDSC[3:0]				Reserved		SSCR	RSEN
								rw						rw	rw

Bits	Fields	Descriptions
31:8	Reserved	Forced by hardware to 0.
7:4	SDSC[3:0]	Select the delayed sample clock of read data 0x0: Select the clock after 0 delay cell 0x1: Select the clock after 1 delay cell …… 0xF: Select the clock after 15 delay cell
3:2	Reserved	Forced by hardware to 0.
1	SSCR	Select sample cycle of read data 0: add 0 extra CK_EXMC cycle to the read data sample clock besides the delay chain 1: add 1 extra CK_EXMC cycle to the read data sample clock besides the delay chain
0	RSEN	Read sample enable 0: Read sample disabled 1: Read sample enabled

49. Digital camera interface (DCI)

49.1. Overview

DCI is a parallel interface to capture video or picture from a camera. It supports various color space such as YUV/RGB, as well as compression format such as JPEG. Support CCIR656 video decoder formats and perform additional processing of the image.

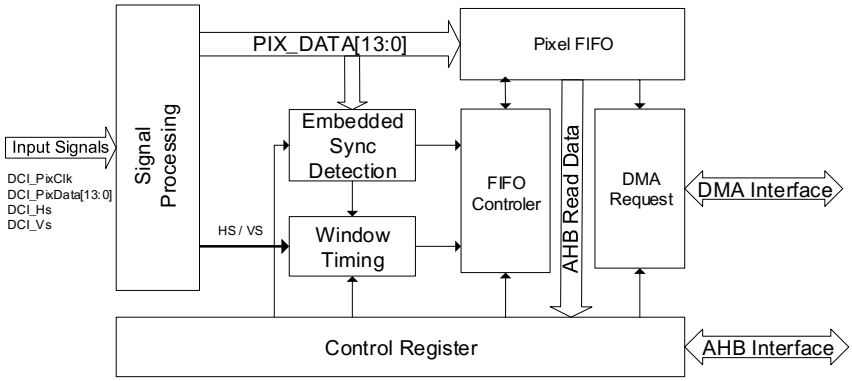
49.2. Characteristics

- Digital video/picture capture
- 8/10/12/14 data width supported
- High transfer efficiency with DMA interface
- Video/picture crop supported
- Various pixel digital encoding formats supported including YcbCr422/RGB565/YUV420/Bayer
- JPEG compression format supported
- Hard/embedded synchronous signals supported
- Support for CCIR656 video interface as well as traditional sensor interface.

49.3. Block diagram

The DCI contains these modules: Signal Processing, Pixel FIFO, FIFO controller, window timing, embedded sync detection, DMA interface and control register.

Figure 49-1. DCI module block diagram



The signal processing module generates useful signals for other internal modules from external input signals. The frequency of HCLK should be 2.5 times higher than DCI_PixClk to ensure the proper operation of signal processing module.

The embedded sync detection module is designed to support embedded synchronization

mode. In DCI embedded synchronization mode, video synchronization information is embedded into pixel data and there is no hardware horizontal or vertical synchronization signal (DCI_HSYNC or DCI_VSYNC). DCI uses embedded sync detection module to extract synchronization information from pixel data, and then recover horizontal and vertical synchronization signals.

The window timing module performs image cutting function. This module calculates a pixel's position using synchronization signals either from DCI interface or embedded sync detection module and then decides whether this pixel data needs to be received according to the configuration of DCI_CWSPOS and DCI_CWSZ registers.

DCI uses a 4 word (32-bit) FIFO to buffer the received pixel data. If DMA mode is enabled, the DMA interface asserts a DMA request every time a 32-bit data is received. Control register provides register interface between DCI and software.

49.4. Signal description

Table 49-1. Pins used by DCI

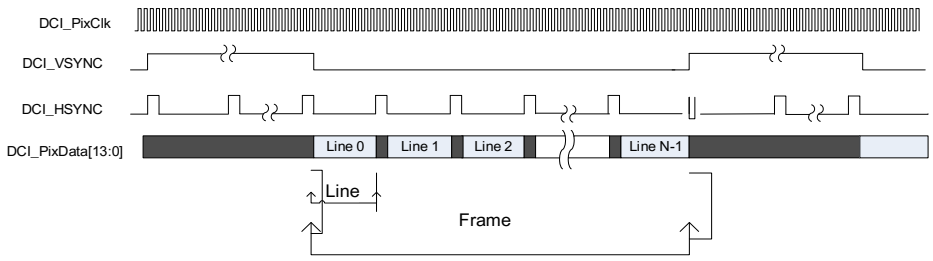
Direction	Pins	Signals	Width	Description
Input	DCI_PIXCLK	DCI_PixClk	1	DCI Pixel Clock
Input	DCI_Dx	DCI_PixData	14	DCI Pixel Data
Input	DCI_HSYNC	DCI_HSYNC	1	DCI Horizontal Synchronization
Input	DCI_VSYNC	DCI_VSYNC	1	DCI Vertical Synchronization

49.5. Function overview

49.5.1. DCI hardware synchronization mode

In DCI hardware synchronization mode (ESM bit in DCI_CTL register is 0), DCI_HSYNC and DCI_VSYNC signals are used to indicate the start of a line and a frame. DCI captures pixel data from DCI_PixData[13:0] at rising or falling edge of DCI_PixClk (clock polarity is configured by CKS bit in DCI_CTL).

Figure 49-2. Hardware synchronization mode



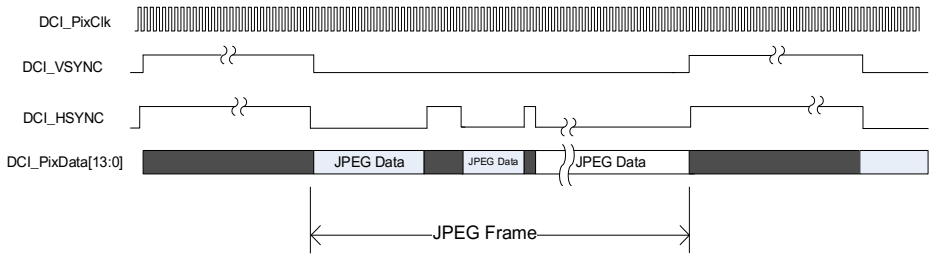
The above figure assumes that the polarities of both DCI_HSYNC and DCI_VSYNC are high during blanking period, so the data on DCI_PixData lines is only valid when both

DCI_HSYNC and DCI_VSYNC are low.

JPEG mode

DCI supports JPEG video/picture compression format in hardware synchronization mode. In JPEG mode (JM bit in DCI_CTL is set), the DCI_VSYNC is used to indicate start of a new frame, and DCI_HSYNC is used as stream data valid signal.

Figure 49-3. Hardware synchronization mode: JPEG format supporting



49.5.2. Embedded synchronization mode

DCI supports embedded synchronization mode. In this mode there are only DCI_PixData and DCI_PixClk signals in DCI interface and the synchronization information is embedded in the pixel data. This mode is enabled by setting ESM bit and clearing JM bit in DCI_CTL register.

In embedded synchronization mode, line and frame synchronization information is encoded into sync code and embedded into the pixel data. There are four kinds of sync code: Line Start(LS), Line End(LE), Frame Start(FS) and Frame End(FE). In this mode the data width is forced to 8 and each sync code is composed by 4-byte sequence: FF-00-00-XY and XY is defined in DCI_SC register. In embedded synchronization mode, the 0xFF and 0x00 should not appear in pixel data to avoid mistake.

In embedded synchronization mode, DCI starts to detect the sync codes after enabled and recover line/frame synchronization information. For example, DCI starts to capture a new frame if it detects a Frame End code and then a Frame Start Code.

When detecting sync code, it is possible to make DCI compare only a few bits of XY byte in FF_00_00_XY sequence by configuring sync code unmask register (DCI_SCUMSK). DCI will only compare bits unmasked by DCI_SCUMSK register. For example: LS in DCI_SC register is A5 and LSM in DCI_SCUMSK is F0, then DCI will only compare the higher 4 bits for LS sync code and thus, FF-00-00-A6 sequence will also be detected as a LS code.

49.5.3. CCIR656 mode

Interlace mode

CCIR656 standard use an embedded timing codec to replace the VSYNC and HSYNC signal. Only the PIXCLK and DCI_PixData[7:0] signals are used in interlace mode. Code (SAV)

represents the beginning of each valid row, and code (EAV) is the end of each valid row. Blanking data inserts are inserted between the EAV and SAV code. The DCI decode the embedded timing codec, according to the information of the field, the original YcbCr image is rearranged in the software.

Table 49-2. Typical one-line data composition

EAV Code				Blanking Video				SAV Code				Active Video			
FF	0	0	XY	Cb	Y	Cr	Y	FF	0	0	XY	Cb	Y	Cr	Y
4 Bytes				280(268) Bytes				4 Bytes				1440 Bytes			

The 4-byte format of the EAV and SAV is specified as follows (represented in hexadecimal below):FF 00 00 XY

The first three bytes are fixed and must be FF 00 00, while the fourth byte (XY) is determined by the field and the hidden information, and its eight bits are defined as follows: 1 F V H P3 P2 P1 P0.

F: marks the field information, which is 0 when transmitting the even field and 1 when transmitting the odd field.

V: Marks the blanking information, which is 1 when the blanking data is transmitted and 0 when the valid video data is transmitted.

H: Label EAV or SAV, where SAV is 0 and EAV is 1.

P0~P3 is the protection bit, and its value depends on F, H and V, which plays the role of calibration. The calculation method is as follows:

Table 49-3. Coding for SAV and EAV

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	**F**	**V**	**H**	**P3**	**P2**	**P1**	**P0**
1	0	0	0	0	0	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	0	1	1
1	0	1	1	0	1	1	0
1	1	0	0	0	1	1	1
1	1	0	1	1	0	1	0
1	1	1	0	1	1	0	0
1	1	1	1	0	0	0	1

The DCI decodes and filters out the timing-coding from the data stream, recovering VSYNC and HSYNC signals for internal use, such as statistical block control. Data is forwarded sequentially to the data receiving without reordering, field 0 followed by field 1. The fields must be reordered in software to get back the original image.

The Change of Field interrupt Flag(COFIF) flag bit will be set when an alternation occurs on the spot. CCIR656 standard images are generally in 625/50 PAL or 525/60 NTSC format. The image is composed of odd and even fields, horizontal and vertical blanking and valid data. A

field consists of three parts, top vertical blanking, valid data and bottom vertical blanking.

The following table shows the frame information in PAL/NTSC system.

Table 49-4. A frame image format definition

Lines		Field/VBlk		Line Description
PAL	NTSC	F	V	
22	19	0	1	Field 0 top vertical blanking
288	240	0	0	Field 0 active video
2	3	0	1	Field 0 bottom vertical blanking
23	20	1	1	Field 1 top vertical blanking
288	240	1	0	Field 1 active video
2	3	1	1	Field 1 bottom vertical blanking
625	525			

Progressive Mode

Images are scanned and arranged in a progressive manner. This activity field is considered to be field 1 and the DCI decoding ignores the F bit in the SAV/EAV code.

Table 49-5. General case of progressive mode

EAV	Blanking	SAV	Blanking	Field 1 (F = 1)
⋮				
EAV	Blanking	SAV	Blanking	
EAV	Blanking	SAV	Data	
⋮				
EAV	Blanking	SAV	Data	

In progressive mode, COFIF will be ignored, but VSIF can generate an interrupt. When the VSYNC signal is provided by the camera sensor, it is called external VSYNC mode, and when the VSIF flag bit is decoded from the embedded code, it is called internal VSYNC mode. DCI can run both internal and external VSYNC mode.

Error Correction for CCIR656 Coding

According to the CCIR coding algorithm, the guard bits in SAV and EAV are coded in a way that allows 1-bit errors to be corrected, or 2-bit errors to be detected by the decoder. This feature is supported by the interlaced mode CCIR decoder in DCI.

For 1-bit error conditions, the user can choose to correct the error automatically, or simply display it as a status flag. For 2-bit error cases, the error will only be displayed as a status flag since the decoder cannot make corrections.

When a CCIR error is detected, the CCEF flag in the DCI_STAT1 register is set. If the CCIR error interrupt is enabled (CCEIE is set), an interrupt is generated and the CCEIF flag in the DCI_INTF register is set. If automatic error correction is enabled (AECEN is set in the DCI_CTL register), 1-bit errors are corrected automatically. 2-bit errors cannot be corrected

and are only indicated by the error flags.

49.5.4. Capture data using snapshot or continuous capture modes

The DCI supports two capture modes: snapshot and continuous capture. Capture mode is configured by SNAP bit in DCI_CTL register.

After correct configuration, enable the DCI and set the CAP bit in the DCI_CTL register, the DCI begins to detect the frame start. It begins to capture data once a frame start is detected. In snapshot mode (SNAP = 1), the DCI automatically stops capturing and clears the CAP bit after a whole frame is captured, while in continuous mode, the DCI prepares to capture the next frame. In continuous mode, the capture frequency is defined by the FR[1:0] bits. For example, if FR[1:0] = 00, the DCI captures every frame, and if FR[1:0] = 01, the DCI captures one in every two frames.

In continuous mode, software may clear the CAP bit at any time while the DCI is capturing data, but the DCI does not stop capturing immediately. It always stops after the current frame ends. Software should read back the CAP bit to check whether the DCI has stopped.

Use the end of frame flag (EFF) to count the captured frames. In CCIR656 interlaced mode, the end of field 1 (the second field of the image, F = 1) is regarded as the end of a frame.

49.5.5. Window function

DCI supports a window function, which is able to crop a part of the image from the captured frame.The window function is enabled by setting the WDEN bit in the DCI_CTL register, and it is disabled in JPEG mode.

The DCI continuously counts and calculates the horizontal and vertical positions of pixels during capturing, compares the positions with the values in the crop window registers (DCI_CWSPOS and DCI_CWSZ), discards the pixels outside the crop window, and only pushes the pixels inside the window into the pixel FIFO.

If a frame ends before the number of vertical lines defined in DCI_CWSZ is reached, the end of frame flag is triggered and the DCI stops capturing. In CCIR656 interlaced mode, each field image is cropped by the window.

49.5.6. Pixel formats, data padding and DMA

DCI supports various pixel digital encoding formats including YcbCr422/RGB565. However, DCI only receives these pixel data, pads these pixels into a word and push into a pixel FIFO. DCI doesn't perform any pixel format conversion or data processing and doesn't care about the detail of pixel format.

DCI uses a 32-bits width data buffer to transfer between DCI interface and pixel FIFO. These are two padding method in this module: byte padding and half-word padding, depending on the data width of DCI interface. Data width is configured by DCIF[1:0] in DCI_CTL register.

The data width is fixed to 8 in JPEG mode and embedded synchronization mode.

The DMA interface sends DMA request when a 32-bit data is received.

Byte padding mode

Byte padding mode is used if data width of DCI interface is 8. In byte padding mode four bytes are filled into the 32-bits width data buffer. In Non-JPEG mode, the DCI pushes the 32-bits buffer's data into the pixel FIFO when the buffer is full or meets the end of a line. In JPEG mode, the DCI pushes the 32-bits buffer's data into the pixel FIFO when the buffer is full or meets the end of a frame.

Table 49-6. Memory view in byte padding mode

D3[7:0]	D2[7:0]	D1[7:0]	D0[7:0]
D7[7:0]	D6[7:0]	D5[7:0]	D4[7:0]

Half-word padding mode

Half-word padding is used if data width of DCI interface is configured into 10/12/14. In this mode each pixel data is extended into 16-bits length by filling zero at higher position, so the 32-bits width data buffer is able to hold two pixel data. DCI pushes the data buffer into pixel FIFO each time the buffer is full or line end.

Table 49-7. Memory view in half-word padding mode

2'b00	D1[13:0]	2'b00	D0[13:0]
2'b00	D3[13:0]	2'b00	D2[13:0]
2'b00	D5[13:0]	2'b00	D4[13:0]
2'b00	D7[13:0]	2'b00	D6[13:0]

49.6. Interrupts

There are several status and error flags in DCI, and interrupts may be asserted from these flags. These status and error flags will assert global DCI interrupt if enabled by corresponding bit in DCI_INTEN. These flags are cleared by writing 1 into DCI_INTC register.

Table 49-8. Status/Error and interrupt flags

Status Flag Name	Description
CCEIF	CCIR Error Interrupt Flag
COFIF	CCIR Change of Field Interrupt Flag
F1IF	CCIR Field 1 Interrupt Flag
F0IF	CCIR Field 0 Interrupt Flag
ELF	End of Line Flag
EFF	End of Frame Flag
OVRF	FIFO Overrun Flag

VSF	Frame VS Blank Flag
ESEF	Embedded Sync Error Flag

49.7. Register definition

DCI base address: 0x4802 0000

49.7.1. Control register (DCI_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															EVSEN
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AECEN	DCIEN	CCMOD	CCEN	DCIF[1:0]		FR[1:0]		VPS	HPS	CKS	ESM	JM	WDEN	SNAP	CAP
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	EVSEN	External vsync enable 0: External vsync mode is disabled 1: External vsync mode is enabled Only works in CCIR progressive mode
15	AECEN	Automatic error correction enable, 1bit correction 0: Automatic error correction is disabled 1: Automatic error correction is enabled Only works in CCIR interlaced mode
14	DCIEN	DCI Enable 0: DCI is disabled 1: DCI is enabled
13	CCMOD	CCIR mode select 0: CCIR progressive mode 1: CCIR interlace mode
12	CCEN	CCIR enable 0: CCIR is disabled 1: CCIR is enabled
11:10	DCIF[1:0]	Digital camera interface format 00: 8-bit data on every pixel clock 01: 10-bit data on every pixel clock 10: 12-bit data on every pixel clock

		11: 14-bit data on every pixel clock
9:8	FR[1:0]	Frame rate FR defines the frame capture rate in continuous capture mode 00: Capture all frames 01: Capture one in 2 frames 10: Capture one in 4 frames 11: Reserved
7	VPS	Vertical polarity selection 0: Low level during blanking period 1: High level during blanking period
6	HPS	Horizontal polarity selection 0: Low level during blanking period 1: High level during blanking period
5	CKS	Clock polarity selection 0: Capture at falling edge 1: Capture at rising edge
4	ESM	Embedded synchronous mode 0: Embedded synchronous mode is disabled 1: Embedded synchronous mode is enabled
3	JM	JPEG mode 0: JPEG mode is disabled 1: JPEG mode is enabled
2	WDEN	Window enable 0: Window is disabled 1: Window is enabled
1	SNAP	Snapshot mode 0: Continuous capture mode 1: Snapshot capture mode
0	CAP	Capture enable 0: Frame not captured 1: Frame is captured

49.7.2. Status register0 (DCI_STAT0)

Address offset: 0x04

Reset value: 0x0000 0003

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Reserved

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													FV	VS	HS
													r	r	r

Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value.
2	FV	FIFO valid 0: No valid pixel data in FIFO 1: Valid pixel data in FIFO
1	VS	VS line status 0: Not in vertical blanking period 1: In vertical blanking period
0	HS	HS line status 0: Not in horizontal blanking period 1: In horizontal blanking period

49.7.3. Status register1 (DCI_STAT1)

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							CCEF	COFF	F1F	F0F	ELF	VSF	ESEF	OVRF	EFF
							r	r	r	r	r	r	r	r	r

Bits	Fields	Descriptions
31:9	Reserved	Must be kept at reset value.
8	CCEF	CCIR error flag 0: No error occurs 1: Error is found on the SAV or EVA codes Only works in CCIR interlaced mode
7	COFF	CCIR change of field flag 0: No change of field event occurs 1: Change of field event occurs

		Only works in CCIR interlaced mode
6	F1F	CCIR field 1 0: Current field is not 1 1: Current field is 1 Only works in CCIR interlaced mode(cleared automatically when current field does not match)
5	F0F	CCIR field 0 0: Current field is not 0 1: Current field is 0 Only works in CCIR interlaced mode(cleared automatically when current field does not match)
4	ELF	End of line flag 0 : No end of line flag 1: A line is captured by DCI
3	VSF	Vsync flag 0: No vsync flag 1: A vsync blanking detected
2	ESEF	Embedded synchronous error flag 0: No embedded synchronous error flag 1: A embedded synchronous error detected
1	OVRF	FIFO overrun flag 0: No FIFO overrun 1: A FIFO overrun occurs
0	EFF	End of frame flag 0: No end of frame flag 1: A frame is captured by DCI

49.7.4. Interrupt enable register (DCI_INTEN)

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							CCEIE	COFIE	F1IE	F0IE	ELIE	VSIE	ESEIE	OVRIE	EFIE
							rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:9	Reserved	Must be kept at reset value.
8	CCEIE	CCIR error Interrupt enable 0: CCIR error flag won't generate interrupt 1: CCIR error flag will generate interrupt
7	COFIE	CCIR change of field interrupt enable 0: CCIR change of field flag won't generate interrupt 1: CCIR change of field flag will generate interrupt
6	F1IE	CCIR field 1 interrupt enable 0: CCIR field 1 flag won't generate interrupt 1: CCIR field 1 flag will generate interrupt
5	F0IE	CCIR field 0 interrupt enable 0: CCIR field 0 won't generate interrupt 1: CCIR field 0 flag will generate interrupt
4	ELIE	End of line interrupt enable 0: End of line flag won't generate interrupt 1: End of line flag will generate interrupt
3	VSIE	Vsync interrupt enable 0: Vsync flag won't generate interrupt 1: Vsync flag will generate interrupt
2	ESEIE	Embedded synchronous error interrupt enable 0: Embedded synchronous error flag won't generate interrupt 1: Embedded synchronous error flag will generate interrupt
1	OVRIE	FIFO overrun interrupt enable 0: FIFO overrun won't generate interrupt 1: FIFO overrun will generate interrupt
0	EFIE	End of frame interrupt enable 0: End of frame flag won't generate interrupt 1: End of frame flag will generate interrupt

49.7.5. Interrupt flag register (DCI_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							CCEIF	COFIF	F1IF	F0IF	ELIF	VSIF	ESEIF	OVRIF	EFIF
							r	r	r	r	r	r	r	r	r

Bits	Fields	Descriptions
31:9	Reserved	Must be kept at reset value.
8	CCEIF	CCIR error interrupt flag Only works in CCIR interlaced mode, error is found on the SAV or EVA codes.
7	COFIF	CCIR change of field interrupt flag Only works in CCIR interlaced mode
6	F1IF	CCIR field 1 interrupt flag Only works in CCIR interlaced mode(cleared automatically when current field does not match)
5	F0IF	CCIR field 0 interrupt flag Only works in CCIR interlaced mode(Cleared automatically when current field does not match)
4	ELIF	End of line interrupt flag
3	VSIF	Vsync interrupt flag
2	ESEIF	Embedded synchronous error interrupt flag
1	OVRIF	FIFO overrun interrupt flag
0	EFIF	End of frame interrupt flag

49.7.6. Interrupt flag clear register (DCI_INTC)

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							CCEFC	COFFC	F1FC	F0FC	ELFC	VSFC	ESEFC	OVRFC	EFFC
							w	w	w	w	w	w	w	w	w

Bits	Fields	Descriptions
31:9	Reserved	Must be kept at reset value.
8	CCEFC	CCIR error flag clear

		Write 1 to clear CCIR error flag
7	COFFC	CCIR change of field flag clear Write 1 to clear CCIR change of field flag
6	F1FC	CCIR field 1 interrupt flag clear Write 1 to clear CCIR field 1 interrupt flag
5	F0FC	CCIR field 0 interrupt flag clear Write 1 to clear CCIR field 0 interrupt flag
4	ELFC	End of line flag clear Write 1 to clear end of line flag
3	VSFC	Vsync flag clear Write 1 to clear vsync flag
2	ESEFC	Clear embedded synchronous error flag Write 1 to clear embedded synchronous error flag
1	OVRFC	Clear FIFO overrun flag Write 1 to clear FIFO overrun flag
0	EFFC	Clear end of frame flag Write 1 to clear end of frame flag

49.7.7. Synchronization codes register (DCI_SC)

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
FE[7:0]								LE[7:0]							
rw								rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LS[7:0]								FS[7:0]							
rw								rw							

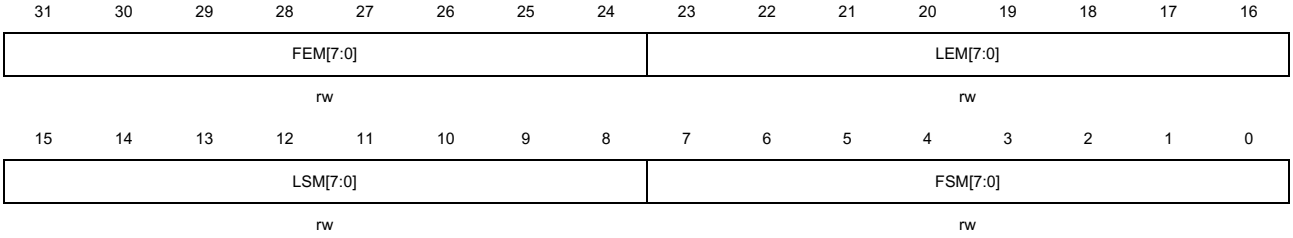
Bits	Fields	Descriptions
31:24	FE[7:0]	Frame end code in embedded synchronous mode
23:16	LE[7:0]	Line end code in embedded synchronous mode
15:8	LS[7:0]	Line start code in embedded synchronous mode
7:0	FS[7:0]	Frame start code in embedded synchronous mode

49.7.8. Synchronization codes unmask register (DCI_SCUMSK)

Address offset: 0x1C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)



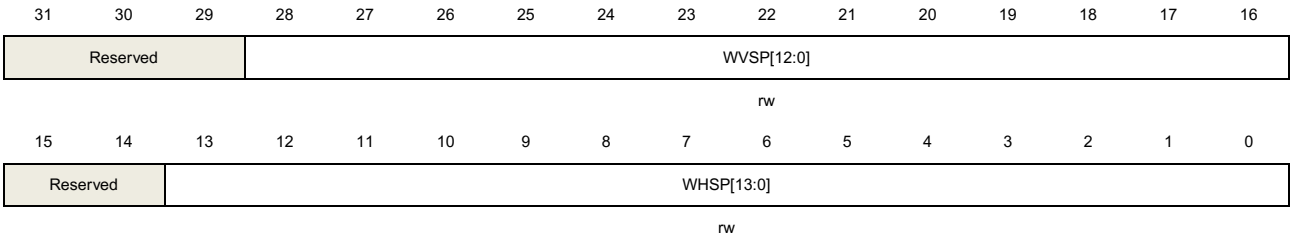
Bits	Fields	Descriptions
31:24	FEM[7:0]	Frame end code unmask bits in embedded synchronous mode
23:16	LEM[7:0]	Line end code unmask bits in embedded synchronous mode
15:8	LSM[7:0]	Line start code unmask bits in embedded synchronous mode
7:0	FSM[7:0]	Frame start code unmask bits in embedded synchronous mode

49.7.9. Cropping window start position register (DCI_CWSPOS)

Address offset: 0x20

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)



Bits	Fields	Descriptions
31:29	Reserved	Must be kept at reset value.
28:16	WVSP[12:0]	Window vertical start position Zero means the first line
15:14	Reserved	Must keep the reset value
13:0	WHSP[13:0]	Window horizontal start position Zero means the first pixel clock in a line

49.7.10. Cropping window size register (DCI_CWSZ)

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		WVSZ[13:0]													
		rw													

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		WHSZ[13:0]													
		rw													

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29:16	WVSZ[13:0]	Window vertical size WVSZ=X means X+1 lines
15:14	Reserved	Must be kept at reset value.
13:0	WHSZ[13:0]	Window horizontal size WHSZ=X means X+1 pixels clock in a line

49.7.11. DATA register (DCI_DATA)

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
DT3[7:0]								DT2[7:0]							
r								r							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DT1[7:0]								DT0[7:0]							
r								r							

Bits	Fields	Descriptions
31:24	DT3[7:0]	Pixel data 3
23:16	DT2[7:0]	Pixel data 2
15:8	DT1[7:0]	Pixel data 1
7:0	DT0[7:0]	Pixel data 0

50. General parallel slave interface(GPSI)

50.1. Overview

The GPSI is a general synchronous input/output slave interface for 8/16-bit parallel data. When the data is valid, it sends a data valid signal by the transmitter. If the receiver is ready to sample the data, it will give a flow control signal. The GPSI shared same functions, GPIO and interrupt vector with DCI(Digital camera interface), so two peripherals cannot be used at the same time.

50.2. Characteristics

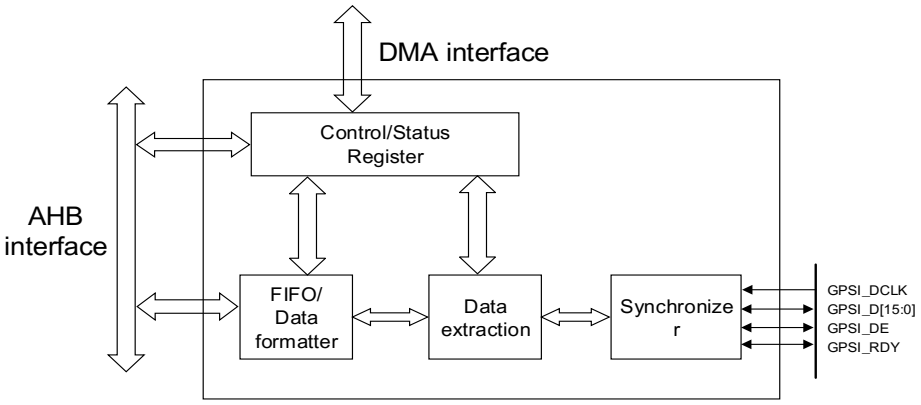
- Supported 8/16-bit parallel data input or output.
- 32-byte FIFO.
- Supported slave mode.
- Alternate function input of data enable(GPSI_DE).
- Alternate function output of ready(GPSI_RDY).

50.3. Function overview

50.3.1. Block diagram

The GPSI is a general synchronous parallel slave interface which is used for sending or receiving high-speed data flow. It has a 16-bit data line(GPSI_Data[15:0]) and a clock line(GPSI_DCLK). Through setting different clock polarities, data can be transmitted or captured on clock rising or falling edge. General DMA channel can transmit 32-bit packed data through the GPSI_DATA register. Hardware can use the GPSI_DE(Data Enable) and GPSI_RDY(Ready) optionally to make data flow be continuous or synchronized. ***Figure 50-1. The GPSI block diagram*** shows internal blocks, interfaces, signals of GPSI block and GPSI top level block with external signals.

Figure 50-1. The GPSI block diagram



50.3.2. Pins and Signals

The GPSI interface consists of 19 pins to transfer parallel data, ***Table 50-1. GPSI input/output pins*** shows GPSI relevant pins and signals shared with DCI(Digital camera interface). ***Table 50-2. GPSI internal input/output signals*** shows internal input and output signals of GPSI. GPSI and DCI can not be used at the same time because these two peripherals are designed with the same circuitry. Therefore, when using the GPSI, the DCI registers cannot be set. If the GPSI Enable bit(GPSI_CTL[14]) is set to 1, GPSI will use the alternate function and interrupt vector. Otherwise, they are used by DCI(Note: the DCI Enable bit and GPSI Enable bit can not be set to 1 at the same time). The GPIO can be configured for alternate function. For example, the GPIO pin of GPSI_DCLK/DCI_PixClk can be set to GPSI_DCLK function if the GPSI Enable bit is set to 1.

Table 50-1. GPSI input/output pins

GPSI signal	Shared DCI signal	In/out type	Description
GPSI_DCLK	DCI_PixClk	Input	GPSI Clock input of parallel data
GPSI_Data[15:0]	DCI_PixData[13:0]	Input/output	GPSI data for transmitting and receiving
GPSI_DE	DCI_HSYNC	Input/output	GPSI data enable signal
GPSI_RDY	DCI_VSYNC	Input/output	GPSI ready signal

Table 50-2. GPSI internal input/output signals

GPSI internal signal	In/out type	Description
GPSI_hclk	Input	AHB clock

GPSI internal signal	In/out type	Description
GPSI_dma	output	DMA request
GPSI_interrupt	output	Interrupt output

50.3.3. Clock

The ratio of AHB clock frequency to GPSI_DCLK frequency must be at least 2.5 to 1. If this value is lower than 2.5, data error or loss might occur. GPSI_DCLK is used to synchronize with data transfers. By setting CKPL bit of GPSI_CTL, GPSI_DCLK can be configured as *Table 50-3. GPSI_DCLK configured by CKPL bit* shown.

Table 50-3. GPSI_DCLK configured by CKPL bit

CKPL bit	Description
0	Input pins are sampled on GPSI_DCLK falling edge.
	Output pins are driven on GPSI_DCLK rising edge.
1	Input pins are sampled on GPSI_DCLK rising edge.
	Output pins are driven on GPSI_DCLK falling edge.

50.3.4. GPSI optional control signals

Data Enable(GPSI_DE) alternate function input

GPSI_DE is optional data enable signal which is driven by data source or transmitter. They use the data validity in active-high or active-low of GPSI_DE to indicate the transferred data is valid during this current cycle. If GPSI_DE is inactive, it indicates that the receiver cannot sample the data at the next clock edge.

Using SIGCFG bits (GPSI_CTL[20:18]) can enable alternate function signal. Through DPL bit (GPSI_CTL[6]), GPSI_DE polarity can be configured. If DPL is set to 1, GPSI_DE is active-high, otherwise GPSI_DE is active-low.

DIRSEL bit of GPSI_CTL register defines the direction of GPSI_DE which is same as data direction.

In receive mode(DIRSEL is cleared to 0) , if GPSI_DE alternate function input is enabled, GPSI_DE signal and data(GPSI_Data[15:0]) both can be sampled in the same GPSI_DCLK edge. If GPSI_DE is active, the sampling data is stored in FIFO. If GPSI_DE is inactive, data will be seen as invalid and discarded.

The transmitter can use GPSI_DE as a signal of valid data, making GPSI_DE inactive when the data in this current cycle is invalid. This function of flow control allows to avoid the underrun errors. *Figure 50-2. Data sampling in receive mode waveform diagram(CKPL =*

0, DPL/RPL = 0) and *Figure 50-3. Data sampling in receive mode waveform diagram(CKPL = 1, DPL/RPL = 0)* shows data sampling when GPSI_DE is low-high in receive mode. CKPL is set 0 or 1 and data sampling is located on blue line edge.

Figure 50-2. Data sampling in receive mode waveform diagram(CKPL = 0, DPL/RPL = 0)

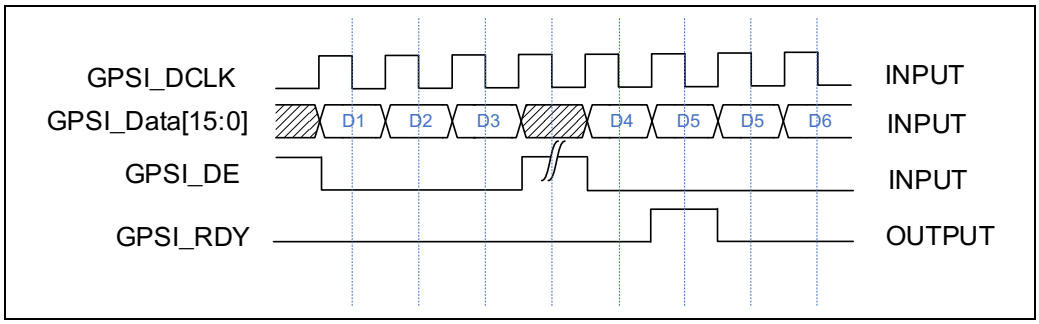
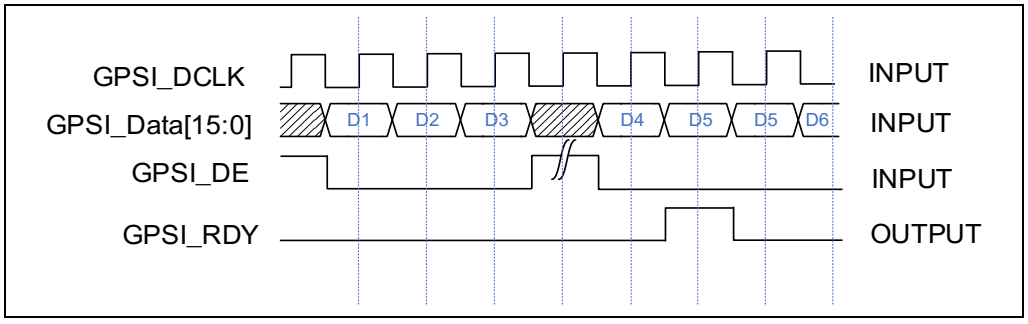


Figure 50-3. Data sampling in receive mode waveform diagram(CKPL = 1, DPL/RPL = 0)



In transmit mode(DIRSEL is set 1), if GPSI_DE alternate function output is enabled, the GPSI drives GPSI_DE and data(GPSI_Data[15:0]) on the same GPSI_DCLK edge. When a 8/16-bit data(by setting GPSIF[1:0]) in the internal FIFO is available to transfer, the data is output by GPSI_Data [15:0] and GPSI_DE is output active on the current GPSI_DCLK edge. But if TX FIFO is empty, GPSI_Data[15:0] values do not change and GPSI_DE becomes inactive. *Figure 50-4. Data output in transmit mode waveform diagram(CKPL = 0, DPL/RPL = 0)* and *Figure 50-5. Data output in transmit mode waveform diagram(CKPL = 1, DPL/RPL = 0)* shows data transfer when GPSI_DE is low-high in transmit mode. CKPL is set 0 or 1 and data output is locate on red line edge.

Figure 50-4. Data output in transmit mode waveform diagram(CKPL = 0, DPL/RPL = 0)

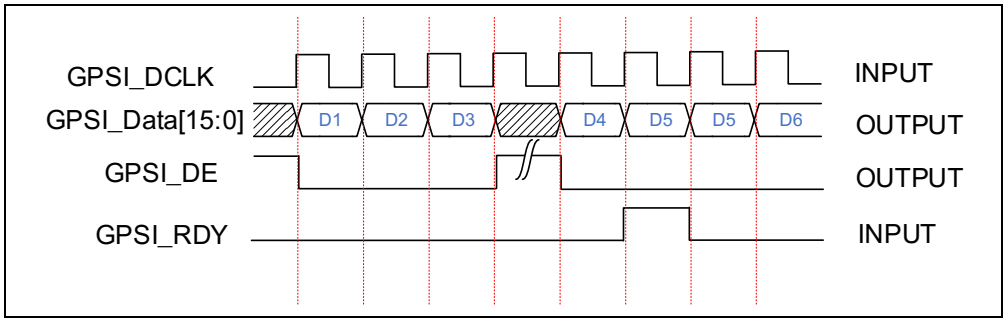
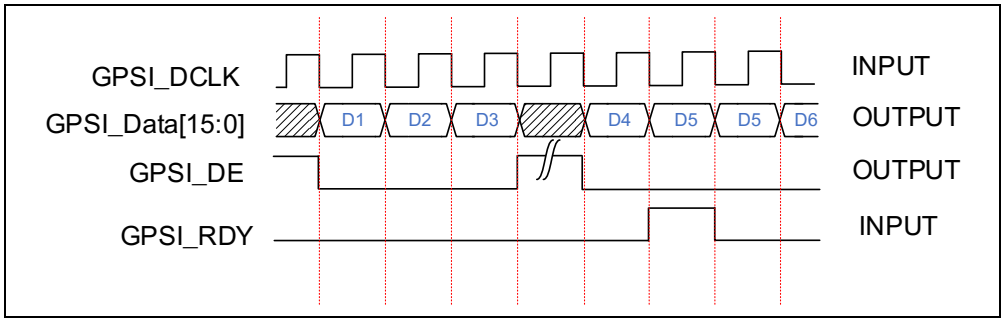


Figure 50-5. Data output in transmit mode waveform diagram(CKPL = 1, DPL/RPL = 0)



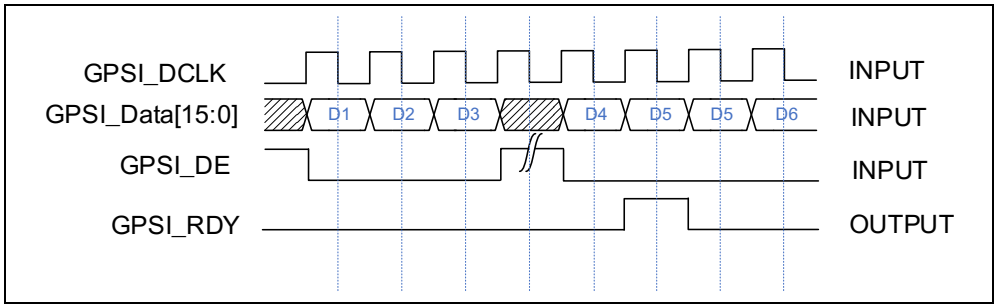
Ready(GPSI_RDY) alternate function output

GPSI_RDY is a optional ready signal which is driven by the receiving device and indicates that whether the data can be accepted during current cycle. When GPSI_RDY is inactive, receiver can not sample the data at the next clock edge.

SIGCFG control bits(GPSI_CTL[20:18]) can enable alternate function of GPSI_RDY. RPL control bit(GPSI_CTL[8]) can configure GPSI_RDY polarity. When RPL is cleared to 0, GPSI_RDY is active-low. And when RPL is set to 1, GPSI_RDY is active-high.DIRSEL bit(GPSI_CTL[31]) also define the direction of GPSI_RDY signal. Compared with GPSI_DE and GPSI_Data signals, GPSI_RDY signal is set in the opposite direction.

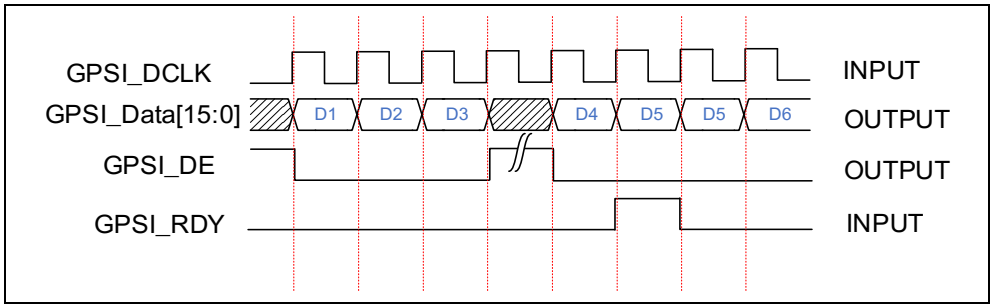
In receive mode(DIRSEL is cleared to 0), when alternate function of GPSI_RDY is enabled(by SIGCFG), the GPSI will drive GPSI_RDY half cycle of one GPSI_DCLK after it samples the data(GPSI_Data[15:0]). ***Figure 50-6. Ready signal in receive mode waveform diagram(CKPL = 0, DPL/RPL = 0)*** shows GPSI_RDY signal in receive mode. If there is enough space in FIFO to accept more data, GPSI_RDY signal is active, but if FIFO is full, GPSI_RDY signal will be inactive. Transmitter must repeat current data at next cycle when detecting that GPSI_RDY is inactive. This function of flow control can make the GPSI avoid overrun errors when the system(by DMA transferring) can not keep up with the data flow.

Figure 50-6. Ready signal in receive mode waveform diagram(CKPL = 0, DPL/RPL = 0)



In transmit mode(DIRSEL is set to 1), when alternate function of GPSI_RDY is enabled(by SIGCFG), GPSI_RDY signal is sampled on the GPSI_DCLK edge which is opposite to the edge of driving GPSI_Data [15:0]. ***Figure 50-7. Ready signal in transmit mode waveform diagram(CKPL = 0, DPL/RPL = 0)*** shows GPSI_RDY signal in transmit mode. If GPSI_RDY signal is inactive, the GPSI will keep the same data(GPSI_Data [15:0]) and GPSI_DE signals. If GPSI_RDY is active, and when the data in TX FIFO is available, it will be output by GPSI_Data [15:0]. However, if there is no data available in TX FIFO, the output data values will not change and GPSI make GPSI_DE signal inactive(if enabled).

Figure 50-7. Ready signal in transmit mode waveform diagram(CKPL = 0, DPL/RPL = 0)



Through GPSI_RDY signal the receiver can control data flow to avoid overrun errors when the system(by DMA transferring) can not keep up with the data flow.

GPSI_DE and GPSI_RDY are both disabled

When GPSI_DE and GPSI_RDY are both disabled(SIGCFG bit is set to 000), GPSI data can be transmitted by GPSI_DCLK. If GPSI_DCLK's initial signal level is high, data will be transmitted on the falling edge of the second clock pulse as ***Figure 50-8. Data transmitted on the falling edge of the second clock pulse*** shown. If GPSI_DCLK's initial signal level is low, data will be transmitted on the falling edge of the first clock pulse as ***Figure 50-9. Data transmitted on the falling edge of the first clock pulse*** shown.

Figure 50-8. Data transmitted on the falling edge of the second clock pulse

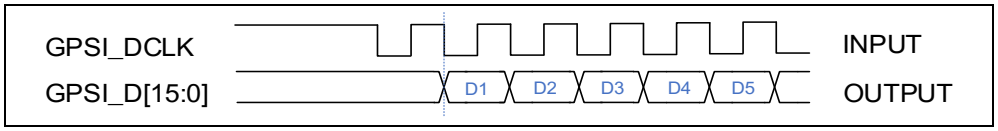
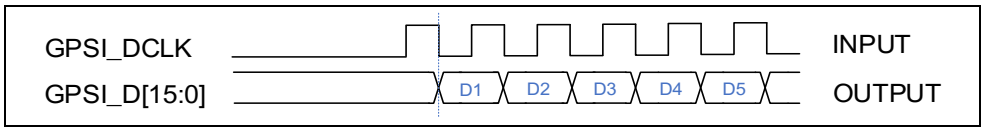


Figure 50-9. Data transmitted on the falling edge of the first clock pulse



Bidirectional GPSI_DE/GPSI_RDY signal

When DPL bit and RPL bit both are set to 1 and SIGCFG bit is set to 111 or 100, GPSI_DE and GPSI_RDY can share with a single pin. In this condition, the selected GPIO corresponding to alternate function must be set as open-drain. The other device must be set to drive open-drain line, and this line must be connected to a weak pull-up.

This configuration make signal bidirectional. When transmitter drives the line low(indicate that the data is not valid) or the receiver drives the line low(indicate that current data is not sampled), both devices can know that the data is not transferred during the current cycle. *Figure 50-10. Bidirectional GPSI_DE/GPSI_RDY waveform diagram(CKPL = 0, DPL/RPL = 0, transmitter drive the GPSI_DE to low)* and *Figure 50-11. Bidirectional GPSI_DE/GPSI_RDY waveform diagram(CKPL = 0, DPL/RPL = 0, receiver drive the GPSI_RDY to low)* shows bidirectional GPSI_DE and GPSI_RDY signals waveform. *Figure 50-12. Bidirectional GPSI_DE/GPSI_RDY connection diagram* shows bidirectional GPSI_DE and GPSI_RDY of master and receiver.

Figure 50-10. Bidirectional GPSI_DE/GPSI_RDY waveform diagram(CKPL = 0, DPL/RPL = 0, transmitter drive the GPSI_DE to low)

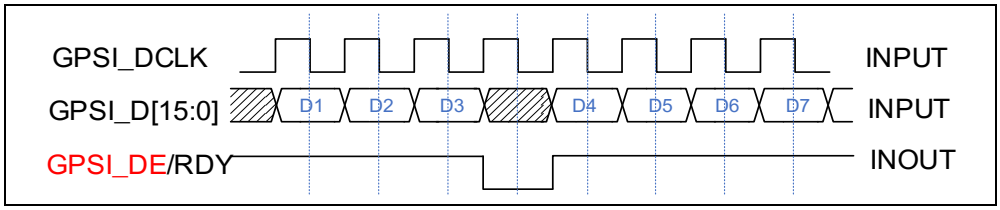


Figure 50-11. Bidirectional GPSI_DE/GPSI_RDY waveform diagram(CKPL = 0, DPL/RPL = 0, receiver drive the GPSI_RDY to low)

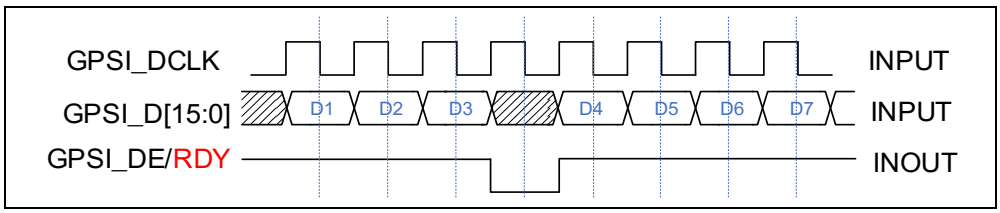
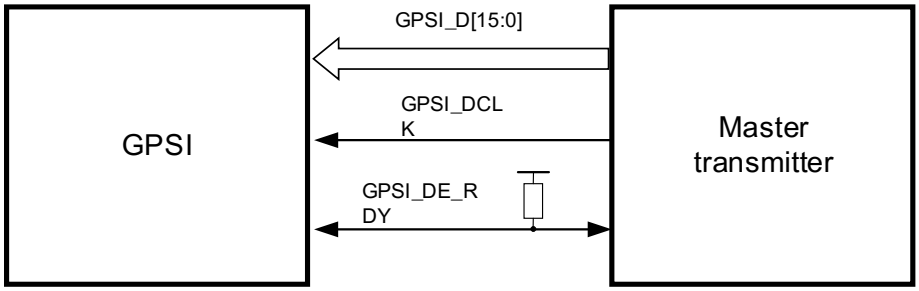


Figure 50-12. Bidirectional GPSI_DE/GPSI_RDY connection diagram



50.3.5. Data Transfer and DMA

Data direction

The DIRSEL bit of GPSI_CTL can be used to set direction of data transfers. When DIRSEL bit is 0(default setting), the GPSI works in receive mode and data direction is input on the data pins. When DIRSEL bit is 1, the GPSI works in transmit mode and data direction is output on the data pins.

DIRSEL bit can be configured only when the GPSI Enable bit is cleared to 0.

Data register and DMA

Data transfers from or to FIFO via GPSI_DATA register in two modes. In receive mode, data must be read from the FIFO by reading GPSI_DATA register. IF FIFO reads more datas which beyond the number of valid bytes, the invalid data will return 0. In transmit mode, data must be written to the FIFO by writing into GPSI_DATA register. When GPSIEN bit(GPSI_CTL[14]) is set to 0, an error will response when users access to GPSI_DATA register.

Two modes both support word(32-bit) and half-word(16-bit) transfer. Byte(8-bit) data can access to GPSI_DATA[7:0] only when GPSI is configured to transfer 8 bits by setting GPSIF bits to 00 in GPSI_CTL register.

To reduce CPU load, DMA can transfer data from or to GPSI FIFO via GPSI_DATA register. 8/16/32-bit data transfers are supported and using 32-bit transfers can optimizes bandwidth and reduces the bus load. By setting DMAEN bit to 1(default setting) in GPSI_CTL, a DMA transfer starts when FIFO is ready for a 32-bit transfer. The ready condition is that there are four valid bytes in receive mode or four empty bytes in transmit mode. The transfer will not be initiated if FIFO do not have at least four bytes, though the DMA is configured to 8-bit transfers.

When CPU directly transmits data from or to FIFO, the 4BRF and 1BRF bit in GPSI_STAT can be used to indicate status. When 4BRF is set to 1, it indicates that FIFO is ready to transfer four bytes. In this time, FIFO has at least four valid bytes in receive mode or at least four free bytes in transmit mode. When 1BRF is set to 1, it indicates that FIFO is ready to transfer one byte. In this time, FIFO has at least one valid byte in receive mode or at least one free byte in transmit mode.

Byte data

Through setting GPSIF[1:0], GPSI parallel interface can transfer 8-bit by using GPSI_DATA[7:0] or transfer 16-bit by using GPSI_Data[15:0]. If 8-bit transfer is configured, the unused GPSI_Data [15:0] can be used as other functions such as GPIO.

When GPSIF[1:0] is set 00, GPSI_DATA [7:0] pins are used to transfer 8 bits. In this condition, GPSI_DATA [15:8] are not used and a 32-bit word transfer needs four GPSI_DCLK cycles.

When a 32-bit word transfers, the least-significant byte(bits 7:0) corresponds to the first byte transferred, and the most-significant byte (bits 31:24) corresponds to the fourth byte transferred. *Table 50-4. Data bytes transfer in 32-bit words(8-bit width)* shows corresponding position of data bytes transfer in 32-bit words.

Table 50-4. Data bytes transfer in 32-bit words(8-bit width)

31:24	23:16	15:8	7:0
D3[7:0]	D2[7:0]	D1[7:0]	D0[7:0]
D7[7:0]	D6[7:0]	D5[7:0]	D4[7:0]

Half-word data

When GPSIF[1:0] is set 11, GPSI_DATA[15:0] pins are used to transfer the16-bit data. In this condition, a 32-bit word transfer needs two GPSI_DCLK cycles. When a 32-bit word transfers, the least-significant half-word(bits 15:0) corresponds to the first half word, and the most-significant half-word(bits 31:16) corresponds to the second half word. *Table 50-5. data bytes transfer in 32-bit words(16-bit width)* shows corresponding position of data bytes transfer in 32-bit words.

Table 50-5. data bytes transfer in 32-bit words(16-bit width)

31:16	15:0
D1[15:0]	D0[15:0]
D3[15:0]	D2[15:0]

FIFO and error

FIFO in GPSI is 8-word depth and can improve transfer performance and avoid overruns and underruns error.

In receive mode, if ready signal(GPSI_RDY) is disabled, an overrun error generates when clock active edge occurs at the same time that FIFO is full. This condition will make input data be lost.

In transmit mode, if data enable signal(GPSI_DE) is disabled, an underrun error generates when clock active edge occurs at the same time that FIFO is empty. This condition will make

output data unpredictable.

The OUIF status bit can indicate that an overrun/underrun occurred. When these events occurs, an interrupt will be generated.

50.4. Interrupt requests

GPSI interrupt is only OVR which is shared with DCI(including interrupt vector). It is equal to global interrupt. When GPSIEN bit (GPSI Enable bit) is set to 1, the interrupt is controlled by GPSI, otherwise the interrupt is controlled by DCI. The enable bit of DCI and GPSI must not be set at the same time. ***Table 50-6. GPSI interrupt requests*** shows GPSI interrupt requests information.

Table 50-6. GPSI interrupt requests

Interrupt	Shared with DCI	Interrupt event	Status bit	Enable control bit	Clear bit
OVR	OVRF	Overrun in receive mode or underrun in transmit mode occur	OUIF	OUIE	OUFC

50.5. Register definition

GPSI base address: 0x4802 0200

50.5.1. Control register (GPSI_CTL)

Address offset: 0x00
Reset value: 0x4000 0000

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DIRSEL	DMAEN	Reserved									SIGCFG[2:0]			Reserved	
rw	rw											rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	GPSIEN	Reserved		GPSIF[1:0]		Reserved	RPL	Reserved	DPL	CKPL	Reserved				
	rw			rw			rw		rw	rw					

Bits	Fields	Descriptions
31	DIRSEL	Data direction. 0: Receive mode, and data is input synchronously with GPSI_DCLK. 1: Transmit mode, and data is output synchronously with GPSI_DCLK.

30	DMAEN	DMA enable bit 0: Disable DMA, and GPSI_DATA register can be accessed directly. 1: Enable DMA(default setting). A DMA channel need be configured to transfer from/to GPSI_DATA.
29:21	Reserved	Must be kept at reset value.
20:18	SIGCFG[2:0]	Data enable and ready setting bits. 000: GPSI_DE and GPSI_RDY are both disabled. 001: Only GPSI_RDY is enabled. 010: Only GPSI_DE is enabled. 011: Both GPSI_RDY and GPSI_DE alternate functions enabled 100: Both GPSI_RDY and GPSI_DE features enabled and GPSI_RDY pin is bidirectional. 101: Only GPSI_RDY functions enabled, and mapped to GPSI_DE pin. 110: Only GPSI_DE functions enabled, and mapped to GPSI_RDY pin. 111: Both GPSI_RDY and GPSI_DE features enabled and GPSI_DE pin is bidirectional.
17:15	Reserved	Must be kept at reset value.
14	GPSIEN	GPSI enable bit. 0: GPSI is disabled. 1: GPSI is enabled. When GPSIEN is cleared to 0, the contents of FIFO are flushed. **Note**: When GPSIEN is 1, the contents of GPSI_CTL cannot be changed, except for GPSIEN bit itself. All configuration bits can be changed before GPSIEN is set to 1. User must configure the DMA controller and GPSI registers correctly before GPSIEN is set to 1. The DCI Enable bit and GPSI Enable bit can not be set to 1 at the same time .
13:12	Reserved	Must be kept at reset value.
11:10	GPSIF[1:0]	Captured data width select. 00: 8-bit data can be captured on every parallel data clock. 01: Reserved and must not be selected. 10: Reserved and must not be selected. 11: 16-bit data can be captured on every parallel data clock.
9	Reserved	Must be kept at reset value.
8	RPL	GPSI_RDY polarity. 0: GPSI_RDY is active-low(when the level of GPSI_RDY is 0, receiver is ready to receive). 1: GPSI_RDY is active-high(when the level of GPSI_RDY is 1, receiver is ready to receive).

7	Reserved	Must be kept at reset value.
6	DPL	GPSI_DE polarity. 0: GPSI_DE is active-low(when the level of GPSI_DE is 0, data is valid). 1: GPSI_DE is active-high(when the level of GPSI_DE is 1, data is valid).
5	CKPL	Parallel data clock polarity. This bit and DIRSEL commonly configure the edge of parallel clock for data capture and data driving. 0: Falling edge active for input or rising edge active for output. 1: Rising edge active for input or falling edge active for output.
4:0	Reserved	Must be kept at reset value.

50.5.2. Status register (GPSI_STAT)

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											2BRF	1BRF	4BRF	Reserved	
											r	r	r		

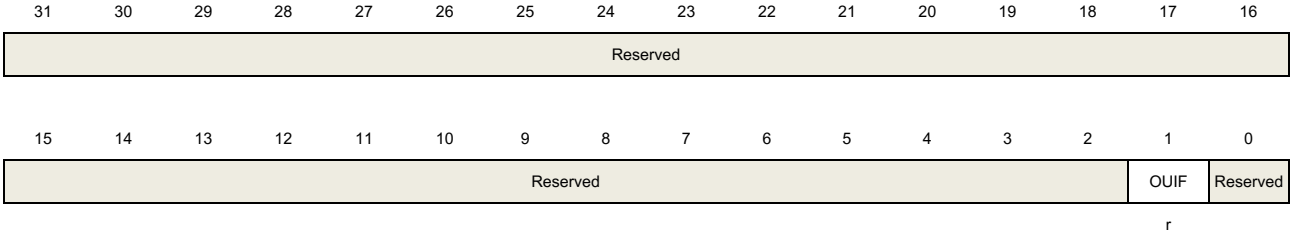
Bits	Fields	Descriptions
31:5	Reserved	Must be kept at reset value.
4	2BRF	FIFO is ready to transfer two bytes. 0: FIFO is not ready to transfer two byte data. 1: FIFO is ready to transfer two byte data. This indicates that FIFO has at least two valid data bytes in receive mode, and at least two byte free is in FIFO in transmit mode.
3	1BRF	FIFO is ready to transfer one byte. 0: FIFO is not ready to transfer one byte data. 1: FIFO is ready to transfer one byte data. This indicates that FIFO has at least one valid data byte in receive mode, and at least one byte free is in FIFO in transmit mode.
2	4BRF	FIFO is ready to transfer four bytes. 0: FIFO is not ready to transfer four byte data.

		1: FIFO is ready to transfer four byte data. This indicates that FIFO has at least four valid data bytes in receive mode, and at least four bytes free are in FIFO in transmit mode.
1:0	Reserved	Must be kept at reset value.

50.5.3. Interrupt status register (GPSI_INTF)

Address offset: 0x08
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)

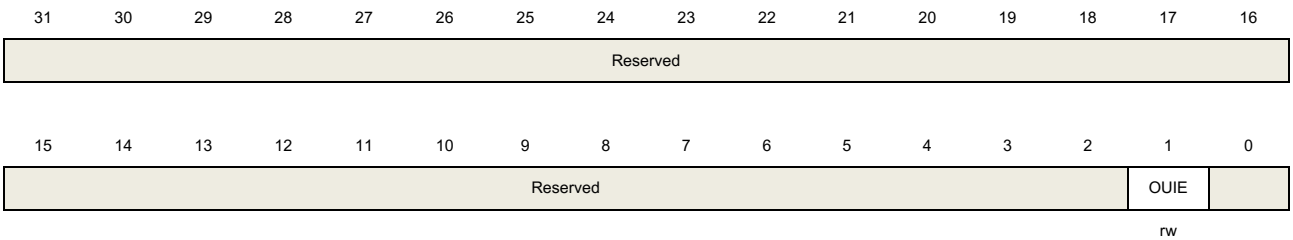


Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	OUIF	Data buffer overrun or underrun interrupt status. 0: No error occurred. 1: An overrun error in receive mode or an underrun error in transmit mode occurred. This bit can be cleared by writing 1 to OUFC bit in GPSI_INTC register.
0	Reserved	Must be kept at reset value.

50.5.4. Interrupt enable register (GPSI_INTEN)

Address offset: 0x0C
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.

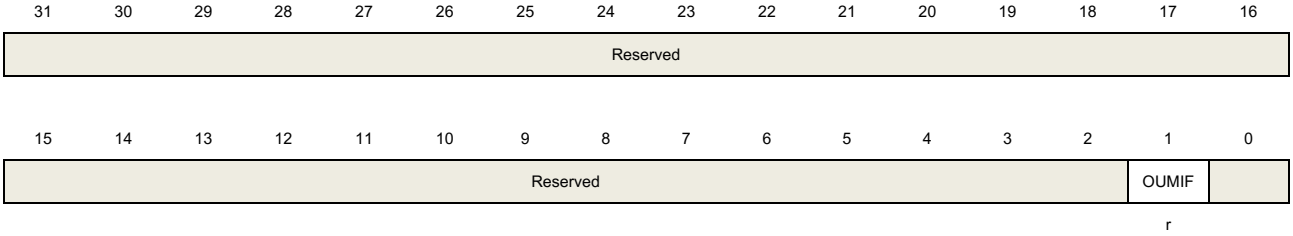
1	OUIE	Data buffer overrun or underrun interrupt enable. 0: No interrupt is enabled. 1: overrun or underrun error interrupt is enabled.
0	Reserved	Must be kept at reset value.

50.5.5. Masked interrupt status register (GPSI_MINTF)

Address offset: 0x10

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)



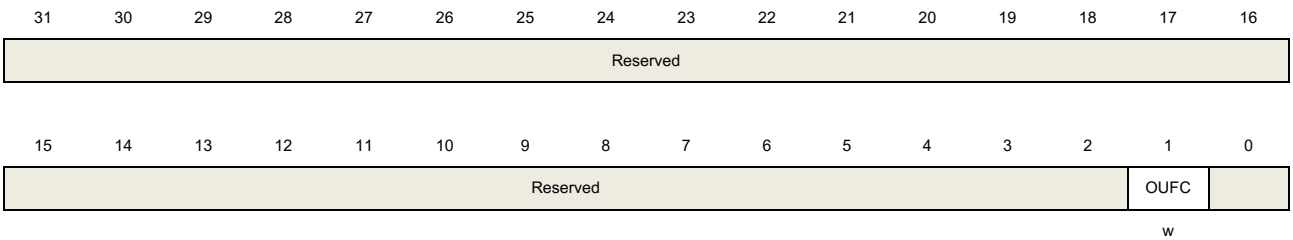
Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	OUMIF	Data buffer overrun or underrun masked interrupt status. 0: No interrupt generates when an overrun or underrun error occurs. 1: An interrupt generates when an overrun or underrun error occurs and OUIE bit is set in GPSI_INTEN register.
0	Reserved	Must be kept at reset value.

50.5.6. Interrupt clear register (GPSI_INTC)

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	OUFC	Clear data buffer overrun or underrun interrupt status. Write 1 to clear OUIF bit in GPSI_INTF register.
0	Reserved	Must be kept at reset value.

50.5.7. Data register (GPSI_DATA)

Address offset: 0x28

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit), half-word (16-bit) or byte(8-bit)



Bits	Fields	Descriptions
31:24	DT3[7:0]	GPSI DATA 3.
23:16	DT2[7:0]	GPSI DATA 2.
15:8	DT1[7:0]	GPSI DATA 1.
7:0	DT0[7:0]	GPSI DATA 0.

51. Receiver of Sony/Philips Digital Interface (RSPDIF)

51.1. Overview

The receiver of Sony/Philips Digital Interface (RSPDIF) module provides the function of receiving and decoding RSPDIF audio data streams.

51.2. Characteristics

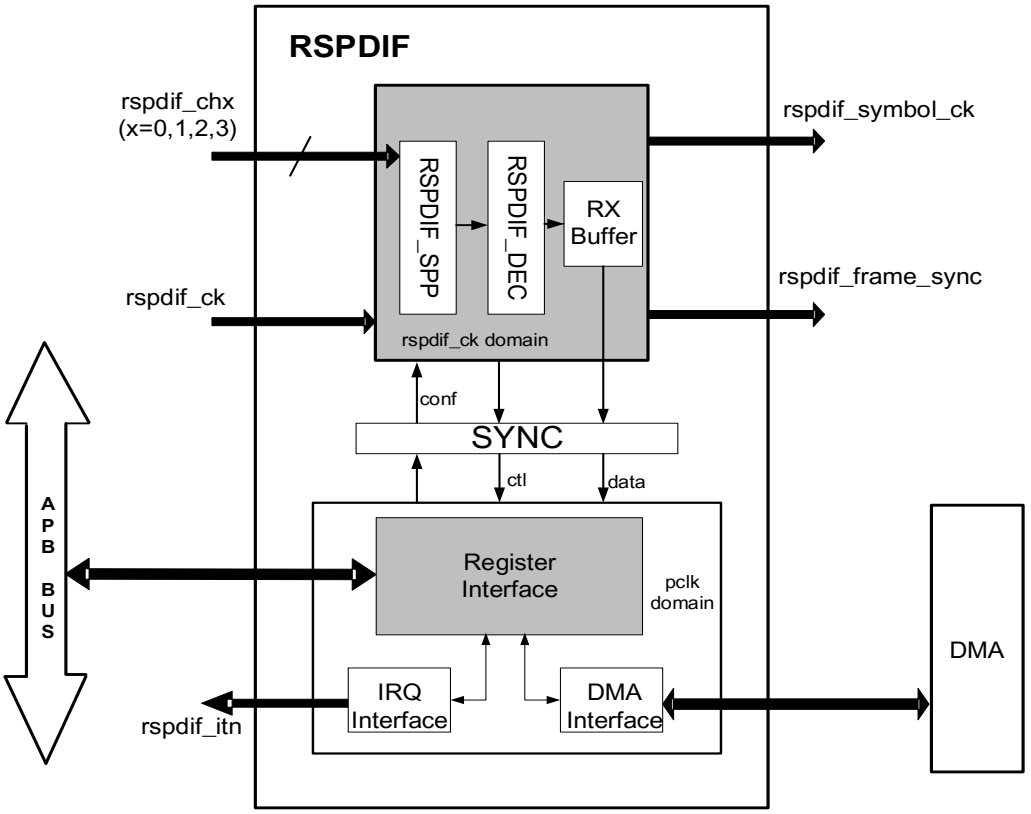
- Supports audio IEC-60958 and IEC-61937.
- Up to 4 inputs available.
- Supports receiving double buffer.
- Dual clock domain：PCLK for register interface and RSPDIF_CK for other parts.
- Supports maximum symbol rate: 12.288 MHz.
- Supports stereo stream from 8 to 192 kHz.
- Supports automatic symbol rate detection.
- Generate symbol clock.
- Check the parity bit of the received data.
- Support multiple data processing methods, which can process audio data and user channel information separately or together.
- Supports using DMA communication to receive audio data and user channel information respectively.
- Supports interrupt function.

51.3. Function overview

51.3.1. RSPDIF block diagram

The receiver of Sony/Philips Digital Interface (RSPDIF) module provides the function of receiving and decoding SPDIF audio data streams. The audio data, channel status (CS) and User data (U) received can be accessed by DMA interface. The RSPDIF transceiver allows the handling of both RSPDIF channel status (CS) and User data (U) and supports the precise measurement of an incoming sampling frequency.

Figure 51-1. RSPDIF block diagram



Legend:

rspdif_frame_sync: RSPDIF frame rate synchronization signal

rspdif_symbol_ck: RSPDIF channel symbol clock

The RSPDIF module is responsible for decoding the S/PDIF stream received from RSPDIF_CH[3:0], and is used to receive S/PD audio data in accordance with the IEC-60958 and IEC-61937 standards. These standards support simple stereo streams at high sampling rates, as well as compressed multi-channel surround sound.

RSPDIF resamples the incoming signal, decodes the Manchester stream, recognizes subframes, frames, and block elements, and passes the decoded data and associated status flags to the RSPDIF register. The RSPDIF section can be fully controlled through the APB bus, and provides a dedicated path for user information and channel information. It can handle two DMA channels, audio sample and channel/user information. Interrupt service can also be used as an alternative to DMA for sending error signals or critical states.

When RSPDIF decodes the incoming audio data stream, it also provides two signals:

- rspdif_frame_sync: The frequency of this signal is equal to the frame rate, and it is inverted every time the subframe preamble is detected by RSPDIF, and the duty cycle is 50%.
- rspdif_symbol_ck: The signal frequency is equal to the symbol rate. The signal can drive internal components in the system, such as SAI ports, and external components, such

as A/Ds or D/As, with clocking control provided via related registers. Please refer to *RSPDIF clock management* for specific generation conditions.

51.3.2. S/PDIF protocol

S/PDIF (Sony/Philips Digital Interface) is a type of digital audio interconnect used in consumer audio equipment to output audio over reasonably short distances. The signal is transmitted over either a coaxial cable with RCA connectors or a fiber optic cable with TOSLINK connectors. S/PDIF interconnects components in home theaters and other digital high-fidelity systems.

S/PDIF is a data link layer protocol as well as a set of physical layer specifications for carrying digital audio signals between devices and components over either optical or electrical cable. The name stands for Sony/Philips Digital Interconnect Format but is also known as Sony/Philips Digital Interface. Sony and Philips were the primary designers of S/PDIF. S/PDIF is standardized in IEC 60958.

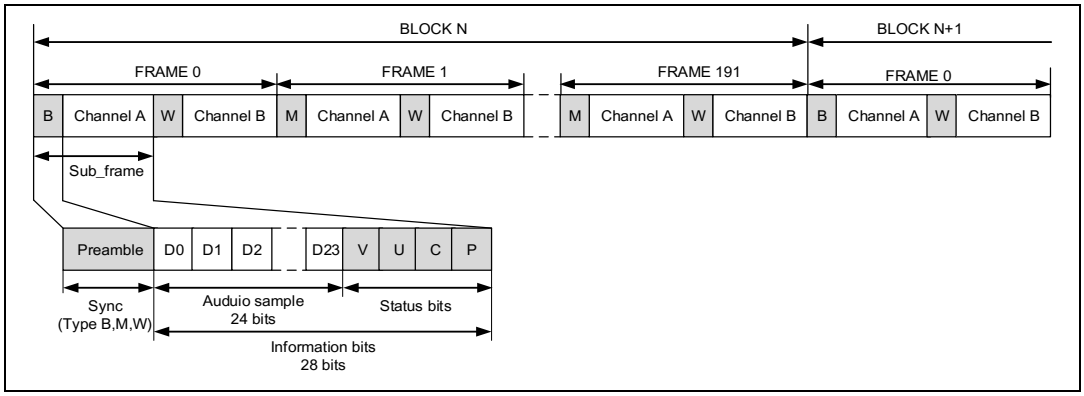
S/PDIF block

A S/PDIF block consists of 192 frames and each frame contains two sub-frames(one for left channel and the other one for right channel), that is, a RSPDIF block contains 384 sub-frames. Each sub-frame contains 32 bits. The data format of the block and the sub-frame is shown in *Table 51-1. Sub-frame pattern* and *Figure 51-2. S/PDIF block and S/PDIF sub-frame format.*

Table 51-1. Sub-frame pattern

BITS	Description
0~3	synchronization preambles, three types: B, M, W. Preambe M represents channel A (left channel) or main is transmitted at this time, preamble W represents channel B (right channel) is transmitted at this time, and preamble B is more special, which represents channel A (left channel) is transmitted at this time, and it is the starting subframe of A block.
4~27	the audio sample word in linear 2's complement representation. The most significant bit (MSB) is always carried by bit 27. When a 20-bit coding range is used, bits 8 to 27 carry the audio sample word with the LSB in bit 8.
28	validity bit "V", indicates if the data is valid
29	user data bit "U", defined by user
30	channel status bit "C"
31	parity bit "P", carries a parity bit such that bits 4 to 31 inclusive carry an even number of ones and an even number of zeroes (even parity).

Figure 51-2. S/PDIF block and S/PDIF sub-frame format

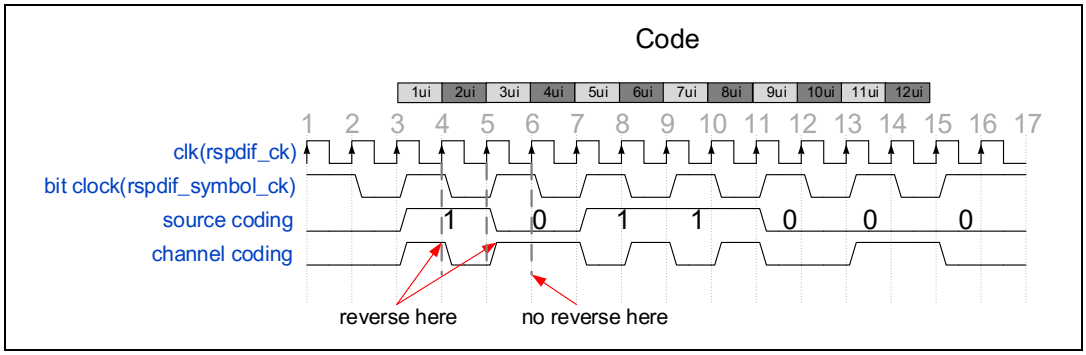


Coding of information bit

When IEC60958 transmits data, bits 4 to 31 are encoded in biphase-mark(Manchester protocol). The principle is to use a clock frequency that is twice the transmission bit rate as a reference, split the original one bit of data into two, and each bit to be transmitted is represented by a symbol containing two consecutive binary states. When the data is 1, in the clock cycle shift a potential (0->1 or 1->0) data into two different potentials, into 10 or 01, When the data is 0, it doesn't switch, it goes to 11 or 00. These binary states are named "UI" (Unit Interval) in the IEC-60958 specification.

Note: A symbol represents a data bit which comprises two unit intervals. The symbols rate means the frequency of bit clock shown in ***Figure 51-3. Information bits coding example.*** The clock frequency used by RSPDIF module is twice of that of bit clock.

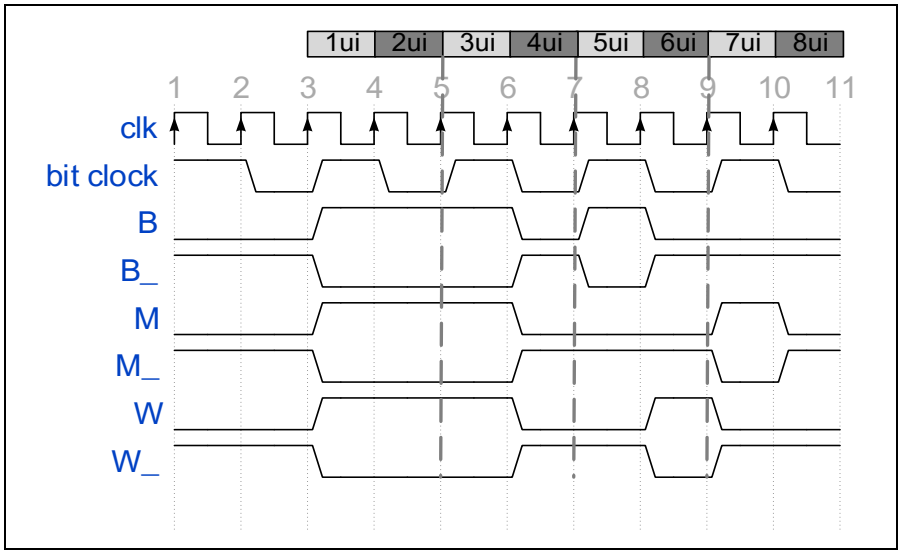
Figure 51-3. Information bits coding example



Synchronization preambles

Whether or not to invert the preamble pattern is determined by the value of the previous half-bit value. For the first "B" preamble of the first frame, this first half-bit value is the line level before transmission is enabled. For other preambles, this first half-bit value is the second half of the parity bit of the previous sub-frame. The preambles patterns B, M and W are shown in the ***Figure 51-4. S/PDIF Preambles.*** The preamble pattern can be concluded in ***Table 51-2. Preamble pattern***.

Figure 51-4. S/PDIF Preambles



Note: When coding data, transition should occur at the end of second UI. However, transition may lack in preamble at the end of second UI.

Table 51-2. Preamble pattern

Preceding state (the previous half-bit value)	0	1	Description
Preamble	Channel coding		
B	11101000	00010111	Channel A data at the start of block
W	11100100	00011011	Channel B data at somewhere in the block
M	11100010	00011101	Channel A data

51.3.3. RSPDIF

RSPDIF mainly includes two parts: RSPDIF_SPP and RSPDIF_DEC. The S/PDIF stream is decoded based on measuring the time interval between two consecutive edges. There are three kinds of time intervals in the S/PDIF stream, shown as ***Table 51-3. RSPDIF time interval***.

Table 51-3. RSPDIF time interval

Time interval	Description
TL	long time interval, having a duration of 3 x UI. It appears only during preambles.
TM	medium time interval, having a duration of 2 x UI. It appears both in some preambles or the information field.
TS	short time interval, having a duration of 1 x UI. It appears both in some preambles or the information field.

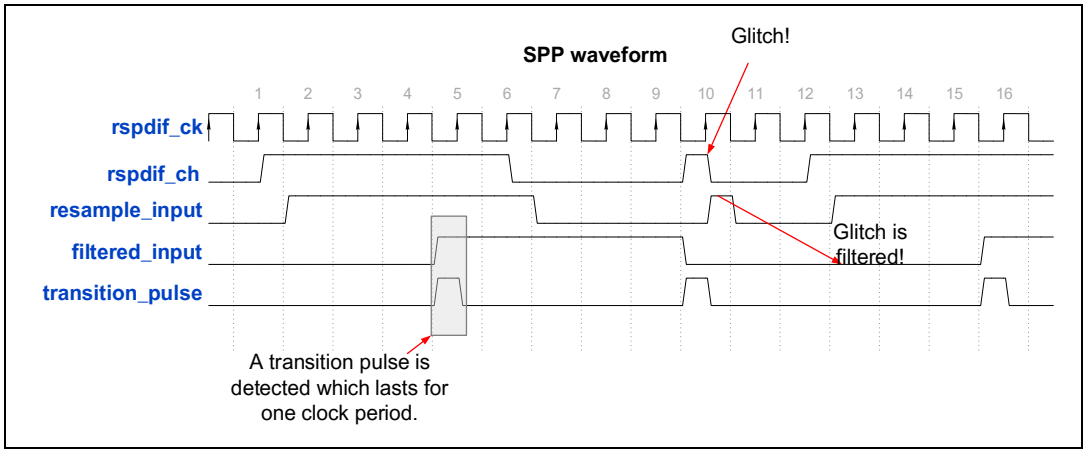
RSPDIF_SPP

RSPDIF_SPP signal preprocessing module is designed to perform noise filtering and rising/falling edge detection.

RSPDIF supports a total of four input signals, select the required input by configuring the RXCHSEL[2:0] bits in RSPDIF_CTL register. The S/PDIF signal received on the selected RSPDIF_CH is re-sampled using the rspdif_ck clock (acquisition clock). A simple filtering is applied in order to eliminate glitches.

This is performed by the stage detecting the edge transitions. During edge conversion detection, when the sequence 0 is followed by two 1s, the rising edge is detected. When the sequence 1 is followed by two 0s, the falling edge is detected. After the rising edge, a sequence of falling edges is expected. After the falling edge, a sequence of rising edges is expected.

Figure 51-5. Noise filtering and rising/falling edge detection



Note: The transition_pulse is an important indicator for the RSPDIF_DEC module to judge the transition types and decode the input bit stream properly.

RSPDIF_DEC

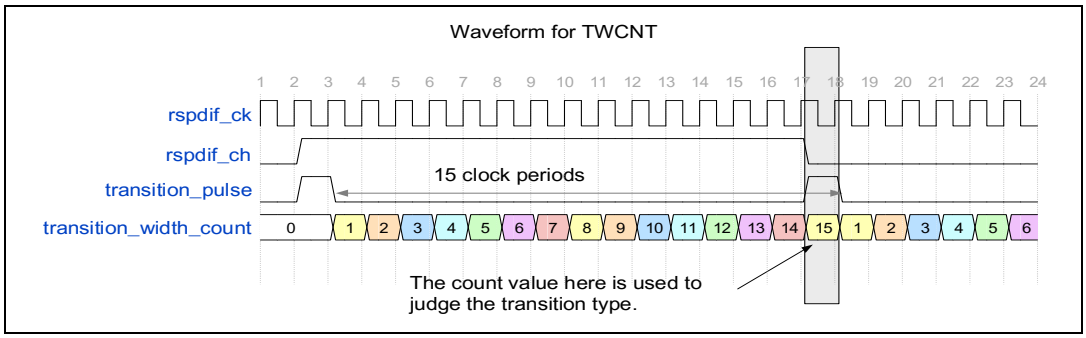
RSPDIF_DEC detects preamble type and decodes information bits. RSPDIF_DEC will pack these data and write into the RX buffer or into RSPDIFCHSTAT register. The R SPDIF_DEC module can perform functions such as time interval estimation, symbol ra te and synchronization estimation, block and subframe header detection, data decodin g and continuous symbol rate tracking.

TWCNT

The TWCNT counter is used to measure the time interval duration. It is clocked by the rspdif_ck signal. On every transition pulse, the counter value is stored and the counter is reset to start counting again. If the time interval between the two transitions is too long and the TWCNT overflows, RSPDIF will stop working and the TMOUTERR flag of the RSPDIF_STAT register is set at this time.

The transition timer is like a watchdog timer, which generates a trigger after 70 conversions of the incoming signal, and stops searching for the longest and shortest conversions. Note that calculating 70 transitions ensures a delay that is a little longer than a subframe.

Figure 51-6. Waveforms of TWCNT



Note: The counter value stored is equal to the actual clock periods between the two transition pulses. For example, as shown in ***Figure 51-6. Waveforms of TWCNT***, the time interval duration between the two transition pulses is 15 RSPDIF_CK clock periods, so the number stored is 15(not 14!). In this way, it is convenient to calculate CKCNT5, WIDTH24 and WIDTH40. When the counter value is reset, the reset value is (NOT 0!).

SYNC

RSPDIF_DEC provides a SYNC module, please refer to ***RSPDIF synchronization process***. SYNC module calculates WIDTH24 and WIDTH40 which are references for calculating the thresholds for judging the transition type. WIDTH24 and WIDTH40 represent the time interval duration of consecutive 24 symbols and 40 symbols respectively. As introduced above, it is clear that:

$$WIDTH24 = 48\ UI \rightarrow TL_{LO} = 1.5\ UI = \frac{WIDTH24}{32} \quad (51\text{-}1)$$

$$WIDTH40 = 80\ UI \rightarrow TH_{HI} = 2.5\ UI = \frac{WIDTH40}{32} \quad (51\text{-}2)$$

TH_{HO} and TL_{LO} is threshold values which are used for judging the transition types and this part shall be introduced in the description of DECODER module and RSPDIF synchronization process in detail. WIDTH24 and WIDTH40 can also be used for generating the symbol clock and this will be introduced ***RSPDIF clock management***.

Additionally, CKCNT5 in RSPDIF_STAT register which is used to estimate the S/PDIF symbol rate is also calculated in SYNC.

DECODER

RSPDIF_DEC provides a DECODER module which has the function of transition coder and preamble detector. It outputs the preamble type (PREF), 28 bit information bits and status bits (C,U,V,P). A recovered symbol clock is also generated during the decoder process of the RSPDIF_CH bit stream. Decoder block receives the current transition width from the TWCNT counter. This block encodes the current transition width by comparing the current transition

width with two different thresholds, names TH_{HI} and TH_{LO}, as shown in *Table 51-4. Transition encoder encoding rules*.

Table 51-4. Transition encoder encoding rules

transition width (TH)	Code	Description
TH < (TH_{LO} - 1)	TS	the data received is half part of data bit '1'
(TH_{LO} - 1)< TH <TH_{HI}	TM	the data received is data bit '0'
TH > TH_{HI}	TL	the data received is the long pulse of preambles
else	-	FRERR flag is set

The calculation methods of THHI and THLO, refer to *Table 51-5. Thresholds calculation*.

Table 51-5. Thresholds calculation

Thresholds	Synchronization
TH_{LO}	WIDTH24 / 32
TH_{HI}	WIDTH40 / 32

RSPDIF synchronization refer to *Figure 51-7. Synchronization flowchart flow* for additional information.

Preamble detector

The decoder provides the function of preamble detection, checking four consecutive conversions of a specific sequence to ensure whether they constitute the preamble part. Assume that TRANS0, TRANS1, TRANS2, and TRANS3 represent the four consecutive transformations of the above encoding. The four converted values are shown in *Table 51-6. Preamble transition sequence*. The absence of this mode indicates that these conversions do not constitute a header, but rather constitute part of the subframe data and can be decoded by a bi-phase decoder.

Table 51-6. Preamble transition sequence

Preamble type	Biphase data pattern	TRANS3	TRANS2	TRANS1	TRANS0
B	11101000	TL	TS	TS	TL
M	11100010	TL	TL	TS	TS
W	11100100	TL	TM	TS	TM

Bi-phase decoder

The Bi-phase decoder uses the conversion information provided by the conversion encoder and the preamble detector to decode the input bi-phase symbol data stream. When the header has been correctly detected, the Bi-phase decoder will decode the following data information, as shown in *Table 51-7. Bi-phase decoding rules*.

Table 51-7. Bi-phase decoding rules

Input	Decode value
TM	0
Two consecutive TS	1
else	FRERR flag is set

51.3.4. RSPDIF synchronization process

When the RXCFG[1:0] in RSPDIF_CTL register is set to 2'b01 or 2'b11, the synchronization process starts. If the WFRXA bit in RSPDIF_CTL register is set to 1, RSPDIF will first detect the activity of the selected RSPDIF_CH line before the synchronization. Only when four transitions are detected on the selected RSPDIF_CH line, RSPDIF will switch to the synchronization phase. This function can avoid synchronization errors effectively. The synchronization process is shown in ***Figure 51-7. Synchronization flowchart flow***. This function is implemented in the RSPDIF_DEC module.

After the synchronization is correctly decoded by RSPDIF, the accurate synchronization threshold is calculated. When RSPDIF can correctly measure the duration of 24 and 40 consecutive symbols, the synchronization is completed, the threshold will be updated, and the SYNDO flag is set to 1.

There may be interference on the RSPDIF_CH line, and it may happen that the synchronization process is not performed correctly. RSPDIF provides a function to set the number of synchronization retries (MAXRT) before synchronization. Until the times set by MAXRT bit are reached, the synchronization has not been performed correctly, and the SYNERR error flag is set. If there is no valid RSPDIF data stream on the RSPDIF_CH line, TWCNT overflows and the TMOUTERR error flag is set.

After synchronization is completed, when the next header "B" is detected, RSPDIF starts to receive channel status ® and user data (U) (see ***Figure 51-8. Synchronization process scheduling***). The user reads the C and U bits through the RSPDIF_CHSTAT register, and configures the RXDF[1:0] and RXSTEOMEN bits according to the C and U bits.

Note: When RXCFG[1:0] = 2'b11, the modification of RXDF[1:0] and RXSTEOMEN bits is invalid. Refer to ***RSPDIF_CTL register access permissions in different states.***

Figure 51-7. Synchronization flowchart flow

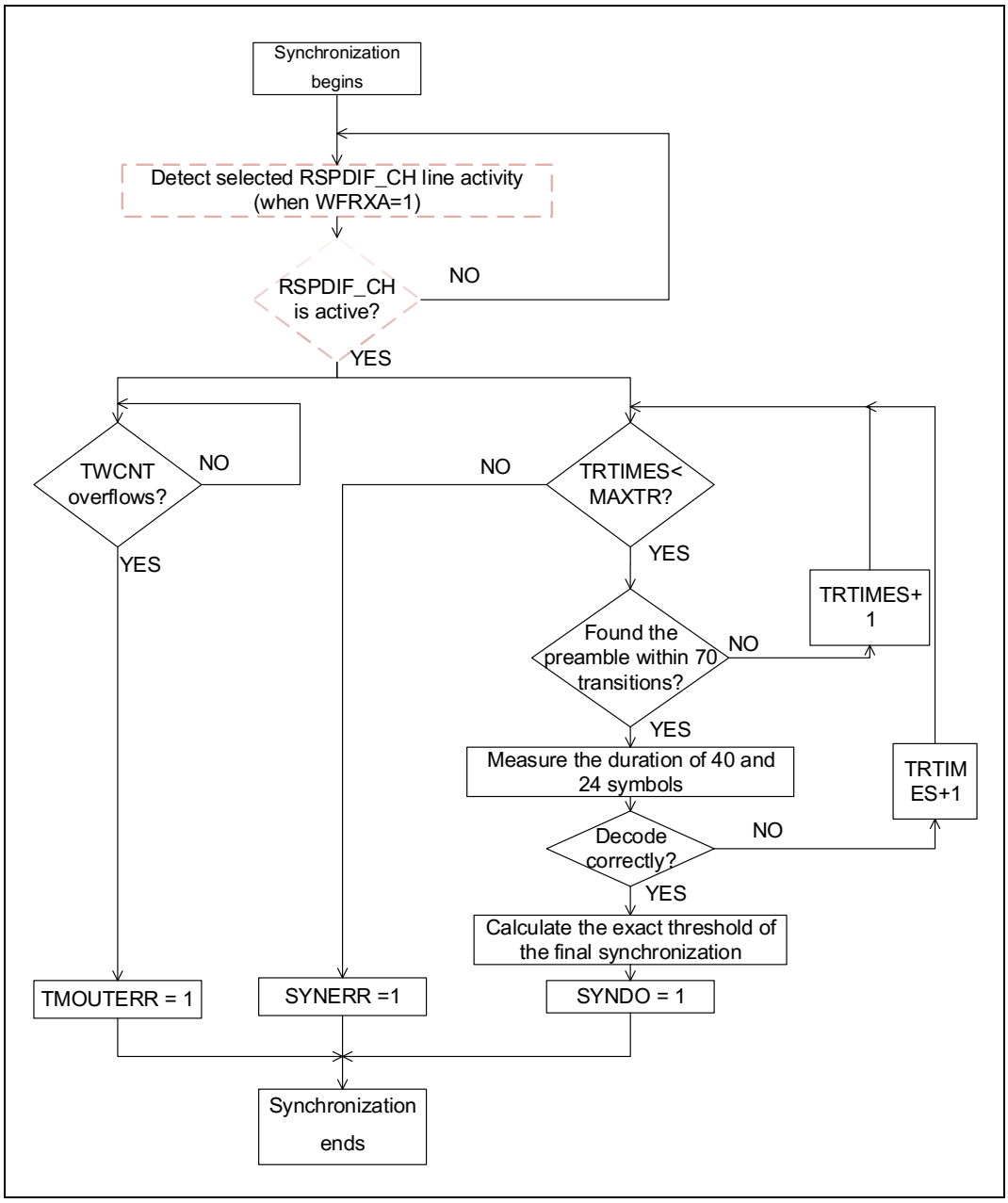
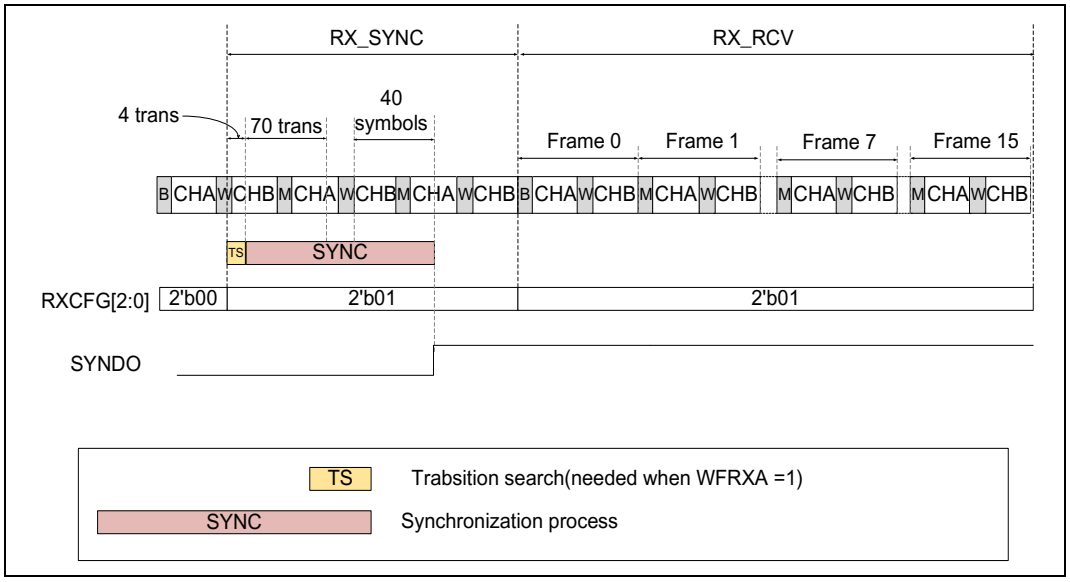


Figure 51-8. Synchronization process scheduling



51.3.5. RSPDIF state machine

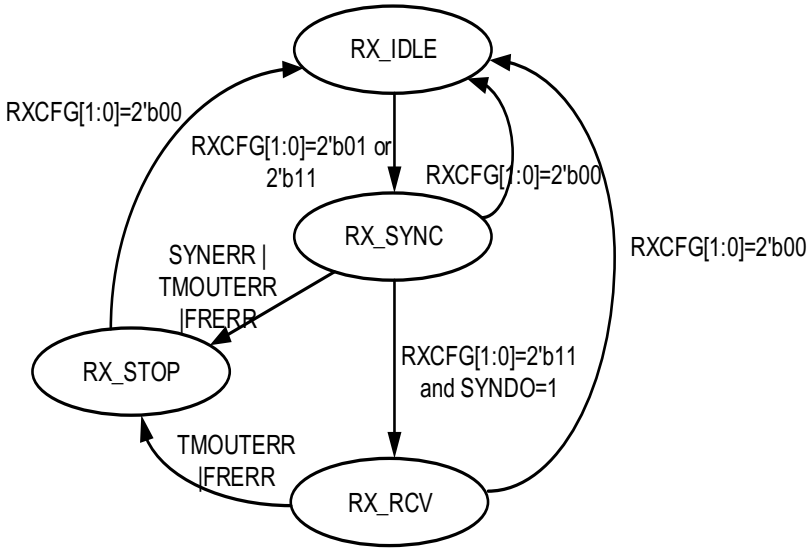
The RSPDIF state machine includes the following states, as shown in *Table 51-8. RSPDIF state*.

Table 51-8. RSPDIF state

RSPDIF state	Description
RX_IDLE	In idle state, RSPDIF is disabled, rspdif_ckdomain is reset, and rspdif_pclk domain function is normal.
RX_SYNC	In synchronization status, RSPDIF is synchronized with the data stream, the threshold is updated regularly, and the user data and channel status can be obtained through interrupt or DMA.
RX_RCV	In receiving status: RSPDIF is synchronized with the data stream, and the threshold is updated regularly. Users, channel status and audio data can be obtained through interrupts or DMA. When RXCFG[1:0] turns to 2'b11, RSPDIF waits for the "B" preamble before starting to save audio data.
RX_STOP	In stop status: RSPDIF is no longer synchronized, user and channel status and audio data reception stop.

The *Figure 51-9. RSPDIF states* shows how to switch between RSPDIF states.

Figure 51-9. RSPDIF states



RX_IDLE

The state machine will enter the RX_IDLE state from the following conditions:

- Set RXCFG[1:0] to 2'b00.
- When in RX_SYNCstate, set RXCFG[1:0] to 2'b00.
- When in RX_RCV state, set RXCFG[1:0] to 2'b00.
- When in RX_STOP state, set RXCFG[1:0] to 2'b00.

The next state of RX_IDLE is:

- RX_SYNC: Set RXCFG[1:0] to 2'b01 or 2'b11.

Note: The software can set RXCFG[1:0] to 0 at any time, and RSPDIF immediately returns to the RX_IDLE state. If the DMA transfer is in progress, it will wait for the DMA transfer completed.

RX_SYNC

The state machine will enter the RX_SYNC state from the following conditions:

- When in RX_IDLE state, set RXCFG[1:0] to 2'b01 or 2'b11.

The next state of RX_SYNC is:

- RX_IDLE : Set RXCFG[1:0] to 2'b00.
- RX_RCV : Set RXCFG[1:0] to 2'b11 and synchronization is completed successfully (SYNDO = 1).
- RX_STOP : Synchronization fails (If the number of synchronization retries has been set, the maximum value has been reached) or the received data has not been decoded correctly (FRERR or SYNERR or TMOUTERR = 1).

Note: When the synchronization is completed, if RXCFG[1:0] = 01, the RSPDIF is still in this state.

RX_RCV

The state machine will enter the RX_RCV state from the following conditions:

- When in RX_SYNC state, set RXCFG[1:0] to 2'b11 and synchronization is completed successfully (SYNDO = 1).

The next state of RX_RCV is:

- RX_IDLE : Set RXCFG[1:0] to 2'b00.
- RX_STOP：The received data has not been decoded correctly (FRERR or TMOUTERR = 1).

RX_STOP

The state machine will enter the RX_STOP state from the following conditions:

- When in RX_SYNC state, synchronization fails (If the number of synchronization retries has been set, the maximum value has been reached) or the received data has not been decoded correctly (FRERR or SYNERR or TMOUTERR = 1).
- When in RX_RCVstate, the received data has not been decoded correctly (FRERR or TMOUTERR = 1).

The next state of RX_STOP is:

- RX_IDLE：Set RXCFG[1:0] to 2'b00.

Note: When RXCFG[1:0] is set to 0, the RSPDIF is disabled, meaning that all the state machines are reset, and RX BUFFER is flushed, as well that flags FRERR, SYNERR and TMOUTERR are reset.

RSPDIF_CTL register access permissions in different states

In the different state stages of the RSPDIF state machine, the access rights to different bit fields of the RSPDIF_CTL register are different. Please refer to ***Table 51-9. RSPDIF_CTL register access permissions in different states*** for details. SPDIF_CTL bit access characteristics in different RSPDIF states. The hardware handles this way to avoid incorrect configuration of the RSPDIF_CTL register. Please note that even user can modify the PTNCPEN, CUNCPEN, VNCPEN and PNCPEN bits in any state of RSPDIF, but these modifications will not affect the value already saved in the DATA register.

Table 51-9. RSPDIF_CTL register access permissions in different states

RSPDIF state	RXCHSEL[2:0]	WFRXA	MAXRT[1:0]	CFCHSEL	DMACBEN	PTNCPEN	CUNCPEN	VNCPEN	PNCPEN	RXDF[1:0]	RXSTEOMEN	DMAREN
RX_IDLE (RXCFG[1:0] = 2'b00)	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

RSPDIF state	RXCHSEL[2:0]	WFRXA	MAXRT[1:0]	CFCHSEL	DMACBEN	PTNCPEN	CUNCPEN	VNCPEN	PNCPEN	RXDF[1:0]	RXSTEOMEN	DMAREN
RX_SYNC (RXCFG[1:0] = 2'b01)	r	r	r	r	rw	rw	rw	rw	rw	rw	rw	rw
RX_RCV (RXCFG[1:0] = 2'b11)	r	r	r	r	rw	rw	rw	rw	rw	r	r	rw

51.3.6. RSPDIF data reception management

RSPDIF double data buffer

RSPDIF implements the double buffering function when data is received. The receiving data double buffer is composed of a 32-bit RX buffer and RSPDIF_DATA register. When the RSPDIF_DATA register is empty and the conversion between the parity bit (P) and the next preamble has been detected, the data in the RX buffer will be immediately transferred to the RSPDIF_DATA register.

The double buffer mechanism of data improves the tolerance of delay. The maximum delay allowed is T_{SAMPLE} - $2T_{pclk}$ - $2T_{rspdif_ck}$ (T_{SAMPLE} is the audio sampling rate of the received stereo audio samples, T_{PCLK} is the period of the rspdif_pclk clock, and T_{rspdif_ck} is the period of rspdif_ck clock.).

The software can obtain the received data by reading the RSPDIF_DATA register or DMA. Recommended DMA operation, please refer to ***DMA function*** for details.

RSPDIF_DATA register

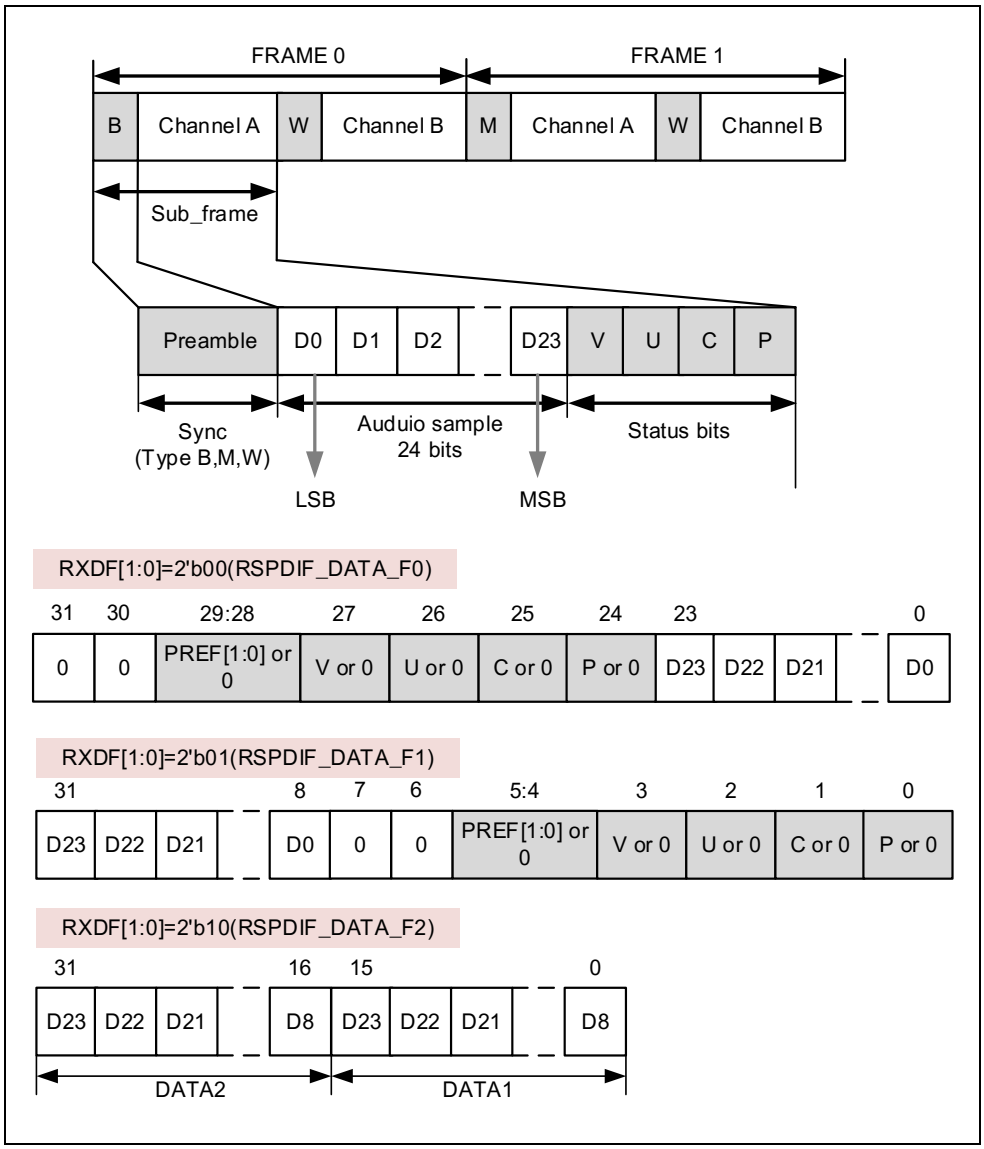
Except for V, U, C, P status bits and headers, each subframe includes up to 24 bits of data. The RSPDIF_DATA register has three formats according to the value of RXDF[1:0] in the RSPDIF_CTL register, and there are three processing methods for the received audio data stream:

- When RXDF[1:0] = 2'b00, the format of the data register is the format described by RSPDIF_DATA_F0, and the data is forced to be right-aligned. The PREF\C\U\V\P bit can be enabled or forced to 0 as required.
- When RXDF[1:0] =2'b01, the format of the data register is the format described by RSPDIF_DATA_F1, and the data is forced to be left-aligned. The PREF\C\U\V\P bit can be enabled or forced to 0 as required.
- When RXDF[1:0] = 2'b11, the format of the data register is the format described by RSPDIF_DATA_F2. This format uses only 16 bits per subframe in non-linear mode. The data of two consecutive sub-frames will be stored in a RSPDIF_DATA register. The format PREF\C\U\V\P bits and data bits cannot be mixed, but the software can still be

obtained by reading the RSPDIF_CHSTAT register or a dedicated DMA channel. When RXSTEOMEN = 1, there is no risk of misalignment (that is, the data from channel A is always stored in RSPDIF_DATA[31:16]). If RXSTEOMEN = 0, there is a risk of misalignment in case of excess. In this case, RSPDIF_DATA[31:16] always contains the oldest value, and RSPDIF_DATA[15:0] always contains the most recent value (*Figure 51-10. RSPDIF_DATA register format*.).

Note: This document describes three data registers: RSPDIF_DATA_Fx (x = 0,1,2), but there is actually only one physical data register.

Figure 51-10. RSPDIF_DATA register format

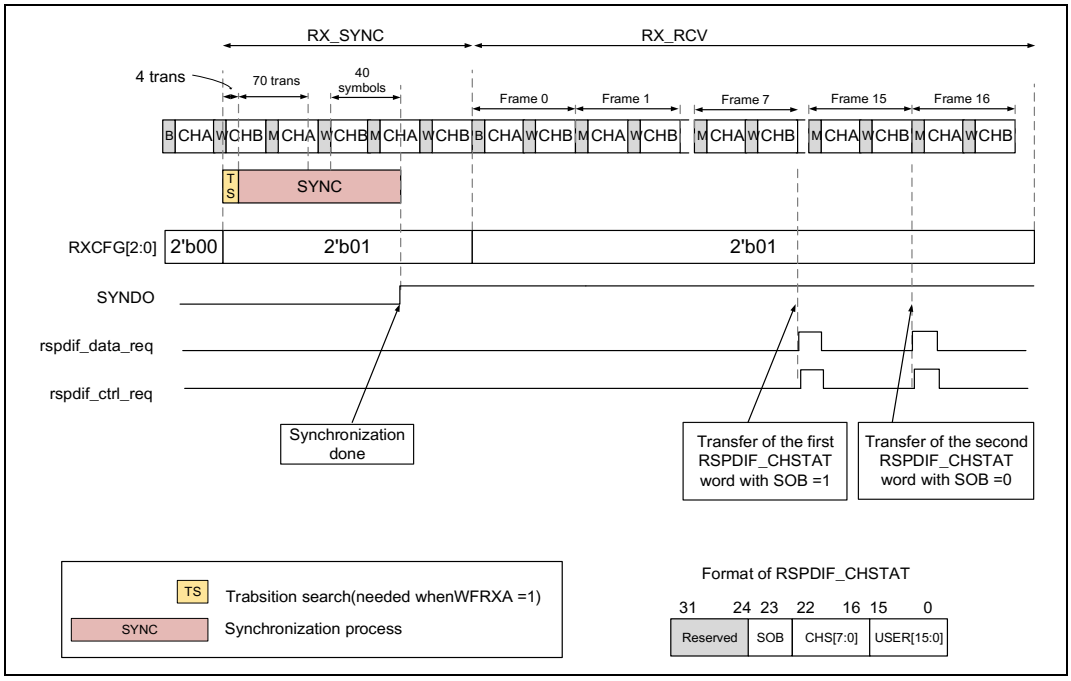


Dedicated control flow

The RSPDIF provides a dedicated channel for receiving user data U and channel status C. By configuring the CFCHSEL bit in the RSPDIF_CTL register, user can select to obtain the channel status bit C from channel A or channel B. When RXCFG[1:0] is set to 2'b01 or 2'b11,

the acquisition will start after the synchronization phase ends. When 8 channel status bits and 16 user data bits are received, pack them and store them in the RSPDIF_CHSTAT register. If the DMACBEN bit of the RSPDIF_CTL register is configured as 1, and the DMA function of the control flow is enabled, a DMA request is triggered. If CHS[0] is the first status bit of the new block, the SOB bit is set to 1. Please refer to ***RX Channel status register (RSPDIF_CHSTAT)***.

Figure 51-11. Channel/user data format



Mixing data and control flow

When RXDF[1:0] = 2'b00 or 2'b01, the RSPDIF_DATA register format is RSPDIF_DATA_F0 or RSPDIF_DATA_F1, which allows the V, U, C, P status bits, preamble and data mixed reception management. By configuring the RSPDIF_CTL register, the user can choose which contents of the V, U, C, P status bits, and preamble are stored in the RSPDIF_DATA register flexibly, as shown in ***Table 51-10. Mixing data and control flow***.

Table 51-10. Mixing data and control flow

BIT	value	Description
PNCPEN	1	Parity error information is masked
	0	Parity error information is copied into RSPDIF_DATA
VNCPEN	1	Validity information is masked
	0	Validity information is copied into RSPDIF_DATA
CUNCPEN	1	Channel status, and user data information are masked
	0	Channel status, and user data information are copied into RSPDIF_DATA
PTNCPEN	1	Preamble type is masked

BIT	value	Description
	0	Preamble type is copied into RSPDIF_DATA

51.3.7. RSPDIF clock management

From the RSPDIF block diagram shown in *Figure 51-1. RSPDIF block diagram*, the RSPDIF block needs the rspdif_pclk clock for the registers interface and the rspdif_ck clock for the RSPDIF DEC module. Rspdif_ck should not be phase locked. In the SYNC block, all signals passing through these clock domains are resynchronized.

In order to decode the incoming S/PDIF stream correctly, the decoding clock rspdif_ck should be at least 11 times higher than the maximum symbol rate or 704 times higher than the audio sampling rate to resample the received data. For example, if the audio rate is 192KHz, and the symbol rate is 12.288MHz (audio rate * number of channels * number of sub-frames), at least 135.2MHz decoding clock (symbol rate * 11) is required. For rspdif_pclk, rspdif_pclk cannot be less than the symbol rate.

Table 51-11. Minimum rspdif_ck frequency and audio sampling rate

Symbol Rate	Minimum rspdif_ck frequency	Audio
3.072 MHz	33.8 MHz	For 48 kHz stream
6.144 MHz	67.6 MHz	For 96 kHz stream
12.288 MHz	135.2 MHz	For 192 kHz stream

In RSPDIF module,

RSPDIF_DEC: RSPDIF_DEC is designed to decode the input RSPDIF bits stream. When RSPDIF is not enable, the clock of RSPDIF_DEC module is gated. The source clock of this module is rspdif_clk.

RSPDIF symbol clock

When RSPDIF is decoding input audio stream, a symbol clock rspdif_symbol_ck is constructed using WIDTH24, WIDTH40 and the value of the symbol boundary. It can be used as a reference core clock for other audio devices (such as SAI or I2S), and can be used for the RSPDIF to I2S bridging function.

When receiving the subframe synchronization preamble, the falling and rising edges of the symbol clock are constructed from the WIDTH24 and WIDTH40 values. When RSPDIF is RX_STOP or RX_IDLE, WIDTH24 and WIDTH40 can also be used to generate the symbol clock. When receiving a subframe, RSPDIF uses symbol boundaries to generate rising edges, use WIDTH24 and WIDTH40 generate falling edges, and the symbol clock's duty cycle is close to 50%.

However, when RSPDIF switches from using WIDTH24 and WIDTH40 to generate a symbol clock to generating a symbol clock from a symbol boundary, the symbol clock duty cycle can be changed, and vice versa. When the symbol clock generation mode is switched or when the S/PDIF signal is re-sampled using the rspdif_ck clock, the symbol clock may be greatly

jittered.

Configure the SCKEN bit in the RSPDIF_CTL register to enable or disable the generation of the symbol clock. Configure the BKSCKEN bit in the RSPDIF_CTL register to enable or disable the generation of the backup symbol clock. However, when the flag SYNERR is set to '1', neither the symbol clock nor the backup clock can be generated, since there is no synchronization. When both SCKEN and BKSCKEN are set to '1', the symbol clock will lose some transitions when the RSPDIF switches from RX_SYNC or RX_RCV to RX_STOP, or RX_IDLE.

For the specific generation conditions of the symbol clock, please refer to *Table 51-12. Conditions of RSPDIF_symbol_ck generation*.

Table 51-12. Conditions of RSPDIF_symbol_ck generation

RSPDIF	SCKEN	BKSCKEN	valid data are received via the selected RSPDIF CH	valid values for WIDTH40 and WIDTH24	WIDTH40 and WIDTH24 contain the valid values from the previous synchronization	no transitions detected on the selected RSPDIF input	complete the Synchronization (SYNDO = '1')	RSPDIF_symb_ck state
Any state	0	x	x	x	x	x	x	Disable
RX_IDLE	1	0	x	x	x	x	1	No output
		1	x	0	x	x	x	
	1	1	x	1	x	x	x	output
RX_SYNC	1	0	x	x	x	x	0	No output
		1	x	0	x	x	0	
	1	0	x	x	x	x	1	output
		1	x	x	x	x	1	
			x	x	1	x	0	
STATE_REV	1	0	x	x	x	1	x	No output
		1	x	x	x	1	x	
	1	0	1	x	x	x	x	output
		1	1	x	x	x	x	
RX_STOP	1	0	x	x	x	x	x	No output
		1	x	0	x	x	x	
	1	1	x	1	x	x	x	output

Note: "0" means that the condition must not be met; "1" means that the condition must be met; "x" means that the condition has no effect on the generation of the symbol clock.

51.3.8. DMA function

RSPDIF can obtain data and control information through the corresponding dedicated DMA channel. Please refer to the *Direct memory access controller (DMA)* for specific channels

used. Set the DMAREN bit in the RSPDIF_CTL register can enable the RSPDIF block data DMA transfer function. When RSPDIF_DATA is not empty, a data transfer request will be sent to the DMA, and the DMA will read the data directly. Set the DMACBEN bit in the RSPDIF_CTL register can enable the RSPDIF block DMA function of channel and user information. For DMA usage of control data, please refer to *Table 51-9. RSPDIF_CTL register access permissions in different states*.

It is not recommended to configure the DMAREN or DMACBEN bit to enable DMA transfer after beginning data reception.

51.3.9. Status、error and interrupt

Status flags

Reception events for data flow (RBNE)

When the RX buffer is not empty, the RBNE bit is set. Indicates that a data is received and when the RSPDIF_DATA register is empty and the conversion between the parity bit (P) and the next preamble has been detected, the data in the RX buffer will be immediately transferred to the RSPDIF_DATA register. The software can obtain this data by reading RSPDIF_DATA register.

Reception event for control flow (CBNE)

When the RSPDIF_CHSTAT is not empty, the CBNE bit is set. Indicates that the control data is received and stored in the RSPDIF_CHSTAT at this time. The software can obtain this data by reading RSPDIF_CHSTAT.

Synchronization done (SYNDO)

When RSPDIF can measure the duration of 24 and 40 consecutive symbols correctly and SYNC is completed, this bit is set. Indicates that the synchronization is completed.

Start of new block interrupt (SYNDB)

When preamble B has been detected, this bit is set. Represented as the beginning of a block.

Note: SYNDB event can only occur when RSPDIF is synchronized to the input stream (SYNDO = 1).

Error flags

Frame error (FERR)

When the conversion sequence is incorrect in the 28 information bits or the preamble appears in an unexpected position or the expected preamble is not received, RSPDIF frame error

occurs and the FRERR bit is set. If the RXDCERRIE bit of the RSPDIF_INTEN register is set, a corresponding interrupt will be generated. Set RXCFG[1:0] to 2'b00 to clear the error flag.

Synchronization error (SYNERR)

When the synchronization fails (if the maximum retries MAXTR is set, MAXTR has been exceeded), RSPDIF synchronization error occurs, and the SYNERR bit is set. If the RXDCERRIE bit of the RSPDIF_INTEN register is set, a corresponding interrupt will be generated. Set RXCFG[1:0] to 2'b00 to clear the error flag.

Timeout error (TMOUTERR)

When no conversion is detected during the 8192 cycle of the rspdif_ck clock and TWCNT overflows, RSPDIF timeout error occurs, and the TMOUTERR bit is set. If the RXDCERRIE bit of the RSPDIF_INTEN register is set, a corresponding interrupt will be generated. Set RXCFG[1:0] to 2'b00 to clear the error flag.

Parity error (PERR)

When the number of 0 or 1 is not an even number among the 28 information bits in a subframe, RSPDIF parity error occurs, and the PERR bit is set. If the PERRIE bit in the SPDIF_INTEN register is set, a corresponding interrupt will be generated. Set the PERRC bit to 1, to clear the PERR flag.

Note: Even if the interrupt is suspended, receiving incoming data will not be suspended, and RSPDIF will continue to send data to RSPDIF_DATA. If the software wants to ensure the consistency between the data read in the RSPDIF_DATA register and the value of the PERR bit, the PNCPEN bit must be set to 0.

Overrun error (RXORERR)

When both RSPDIF_DATA and RX buffer are full, while RSPDIF_DEC still writes a new data into the RX buffer, RSPDIF overrun error occurs, RXORERR is set, and this new data will not be written into the buffer. If the RXORERRIE bit of the RSPDIF_INTEN register is set, a corresponding interrupt will be generated. Set the RXORERRC bit to 1 to clear the RXORERR flag.

Note: Even if the RXORERR flag is suspended, when RXSTEOMEN = 0 and the RX buffer is empty, the next incoming data will still be stored. See ***Figure 51-12. RSPDIF overrun error when RXSTEOMEN = 0 and RXDF[1:0] = 2'b0x*** and ***Figure 51-13. RSPDIF overrun error when RXSTEOMEN = 0 and RXDF[1:0] = 2'b10***.

Figure 51-12. RSPDIF overrun error when RXSTEOMEN = 0 and RXDF[1:0] = 2'b0x

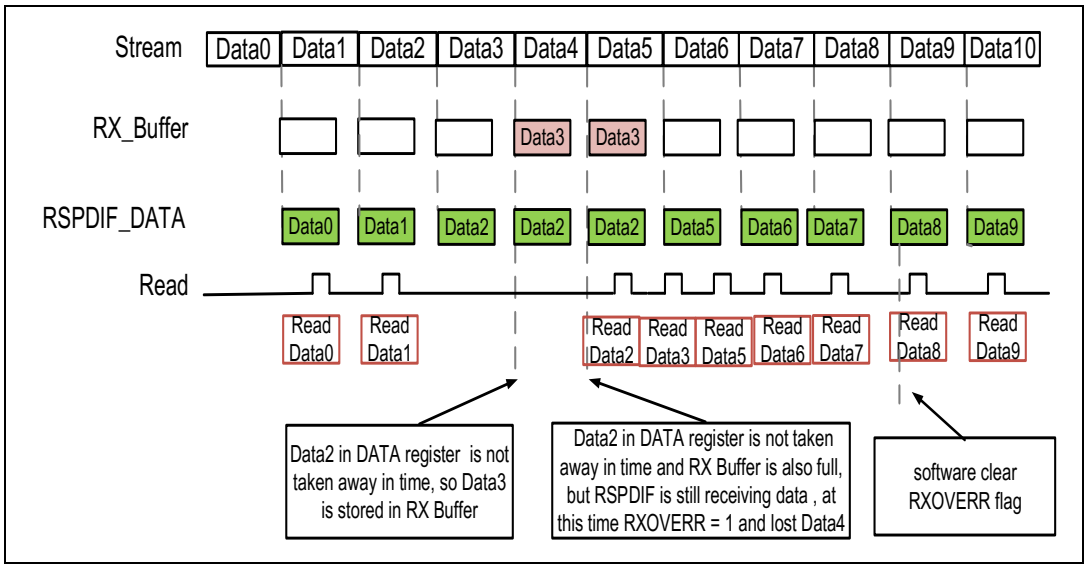
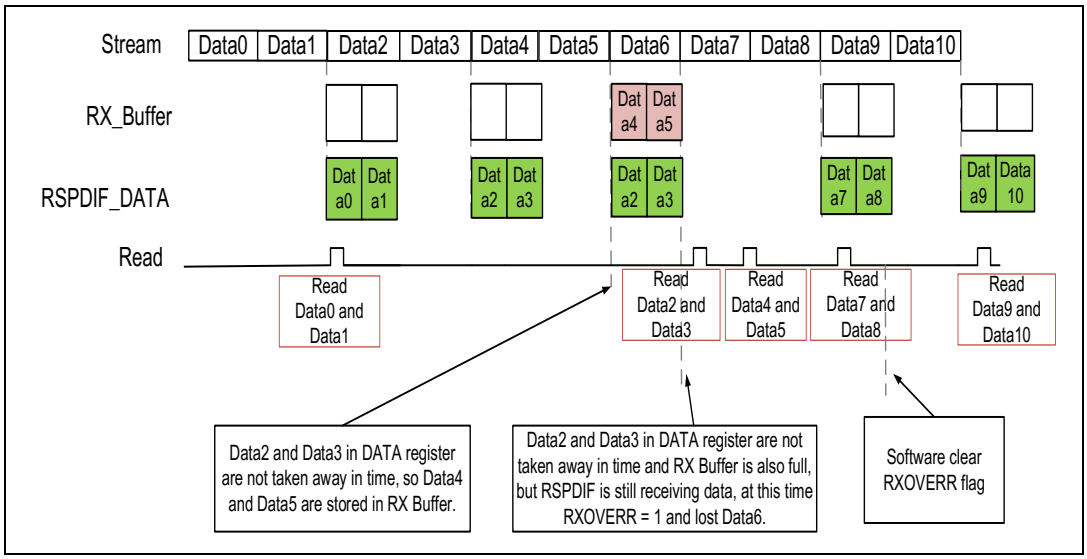


Figure 51-13. RSPDIF overrun error when RXSTEOMEN = 0 and RXDF[1:0] = 2'b10



If RXSTEOMEN = 1, RSPDIF transmits stereo data. In order to avoid the problem of misalignment of the left and right channels, RSPDIF must discard the second data, even if there is space in the RX buffer. The incoming data can be written to the RX buffer normally, even if the RXORERR flag is still suspended. Refer to ***Figure 51-14. RSPDIF overrun error when RXSTEOMEN = 1 and RXDF[1:0] = 2'b0x*** and ***Figure 51-15. RSPDIF overrun error when RXSTEOMEN = 1 and RXDF[1:0] = 2'b10***.

Figure 51-14. RSPDIF overrun error when RXSTEOMEN = 1 and RXDF[1:0] = 2'b0x

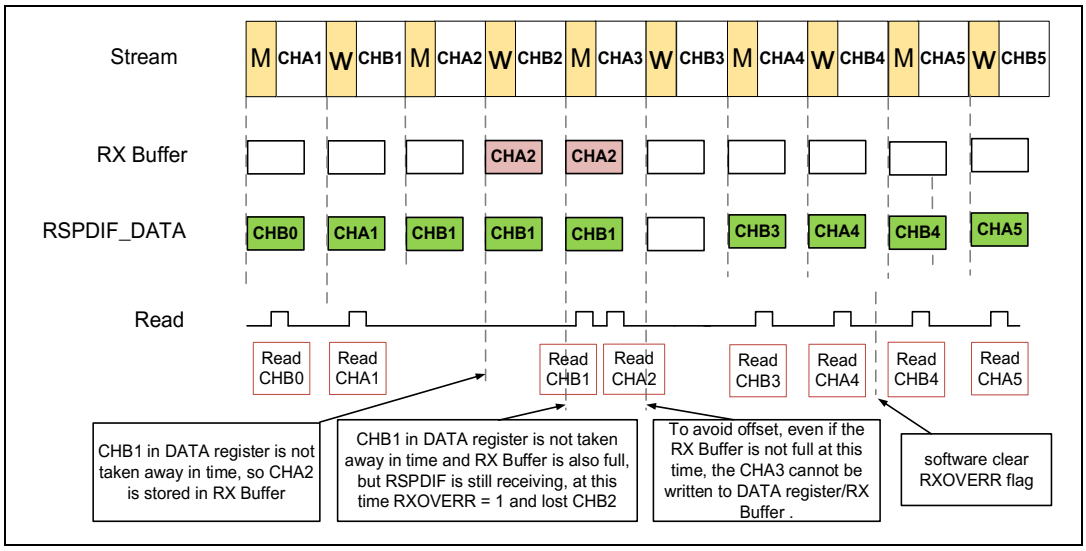
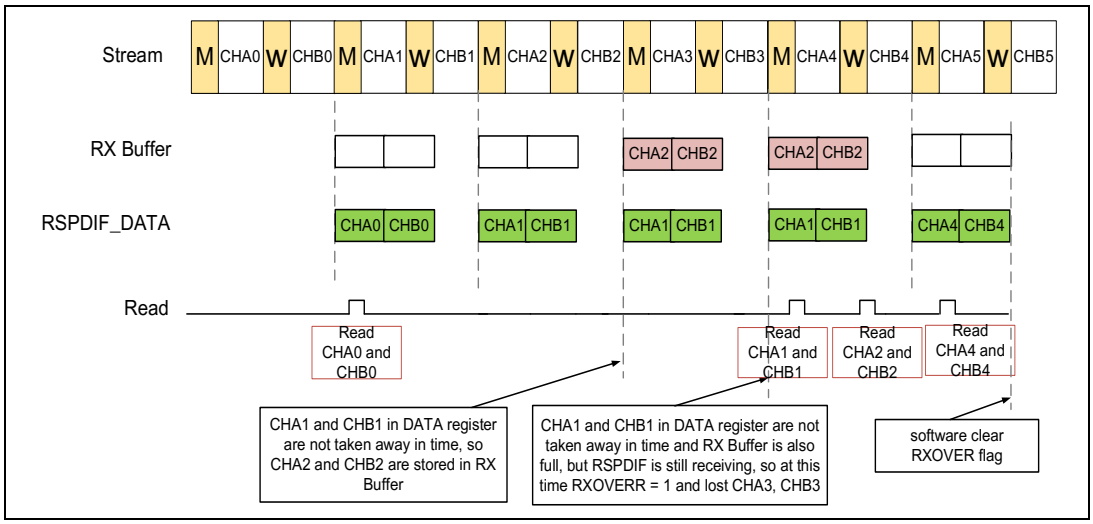


Figure 51-15. RSPDIF overrun error when RXSTEOMEN = 1 and RXDF[1:0] = 2'b10



Interrupts

The RSPDIF interrupt events and flags are listed in *Table 51-13. RSPDIF interrupt events*.

Table 51-13. RSPDIF interrupt events

Flag	Interrupt event	Clear method	Enable control bit
SYNDO	SYNC is completed	Set SYNDBC bit in RSPDIF_STATC register to 1	SYNDOIE
RBNE	RSPDIF receive buffer is not empty	Read RSPDIF_DATA register	RBNEIE
PERR	In the 28 information bits of one subframe, 0 and 1 are not even numbers	Set PERRC bit in RSPDIF_STATC register to 1	PERRIE
RXORERR	Both RSPDIF_DATA and RX buffer are	Set RXORERRC bit in	RXORERRIE

Flag	Interrupt event	Clear method	Enable control bit
	full, and RSPDIF_DEC still write a new data to RX buffer	RSPDIF_STATC register to 1	
CBNE	RSPDIF control flow receive buffer is not empty	Read RSPDIF_CHSTAT register	CBNEIE
SYNDB	Preamble B has been detected	Set SYDBC bit in RSPDIF_STATC register to 1	SYNDBIE
SYNERR	The synchronization fails	Clear RXCFG bits in RSPDIF_CTL register	RXDCERRIE
FRERR	The conversion sequence is incorrect in the 28 information bits or the preamble appears in an unexpected position or the expected preamble is not received		
TMOUTERR	TWCNT overflows		

51.4. Register definition

RSPDIF base address: 0x4000 4000

51.4.1. Control register (RSPDIF_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										BKSCKEN	SCKEN	Reserved	RXCHSEL[2:0]		
										rw	rw			rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	WFRXA	MAXRT[1:0]		CFCHSEL	DMACBEN	PTNCPEN	CUNCPEN	VNCPEN	PNCPEN	RXDF[1:0]		RXSTEOMEN	DMAREN	RXCFG[1:0]	
	rw	rw		rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	

Bits	Fields	Descriptions
31:22	Reserved	Must be kept at reset value.
21	BKSCKEN	Backup symbol clock enable This bit is set and cleared by software. 0: The RSPDIF does not generate a backup symbol clock. 1: The RSPDIF generates a backup symbol clock if SCKEN = 1 Please refer to ***RSPDIF clock management*** for information on how to program this field.
20	SCKEN	Symbol clock enable This bit is set and cleared by software. 0: The RSPDIF does not generate a symbol clock 1: The RSPDIF generates a symbol clock
19	Reserved	Must be kept at reset value.
18:16	RXCHSEL[2:0]	RSPDIF input channel selection 000: RSPDIF_CH0 selected 001: RSPDIF_CH1 selected 010: RSPDIF_CH2 selected 011: RSPDIF_CH3 selected 100 ~ 111: reserved
15	Reserved	Must be kept at reset value.
14	WFRXA	Wait for the four valid transition signal of the selected RSPDIF channel

		This bit is set and cleared by software. 0: Before starting the synchronization process, RSPDIF does not wait for the valid conversion signal from the selected RSPDIF channel. 1: Before starting the synchronization process, RSPDIF wait for the valid conversion signal from the selected RSPDIF channel.
13:12	MAXRT[1:0]	Maximum number of retries allowed during the RSPDIF synchronization phase 00: Do not allow retry (only one chance) 01: Allow up to 3 retries 10: Allow up to 15 retries 11: Allow up to 63 retries
11	CFCHSEL	The control flow acquires channel state source selection This bit is set and cleared by software. 0: Gets channel status from channel A 1: Gets channel status from channel B
10	DMACBEN	Control buffer DMA enable for control flow This bit is set and cleared by software. 0: Disable control flow DMA mode. 1: Enable control flow DMA mode. When this bit is set, a DMA request is generated whenever the CBNE flag is set.
9	PTNCPEN	Preamble type bits no copy enable bit This bit is set and cleared by software. 0: Copy the preamble type bits into the RSPDIF_DATA 1: Do not copy the preamble type bits into the RSPDIF_DATA, but write 0 instead.
8	CUNCPEN	Channel status and user bits no copy enable bit This bit is set and cleared by software. 0: Copy the channel status and user bits into the RSPDIF_DATA. 1: Do not copy the channel status and user bits into the RSPDIF_DATA, but write 0 instead.
7	VNCPEN	Validity bit no copy enable bit This bit is set and cleared by software. 0: Copy the validity bit into the RSPDIF_DATA. 1: Do not copy the validity bit into the RSPDIF_DATA, but write 0 instead.
6	PNCPEN	Parity error bit no copy enable bit This bit is set and cleared by software. 0: Copy the parity error bit into the RSPDIF_DATA. 1: Do not copy the parity error bit into the RSPDIF_DATA, but write 0 instead.
5:4	RXDF[1:0]	RX data format selection This bit is set and cleared by software. 00: The data format is described in the RSPDIF_DATA_F0 register, audio data is right aligned (LSB)

		01: The data format is described in the RSPDIF_DATA_F1 register, audio data is left aligned (MSB) 10: The data format is described in the RSPDIF_DATA_F2 register, which packs two 16-bit audio data into one 32-bit data 11: reserved
3	RXSTEOMEN	RX stereo mode enable This bit is set and cleared by software. 0: MONO mode 1: STEREO mode This bit is used in case of overrun situation in order to handle misalignment
2	DMAREN	Receiver DMA enable for data flow This bit is set and cleared by software. 0: Disable reception DMA mode. 1: Enable reception DMA mode. When this bit is set, the DMA request is generated whenever the RBNE flag is set.
1:0	RXCFG[1:0]	RSPDIF configuration 00: Disable RSPDIF 01: Enable RSPDIF synchronization only 10: Reserved 11: Enable RSPDIF

51.4.2. Interrupt enable register (RSPDIF_INTEN)

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									RXDCERRIE	SYNDOIE	SYNDBIE	RXORERRIE	PERRIE	CBNEIE	RBNEIE
									rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value.
6	RXDCERRIE	RSPDIF data decoding error interrupt enable This bit is set and cleared by software. 0: Disable 1: Enable When this bit is set, a RSPDIF data decoding error interrupt is generated if

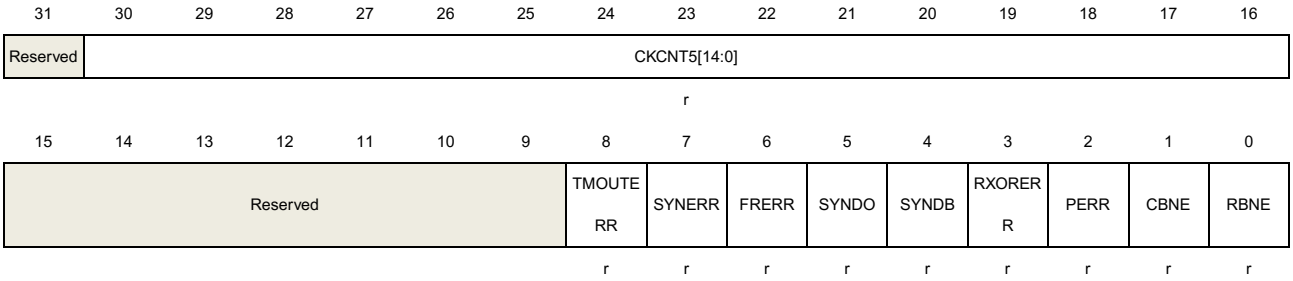
		SYNERR, TMOUTERR or FRERR bit in the RSPDIF_STAT register is set.
5	SYNDOIE	Synchronization done interrupt enable This bit is set and cleared by software. 0: Disable 1: Enable When this bit is set, a synchronization done interrupt is generated if SYNDO bit in the RSPDIF_STAT register is set.
4	SYNDBIE	Synchronization block detected interrupt enable This bit is set and cleared by software. 0: Disable 1: Enable When this bit is set, a synchronization block detected interrupt is generated if SYNDB in the RSPDIF_STAT register is set .
3	RXORERRIE	RX overrun error interrupt enable This bit is set and cleared by software. 0: Disable 1: Enable When this bit is set, a RX overrun error interrupt is generated if RXORERR in the RSPDIF_STAT register is set.
2	PERRIE	Parity error interrupt enable This bit is set and cleared by software. 0: Disable 1: Enable When this bit is set, a parity error interrupt is generated if PERR in the RSPDIF_STAT register is set.
1	CBNEIE	RSPDIF_CHSTAT register no empty interrupt enable This bit is set and cleared by software. 0: Disable 1: Enable When this bit is set, a RSPDIF_CHSTAT register no-empty interrupt is generated if CBNE in the RSPDIF_STAT register is set.
0	RBNEIE	RSPDIF_DATA register no empty interrupt enable This bit is set and cleared by software. 0: Disable 1: Enable. When this bit is set, a RSPDIF_DATA register no-empty interrupt is generated if RBNE in the RSPDIF_STAT register is set.

51.4.3. Status register (RSPDIF_STAT)

Address offset: 0x08

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.
30:16	CKCNT5[14:0]	The number of consecutive time clock cycles of 5 symbols counted using rspdif_ck This value can be used to estimate the S/PDIF symbol rate. Its accuracy is limited by the frequency of rspdif_ck. For example if the rspdif_ck is fixed to 84 MHz, and CKCNT5 = 147d. The estimated sampling rate of the S/PDIF stream is: Fs = 5 x rspdif_ck / (CKCNT5 x 64) ~ 44.6 kHz, so the closest standard sampling rate is 44.1 kHz. Note that CKCNT5 is updated by the hardware when SYNDO goes high, and then every frame.
15:9	Reserved	Must be kept at reset value.
8	TMOUTERR	Time out error 0: No timeout error is detected 1: Timeout error is detected When the counter TWCNT value reaches its maximum value, the bit is set to 1 by the hardware. It indicates that the time between transitions is too long. This usually means that there is no valid signal on the RSPDIF_CH input. If RXDCERRIE=1 in the RSPDIF_INTEN register, an interrupt is generated. Set RXCFG[1:0] to 2'b00 in RSPDIF_CTL register to clear this bit.
7	SYNERR	Synchronization error 0: No synchronization error is detected 1: Synchronization error is detected When synchronization fails due to the number of MAXRT retries, the bit is set to 1 by the hardware. If RXDCERRIE=1 in the RSPDIF_INTEN register, an interrupt is generated. Set RXCFG[1:0] to 2'b00 in RSPDIF_CTL register to clear this bit.
6	FRERR	Frame error 0: no Manchester coding error detected 1: Manchester coding error detected

		When an error occurs during receiving data, the bit is set to 1 by the hardware. Preamble does not appear in expected position, short transitions are not grouped by pair. This bit can be set to 1 by the hardware only when synchronization is complete (SYNDO = 1). If RXDCERRIE=1 in register RSPDIF_INTEN, an interrupt is generated. Set RXCFG[1:0] to 2'b00 in RSPDIF_CTL register to clear this bit.
5	SYNDO	Synchronization done 0: Synchronization is pending 1: Synchronization is completed When the synchronization phase is properly completed, the bit is set to 1 by the hardware.. If SYNDOIE = 1 in the RSPDIF_INTEN register, an interrupt is generated. Set SYNDOC bit in RSPDIF_STATC register to clear this bit.
4	SYNDB	Synchronization block detected 0: No "B" preamble detected 1: "B" preamble has been detected When a "B" preamble has been detected, the bit is set to 1 by the hardware. If SYNDBIE = 1 in the RSPDIF_INTEN register, an interrupt is generated. Set SYNDBC bit in RSPDIF_STATC register to clear this bit.
3	RXORERR	RX overrun error 0: No Overrun error 1: Overrun error is detected When RSPDIF is still receiving data while RBNE = 1 and both RSPDIF_DATA and RX buffer are full, the bit is set to 1 by the hardware. If RXORERRIE=1 in the RSPDIF_INTEN register, an interrupt is generated. Set RXORERRC bit in RSPDIF_STATC register to clear this bit. **Note:** When this bit is set, the RSPDIF_DATA register content is not lost but the last data received are lost.
2	PERR	Parity error 0: No parity error 1: Parity error When the received subframe contains an odd number of 0 and 1 in the data and status bits, the bit is set to 1 by the hardware. If PERRIE = 1 in the RSPDIF_INTEN register, an interrupt is generated. Set PERRC bit in RSPDIF_STATC register to clear this bit.
1	CBNE	RSPDIF_CHSTAT register is not empty 0: No control word available on RSPDIF_CHSTAT register 1: A control word is available on RSPDIF_CHSTAT register When there is control flow data in the RSPDIF_CHSTAT register, the bit is set to 1 by the hardware. If CBNEIE = 1 in the RSPDIF_INTEN register, an interrupt is generated.

		Read RSPDIF_CHSTAT register to clear this bit.
0	RBNE	RX buffer is not empty 0: Data is not received 1: Received data is ready to be read. When a valid data is available into RX buffer, the bit is set to 1 by the hardware. If RBNEIE = 1 in the RSPDIF_INTEN register, an interrupt is generated. Read RSPDIF_DATA register to clear this bit.

51.4.4. Status flag clear register (RSPDIF_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										SYNDOC	SYNDBC	RXORERRC	PERRC	Reserved	
										w	w	w	w		

Bits	Fields	Descriptions
31:6	Reserved	Must be kept at reset value.
5	SYNDOC	clears the synchronization done flag Set this bit to clear the SYNDO flag in the RSPDIF_STAT register.
4	SYNDBC	clears the synchronization block detected flag Set this bit to clear the SYNDB flag in the RSPDIF_STAT register.
3	RXORERRC	clears the RX overrun error flag Set this bit to clear the OVR flag in the RSPDIF_STAT register.
2	PERRC	clears the parity error flag Set this bit to clear the PERR flag in the RSPDIF_STAT register.
1:0	Reserved	Must be kept at reset value.

51.4.5. RX data register (RSPDIF_DATA)

Address offset: 0x10

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

According to the value of RXDF[1:0], the receive data register has three different formats.

When RXDF[1:0]=2'b00, the format is as follows:

RX data register format 0 (RSPDIF_DATA_F0)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		PREF[1:0]		C	U	V	P	DATA[23:16]							
		r		r	r	r	r	r							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA[15:0]															
r															

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29:28	PREF[1:0]	Preamble type bits These bits indicate the preamble received. 00: not used 01: Preamble B received 10: Preamble M received 11: Preamble W received Note that if PTNCPEN = 1, this field is forced to zero
27	C	Channel status bit Contains the received channel status bit, if CUNCPEN = 0, otherwise it is forced to 0
26	U	User bit Contains the received user bit, if CUNCPEN = 0, otherwise it is forced to 0
25	V	Validity bit Contains the received validity bit if VNCPEN = 0, otherwise it is forced to 0
24	P	Parity error bit Contains a copy of PERR bit if PNCPEN = 0, otherwise it is forced to 0
23:0	DATA[23:0]	Data value Contains the 24 received data bits, right-aligned.

When RXDF[1:0]=2'b01, the format is as follows:

RX data register format 1 (RSPDIF_DATA_F1)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA [23:8]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA [7:0]								Reserved		PREF[1:0]		C	U	V	P
r										r		r	r	r	r

Bits	Fields	Descriptions
31:8	DATA[23:0]	Data value Contains the 24 received data bits, left-aligned.
7:6	Reserved	Must be kept at reset value.
5:4	PREF[1:0]	Preamble type bits These bits indicate the preamble received. 00: not used 01: Preamble B received 10: Preamble M received 11: Preamble W received Note that if PTNCPEN = 1, this field is forced to zero
3	C	Channel status bit Contains the received channel status bit, if CUNCPEN = 0, otherwise it is forced to 0
2	U	User bit Contains the received user bit, if CUNCPEN = 0, otherwise it is forced to 0
1	V	Validity bit Contains the received validity bit if VNCPEN = 0, otherwise it is forced to 0
0	P	Parity error bit Contains a copy of PERR bit if PNCPEN = 0, otherwise it is forced to 0

When RXDF[1:0]=2'b10, the format is as follows:

RX data register format 2 (RSPDIF_DATA_F2)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA2[15:0]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA1[15:0]															
r															

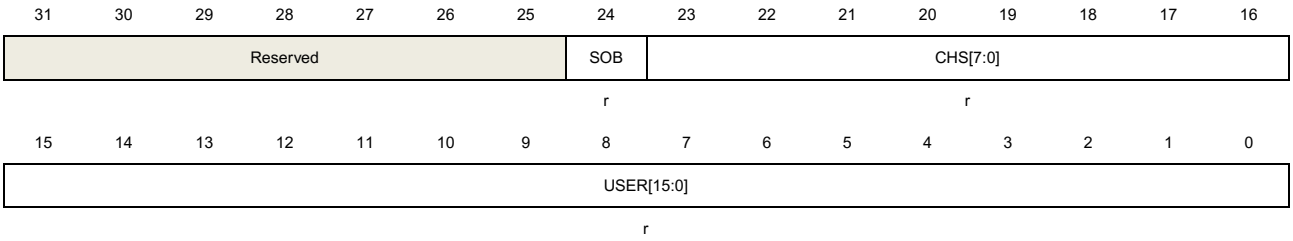
Bits	Fields	Descriptions
31:16	DATA2[15:0]	When in steo mode: this field contains the channel A data. When in mono mode: this field contains the oldest value.
15:0	DATA1[15:0]	When in steo mode: this field contains the channel B data. When in mono mode: this field contains the more recent value.

51.4.6. RX Channel status register (RSPDIF_CHSTAT)

Address offset: 0x14

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



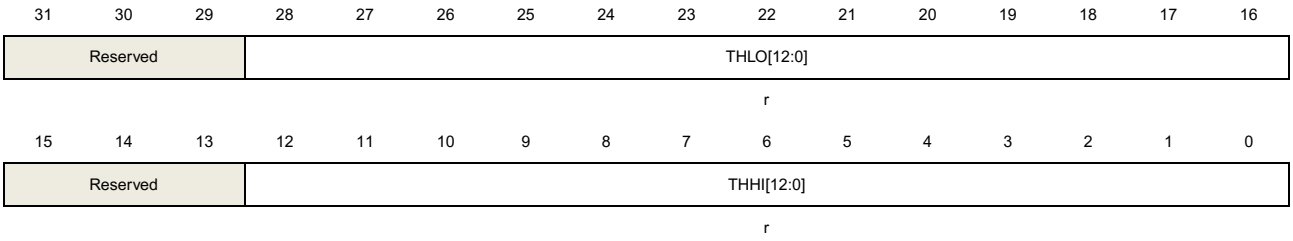
Bits	Fields	Descriptions
31:25	Reserved	Must be kept at reset value.
24	SOB	Start of block This bit indicates if the bit CHS[0] corresponds to the first bit of a new block 0: CHS[0] is not the first bit of a new block 1: CHS[0] is the first bit of a new block
23:16	CHS[7:0]	Channel status information Bit CHS[0] is the oldest value
15:0	USER[15:0]	User data information Bit USR[0] is the oldest value, and comes from channel A, USR[1] comes channel B. So USR[n] bits come from channel A is n is even, otherwise they come from channel B.

51.4.7. RX data threshold register (RSPDIF_DTH)

Address offset: 0x18

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.
28:16	THLO[12:0]	Low threshold

		THLO = 1.5 x UI / T_{rspdif_ck} This field contains the current low threshold estimation. This value can be used to estimate the sampling rate of the received stream. The accuracy of THLO is limited to a period of the rspdif_ck. The sampling rate can be estimated as follow: Sampling Rate = [2 x THLO x T_{rspdif_ck} +/- T_{rspdif_ck} x 2/3 Note that THLO is updated by the hardware when SYNDO goes high, and then every frame.
15:13	Reserved	Must be kept at reset value.
12:0	THHI[12:0]	High threshold THHI = 2.5 x UI / T_{rspdif_ck} This field contains the current high threshold estimation. This value can be used to estimate the sampling rate of the received stream. The accuracy of THHI is limited to a period of the rspdif_ck. The sampling rate can be estimated as follow: Sampling Rate = [2 x THHI x T_{rspdif_ck} +/- T_{rspdif_ck}] x 2/5 Note that THHI is updated by the hardware when SYNDO goes high, and then every frame.

52. Serial Audio Interface (SAI)

52.1. Overview

The Serial Audio Interface (SAI) is designed to target a wide range of commonly used audio protocols, both in mono and stereo modes, such as I2S, PCM/DSP, AC'97 (Audio Codec '97), LSB or MSB-justified and TDM (Time Division Multiplexing). The audio module can be output as SPDIF (Sony/Philips Digital Interface) when configured in transmitter mode.

To realize these multitudes of configuration, two identically independent audio sub-blocks are implemented. Each audio sub-block contains up to 4 IO pins (SD, SCK, FS, and MCLK). Parts of these pins are designed to be shareable when the two audio sub-blocks are configured to be synchronous with each other.

The SAI can be configured to any of the master/slave and transmitter/receiver combination, full/half-duplex operating mode depends on synchronous/asynchronous configuration of the audio sub-blocks.

52.2. Characteristics

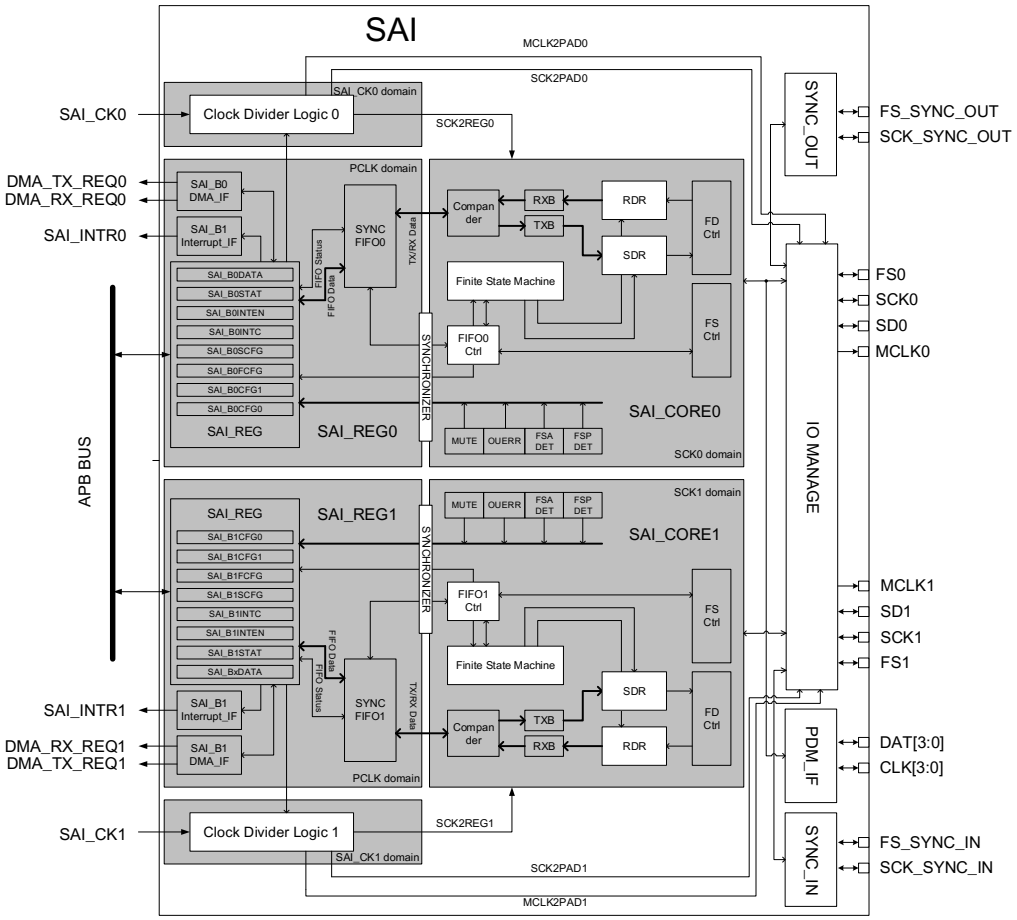
- Two independent audio sub-blocks.
- Each audio sub-block can be configured as any of the master/slave and transmitter/receiver combination with 8-word FIFO.
- Local clock divider logic to satisfy the various audio sampling rates.
- Flexible audio protocol configuration such as I2S, PCM/DSP, AC'97, LSB or MSB-justified and TDM.
- **PDM (Pulse Density Modulation) interface, supporting** up to 4 microphone pairs (GD32H77x_78x support 3 microphone pairs).
- Mono/Stereo audio capability with mute option.
- **SPDIF output in transmitter mode.**
- **A-law and Mu-law compander.**
- Frame Synchronization configuration (active level, active length and offset).
- Each audio frame contains up to 16 configurable slots.

- Slot length is flexible, and can be configured as active or inactive.
- Each slot can hold a data of size 8-, 10-, 16-, 20-, 24-, and 32-bits with configurable first bit offset, and configurable LSB or MSB data transfer.
- Serial clock strobe edge selection (SCK).
- Error flag and interrupt sources
  - FIFO overrun and underrun
  - Frame synchronization advanced detection in slave mode.
  - Frame synchronization postpone detection in slave mode.
  - AC'97 codec not ready.
  - Wrong clock configuration
  - Mute detection in receiver mode.
- Each audio sub-block has its own DMA interface. Support slave mode with a frequency up to 4MHz

52.3. Function overview

52.3.1. Block diagram

Figure 52-1. block diagram



Note: GD32H77x_78x PDM interface signals are only valid for DAT[2:0] and CLK[1:0].

The flexible audio transceiver is the integration of two identically independent sub-blocks, with an IO management module attached to their outputs. Each audio sub-block is composed of three isolated timing domain, the SAI_CK, FCLK, and PCLK domain. Clock divider logic which defines the audio sampling rate resides in SAI_CK domain, its clock output is wired to FCLK domain where SAI main functional state machine, compress/decompress, transmission/reception logic and interrupt generation logic is. The main control registers and the synchronous FIFO is located in the PCLK domain. The synchronous FIFO can be accessed either by ARM CPU APB Bus or by DMA controller.

Each of the audio sub-block can be configured as any combination of master/slave and transmitter/receiver pair, the Frame Synchronization (FS) and Serial Clock (SCK) are generated in master mode, and received in slave mode from an external master or another audio sub-block in synchronous mode. The Master Clock (MCLK) generated only in master

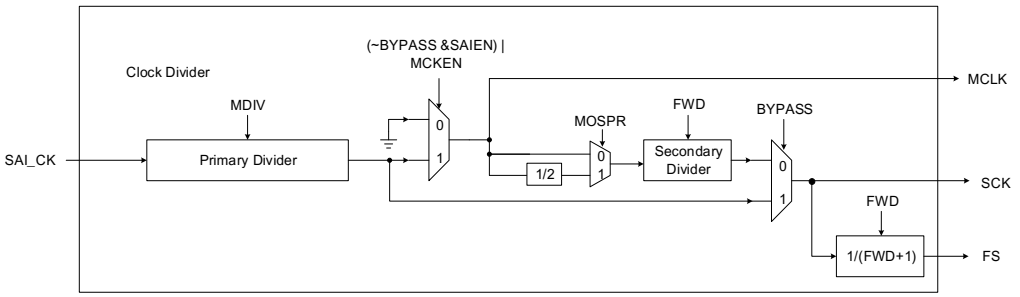
mode for external DAC/ADC operations. There is one exception that, when SAI is configured in AC'97 protocol, FS is forced to be an output, independent of master/slave configurations. The Serial Data (SD) IO pin is configured as output in transmission and input in reception.

The IO Management block controls the IO pins of each audio sub-block, when the two sub-blocks are declared synchronous with each other, FS, SCK and MCLK can be shared, those pins of the synchronous sub-block are freed and can be used as general purpose IO.

52.3.2. Clock divider

The clock divider logic which is present in both SAI audio sub-blocks turns on only when they are configured as master devices, otherwise it is off and both MCLK and SCK output stays low. The divider's source clock SAI_CK is recommended to take on 2 specific values, 45.1584 MHz and 49.152 MHz to generate standard audio sampling rate. The clock divider logic is composed of a primary clock divider which is used to generate the required master clock (MCLK) and a secondary clock divider which is used to generate the bit clock (SCK), its block diagram is as follows *Figure 52-2. Clock divider logic*.

Figure 52-2. Clock divider logic



The primary clock divider's ratio MDIV is a direct link to Master Clock Divider Ratio control field inside SAI control register, its output frequency could be obtained by the following formula.

$$F_{MCLK} = \begin{cases} \frac{f_{SAI_CK}}{MDIV}, & MDIV \neq 0 \\ f_{SAI_CK}, & MDIV = 0 \end{cases} \quad (52\text{-}1)$$

Note: The formula above holds only when BYPASS is not valid, SAI is on and MCLKEN is on, otherwise, MCLK stays low level.

The secondary clock divider's ratio is determined by the frame width (FWD) control field in the SAI_BxFCFG register and the master clock oversampling rate (MOSPR) control field in the SAI_BxCFG0 register, when BYPASS is off, the secondary divider's input is linked to the primary divider's output stage. The following formula determines the relationship between SAI_CK and bit clock (SCK) sampling rate.

$$F_{SCK} = \frac{f_{SAI_CK} \times (FWD + 1)}{MDIV \times (MOSPR + 1) \times 256} \quad (52\text{-}2)$$

The frame synchronization frequency is given by:

$$f_{FS} = \frac{f_{SAI_CK}}{MDIV \times (MOSPR + 1) \times 256} \quad (52\text{-}3)$$

When BYPASS is set, the master clock (MCLK) is turned off with a fixed output value of 0, while the bit clock (SCK) is linked directly to SAI_CK. In addition, there is no restriction on the frame length value as long as the frame length is bigger or equal to 8.

When BYPASS is cleared, It is necessary to set (FWD + 1) equals to the result of an exponential function of base 2 in master mode to guaranty T_{SCK} divisible by T_{MCLK}.

Some of the commonly used audio sampling rate configurations when frame width is 256-bit are listed in the ***Table 52-1. Commonly used audio sampling rate***.

Table 52-1. Commonly used audio sampling rate

SAI_CK Clock Rate	Standard Audio Sampling Rate	Master Clock Divide Ratio
192kHz x 256	192 kHz	MDIV = 1
	96 kHz	MDIV = 2
	48 kHz	MDIV = 4
	16 kHz	MDIV = 12
	8 kHz	MDIV = 24
44.1kHz x 256	44.1 kHz	MDIV = 1
	22.05 kHz	MDIV = 2
	11.025 kHz	MDIV = 4

52.3.3. Operating mode

The SAI audio sub-blocks can be configured in any combination of the master/slave and transmitter/receiver pair independently.

Master

Frame synchronization (FS) is always generated by the master at the start of a frame as long as the FIFO is not empty to indicate frame start or channel identification. Serial clock (SCK) and master clock (MCLK) are other signals generated by the master, SCK is used exclusively by the slave, acting as its bit clock. Unlike FS, SCK and MCLK's generation is not conditioned by the emptiness of the FIFO, they are generated as soon as the audio sub-block is enabled.

Slave

Slave receives FS and SCK signal from the master, its source depends on whether the audio sub-block is declared synchronous or asynchronous. When asynchronous mode is chosen, the FS and SCK signals source is connected directly to the chip level IO ports, while when synchronous mode is chosen, the FS and SCK signal is wired to another audio sub-block's FS and SCK signal. Users must always enable the slave before the master; otherwise the slave will not receive the complete data from the master.

Transmitter

When the audio sub-block is configured as transmitter, serial data (SD) is an output. If the FIFO is still empty after the audio sub-block is enabled, a 0 value is sent, and the underrun flag (OUERR) is raised.

Receiver

When the audio sub-block is configured as receiver, serial data (SD) is an input. Slave receiver always looks at the FS signal, when the first active edge is observed, it stores the received data, the coordination of the following data reception is handled by the internal finite state machine, and the receiver terminates reception at the end of a frame where a disable signal has already been set.

52.3.4. SAI synchronization mode

SAI synchronization mode includes two modes: internal synchronization mode and external synchronization mode.

Internal synchronization

The internal synchronization mode has the advantage of reducing the number of external pins occupied during communication. The SAI sub-blocks SAI_B0 and SAI_B1 run synchronously, and they will share the SAI_FS and SAI_SCK signals, thereby releasing the GPIO pins of SCKx, FSx and MCLKx.

The SAI sub-block in internal synchronization mode can be configured in the following modes in full-duplex communication:

1. SAI_B0 (or SAI_B1) is configured as the master module, and SAI_B1 (or SAI_B0) is configured as the slave module.
2. Both SAI_B0 and SAI_B1 are configured as slave modules.
3. SAI_B0 (or SAI_B1) is configured as an asynchronous module, and SAI_B1 (or SAI_B0) is configured as a synchronous module.

Note: Due to the internal resynchronization phase, the frequency of the PCLK APB is required to be above twice the bit rate clock frequency.

External synchronization

External synchronization means that the SAI audio sub-block is synchronized with other SAIs. The SYNO[1:0] in the SAI_SYNCFG register is configured to determine the synchronization source that provides FS and SCK signals for other SAIs(SAI_B0 or SAI_B1). By configuring SYNI[1:0] in the SAI_SYNCFG register, it is determined which SAI is selected by the SAI receiving the synchronization signal for synchronization. By configuring the SYNCMOD[1:0] in the SAI_BxCFG0 register, specify whether the SAI audio sub-block is synchronized with other SAIs.

If both audio sub-blocks in SAI need to be synchronized with another SAI, either configure each audio sub-block to synchronize with the other SAI through the SYNCMOD[1:0] bits, or configure one audio sub-block to synchronize with the other SAI through the SYNCMOD[1:0] bits and then configure the other audio sub-block to synchronize with the first audio sub-block through the SYNCMOD[1:0] bits.

Table 52-2. External synchronization configuration is a reference for external synchronization configuration.

Table 52-2. External synchronization configuration

SAI module	SYNI =3	SYNI =2	SYNI =1	SYNI =0
SAI0	SAI3 sync.	SAI2 sync.	SAI1 sync.	Reserved
SAI1	SAI3 sync.	SAI2 sync.	Reserved	SAI0 sync.
SAI2	SAI3 sync.	Reserved	SAI1 sync.	SAI0 sync.
SAI3	Reserved	SAI2 sync.	SAI1 sync.	SAI0 sync.

52.3.5. Frame configuration

Frame synchronization

Frame synchronization is the coordination signal between master and slave to initiate a transfer. A number of parameters were implemented to manipulate with its waveform.

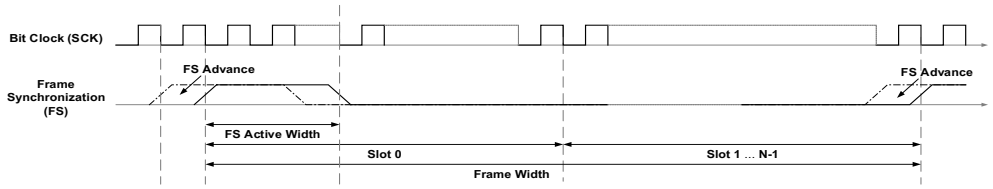
Frame synchronization advancement

Frame synchronization active edge could be aligned with the start of the first bit of the first slot, or one bit clock (SCK) cycle in advance, depending on the control field FSOST in SAI_BxFCFG register. *Figure 52-3 FS active width* shows how the FS waveform is changed.

Frame synchronization active width

Frame synchronization active width as shown in the following diagram depends on the control field FSAWD in SAI_BxFCFG register, its actual width equals (FSAWD + 1) SCK clock cycles. The minimum is 1 SCK clock cycle, while the maximum is 128 SCK clock cycles, which is half the maximum frame width. When FSFUNC is set, FS determines not only frame start, but also channel identification, (FSAWD+1) has to set equals half the frame width, otherwise the audio sub-block's function is not guaranteed.

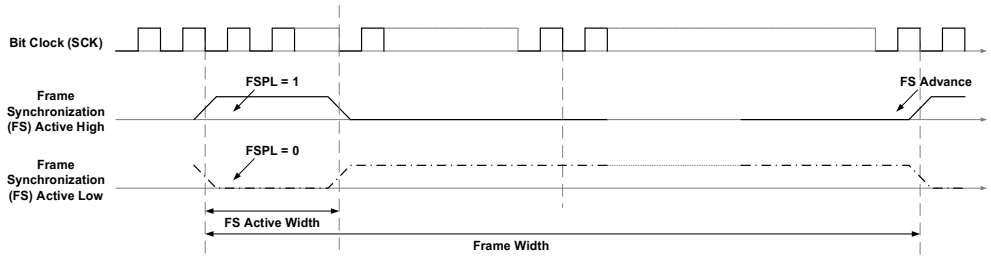
Figure 52-3 FS active width



Frame synchronization polarity

Frame synchronization active level could be configured through FSPL control field in SAI_BxFCFG register, as shown in the ***Figure 52-4 FS polarity***.

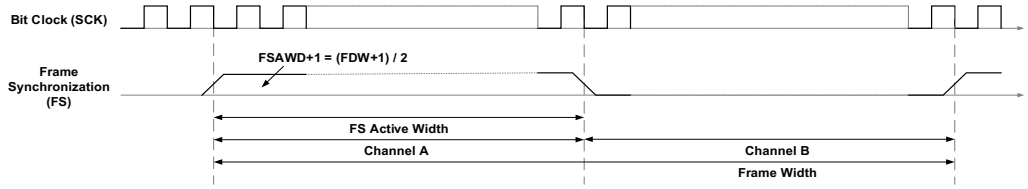
Figure 52-4 FS polarity



Frame synchronization function

Frame synchronization function definition is configured though the FSFUNC in SAI_BxFCFG register. Two specific functions could be selected, when FSFUNC is set to 1, FS not only represent frame start, but also channel number identification, in this case, frame active width (FSAWD + 1) should be configured to half of the frame width, as shown in ***Figure 52-5 FS function***, otherwise the audio sub-block behavior is not guaranteed. When FSFUNC is set to 0, FS only represents frame start.

Figure 52-5 FS function



Frame width

Frame width cannot be lower than 8-bit, which corresponding to one byte of data, and cannot exceed 256 bits.

In master mode, if BYPASS is cleared, frame length (FWD+1) should be set to a value equals the result of a base 2 exponential function within 8 to 256 to guarantee an integer number of MCLK cycles within one SCK cycle, this is must for correct external DAC/ADC operations. Otherwise error clock flag (ERRCK) is raised in SAI_BxSTAT register, and an interrupt is generated if error clock interrupt enable (ERRCKIE) is set in SAI_BxINTEN register. While if BYPASS is set, there will be no restriction on frame length configuration, master clock is automatically disabled.

In slave mode, frame length configuration is used to coordinate the internal finite state machine knowing the start and the end of an active frame. Another utility is used for advanced or postponed frame synchronization detection, an error flag is raised if this situation do occur and an interrupt is generated if the corresponding interrupt enable bit is set, refer to chapters

Error flags and *Interrupts* for more details.

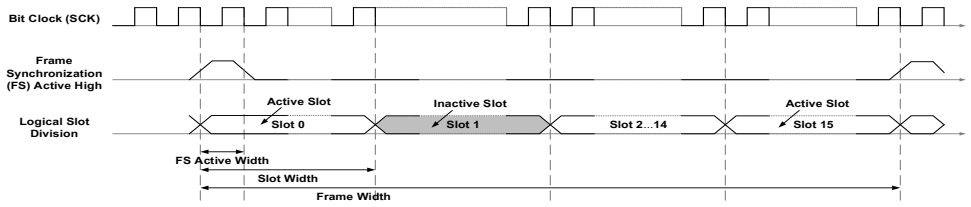
52.3.6. Slot configuration

Each SAI frame could be logically divided up to 16 slots, each slot's activation status and their distribution is further controlled through the configuration registers. Slot width can be configured to 16-, 32-bits or the same as data width through the SLOTWD control field in the SAI_BxSCFG register.

Slot activation

Each slot's activation status could be managed independently through slot activation vector (SLOTAV) in SAI_BxSCFG register. SLOTAV is a 16-bits wide control field, and each bit controls the corresponding slot's activation status. The logical division of slots could be shown in the *Figure 52-6 Slot activation*.

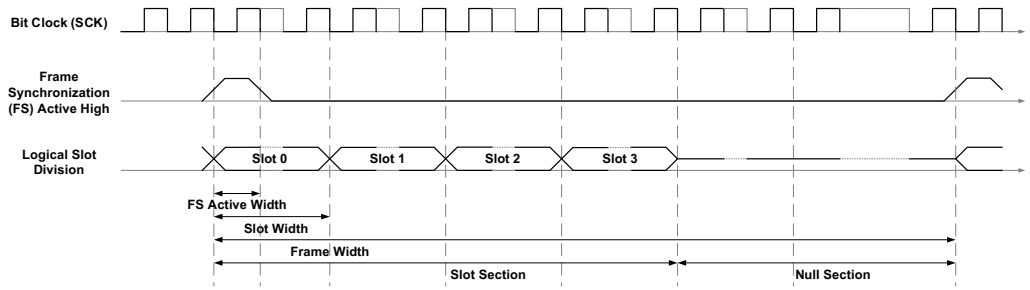
Figure 52-6 Slot activation



Slot distribution

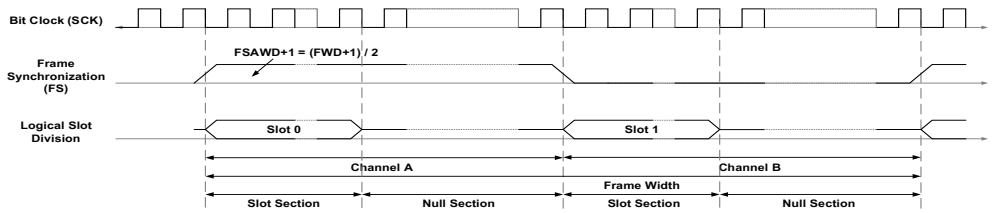
There can be not slot distribution under a specific circumstance that the product of slot number and slot width is less than frame width. Slot section is defined where there is slot distribution; otherwise, it is called a null section. When FSFUNC set to 0, where FS only signals frame start, the null section follows the last slot until the start of the next frame, as shown in the *Figure 52-7 Slot distribution when FSFUNC = 0.*

Figure 52-7 Slot distribution when FSFUNC = 0



When FSFUNC = 1, where FS not only signals frame start, but also channel number identification, slot section and null section are evenly distributed to both channels. Null section fills the space between the last slots of the current channel till the start of the next channel's slot.

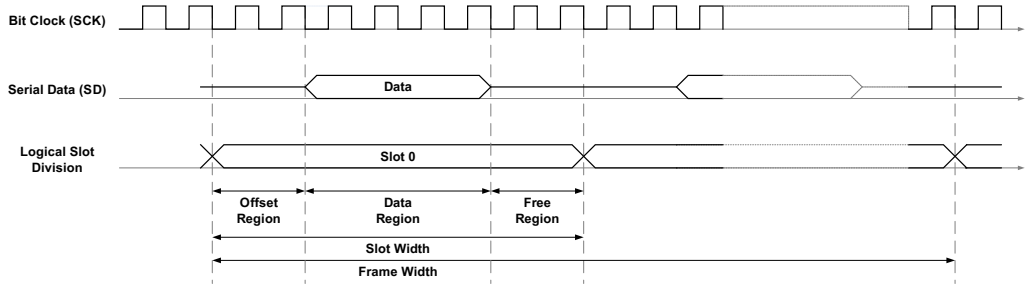
Figure 52-8 Slot distribution when FSFUNC = 1



Serial data output management on inactive slots

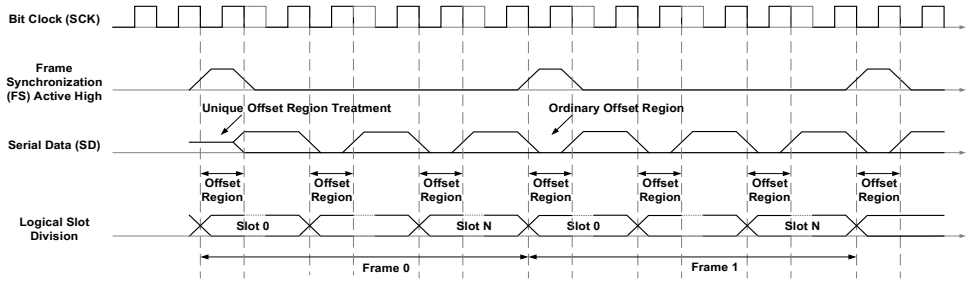
Serial data (SD) output behavior in the vicinity of inactive slots could be set according to the management policy defined in the serial data output mode (SDOM) bit in SAI_BxSCFG register, either SAI releases or drives a 0 value to the output. The SD output behavior in the offset and free regions needs special attention in the following three cases. Slot partition convention, namely offset, data and free region used in this manual is depicted in the ***Figure 52-9 Slot partition convention***.

Figure 52-9 Slot partition convention



Firstly, SD output during the offset region is decided by SDOM, if SDOM is set to 1, SAI will release the output, else a 0 is sent on the SD output, the diagram ***Figure 52-10 Offset region treatment*** shows the differences.

Figure 52-10 Offset region treatment

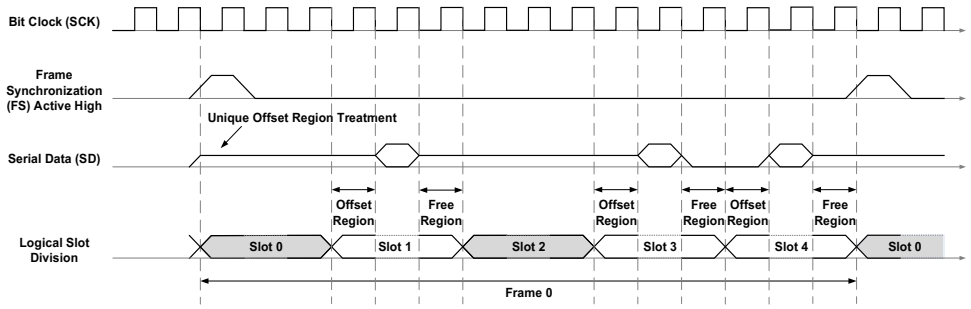


Secondly, SD output during the free region of the last slot has its reference on the activation status of the first slot, when SDOM is set, if slot 0 is inactive, SD output will be released, else a 0 is sent on the SD output. When SDOM is cleared, SD output is low independent of other slots activation status.

Lastly, SD output behavior during the offset and free region of slots in the middle of a frame has its reference on their next slot's activation status. If the following slot is inactive, and free

region is present, SD output will be release when SDOM set to 1, and drive to 0 if not. SD output behavior of offset and free region near active and inactive slot is shown in the ***Figure 52-11 SD output management***.

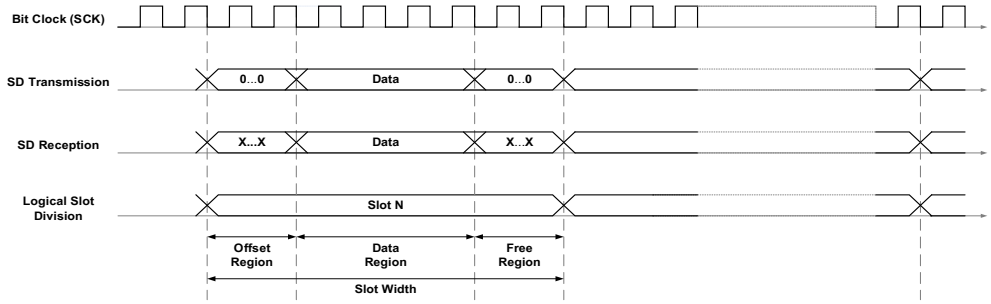
Figure 52-11 SD output management



52.3.7. Data configuration

Data width is also flexible, it can take on the value of 8-, 10-, 16-, 20-, 24-bits and 32-bits wide, which is configured through the data width (DATAWD) control field in the SAI_BxCFG0 register. Data inside an active slot could be shifted backward by setting the data offset (DATAOST), also located in the SAI_BxSCFG register. As described in the SD output management section (refer to ***Frame configuration***), the space between the start of a slot and the first bit of the data is called the offset region, and the space between the last bit of the data and the end of the slot is the free region. When the audio sub-block is configured as transmitter, and offset or free region is present, a 0 is sent through the SD output, the actual behavior of the SD line not only depends on the output value, but also on the line management condition and the slot activation status of nearby slots. When the audio sub-block is configure as receiver, and offset or free region is present, data reception during these regions will be ignored. Data transmission and reception are shown in ***Figure 52-12 Data configuration***.

Figure 52-12 Data configuration



52.3.8. Internal FIFO

An 8-word depth internal FIFO is implemented inside each SAI audio sub-block independently to increase transaction efficiency, these FIFO could be accessed by both CPU and DMA. FIFO request interrupt mechanism used to call for CPU and DMA access, request is raised depending on operating mode, FIFO threshold, FIFO status, and DMA burst size. FIFO

request interrupt generation is summarized in the *Table 52-3 FIFO request generation conditions* and cleared if the underlying conditions are not satisfied.

Table 52-3 FIFO request generation conditions

Transmitter: OPTMOD[0] = 0				Receiver: OPTMOD[0] = 1			
FIFO Threshold	FFTH	FIFO Status	FFSTAT	FIFO Threshold	FFTH	FIFO Status	FFSTAT
Empty	= 000	Empty	= 000	Empty	= 000	Not Empty	>= 001
1/4 Full	= 001	<1/4 Full	<010	1/4 Full	= 001	>= 1/4 Full	>= 010
1/2 Full	= 010	<1/2 Full	<011	1/2 Full	= 010	>= 1/2 Full	>= 011
3/4 Full	= 011	<3/4 Full	<100	3/4 Full	= 011	>= 3/4 Full	>= 100
Full	= 100	Not Full	<101	Full	= 100	Full	= 101

FIFO flush is done through the FLUSH control field in SAI_BxCFG1 register, when FLUSH is set, all the data content in the FIFO will be cleared, and read/write pointer reset to 0.

Note: DMA request generation is implicitly depended on FIFO request, detailed information will be provided in DMA interface section.

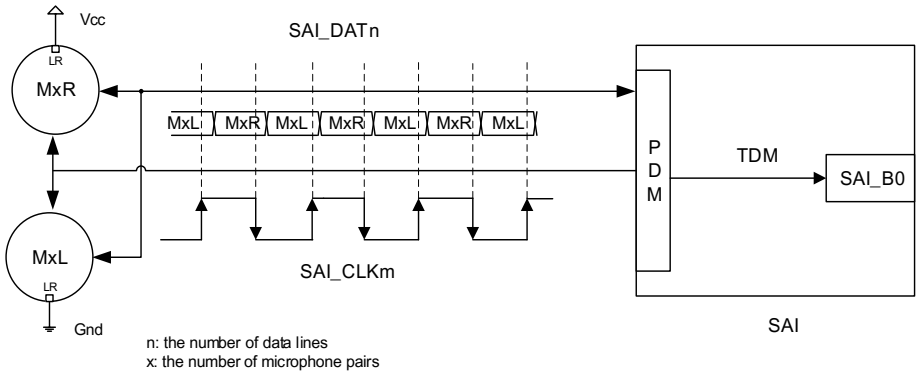
52.3.9. PDM interface

Digital microphones can realize data transmission through the PDM interface. The PDM interface supports up to 4 pairs of digital microphones in parallel, and can only be used in connection with SAI_B0 configured in TDM master mode.

Timing and connection

The schematic diagram of microphones connected by PDM interface is shown in the *Figure 52-13. PDM typical connection and timing*.

Figure 52-13. PDM typical connection and timing



Note: GD32H77x_78x PDM interface signals are only valid for DAT[2:0] and CLK[1:0].

The microphone LR pin connected to VCC is the left microphone channel, and the one connected to GND is the right microphone channel. The left channel and the right channel microphone share one clock line SAI_CLK, whose timing is generated by the TDM interface

of SAI_B0. The left channel data is sampled on the rising edge, and the right channel data is sampled on the falling edge to complete the two-channel data collection.

Note: The PDM interface can only be used in connection with SAI_B0 configured in TDM master mode.

PDM interface enable

The PDM interface enabling process is as follows:

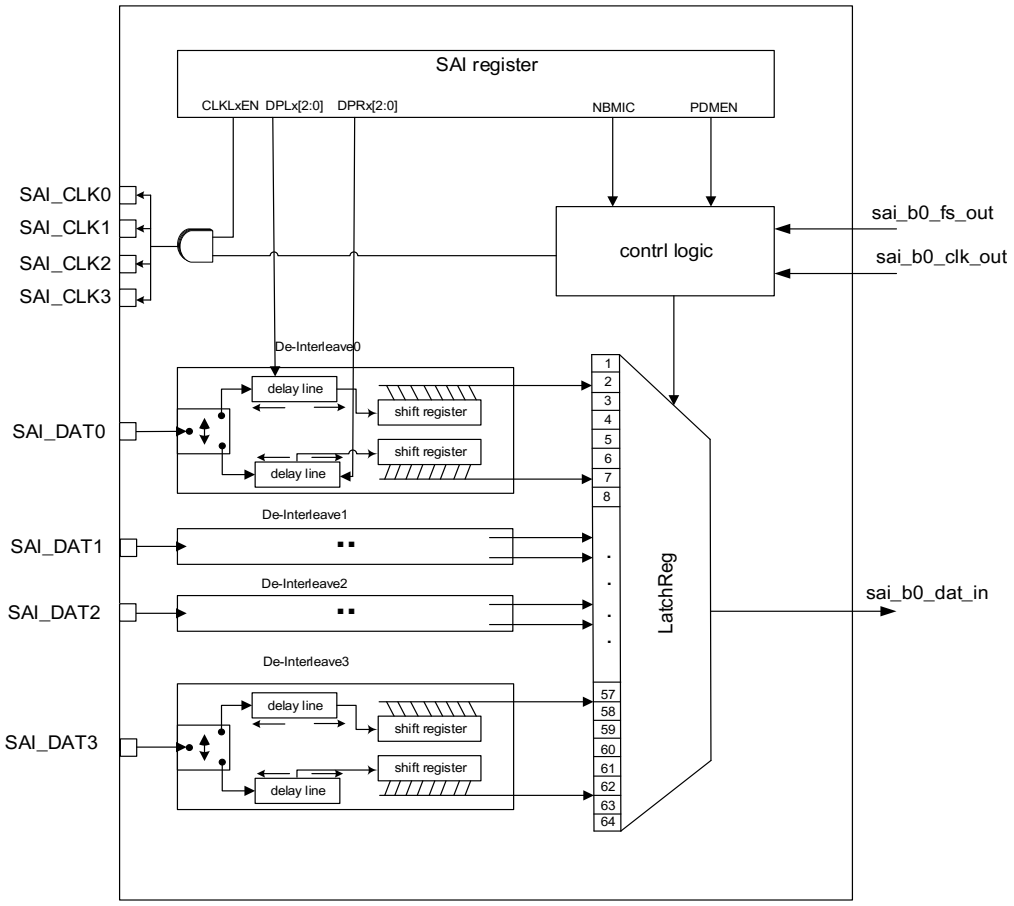
1. Configure TDM as master mode, configure OPTMOD[1:0] of SAI_B0CFG0 register as master receiver, and PROT[1:0] as polymorphic protocol.
2. Configure the PDM interface: select the number of microphones by configuring the MICNUMSEL[1:0] bits of the SAI_PDMCTL register, and configure the CLKL0EN/CLKL1EN bits in the SAI_PDMCTL register to enable the clock lines.
3. Configure the PDMEN bit in the SAI_PDMCTL register to enable the PDM interface.
4. Configure the SAIEN bit in the SAI_B0CFG0 register and enable SAI_B0.

Data processing

The data processing sequence in PDM is as follows:

1. SAI_B0 generates the clock to generate the bit stream clock via the TDM link to the PDM interface
2. By configuring the DPLx[2:0] and DPRx[2:0] bits of the SAI_PDMCFG register, the PDM interface is implemented to interleave and delay the bit stream data SAI_DATx generated by the microphone, thereby adjusting the delay generated by the microphone.
3. The shift register converts the data stream into bytes, and transmits the data to SAI_B0 through the TDM link.

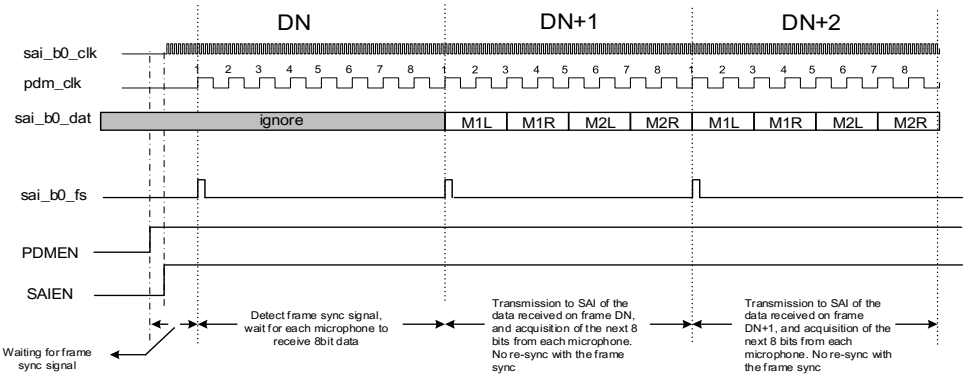
Figure 52-14. PDM data processing diagram



Data transfer start process

After the PDM interface is enabled, the microphone data sampling starts after the frame synchronization event occurs. After 8 SAI_CK clocks, the microphone data will be transmitted to the SAI through the TDM interface. The ***Figure 52-15. The start-up process of PDM data transmission*** shows the start-up process of PDM data transmission..

Figure 52-15. The start-up process of PDM data transmission



Data format

The data format from the microphone data is mainly related to the following configuration,

1. Configure MICNUMSEL[1:0] in the SAI_PDMCTL register to set the number of microphones.
2. Configure SLOTWD[1:0] in the SAI_B0SCFG register to set the slot width.
3. Configure the SHIFTDIR bit in the SAI_B0CFG0 register to set the MSB/LSB of the transmitted data.

Table 52-4 Get the microphone data read register times under different configurations

Number of microphones	Slot width	The number of times to read the BxDATA register to obtain the microphone data
8	32	2
	16	4
	8	8
4	32	1
	16	2
	8	4
2	16	1
	8	2

Note: GD32H77x_78x supports 3 pairs of microphones.

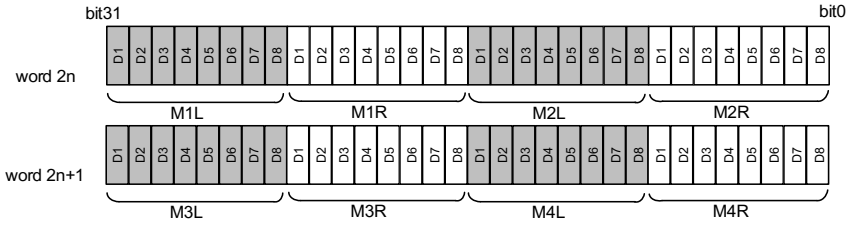
The following figure shows the schematic diagram of register data arrangement under different number of microphones and slot width.

Figure 52-16. Eight microphones, under different slot widths, BxDATA register data

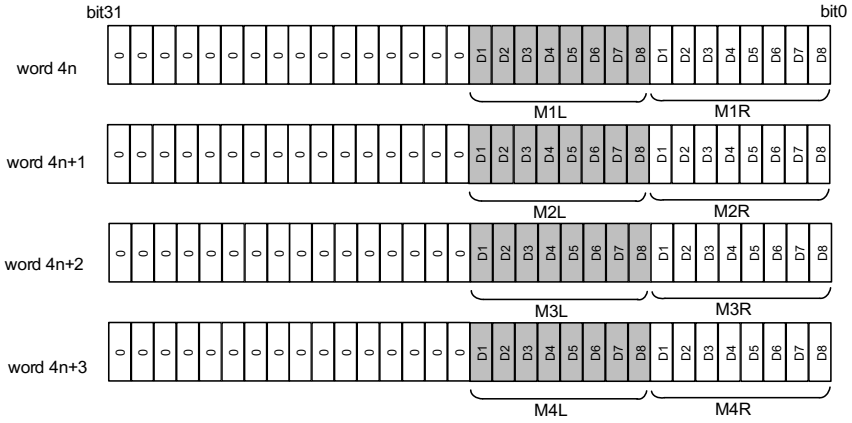
format

Eight microphone configuration

32-bit slot width SHIFTDIR = 0



16-bit slot width SHIFTDIR = 0



8-bit slot width SHIFTDIR = 0

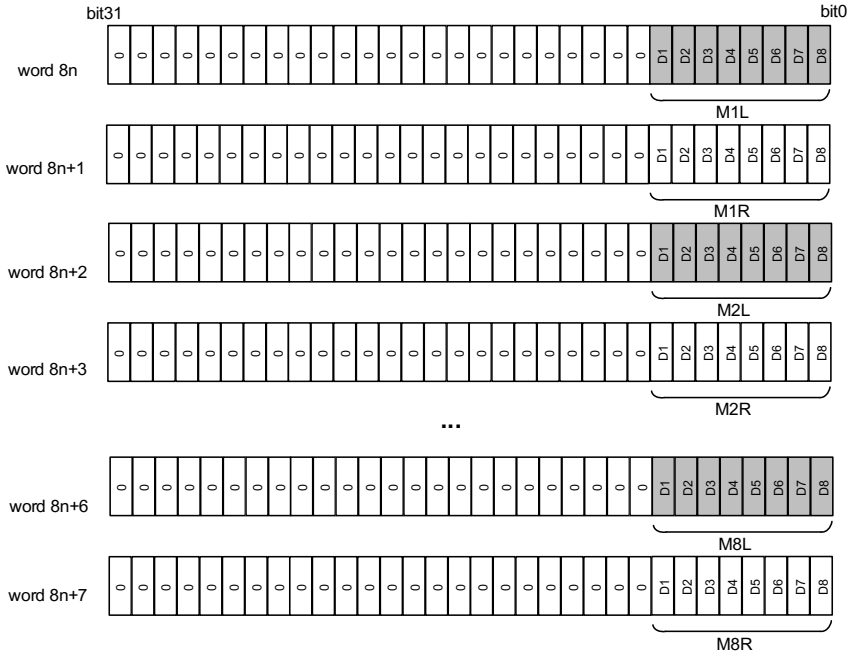


Figure 52-17. Four microphones, under different slot widths, BxDATA register data format

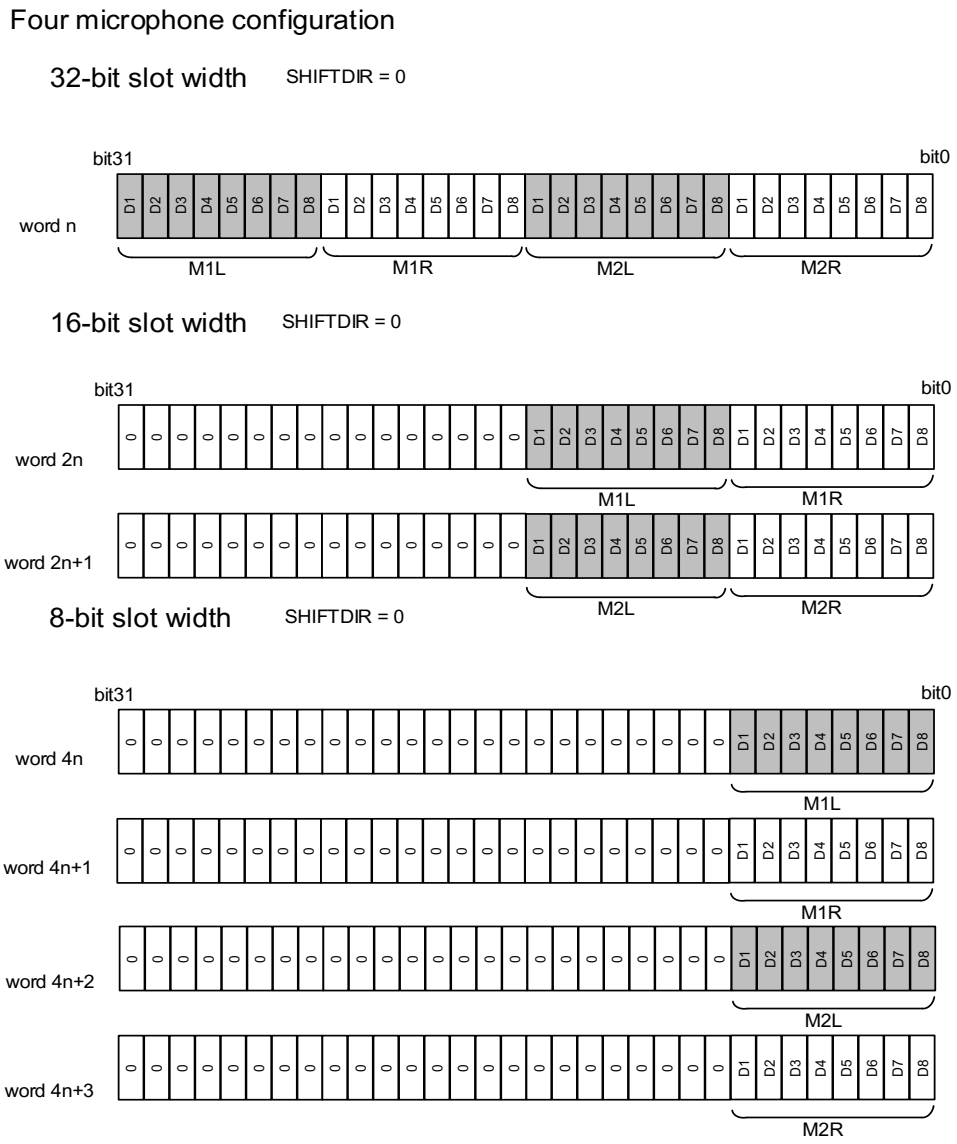
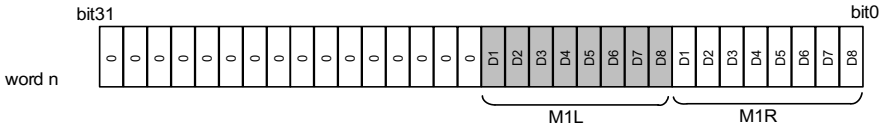


Figure 52-18. Dual microphones, under different slot widths, BxDATA register data

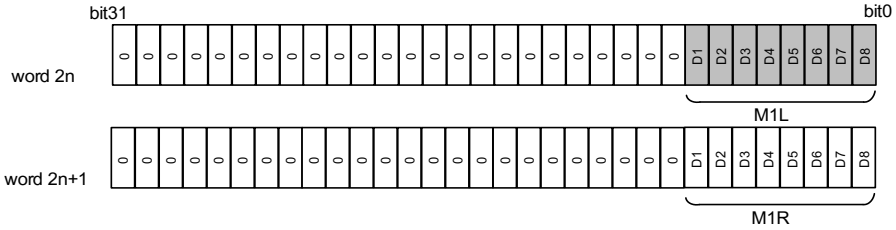
format

Dual microphone configuration

16-bit slot width SHIFTDIR = 0



8-bit slot width SHIFTDIR = 0



TDM configuration

For the TDM configuration items of the PDM interface, please refer to ***Table 52-5. TDM configuration***.

Table 52-5. TDM configuration

Registers	Bit fields	Values	Descriptions
SAI_B0CFG0	OPTMOD[1:0]	0b01	Mode must be MASTER receiver
	PROT[1:0]	0b00	Polymorphic for TDM
	DATAWD[2:0]	x	Data width.
	SHIFTDIR	x	Data is shifted with MSB or LSB first.
	SAMPEDGE	0	Data is sampled on SCK falling or rising edge.
	MONO	0	Stereo or Mono mode selection.
	BYPASS	1	Clock divider logic bypass.
	MDIV[5:0]	x	Master clock divider ratio.
SAI_B0FCFG	FWD[7:0]	x	Frame width
	FSAWD[6:0]	0	Frame synchronization active width.
	FSFUNC	0	FS only defines frame start.
	FSPL	1	FS high active polarity.
	FSOST	0	FS active edge asserted at the beginning of the first bit of the first slot.
SAI_B0SCFG	DATAOST[4:0]	0	Data offset
	SLOTWD[1:0]	0	Slot width equals' data width
	SLOTNUM[3:0]	x	Slot number within frame
	SLOTAV[15:0]	x	To be adjusted according to SLOTNUM

Note: When configuring PDM, the clock frequency, frame length, and slot size need to comply with the following three requirements:

1. The clock frequency configuration follows the following formula:

$$f_{SCK_B0} = f_{PDM_CLK} * (MICNUMSEL+1) * 2 \quad (52\text{-}4)$$

2. The frame width configuration follows the following formula:

$$FWD = (16 * (MICNUMSEL+1)) - 1 \quad (52\text{-}5)$$

10.1.18 Slot width needs to be configured as an integer multiple of (FWD+1).

52.3.10. AC'97 link controller

AC'97 link controller mode is selected via the PROT configuration in the SAI_BxCFG0 register. When this protocol is selected, many of the configuration field are ignored, include data shift direction, data width, most of frame and slot configurations, and part of interrupt control fields. The detail could be seen in the registers definition section.

AC'97 has a constant frame length of 256-bits, divided into 13-slots, the first slot is fixed at 16-bits width, and the rest 12 is fixed at 20-bits wide. User's must set the data width (DATAWD) control field in the SAI_BxCFG0 register to 16- or 20-bits wide, otherwise the audio sub-block's behavior is not guaranteed.

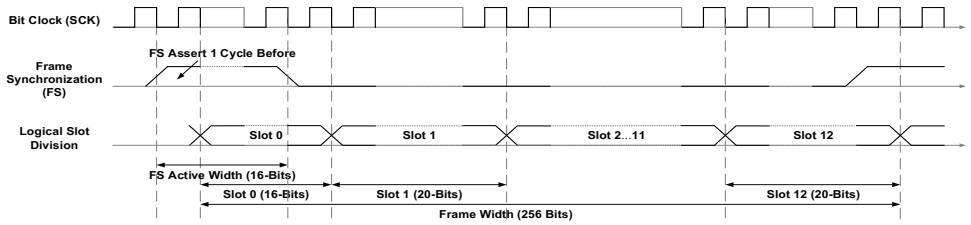
Bit2 of the TAG (slot 0) is reserved, no matter what value is written into the TAG, bit2 of the send data is 0.

Bit3 to 14 of the TAG (slot 0) act as the slot activation vector (SLOTAV) in polymorphic protocol, where the TAG slot (slot 0) is always active, bit3 correspond to slot 12, and bit14 corresponds to slot 1.

Bit15 of the TAG (slot 0) is codec ready status indication, when audio sub-block is configure as receiver, and a TAG received with Bit 15 low, indicate an audio codec not ready (ACNRDY) status, and ACNRDY flag is set accordingly. An interrupt is generated if both ACNRDY flag and audio codec not ready interrupt enable (ACNRDYIE) are both set.

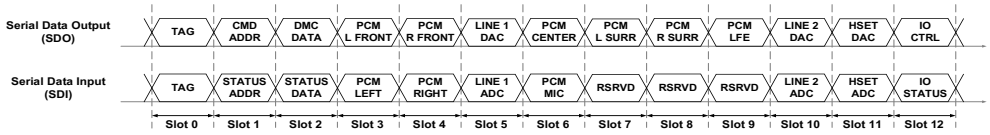
Frame synchronization active edge is asserted 1 clock cycle before the first bit of data, as shown in the ***Figure 52-19 AC'97 slot partition***.

Figure 52-19 AC'97 slot partition



The ***Figure 52-20 AC'97 tag definition*** give an overview of AC'97 slot partition.

Figure 52-20 AC'97 tag definition



The *Table 52-6 AC'97 transmitter tag definition* summarizes each slot's definition and meaning.

When AC'97 link controller acting as transmitter.

Table 52-6 AC'97 transmitter tag definition

Slot	Name	Description
0	SDO TAG	MSBs indicate which slot contain valid data; LSBs convey Codec ID
1	Control CMD ADDR write port	Read/write command bit plus 7-bit Codec register address
2	Control DATA write port	16-bit command register write data
3,4	PCM L&R DAC playback	16-, 18-, 20-bit PCM data for Left and Right channels
5	Modem Line 1 DAC	16-bit modem data for modem line 1 output
6,7,8,9	PCM Center, Surround L&R, LFE	16-, 18-, 20-bit PCM data for Center, Surround L&R, LFE channels
10	Modem Line 2 DAC	16-bit modem data for modem Line 2 output
11	Modem handset DAC	16-bit modem data for modem Handset output
12	Modem IO control	GPIO write port for modem control
10-11	SPDIF Out	Optional AC-link bandwidth for SPDIF output
6-12	Double rate audio	Optional AC-link bandwidth for 88.2 or 96kHz on L, C, R channels. Actual slots used are controlled by the DRSS bits.

When AC'97 link controller acting as receiver.

Table 52-7 AC'97 receiver tag definition

Slot	Name	Description
0	SDI TAG	MSBs indicate which slots contain valid data
1	STATUS ADDR read port	MSBs echo register address; LSBs indicate which slots request data
2	STATUS DATA read port	16-bit command register read data
3,4	PCM L&R ADC record	16-, 18- or 20-bit PCM data from Left and Right inputs
5	Modem Line 1 ADC	16-bit modem data from modem Line 1 input
6	Dedicated Microphone ADC	16-, 18- or 20-bit PCM data from optional 3rd ADC input
7,8,9	Vendor reserved	Vendor specific (enhanced input for docking, array mic, etc.)
10	Modem Line 2 ADC	16-bit modem data from modem Line 2 input

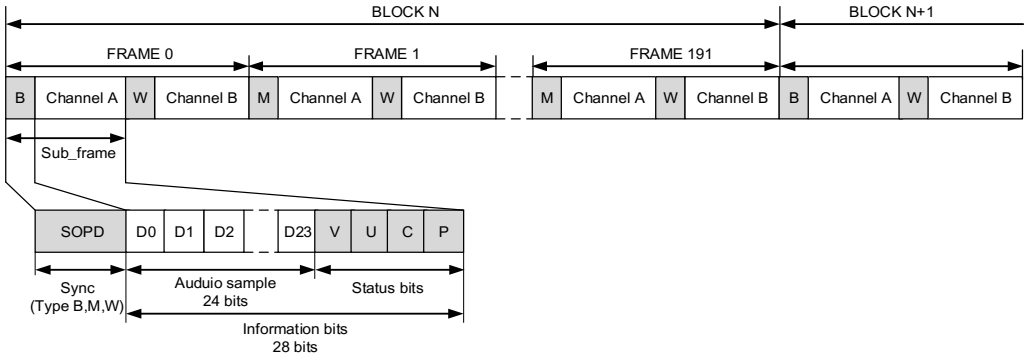
Slot	Name	Description
11	Modem handset input ADC	16-bit modem data from modem Handset input
12	Modem IO status	GPIO read port for modem status

52.3.11. SPDIF Output

SPDIF (Sony/Philips Digital Interface) is a type of digital audio interconnect used in consumer audio equipment to output audio over reasonably short distances. SPDIF is standardized in IEC 60958.

Figure 52-21 SPDIF data shows SPDIF block format and sub_frame format.

Figure 52-21 SPDIF data format



Each SPDIF block contains 192 frames of data, and each frame is composed of a left channel subframe (32 bits) and a right channel subframe (32 bits). Each subframe is composed of 4bit SOPD mode, 24bit data information and 4bit status information.

SOPD mode coding reference *Table 52-8. SOPD mode*

Table 52-8. SOPD mode

Preceding state	0	1	Description
Preamble	Channel coding		
B	11101000	00010111	Channel A data at the start of block
W	11100100	00011011	Channel B data at somewhere in the block
M	11100010	00011101	Channel A data

The data filling of SPDIF data transmission in the SAI_BxDATA register should follow: SAI_BxDATA[26:24] contains the channel status bit, user bit and validity bit, SAI_BxDATA[23:0] contains the 24-bit data of the channel under consideration.

Note: If the data size is 20/16 bits, the data should be mapped to SAI_BxDATA[23:4] / SAI_BxDATA [23:8].

SPDIF is selected by configuring PROT[1:0] = 01 in the SAI_BxCFG0 register, and the operating mode is then forced to master transmitter. At the same time, the DATAWD[2:0] data bit width setting in the SAI_BxCFG0 register will be ignored and forced to be set to 24 bits.

The symbol rate is configured through the clock generator. And encoded through the Manchester protocol.

The SAI first sends the adequate preamble for each sub-frame in a block. The SAI_BxDATA is then sent on the SD line (Manchester coded)21402140. The SAI ends the sub-frame by transferring the Parity bit calculated as described in ***Table 52-9 Parity bit calculation***.

Table 52-9 Parity bit calculation

SAI_BxDATA [26:0]	Parity bit P value transferred
Odd number of 0	0
Odd number of 1	1

For the SPDIF generator, the SAI shall provide a bit clock equal to twice of the symbol-rate.

More generally, the relationship between the audio sampling rate (FS) and the bit-clock rate (FSCK_x) is given by the formula:

$$F_s = \frac{F_{SCK_x}}{128} \quad (52\text{-}6)$$

The bit clock rate is obtained as follows:

$$F_{SCK_x} = \frac{F_{SAI_CK_x}}{MDIV} \quad (52\text{-}7)$$

Note: The above formulas are valid only if BYPASS is set to 1 in SAI_BxCFG0.

52.3.12. Stereo/Mono

The SAI audio sub-block could be switched between stereo mode and mono mode by the MONO configuration in SAI_BxCFG0 register, note that when mono mode is selected, slot number must be configured to 2, else the audio sub-block behavior is not guaranteed.

When the audio sub-block is configured as transmitter, data sent during the first slot (slot 0) will be duplicated to the second slot (slot 1), in this case, the FIFO is access half as much as in stereo mode.

When the audio sub-block is configured as receiver, data received during the first slot is pushed into the FIFO, and data received during the second slot is discarded.

52.3.13. Mute

Users could set the mute property anywhere during an on-going frame through the MT bit in the SAI_BxCFG1 register, but mute will only take effect on the state of the next frame.

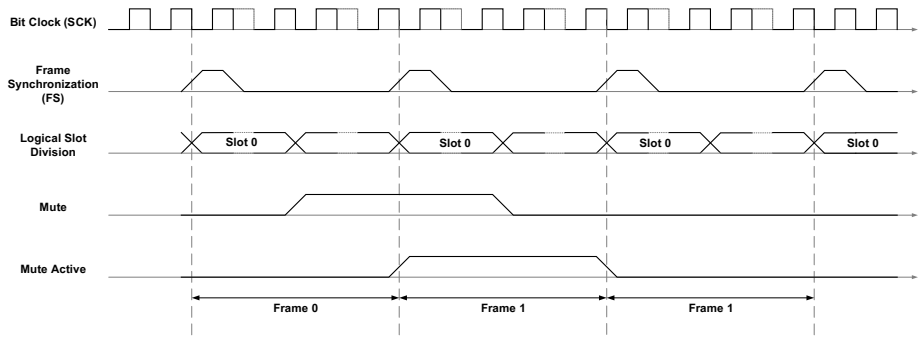
When the SAI audio sub-block is configured as transmitter and mute is configured, data is still read from the FIFO, put into the shift register as usual when mute take effect at next frame, the only difference is that the SD output if forced to a specific value determined by the mute value (MTVAL) configuration also located in the SAI_BxCFG1 register. When MTVAL is set

to 0, a 0 value is forced on the SD output during a mute frame. On the contrary, when MTVAL is set to 1, SD output behavior is further depended upon the slot number (SLOTNUM) configuration. When slot number is less or equals to 2, frame contents transmitted before mute active is repeated in mute frames. When slot number is greater than 2, SD output is forced to 0.

SAI audio sub-block that is configured in receiver mode could detect mute frame and generate interrupt accordingly. A mute frame counter is implemented in each audio sub block, a frame received with each active slot's data equals to 0 will be regarded as a mute frame, and the internal mute frame counter is incremented. This mute frame counter is reset when SAI audio sub-block is disabled or when a frame is not a mute frame. If the number of successive mute frame received amount to a number equals to the mute frame count (MTFCNT) settings in the SAI_BxCFG1 register, the mute detection flag (MTDET) will be set in the SAI_BxSTAT register, and an interrupt is generated if mute detection interrupt enable (MTDETIE) is turned on in the SAI_BxINTEN register.

Mute frame activation is shown in the *Figure 52-22 Mute frame activation*.

Figure 52-22 Mute frame activation



SD output behavior under different configuration is summarized in the *Table 52-10 Mute frame composition*.

Table 52-10 Mute frame composition

Slot Number	Mute Value = 1	Mute Value = 0
<= 2	Frame before Mute active is repeated on SD Output	Forced to 0
>2	Forced to 0	Forced to 0

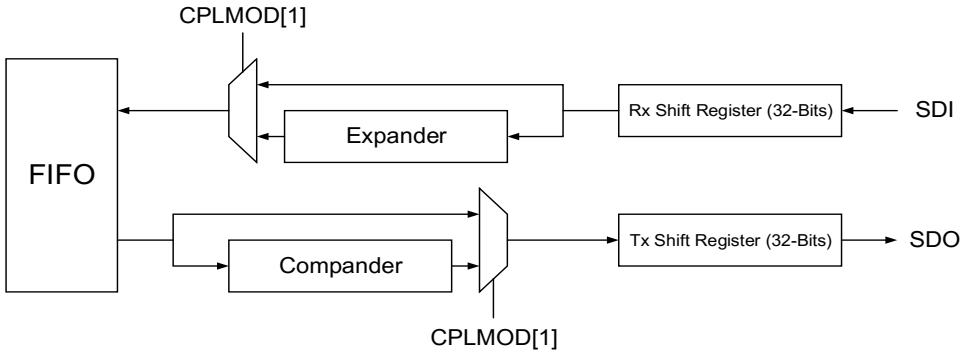
52.3.14. Compander

Compander is simply a system in which information is first compressed, transmitted through a bandwidth limited channel, and expanded in the receiving end. It is frequently used to reduce the bandwidth requirement for transmitting telephone quality speech, by reducing the 13-bit to 8-bit code words, which consist of 1 sign bit, 3-bit chord and 4-bit step. Two international standards for encoding signal data to 8-bit codes are A-law and Mu-law. A-law is the accepted European standard, while Mu-law is the accepted standard in the United

States and Japan.

Both A-law and Mu-law are implemented by the SAI, its configuration is located in the SAI_BxCFG1 register. The audio sub-block chooses between compressing and expanding according to the operating mode (OPTMOD). When the audio sub-block is configured as transmitter, compressing is selected. On the contrary, when configured as receiver, expanding mode is choose. Users can select between 1's or 2's complement as their default data representation as long as he has informed the SAI of his choice through the complement mode (CPLMOD) configuration in SAI_BxCFG1 register. In transmitter mode, no matter what compander mode is selected, the hardware first convert the complement representation to signed magnitude representation, and the feed into the compander. While in receiver mode, the linear output data is converted from sign-magnitude to complement representation, and then stored in the FIFO.

Figure 52-23 Compander data path



A-law compander

A-law is the CCITT recommended companding standard used across Europe, limiting the linear sample values to 12 magnitude bits. The following chart illustrates the A-law encoding algorithm, the linear input data is in the signed-magnitude representation, with S representing the sign, and the following 12-bit representing the magnitude. The encoded output is 8-bit wide, with MSB representing the sign. The sign bit S on both sides of the ***Table 52-11 A-law encoding*** takes on the same value. A, B, C, and D take on the value of 0 or 1, and X represent don't cares.

Table 52-11 A-law encoding

Linear Input Data													A-law Encoded Output							
S	0	0	0	0	0	0	0	A	B	C	D	X	S	0	0	0	A	B	C	D
S	0	0	0	0	0	0	1	A	B	C	D	X	S	0	0	1	A	B	C	D
S	0	0	0	0	0	1	A	B	C	D	X	X	S	0	1	0	A	B	C	D
S	0	0	0	0	1	A	B	C	D	X	X	X	S	0	1	1	A	B	C	D
S	0	0	0	1	A	B	C	D	X	X	X	X	S	1	0	0	A	B	C	D
S	0	0	1	A	B	C	D	X	X	X	X	X	S	1	0	1	A	B	C	D
S	0	1	A	B	C	D	X	X	X	X	X	X	S	1	1	0	A	B	C	D
S	1	A	B	C	D	X	X	X	X	X	X	X	S	1	1	1	A	B	C	D

After the input data is encoded through the logic defined in the table, an inversion pattern is applied to the 8-bit code to increase the density of transitions on the transmission line, a benefit to hardware performance. The inversion pattern is applied by XOR'ing the 8-bit code with 0x55.

Decoding the A-law encoded data is essentially a matter of reversing the steps in the encoding. The ***Table 52-12 A-law decoding*** illustrate the A-law decoding algorithm, applied after reversing the inversion pattern. The least significant bits discarded in the encoding process are approximated by the median value of the interval. This is shown in the output section by the trailing 1…0 pattern after the D bit.

Table 52-12 A-law decoding

A-law Encoded Input								Linear Output data												
S	0	0	0	A	B	C	D	S	0	0	0	0	0	0	0	A	B	C	D	1
S	0	0	1	A	B	C	D	S	0	0	0	0	0	0	1	A	B	C	D	1
S	0	1	0	A	B	C	D	S	0	0	0	0	0	1	A	B	C	D	1	0
S	0	1	1	A	B	C	D	S	0	0	0	0	1	A	B	C	D	1	0	0
S	1	0	0	A	B	C	D	S	0	0	0	1	A	B	C	D	1	0	0	0
S	1	0	1	A	B	C	D	S	0	0	1	A	B	C	D	1	0	0	0	0
S	1	1	0	A	B	C	D	S	0	1	A	B	C	D	0	0	0	0	0	0
S	1	1	1	A	B	C	D	S	1	A	B	C	D	1	0	0	0	0	0	0

Mu-Law compander

The United States and Japan use the Mu-law companding, limiting the linear sample value to 13 magnitude bits. The encoding and decoding process for Mu-law is similar to A-law. There are, however, a few notable differences:

1. Mu-law encoders typically operate on 13-bit magnitude data, as opposed to 12-bit magnitude data with A-law.
2. Before chord determination a bias value of 33 is added to the absolute value of the linear input data to simplify the chord and step calculations.
3. The definition of sign bit is reversed, that is, input and output sign bit takes on the opposite sign.

4. The inversion pattern is applied to all bits in the 8-bit code.

The ***Table 52-13 Mu-law encoding*** illustrate Mu-law encoding algorithm. The sign bit S of the linear input data takes on the opposite value from the sign bit of the encode data.

Table 52-13 Mu-law encoding

Linear Input Data														Mu-law Encoded output							
S	0	0	0	0	0	0	0	1	A	B	C	D	X	~S	0	0	0	A	B	C	D
S	0	0	0	0	0	0	1	A	B	C	D	X	X	~S	0	0	1	A	B	C	D
S	0	0	0	0	0	1	A	B	C	D	X	X	X	~S	0	1	0	A	B	C	D
S	0	0	0	0	1	A	B	C	D	X	X	X	X	~S	0	1	1	A	B	C	D
S	0	0	0	1	A	B	C	D	X	X	X	X	X	~S	1	0	0	A	B	C	D
S	0	0	1	A	B	C	D	X	X	X	X	X	X	~S	1	0	1	A	B	C	D
S	0	1	A	B	C	D	X	X	X	X	X	X	X	~S	1	1	0	A	B	C	D
S	1	A	B	C	D	X	X	X	X	X	X	X	X	~S	1	1	1	A	B	C	D

After the input data is encoded through the algorithm defined in the chart, an inversion pattern is applied to the 8-bit code to increase the density of the transmission line, a benefit to the hardware performance. The inversion pattern is applied by XOR'ing the 8 bit code with 0xFF.

Decoding the Mu-law is essentially a matter of reversing the steps in the encoding. The ***Table 52-14 Mu-law decoding*** illustrates the Mu-law decoding process, applied after reversing the inversion pattern. The least significant bits discarded in the encoding process are approximated by the median value of the interval. This is shown in the output section by the trailing 1…0 pattern after the D bit.

Table 52-14 Mu-law decoding

Mu-law Encoded Input								Linear Output Data												
S	0	0	0	A	B	C	D	~S	0	0	0	0	0	0	0	1	A	B	C	D	1
S	0	0	1	A	B	C	D	~S	0	0	0	0	0	0	1	A	B	C	D	1	0
S	0	1	0	A	B	C	D	~S	0	0	0	0	0	1	A	B	C	D	1	0	0
S	0	1	1	A	B	C	D	~S	0	0	0	0	1	A	B	C	D	1	0	0	0
S	1	0	0	A	B	C	D	~S	0	0	0	1	A	B	C	D	1	0	0	0	0
S	1	0	1	A	B	C	D	~S	0	0	1	A	B	C	D	1	0	0	0	0	0
S	1	1	0	A	B	C	D	~S	0	1	A	B	C	D	1	0	0	0	0	0	0
S	1	1	1	A	B	C	D	~S	1	A	B	C	D	0	0	0	0	0	0	0	0

52.3.15. Output drive

SAI can drive each audio sub-block's frame synchronization (FS), serial clock (SCK), and serial data (SD) independently of SAI enable status through the configuration of output drive (ODRIV) in SAI_BxCFG0 register.

Output drive settings must be programmed after SAI register configuration and before SAI is enabled.

52.3.16. IO management

IO management module is connected to both SAI audio sub-blocks, it is the only medium where they are connected. When audio sub-block is configured synchronous with the other sub-block through the synchronization mode (SYNCMOD) bit in SAI_BxCFG0 register, FS, SCK and MCLK pin could be shared, those pins of the synchronous sub-block are freed and left as general purpose Ios. When one audio sub-block is declared synchronous to the other audio sub-block, it must be configure as slave.

This function is especially useful in duplex mode, where one sub-block act as transmitter, while the other act as receiver, the synchronous sub-block receive its FS and SCK signal through the IO management module, which comes from the asynchronous module if it is declared as master or from external IO if it is declared as slave.

52.3.17. DMA interface

Each SAI audio sub-block has its own DMA interface. DMA access is enable through DMA enable (DMAEN) bit in the SAI_BxCFG0 register. DMA request is generated together with FIFO request (FFREQ), whose status depend on FIFO threshold (FFTH) and FIFO status (FFSTAT), this is especially important when DMA burst transaction is used. When the audio sub-block is configured in transmitter mode, FIFO threshold must be set to a value such that enough empty space is left for a complete DMA burst write operation in the worst case, otherwise FIFO overrun might occur. When the audio sub-block is configured in receiver mode, FIFO threshold must be set to a value such that enough data in the FIFO of a complete DMA burst read operation to avoid FIFO underrun condition.

DMA direction is linked to the audio sub-block operating configuration. When configured as transmitter, the DMA request a data load to the internal FIFO by writing the data register SAI_BxDATA. When configured as receiver, the DMA request a data read from the internal FIFO by reading the data register SAI_BxDATA.

Note: DMA SAI channel must be enabled after SAI register configuration.

52.3.18. Enable/Disable

SAI audio sub-block is enabled through the SAIEN bit in the SAI_BxCFG0 register, users must ensure this is done after the audio sub-block is configured, on-the-fly configuration is not supported, if done so, hardware's behavior is not guaranteed. Slave audio sub-block must be enabled before master audio sub-block.

Users can disable the audio sub-block anywhere during an active frame transfer, but it will only be completely disabled at the end of current frame.

52.3.19. Error flags

Clock error configuration detection

Clock error configuration detection mechanism in enabled only when the audio sub-block is configured as master and clock divider bypass (BYPASS) is clear. In this operating mode, users have to guarantee that the frame length (FWD+1) equals the results of an exponential function of base 2 within 8- and 256-bits range, otherwise the clock error flag (ERRCK) will be set in the status register SAI_BxSTAT. The frame length must be a power of 2 is to ensure there is an integer number of master clock (MCLK) cycles within each bit clock cycle (SCK), for better sound quality.

An interrupt is generated if the clock error configuration detection interrupt enable (ERRCKIE) bit in the interrupt enable register SAI_BxINTEN is set.

When Clock error is detected, it automatically disables the audio sub-block via clearing the SAIEN in the SAI_BxCFG0 register.

Audio codec not ready detection

Audio codec not ready status is checked only when AC'97 protocol is used and receiver operating condition is choose. The audio sub-block determines audio codec ready status from the TAG (the first slot). When bit 15 of the TAG is 0, audio codec not ready flag (ACNRDY) in the status register SAI_BxSTAT is set, an interrupt is generated if audio codec not ready interrupt enable (ACNRDYIE) in the interrupt enable register SAI_BxINTEN is set. When codec not ready is detected, data contained in the following slots of the current frame will not be loaded into the FIFO.

Audio codec not ready detected flag is clear by setting the audio codec not ready detected clear (ACNRDYC) bit in the SAI_BxINTC register.

Frame synchronization advanced detection

Frame synchronization advanced detection mechanism is enabled only when the audio sub-block is configured as a slave, since slave receives the FS signal, and its arrival time is crucial to correct data interpretation. Frame synchronization advanced detection is possible because frame length, frame active polarity and frame offset are determined before audio sub-block is enabled.

Frame synchronization advancement has no effect on the current frame since FS active edge is only anticipated at the end of the frame.

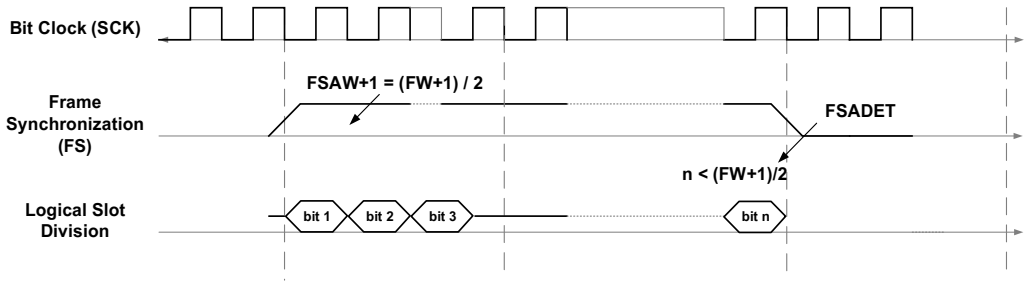
An interrupt is generated when both frame synchronization advanced detected flag (FSADET) in status register SAI_BxSTAT and frame synchronization advanced detection interrupt enable (FSADETIE) in interrupt enable register SAI_BxINTEN are set.

To resynchronize, the following steps are needed:

1. Audio sub-block should be disabled, users must wait for the SAIEN control field of the corresponding sub-block is completely disabled.
2. The internal FIFO should be flushed by setting the FLUSH control field.
3. Enable the audio sub-block again by setting SAIEN.
4. Wait for FS to restart synchronization.

Note: This flag is not generated in AC'97 or SPDIF configuration mode. Since AC'97 is only a link controller, FS is generated even if the audio sub-block is configured as a slave.

Figure 52-24 Frame synchronization advanced detection



Frame synchronization postponed detection

Frame synchronization postponed detected mechanism is enabled only when the audio sub-block is configured as a slave, since the slave receives the FS signal, and its arrival time is crucial to correct data interpretation. Frame synchronization postpone detection is possible because frame length, frame active polarity and frame offset are determined before audio sub-block is enabled.

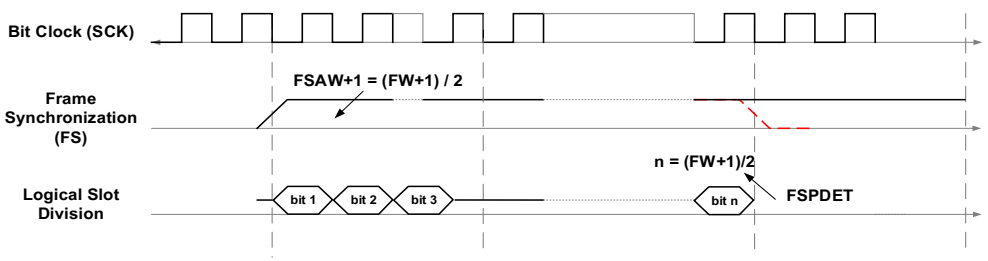
Frame synchronization postpone might be caused by late generation of the master device, external delay or noise induced glitches. This incorrect FS timing could disrupt audio sub-block's internal finite state machine, thus corrupting correct transactions.

An interrupt is generated when both frame synchronization postpone detected flag (FSPDET) in status register SAI_BxSTAT and frame synchronization postpone detection interrupt enable (FSPDETIE) in interrupt enable register SAI_BxINTEN are set.

To resynchronize with the master, the steps for resynchronization should be applied.

Note: This flag is not generated in AC'97 or SPDIF configuration mode. Since AC'97 is only a link controller, FS is generated even if the audio sub-block is configured as a slave.

Figure 52-25 Frame synchronization postponed detection



FIFO overrun or underrun detection

FIFO overrun and underrun flag (OUERR) occupies the same bit in status register SAI_BxSTAT since each audio sub-block can be configured into either transmitter or receiver.

When the audio sub-block is configures as transmitter, underrun can happen during an active frame transaction when FIFO is empty and a new slot of empty data is sent. An interrupt is generated if the overrun or underrun interrupt enable (OUERRIE) in the interrupt enable register SAI_BxINTEN is set. If underrun happens, a re-synchronization procedure is showed in the following steps:

1. Audio sub-block should be disabled, users must wait until SAIEN control field of the corresponding sub-block is completely disabled.
2. The internal FIFO should be flushed by setting the FLUSH control field.
3. Fill the correct data to be transferred in into the FIFO.
4. Enable the audio sub-block again by setting SAIEN.

Underrun flag is cleared by setting the overrun or underrun clear (OUERRC) bit in the interrupt clear register SAI_BxINTC.

When the audio sub-block is configured as receiver, overrun can happen during an active frame transaction when FIFO is full and a new slot of data is received. When overrun happens, the newly received data is discarded, nothing will be written into the FIFO. An interrupt is generated if the overrun or underrun interrupt enable (OUERRIE) in the interrupt enable register SAI_BxINTEN is set.

Overrun flag is also cleared by setting the OUERRC bit in the SAI_BxINTC register.

Note: When DMA is enabled, users have to guarantee correct DMA configuration, especially if DMA burst is used, otherwise overrun and underrun can both happen in transmitter or receiver operating mode.

52.3.20. Interrupts

The *Table 52-15 Interrupt control* summarizes all the interrupt sources present in each audio sub-block.

Table 52-15 Interrupt control

Interrupt source	Interrupt partition	Interrupt raise condition	Interrupt Enable Control	Interrupt Clear Control
FFREQ	Request	Any	FFREQIE	Read/Write SAI_BxDATA
MTDET	Mute	Receiver	MTDETIE	MTDETC
ERRCK	Error	Master; BYPASS = 0	ERRCKIE	ERRCKC
ACNRDY	Error	Receiver; PROT = AC'97	ACNRDYIE	ACNRDYC
FSADET	Error	Slave; PROT ! =AC'97	FSADETIE	FSADETC
FSPDET	Error	Slave; PROT ! =AC'97	FSPDETIE	FSPDETC

OUERR	Error	Any	OUERRIE	OUERRC

To recover from error interrupt, the following procedure should be applied:

1. Disable the corresponding interrupt.
2. Configure SAI function registers.
3. Enable interrupt.
4. Enable SAI audio sub-block.

52.4. Register definition

SAI0 base address: 0x4001 5800

SAI1 base address: 0x4001 5C00

SAI2 base address: 0x4001 6000

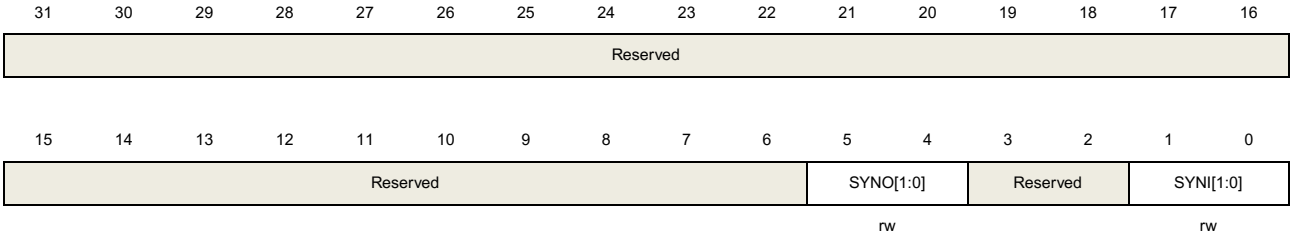
SAI3 base address: 0x4001 6800

52.4.1. Synchronize configuration register (SAI_SYNCFG)

Address offset: 0x00

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:6	Reserved	Must be kept at reset value.
5:4	SYNO[1:0]	Synchronization outputs 00: No synchronization output signals. 01: Block 0 used for further synchronization for others SAI 10: Block 1 used for further synchronization for others SAI 11: Reserved. These bits must be set when both audio sub-blocks (0 and 1) are disabled. **Note:** SYNO[1:0] should be configured as No synchronization output signals when audio block is configured as SPDIF.
3:2	Reserved	Must be kept at reset value.
1:0	SYNI[1:0]	Synchronization inputs. Refer to ***Table 52-2. External synchronization configuration***. These bits must be set when both audio sub-blocks (0 and 1) are disabled. They are meaningful if one of the two audio blocks is defined to operate in synchronous mode with an external SAI (SYNCMOD[1:0] = 10 in SAI_BxCFG0).

52.4.2. Block x configuration register0 (SAI_BxCFG0) (x = 0,1)

Address offset: 0x04 + 0x20 * x

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				MCLKEN	MOSPR	MDIV[5:0]						BYPASS	Reserved	DMAEN	SAIEN
				rw	rw	rw						rw		rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		ODRIV	MONO	SYNCMOD[1:0]		SAMPEDGE	SHIFTDIR	DATAWD[2:0]			Reserved	PROT[1:0]		OPTMOD[1:0]	
		rw	rw	rw		rw	rw	rw				rw		rw	

Bits	Fields	Descriptions
31:28	Reserved	Must be kept at reset value.
27	MCLKEN	The master clock enable 0: The master clock is disabled 1: The master clock is enabled independently of SAIEN bit
26	MOSPR	The master clock oversampling rate. 0: MCLK = 256 * F_{fs} 1: MCLK = 512 *F_{fs}
25:20	MDIV[5:0]	Master clock divider ratio. 000000: Primary frequency divider logic bypass, Otherwise, please refer to formula (521) in chapter Clock divider for output frequency calculationClock divider **Note**: This control field is ineffective when SAI configured in slave mode. **Note**: This control field has to be set before SAI is enabled.
19	BYPASS	Clock divider logic bypass. 0: Clock divider ratio is applied to both primary and secondary divider logic. 1: Clock divider logic is bypassed.
18	Reserved	Must be kept at reset value.
17	DMAEN	DMA enable. 0: DMA disabled 1: DMA enabled **Note**: DMAEN must be set after OPTMOD control field if SAI is configured as receiver to avoid unnecessary DMA request, since SAI is transmitter after reset.
16	SAIEN	SAI sub-block enable. 0: SAI sub-block is disabled. 1: SAI sub-block is enabled, this control field can only be set when SAIEN is 0. **Note**: The SAI sub-block is completely disabled (SAIEN read as 0) only at the end of the current frame if software has already issued a disabled request somewhere during the on-going transfer. **Note**: SAI slave must be enabled before SAI master.

15:14	Reserved	Must be kept at reset value.
13	ODRIV	Output Drive. 0: SAI sub-block output driven only when SAIEN is set. 1: SAI sub-block output driven according to ODRIV setting. **Note**: This control field has to be set after SAI configuration but before SAI sub-block enabled.
12	MONO	Stereo and Mono mode selection. 0: Stereo mode. 1: Mono mode. Mono mode requires slot number equals to 2, in transmitter mode, the first slots data is copied to the second slot, while in receiver mode, the second slot's data is ignored.
11:10	SYNCMOD[1:0]	Synchronization mode. 00: Asynchronous with the other sub-block. 01: Synchronous with the other sub-block, when this mode is selected, users must configure the operating mode to slave. 10: Synchronous with an external SAI audio sub-block, when this mode is selected, users must configure the operating mode to slave. 11: Reserved. **Note**: It is required to be configured when audio sub-block is disabled. **Note**: If the protocol select as SPDIF, the mode should be configured as asynchronous.
9	SAMPEDGE	Sampling clock edge. 0: Data sampled on SCK falling edge. 1: Data sampled on SCK rising edge. **Note**: This control field is ignored in SPDIF mode. **Note**: It is required to be configured when audio sub-block is disabled.
8	SHIFTDIR	Shift direction. 0: Data is shifted with MSB first. 1: Data is shifted with LSB first. **Note**: This control field is ignored in AC'97 mode, since shift direction is forced to MSB first. **Note: This control field is ignored in SPDIF mode, since data is forced to LSB first.**
7:5	DATAWD[2:0]	Data width. 000: Reserved. 001: Reserved. 010: 8-bits wide 011: 10-bits wide. 100: 16-bits wide. 101: 20-bits wide.

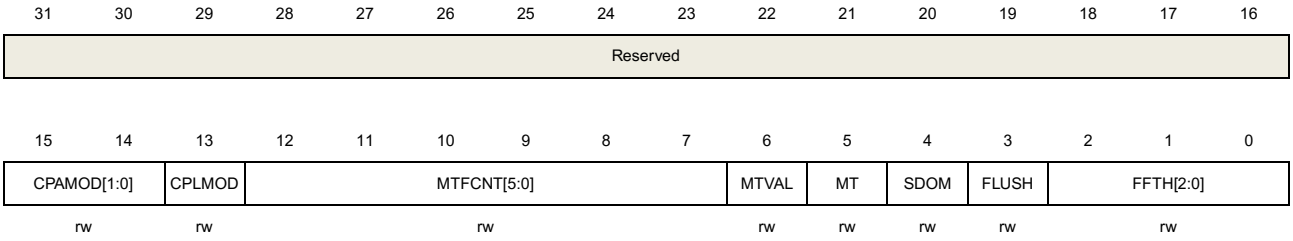
		110: 24-bits wide. 111: 32-bits wide. In compander mode, data width is fixed to 8-bits width by the algorithm itself. **Note**: It is required to be configured when audio sub-block is disabled. **Note**: This control field is ignored in SPDIF mode. **Note**: In case AC'97 protocol is selected, only 16- or 20-bits is viable, otherwise audio sub-block's behavior is not guaranteed.
4	Reserved	Must be kept at reset value.
3:2	PROT[1:0]	Protocol selection. 00: Polymorphic. 01: SPDIF. 10: AC'97. 11: Reserved. Polymorphic configuration allows the user to tweak with all the frame and slot configuration options to form his protocol of choice such as I2S, LSB/MSB justified, TDM, PCM/DSP and so on. **Note**: It is required to be configured when audio sub-block is disabled.
1:0	OPTMOD[1:0]	Operating mode. 00: Master transmitter. 01: Master receiver. 10: Slave transmitter. 11: Slave receiver. If the protocol select as SPDIF, the operating mode will be forced to configure as master transmitter. **Note**: It is required to be configured when audio sub-block is disabled.

52.4.3. Block x configuration register1 (SAI_BxCFG1) (x = 0,1)

Address offset: 0x08 + 0x20 * 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.
15:14	CPAMOD[1:0]	Compander mode.

		00: No compansion applied. 01: Reserved. 10: Mu-law algorithm. 11: A-law algorithm. ITU-T G.711 defines two main compansion algorithm, the Mu-law and A-law, which encode 13- and 12-bits signed linear PCM respectively to logarithmic 8-bit samples. The former gives more resolution to higher range signals while the later provides more quantization levels at lower signal levels. The compression or expansion mode is selected via control field OPTMOD[0], when the audio sub-block is configured as transmitter, compression is automatically applied, while expansion is performed when configured as receiver. The complement mode is selected via control field CPLMOD. **Note**: Compansion should be applied when TDM protocol is configured.
13	CPLMOD	Complement mode. This control field is used to select the complement option in compansion mode. 0: Data represented in 1's complement form. 1: Data represented in 2's complement form.
12:7	MTFCNT[5:0]	Mute frame count. This control field is only used in reception mode. When the number of consecutive mute frame received equals to MTFCNT, MTDET flag is set and an interrupt is generated if MTDETIE is set.
6	MTVAL	Mute value. 0: 0 is sent via the Serial Data (SD) line when mute is on. 1: If SLOTNUM is less or equals to two, last frame is sent via the Serial Data (SD) line when mute is on, Otherwise 0 is sent via the Serial Data (SD) line in mute frames. **Note**: This control field is meaningful only when the audio sub-block is configured as transmitter, and is ignored in SPDIF mode. **Note**: The receiver can only detect 0 valued mute frame, when MTVAL set to 1 and the last frame is repeatedly sent during the mute, the receiver will receive the data as it is, the mute frame counter will not amount, and so no MTDET flag will ever be set.
5	MT	Mute. 0: Mute mode off. 1: Mute mode on. Note: This control field is meaningful only when the audio sub-block is configured as transmitter, SD output is determined by MTVAL when Mute is on. Note: Mute will take effect at next frame if mute is set somewhere inside the on-going frame.
4	SDOM	Serial data output management. 0: SD output is driven entirely during the audio frame.

		1: SD output is released near inactive slots. **Note**: If the data offset of the first slot of the first frame is not equals to zero, SD will remain released until the first active data bit. If the current frame is not the first frame of a continuous transfer, whether the offset section of the first slot is released or not depends on the last slot of the previous frame. If the last slot is active, then the offset section is driven, otherwise it is released. **Note**: If data offset plus data width is still lower than the slot width, than the space between the last bit of data and the end of slot is known as the free region. SD output could be released during the free region depends on whether the following slot is active, if the current slot is the last slot, then its free region behavior depends on the first slot of current frame, it is independent on whether the current frame is the last frame. **Note**: Whether SD outputs is driven or not on the free region precede a slot and the offset section followed by a slot is determined by that slot only, if that slot is active, the SD line will be driven, otherwise it will be released.
3	FLUSH	FIFO flush. 0: No FIFO flush. 1: Perform FIFO flush. **Note**: FIFO flush clears all data content inside the FIFO and reset read/write pointers. FIFO flush should be performed when SAI is disabled.
2:0	FFTH[2:0]	FIFO threshold. 000: FIFO empty. 001: FIFO quarter full. 010: FIFO half full. 011: FIFO three quarter full. 100: FIFO full. 101: Reserved. 110: Reserved. 111: Reserved. **Note**: FIFO threshold is used in conjunction with FIFO status (FFSTAT) control field to determine FIFO request (FFREQ) is CPU and DMA mode.

52.4.4. Block x frame configuration register (SAI_BxFCFG) (x = 0,1)

Address offset: 0x0C + 0x20 * x

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													FSOST	FSPL	FSFUNC
													rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	FSAWD[6:0]							FWD[7:0]							

rw rw

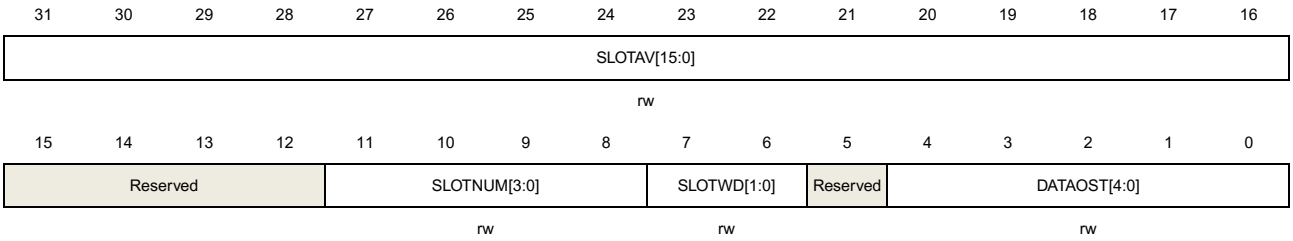
Bits	Fields	Descriptions
31:19	Reserved	Must be kept at reset value.
18	FSOST	Frame synchronization offset. 0: FS active edge asserted at the beginning of the first bit of the first slot. 1: FS active edge asserted one bit cycle before normal FS when FSOST is 0. **Note**: This control field must be configured before the audio sub-block is enabled, and it is meaningless when protocol is set to AC'97 or SPDIF.
17	FSPL	Frame synchronization active polarity. 0: FS low active polarity 1: FS high active polarity. **Note**: This control field must be configured before the audio sub-block is enabled, and it is meaningless when protocol is set to AC'97 or SPDIF.
16	FSFUNC	Frame synchronization function. 0: FS only defines frame start. 1: FS define both frame start and channel number. **Note**: This control field must be configured before the audio sub-block is enabled. **Note**: When FSFUNC is set, slot number (SLOTNUM + 1) within a frame must be even, in this case half of the slot will be allocated to channel A, the other half to channel B. If the combination of all slots allocated to one channel is smaller than half the frame width, SD output will be release where slot is undefined, independent of SDOM. **Note**: When FSFUNC is set, FS active width (FSAWD + 1) must be configure as half the length of a frame.
15	Reserved	Must be kept at reset value.
14:8	FSAWD[6:0]	Frame synchronization active width. **Note**: This control field must be configured before the audio sub-block is enabled, and it is meaningless in AC'97 mode. **Note**: This control field specified the active width of FS in terms of (FSAWD + 1) SCK clock cycles.
7:0	FWD[7:0]	Frame width **Note**: This control field must be configured before the audio sub-block is enabled, and it is meaningless in AC'97 mode. **Note**: This control field specified the frame width in terms of (FWD + 1) SCK clock cycles, When the audio sub-block is configured in master mode, and BYPASS = 0, (FWD+1) must equals the results of an exponential function of base 2 within 8- and 256-bits range.

52.4.5. Block x slot configuration register (SAI_BxSCFG) (x = 0,1)

Address offset: 0x10 + 0x20 * x

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:16	SLOTAV[15:0]	Slot activation vector. 0: Slot inactive. 1: Slot active. Each bit in the SLOTAV vector is aligned to the slot number from 0 to 15, if SLOTNUM is less than 15, the unaligned vector bits is ignored. **Note**: This control field must be configured before the audio sub-block is enabled. **Note**: This control field is meaningless in AC'97 or SPDIF mode.
15:12	Reserved	Must be kept at reset value.
11:8	SLOTNUM[3:0]	Slot number within frame. The actual slot number with in a frame is (SLOTNUM + 1), and cannot exceed 16. When FSFUNC is set, slot number must be even for equals' allocation to both channels. **Note**: This control field must be configured before the audio sub-block is enabled. **Note**: This control field is meaningless in AC'97 mode.
7:6	SLOTWD[1:0]	Slot width. 00: Slot width equals' data width. 01: Slot width of 16-bits. 10: Slot width of 32-bits. 11: Reserved. Slot width must be greater or equal to data width to hold the data, Otherwise the behavior is not guaranteed. **Note**: This control field must be configured before the audio sub-block is enabled. **Note**: This control field is meaningless in AC'97 or SPDIF mode.
5	Reserved	Must be kept at reset value.
4:0	DATAOST[4:0]	Data offset. Defines when the first bit of data appear within an active slot, in transmission mode, the SD output in the offset and free region depends on SDOM and nearby slot

activation status, 0 or Hi-Z will be outputted. While in reception mode, data content on the offset and free region will be ignored.

Note: This control field must be configured before the audio sub-block is enabled.

Note: This control field is meaningless in AC'97 mode.

52.4.6. Block x interrupt enable register (SAI_BxINTEN) (x = 0,1)

Address offset: 0x14 + 0x20 * 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									FSPDETIE	FSADETIE	ACNRDYIE	FFREQIE	ERRCKIE	MTDETIE	OUERRIE
									rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value.
6	FSPDETIE	Frame synchronization postpone detection interrupt enable. 0: Interrupt disabled. 1: Interrupt enabled. An interrupt is generated if FSPDET and FSPDETIE are both set. **Note**: This control field is meaningless when the audio sub-block is configured as master. **Note**: This control field is meaningless in AC'97 or SPDIF mode.
5	FSADETIE	Frame synchronization advanced detection interrupt enable. 0: Interrupt disabled. 1: Interrupt enabled. An interrupt is generated if FSADET and FSADETIE are both set. **Note**: This control field is meaningless when the audio sub-block is configured as master. **Note**: This control field is meaningless in AC'97 or SPDIF mode.
4	ACNRDYIE	Audio codec not ready interrupt enable. 0: Interrupt disabled. 1: Interrupt enabled. An interrupt is generated if ACNRDY and ACNRDYIE are both set. **Note**: This control field has a meaning only when the audio sub-block is configured as receiver. **Note**: This control field has a meaning only when AC'97 mode is selected.

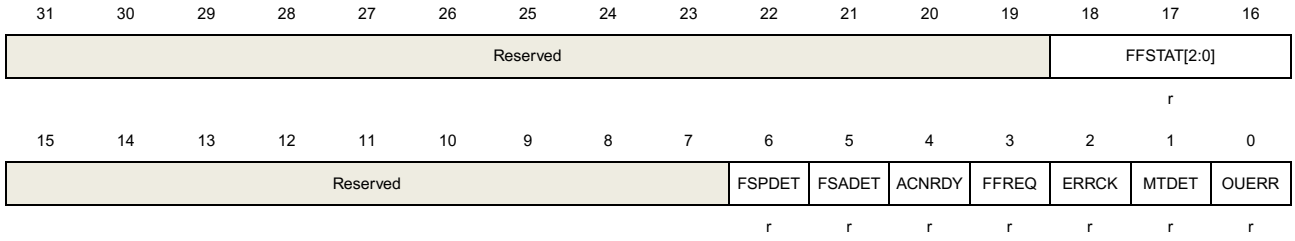
3	FFREQIE	FIFO request interrupt enable. 0: Interrupt disabled. 1: Interrupt enabled. An interrupt is generated if FFREQ and FFREQIE are both set. **Note**: When the audio sub-block is configured as receiver, OPTMOD must be set before FFREQIE is enabled to guarantee no false FIFO request is generated since the sub-block is transmitter after reset.
2	ERRCKIE	Error clock interrupt enable. It is set and cleared by software. 0: Interrupt disabled. 1: Interrupt enabled. An interrupt is generated if ERRCK and ERRCKIE are both set. **Note**: This control field is relevant only if the audio sub-block is configured as master and the BYPASS is set to 0 in the clock divide logic. **Note**: This control field is used when TDM mode is selected, it is meaningless in other modes.
1	MTDETIE	Mute detection interrupt enable. 0: Interrupt disabled. 1: Interrupt enabled. An interrupt is generated if MTDET and MTDETIE are both set. Note: this control field has a meaning only when the audio sub-block is configured as receiver.
0	OUERRIE	FIFO overrun or underrun interrupt enable. 0: Interrupt disabled. 1: Interrupt enabled. An interrupt is generated if OUERR and OUERRIE are both set.

52.4.7. Block x status register (SAI_BxSTAT) (x = 0,1)

Address offset: 0x18 + 0x20 * x

Reset value: 0x0000 0008

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:19	Reserved	Must be kept at reset value.
18:16	FFSTAT[2:0]	FIFO status.

		Indicate the full/empty status of the FIFO, and it is controlled solely by hardware, where different evaluation standards is present according to audio sub-blocks operating mode. In case OPTMOD is configured as receiver: 000: Empty. 001: Empty <FIFO_Level<= 1/4_Full. 010: 1/4_Full <FIFO_Level<= 1/2_Full. 011: 1/2_Full <FIFO_Level<= 3/4_Full. 100: 3/4_Full <FIFO_Level< Full 101: Full. And when OPTMOD is configured as transmitter: 000: Empty. 001: Empty <FIFO_Level< 1/4_Full. 010: 1/4_Full <= FIFO_Level< 1/2_Full. 011: 1/2_Full <= FIFO_Level< 3/4_Full. 100: 3/4_Full <= FIFO_Level< Full 101: Full.
15:7	Reserved	Must be kept at reset value.
6	FSPDET	Frame synchronization postponed detection. 0: Correct FS edge reception. 1: FS edge reception postponed. FS edge reception postponed could generate an interrupt if FSPDETIE is set. This flag is cleared by FSPDETC control field. **Note**: This control field has a meaning only when the audio sub-block is configured as slave.
5	FSADET	Frame synchronization advanced detection. 0: Correct FS edge reception. 1: FS edge reception advanced. FS edge reception advanced could generate an interrupt if FSADETIE is set. This flag is cleared by FSADETC control field. **Note**: This control field has a meaning only when the audio sub-block is configured as slave.
4	ACNRDY	Audio codec not ready. 0: AC'97 codec ready. 1: AC'97 codec not ready. Bit-15 in TAG slot of each frame is the AC'97 audio codec ready indication bit. 0 indicate the audio codec is not ready, on the other hand, 1 means ready. AC'97 codec not ready could generate an interrupt if ACNRDYIE is set. This flag is cleared by ACNRDYC control field. **Note**: This control field is used only in AC'97 mode.
3	FFREQ	FIFO request

		0: No FIFO request. 1: FIFO write or read request. FIFO request could generate an interrupt if FFREQIE is set. The request type depend on audio sub-block configuration, if OPTMOD is configured as transmitter and all conditions met, write request in generated, else if receiver mode is selected, read request is generated.
2	ERRCK	Error clock. 0: Correct clock configuration 1: Error clock configuration. Error clock configuration could generate an interrupt if ERRCKIE is set. This flag is cleared by ERRCKC control field. This control field has a meaning only when the audio sub-block is configured as master and BYPASS is set to 0.
1	MTDET	Mute detection. 0: No mute detected 1: Mute detected. Mute detection could generate an interrupt if MTDETIE is set. This flag is cleared by MTDETC control field. When the number of frame whose slots received is all 0 value reaches the frame number defined in MTFCNT, mute detection flag is set. Mute could not be detected when slot number is less than or equal to 2 and MTVAL is set, in the transmitter, where the frame before mute is transferred repeatedly.
0	OUERR	Overrun or underrun. 0: No FIFO overrun or underrun detected 1: FIFO overrun or underrun detected. FIFO overrun or underrun interrupt could be generated if OUERRIE is set. This flag is cleared by OUERRC control field. FIFO overrun is generated only when the audio sub-block is configured as receiver, the received data fill the FIFO and the FIFO is full. FIFO underrun is generated only when the audio sub-block is configured as transmitter, the FIFO is empty and a transfer request is present.

52.4.8. Block x interrupt flag clear register (SAI_BxINTC) (x = 0,1)

Address offset: 0x1C + 0x20 * 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	FSPDETC	FSADETC	ACNRDYC	Reserved	ERRCKC	MTDETC	OUERRC
	w	w	w		w	w	w

Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value.
6	FSPDETC	Frame synchronization postponed detection interrupt clear. Writing 1 clears FSPDET flag. **Note**: This control field is not used in AC'97 or SPDIF mode. **Note**: This field always reads as 0.
5	FSADETC	Frame synchronization advanced detection interrupt clear. Writing 1 clears FSADET flag. **Note**: This control field is not used in AC'97 or SPDIF mode. **Note**: This field always reads as 0.
4	ACNRDYC	Audio codec not ready interrupt clear. Writing 1 clears ACNRDY flag. **Note**: This control field is only used in AC'97 mode. **Note**: This field always reads as 0.
3	Reserved	Must be kept at reset value.
2	ERRCKC	Clock error interrupt clear. Writing 1 clears ERRCK flag. **Note**: This control field is only used when the audio sub-block is configured as master and BYPASS is set to 0. Note: This field always reads as 0.
1	MTDETC	Mute detection interrupt clear. Writing 1 clears MTDET flag. **Note**: This field always reads as 0.
0	OUERRC	Overrun or underrun interrupt clear. Writing 1 clears OUERR flag. **Note**: This field always reads as 0.

52.4.9. Block x data register (SAI_BxDATA) (x = 0,1)

Address offset: 0x20 + 0x20 * 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
DATA[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA[15:0]															
rw															

Bits	Fields	Descriptions
31:0	DATA[31:0]	Data. Write and read operations are performed on the FIFO directly.

52.4.10. PDM control register (SAI_PDMCTL)

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						CLKL1EN	CLKL0EN	Reserved		MICNUMSEL[1:0]		Reserved			PDMEN
						rw	rw			rw					rw

Bits	Fields	Descriptions
31:10	Reserved	Must be kept at reset value.
9	CLKL1EN	PDM clock line 1 enable. 0: PDM clock line 1 disable 1: PDM clock line 1 enable
8	CLKL0EN	PDM clock line 0 enable. 0: PDM clock line 0 disable 1: PDM clock line 0 enable
7:6	Reserved	Must be kept at reset value.
5:4	MICNUMSEL[1:0]	Select microphones number 00: 2 microphones 01: 4 microphones 10: 6 microphones 11: 8 microphones
3:1	Reserved	Must be kept at reset value.
0	PDMEN	PDM enable 0: PDM disable 1: PDM enable

52.4.11. PDM configuration register (SAI_PDMCFG)

Address offset: 0x48

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	DPR3[2:0]			Reserved	DPL3[2:0]			Reserved	DPR2[2:0]			Reserved	DPL2[2:0]		
	rw				rw				rw				rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	DPR1[2:0]			Reserved	DPL1[2:0]			Reserved	DPR0[2:0]			Reserved	DPL0[2:0]		
	rw				rw				rw				rw		

Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30:28	DPR3[2:0]	Group 3 right channel microphone data flow delay period 000: No delay 001: delay 1 T_{SAI_CK} period 010: delay 2 T_{SAI_CK} periods … 111: delay 7 T_{SAI_CK} periods
27	Reserved	Must be kept at reset value.
26:24	DPL3[2:0]	Group 3 left channel microphone data flow delay period 000: No delay 001: delay 1 T_{SAI_CK} period 010: delay 2 T_{SAI_CK} periods … 111: delay 7 T_{SAI_CK} periods
23	Reserved	Must be kept at reset value.
22:20	DPR2[2:0]	Group 2 right channel microphone data flow delay period 000: No delay 001: delay 1 T_{SAI_CK} period 010: delay 2 T_{SAI_CK} periods … 111: delay 7 T_{SAI_CK} periods
19	Reserved	Must be kept at reset value.
18:16	DPL2[2:0]	Group 2 left channel microphone data flow delay period 000: No delay 001: delay 1 T_{SAI_CK} period 010: delay 2 T_{SAI_CK} periods

		… 111: delay 7 T_{SAI_CK} periods
15	Reserved	Must be kept at reset value.
14:12	DPR1[2:0]	Group 1 right channel microphone data flow delay period 000: No delay 001: delay 1 T_{SAI_CK} period 010: delay 2 T_{SAI_CK} periods … 111: delay 7 T_{SAI_CK} periods
11	Reserved	Must be kept at reset value.
10:8	DPL1[2:0]	Group 1 left channel microphone data flow delay period 000: No delay 001: delay 1 T_{SAI_CK} period 010: delay 2 T_{SAI_CK} periods … 111: delay 7 T_{SAI_CK} periods
7	Reserved	Must be kept at reset value.
6:4	DPR0[2:0]	Group 0 right channel microphone data flow delay period 000: No delay 001: delay 1 T_{SAI_CK} period 010: delay 2 T_{SAI_CK} periods … 111: delay 7 T_{SAI_CK} periods
3	Reserved	Must be kept at reset value.
2:0	DPL0[2:0]	Group 0 left channel microphone data flow delay period 000: No delay 001: delay 1 T_{SAI_CK} period 010: delay 2 T_{SAI_CK} periods … 111: delay 7 T_{SAI_CK} periods

53. Universal serial bus High-Speed interface (USBHS)

GD32H77x_H78x series supports up to 2 USBHS,USBHS0 and USBHS1. Some series have only one USBHS0. Please refer to the GD32H7 Selection Manual for details on how many USBHS are supported.

53.1. Overview

USB High-Speed (USBHS) controller provides a USB-connection solution for portable devices. USBHS supports both host and device modes, as well as OTG mode with HNP (Host Negotiation Protocol) and SRP (Session Request Protocol). USBHS contains an embedded USB PHY internal which can be configured as High-Speed or Full-Speed. USBHS supports all the four types of transfer (control, bulk, Interrupt and isochronous) defined in USB 2.0 protocol. There is also a DMA engine operating as an AHB bus master in USBHS to speed up the data transfer between USBHS and system. For Full-Speed operation, battery charging detection (BCD), attach detection protocol (ADP), and link power management (LPM) are also supported.

Note: USBHS1 can not support OTG mode, and it contains an embedded PHY internal which can be configured as Full-Speed only. It also can not support BCD and ADP.

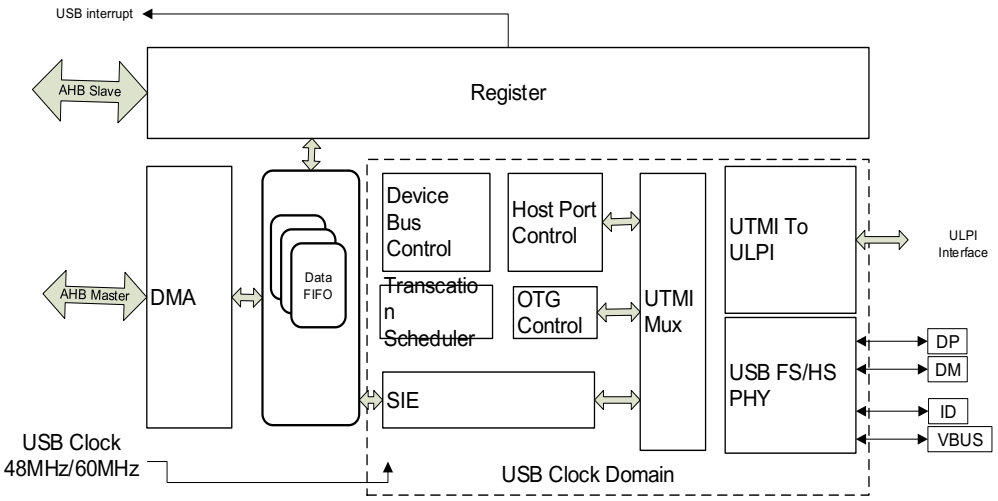
53.2. Characteristics

- Supports USB 2.0 Host mode at High-Speed(480Mb/s), Full-Speed(12Mb/s) or Low-Speed(1.5Mb/s)
- Supports USB 2.0 device mode at High-Speed(480Mb/s) or Full-Speed(12Mb/s)
- Supports OTG protocol with HNP (Host Negotiation Protocol) and SRP (Session Request Protocol) (just for USBHS0)
- Supports all the 4 types of transfer: control, bulk, Interrupt and isochronous
- Supports high-bandwidth interrupt and isochronous transfers
- Includes USB transaction scheduler in host mode to handle USB transaction request efficiently
- Includes a 4KB FIFO RAM
- Supports 16 channels in host mode
- Contains 2 transmit FIFOs (periodic and non-periodic) and one receive FIFO (shared by all channels) in host mode
- Contains 8 transmit FIFOs (one for each IN endpoint) and one receive FIFO (shared by all OUT endpoints) in device mode
- Supports PING protocol in host mode when operates at High-Speed
- Supports 8 OUT and 8 IN Endpoints in device mode
- Supports remote-wakeup in device mode
- Include a USB PHY with OTG protocol supported (just for USBHS0)

- Include an internal DMA scheduler and engine to perform data copy between USBHS and system per the application's request
- Time intervals of SOFs is dynamic adjusted in host mode
- Able to output SOF pulse to PAD
- Able to detect ID level and VBUS voltage (just for USBHS0)
- Needs external component to supply power for connected USB device in Host mode or OTG A-device (just for USBHS0)
- Supports charging port detection (BCD) described in Battery Charging Specification Revision 1.2 (just for USBHS0)
- Supports Attach Detection Protocol (ADP) described in USB On-The-Go Supplement, Revision 2.0 (just for USBHS0)
- Supports Link Power Management (LPM) described in USB 2.0 Link Power Management Addendum and Errata for USB 2.0 ECN: Link Power Management (LPM)

53.3. Block diagram

Figure 53-1. USBHS block diagram



This series has two USB HS modules(USBHS0 and USBHS1), And both of which support ULPI interface. And high-speed operation is allowed with external HS transceivers.

53.4. Signal description

Table 53-1. USBHS signal description

I/O port	Type	Description	Note
VBUS	input	bus power port	for internal PHY only
DM	input/output	differential data line - port	for internal PHY only
DP	input/output	differential data line + port	for internal PHY only
ID	input	USB identification: Mini	for internal PHY only

I/O port	Type	Description	Note
		connector identification port	
ULPI_D[7:0]	input/output	ULPI Data line	for external ULPI PHY
ULPI_NXT	input	ULPI next line	for external ULPI PHY
ULPI_DIR	input	ULPI Direction	for external ULPI PHY
ULPI_STP	output	ULPI Stop	for external ULPI PHY
ULPI_CLK	input	ULPI Clock	for external ULPI PHY

53.5. Function overview

53.5.1. USBHS PHY selection, clocks and working modes

USBHS can operate as a host, a device or a DRD (Dual-role-Device) and supports two types of connection: internal embedded PHY and external ULPI PHY. The application choose to use either the internal embedded PHY or the external ULPI PHY according to the demand.

The application may limit the maximum speed of the internal PHY or the external ULPI PHY to Full-Speed using SPDFSLS bit in USBHS_HCTL register in host mode or DS[1:0] in USBHS_DCFG register in device mode.

Table 53-2. USBHS supported speeds

Register configuration		Host supported speed	Device support speed
EMBPHY_FS=1 EMBPHY_HS=0 (Internal FS PHY)		Full-Speed Low-Speed	Full-Speed
EMBPHY_FS=0 EMBPHY_HS=1 (Internal HS PHY) (just for USBHS0)	DS =01 (device mode) SPDFSLS=1(host mode)	Full-Speed Low-Speed	Full-Speed
	DS =00(device mode) SPDFSLS=0(host mode)	High-Speed Full-Speed Low-Speed	High-Speed Full-Speed
EMBPHY_FS=0 EMBPHY_HS=0 (External ULPI PHY)	DS =01 (device mode) SPDFSLS=1(host mode)	Full-Speed Low-Speed	Full-Speed
	DS =00(device mode) SPDFSLS=0(host mode)	High-Speed Full-Speed Low-Speed	High-Speed Full-Speed

The application control the working modes of USBHS: force host, force device by setting FHM and FDM bits in USBHS_GUSBCS register. When both bits are cleared, USBHS works in OTG mode, which is the default mode after system reset.

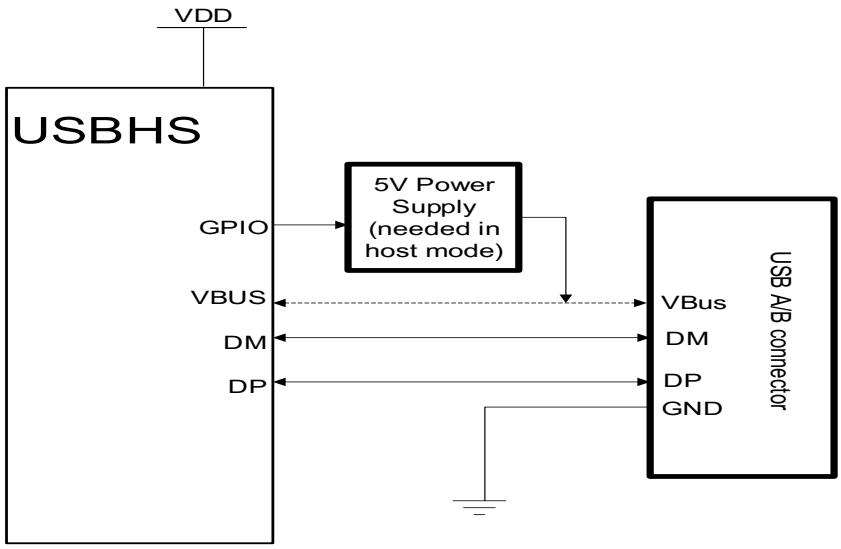
Internal embedded PHY

USBHS includes an internal embedded PHY. The internal embedded PHY supports High-Speed, Full-Speed and Low-Speed in host mode, and High-Speed, Full-Speed in device

mode, supports OTG protocol with HNP and SRP. Software needs to set EMBPHY_FS bit and reset EMBPHY_HS in USBHS_GUSBCS register to use this PHY in FS mode, or reset EMBPHY_FS bit and set EMBPHY_HS to use this PHY in HS mode. If internal embedded PHY is selected, the USB clock used for the USBHS needs to be 48MHz in FS mode and 60MHz in HS mode. The 48MHz USB clock is generated from internal clocks in system, and its source and divider factors are configurable in RCU, while the 60MHz USB clocks is generated from 480MHz PLLUSB.

The pull-up and pull-down resistors are already integrated into the internal PHY and controlled by USBHS automatically based on the current mode (host, device or OTG mode) and connection status. A typical connection using internal PHY is shown in ***Figure 53-2. Connection using internal embedded PHY with host or device mode.***

Figure 53-2. Connection using internal embedded PHY with host or device mode



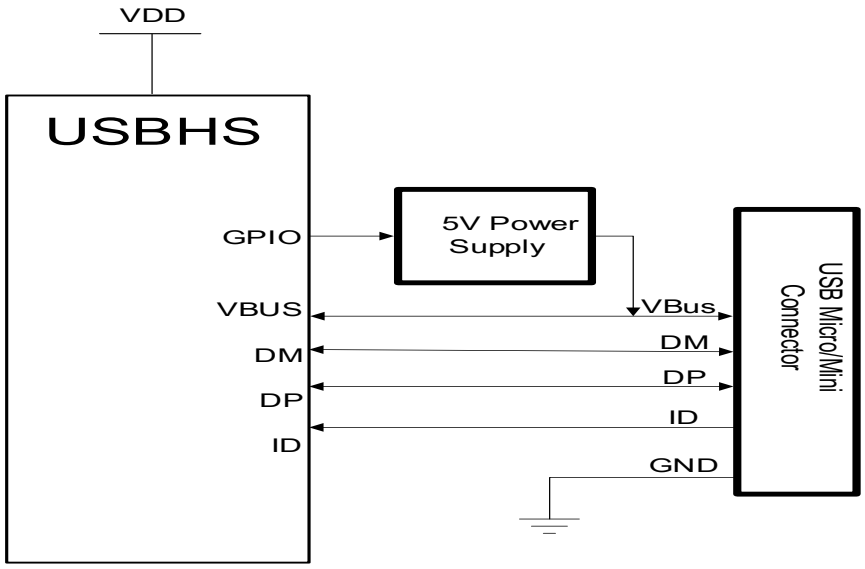
When USBHS works in host mode (FHM bit is set and FDM bit is cleared), the VBUS is 5V power pin defined in USB protocol. The internal PHY cannot supply 5V VBUS power and only has some voltage comparers, charge and dis-charge circuit on VBUS line. If application needs to supply USB power, an external power supply IC is needed. The VBUS connection between USBHS and the USB connector can be omitted in host mode because USBHS doesn't detect the voltage level on VBUS pin and always assumes that the 5V power is present.

When USBHS works in device mode (FHM bit is cleared and FDM bit is set), the VBUS detection circuit is decided by VDEN bit in USBHS_GCCFG register. If the device does not need to detect the voltage on VBUS pin, it may set the VDEN bit and free the VBUS pin for other use. Otherwise, the VBUS connection cannot be omitted, and USBHS continuously monitor the VBUS voltage and will immediately switch off the pull-up resistor on DP line once the VBUS voltage falls below the needed valid value. This will cause a disconnection.

The OTG mode connection is described in the ***Figure 53-3. Connection using internal embedded PHY with OTG mode***. When USBHS works in OTG mode, the FHM, FDM bits in USBHS_GUSBCS and VDEN bit in USBHS_GCCFG should be cleared. In this mode, the

USBHS needs all the four pins: DM, DP, VBUS and ID, and uses several voltage comparers to monitor the voltage on these pins. USBHS also includes VBUS charge and discharge circuit to perform SRP request described in OTG protocol. The OTG A-Device or B-Device is decided by the level of ID pins. USBHS controls the pull-up or pull-down resistor during the HNP protocol.

Figure 53-3. Connection using internal embedded PHY with OTG mode



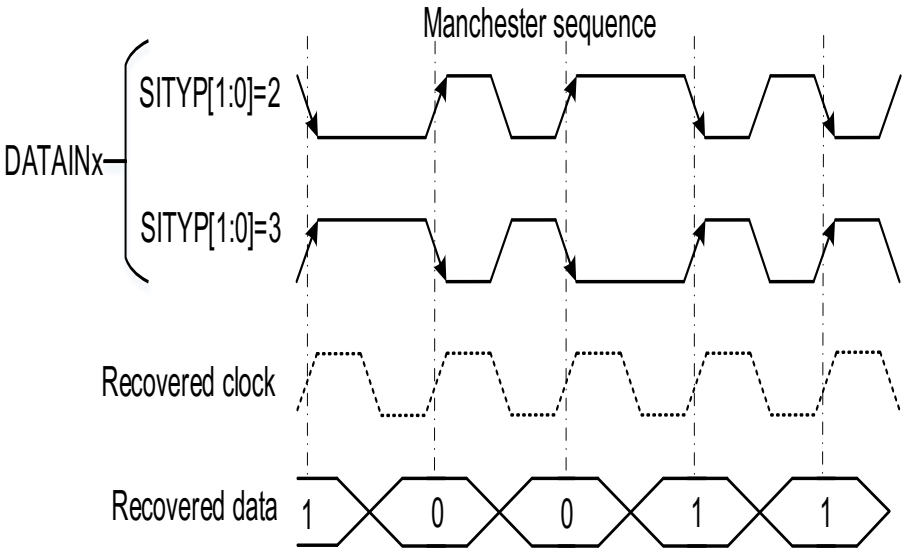
Note: USBHS1 can not support OTG mode.

External ULPI PHY

USBHS provides a ULPI interface for external PHY integration. An external High-Speed ULPI PHY is needed to support High-Speed USB applications. With external ULPI PHY, USBHS supports High-Speed host and device, all the modes described in internal embedded PHY.

Software needs to clear the EMBPHY_FS bit and EMBPHY_HS bit in USBHS_GUSBCS register to enable the ULPI interface. When ULPI mode enabled, the USB clock which introduced from the ULPI_CLK pin needs to be 60MHz.Software can switch on or off the 60MHz ULPI clock in RCU.

Figure 53-4. Connection using external ULPI PHY

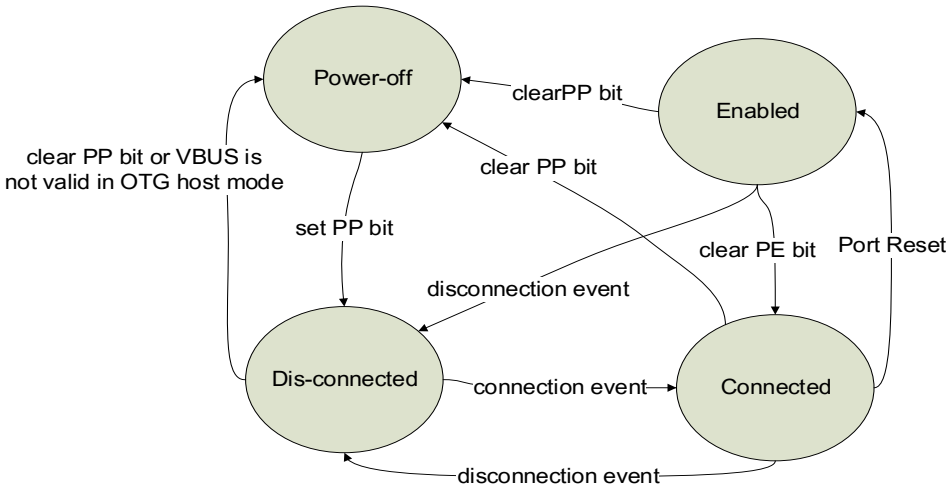


53.5.2. USB host function

USB Host Port State

Host application may control state of the USB port via USBHS_HPCS register. After system initialization the USB port, keep it at power-off state. After PP bit is set by software, the USB PHY (either internal or external) is powered on, and the USB port changes into disconnected state. After a connection is detected, USB port changes into connected state. The USB port changes into enabled state after a port reset is performed on USB bus.

Figure 53-5. State transition diagram of host port



Connection, Reset and Speed identification

As a USB host, USBHS will trigger a connection flag for application after a connection is

detected and will trigger disconnection flag after a disconnection event.

PRST bit is used for USB reset sequence. Application may set this bit to start a USB reset and clear this bit to finish the USB reset. This bit only takes effect when port is at connection or enabled state.

The USBHS performs speed identification during connection and reset, and the speed information is reported in PS[1:0] bits in USBHS_HPCS register.

If the maximum supported speed is configured to Full-Speed (SPDFSLS = 1), USBHS only performs speed-identification during device connection process and it identifies the device speed from the voltage level of DM or DP. As is described in USB protocol, Full-Speed device pulls up DP line while Low-Speed device pulls up DM line.

If the maximum supported speed is configured to High-Speed (SPDFSLS = 0), USBHS first performs speed-identification during connection. If a Full-Speed connection is detected, the USBHS will try to perform High-Speed identification (CHIRP sequence described in USB 2.0 protocol) during each USB reset sequence after the connection event. So the application on host should perform a USB reset after a connection event and check the PS[1:0] bits again if it desires to support High-Speed device.

Suspend and resume

USBHS supports suspend state and resume operation. When USBHS port is at enabled state, writing 1 to PSP bit in USBHS_HPCS register will cause USBHS to enter suspend state. In suspend state, USBHS stops sending SOFs on USB bus and this will cause the connected USB device to enter suspend state after 3ms. Application can set the PREM bit in USBHS_HPCS register to start a resume sequence to wake up the suspended device and clear this bit to stop the resume sequence. The WKUPIF bit in USBHS_GINTF and the USBHS wake up interrupt will be triggered if a host in suspend state detects a remote wakeup signal.

SOF generate

USBHS sends SOF tokens on USB bus in host mode. As described in USB 2.0 protocol, SOF packets are generated (by the host controller or hub transaction translator) every 1ms for Full-Speed links, and every 125 μs for High-Speed links.

Each time after USBHS enters into enabled state, it will send SOF packet using the time defined by USB 2.0 protocol. While, application may adjust the length of a frame or a micro-frame by writing to FRI[15:0] in USBHS_HFT registers. The FRI bits define the number of USB clock cycles in a frame or micro-frame and application should calculate the value based on the frequency of USB clock used by USBHS. The FRT[15:0] bits reflect the remaining clock cycles of the current frame or micro-frame and stops to change during suspend state.

USBHS is able to generate a pulse signal each SOF packet and output it to a pin. The pulse length is 12 HCLK cycle. If application desires to use this function, it needs to set SOFOEN bit in USBHS_GCCFG register and configure the related pin registers in GPIO.

USB Channels and Transactions

USBHS includes 16 independent channels in host mode. Each channel is able to communicate with an endpoint in USB device. The transfer type, direction, packet length and other information is configured in channel related registers such as USBHS_HCHxCTL and USBHS_HCHxLEN.

USBHS supports all the four kinds of transfer types: control, bulk, interrupt and isochronous. USB 2.0 protocol divides these transfers into 2 kinds: non-periodic transfer (control and bulk) and periodic transfer (interrupt and isochronous). Based on this, USBHS includes two request queues: periodic request queue and non-periodic request queue, in order to perform efficient transaction schedule. A request entry in a request queue described above may represent a USB transaction request or a channel operation request.

In non-DMA mode, application needs to write packet into data FIFO via AHB register interface if it wants to start an OUT transaction on USB bus. USBHS hardware will automatically generate a transaction request entry in request queue after the application writes a whole packet. In DMA mode, application only needs to configure the channel property and channel data buffer address, and the DMA engine in USBHS performs the packet data copy and request entry generation. USBHS automatically generate IN request entries when the application enable an IN channel.

The request entries in request queue are processed in order by transaction control module. USBHS always try to process periodic request queue first, then process non-periodic request queue.

After a start of frame USBHS begins to process periodic queue until the queue is empty or bus time required by the current periodic request is not enough, and then process the non-periodic queue. This strategy ensures the bandwidth of periodic transactions in a frame or micro-frame. Each time the USBHS reads and pop a request entry from request queue. If this is a channel disable request, it immediately disables the channel and prepare to process next entry.

If the current request is a transaction request and the USB bus time is enough for this transaction, USBHS will employ SIE to generate this transaction on USB bus.

When the required bus time by the current request is not enough in the current frame, if this is a periodic request, USBHS stops the processing of periodic queue and starts to process non-periodic request. If this is a non-periodic queue the USBHS will stop to process any queue and wait until the end of current frame.

LPM

This addendum for USB defines power management states (LPM states) and mechanisms to affect state changes that are used by hosts and hubs to efficiently manage bus and system power. LPM simply adds a new feature and bus state sleep state (L1) that co-exists with the USB2.0 defined suspend (L2)/resume.

L1 is similar to L2 but supports finer granularity in use. Entry to L1 is started by a request to

a hub or host port to transition to L1. A LPM transaction is sent to the downstream device. The requested transition can only occur if the device response with an ACK handshake. Exit from L1 is via remote wake, resume signaling, reset signaling or disconnect. Either the host or device can initiate resume signaling when in L1. Although the signaling levels of resume are the same as L2, the duration of the signaling and transitional latencies associated with the L1 to L0 (active state) transition are much shorter.

53.5.3. USB device function

USB Device Connection

In device mode USBHS stays at power-off state after initialization. After connected to a USB host with 5V power supply present on VBUS pin or setting VDEN bit in USBHS_GCCFG register, USBHS enters into powered state. USBHS begins to switch on the pull-up resistor on DP line, host side will detect a connection event.

Reset and Speed-Identification

The USB host always starts a USB reset after it detects a device connection, USBHS in device mode will trigger a reset interrupt for software after it detects the reset event on USB bus.

If the maximum supported speed is configured to Full-Speed (DS[1:0] = 01 in USBHS_DCFG register), USBHS will operate as a Full-Speed device. If the maximum supported speed is configured to High-Speed (DS[1:0] = 00 in USBHS_DCFG register), USBHS device tries to start a speed-identification (a chirp handshake described in USB 2.0 protocol) with host during reset sequence. If the chirp handshake with host successes, the device enters High-Speed mode, otherwise, remains at Full-Speed mode.

After reset sequence, speed-identification process completes, USBHS triggers an ENUMF interrupt in USBHS_GINTF register and reports current enumerated device speed in ES bits in USBHS_DSTAT register. If software want to implement a High-Speed device, it must wait ENUMF interrupt first, then read the ES[1:0] bits to get the speed-identification result.

As required by USB 2.0 protocol, USBHS doesn't support Low-Speed in device mode.

Suspend and Wake-up

A USB device will enter into suspend state after the USB bus stays at IDLE state and has no change on data lines for 3ms. When USB device is in suspend state, software can switch off most of its clock to save power. The USB host is able to wake up the suspended device by generating a resume signal on USB bus. USBHS is able to detect the resume signal and triggers the WKUPIF flag in USBHS_GINTF register and the USBHS wake up interrupt.

In suspend mode, USBHS is also able to remote wake-up the USB bus. Software may set RWKUP bit in USBHS_DCTL register to sends a remote-wake-up signal, and if remote-wake up is supported in USB host, the host will begin to send resume signal on USB bus.

Soft Disconnection

USBHS supports soft disconnection. After the device is power on, USBHS will switch on the pull-up resistor on DP line and this will cause the host to detect the connection. Then, software is able to force a disconnection by setting the SD bit in USBHS_DCTL register. After the SD bit is set, if the current device speed is High-Speed, USBHS will first return back to Full-Speed device and then switch off the pull-up resistor on DP line, and if current speed is Full-Speed, USBHS will directly switch off the pull-up resistor. This will cause USB host to detect a disconnection on USB bus.

SOF tracking

When USBHS receives a SOF packet from USB bus, it triggers a SOF interrupt and begins to count the bus time by using local USB clock. The frame number of the current frame is reported in FNRSOF[13:0] in USBHS_DSTAT register. When the USB bus time reaches EOF1 or EOF2 point (End of Frame, described in USB 2.0 protocol), USBHS will trigger a interrupt EOPFIF in USBHS_GINTF register. Software is able to use these flags and registers to get current bus time and position information.

BCD

Charging port detection (BCD) described in Battery Charging Specification Revision 1.2 is supported. In order for PD (portable device) to determine how much current it is allowed to draw from an upstream USB port, there need to be mechanisms that allow the PD to distinguish between a Standard Downstream Port and a Charging Port.

In BCD mechanisms, USB VBUS detection (VD), data contact detection (DCD), primary detection (PD) and secondary detection (SD) are included. The control and configuration bits about BCD is reported in USBHS_GCCFG register.

Note: USBHS1 can not support BCD.

53.5.4. OTG function overview

Note: USBHS1 can not support OTG.

USBHS supports OTG function described in OTG protocol 1.3/2.0. OTG function includes SRP and HNP protocols.

Only USBHS0 in this series supports OTG mode.

10.1.19 Device and B-Device

10.1.20 Device is an OTG capable USB device with a Standard-A or Micro-A plug inserted into its receptacle. The A-Device supplies power for VBUS and it is host at the start of a session. B-Device is an OTG capable USB device with a Standard-B, Micro-B or Mini-B plug inserted into its receptacle, or a captive cable ending in a Standard-A plug. The B-Device is a peripheral at the start of a session. USBHS uses the voltage level of ID pin to judge A-Device or B-Device. The ID status is reported in IDPS bit in USBHS_GOTGCS register. For the details of states transfer between A-Device and B-Device, please refer to OTG 1.3/2.0 protocol.

HNP

The Host Negotiation Protocol (HNP) allows the host function to be transferred between two directly connected On-The-Go devices and eliminates the need for a user to switch the cable connections in order to allow a change in control of communications between the devices. HNP will typically be initiated by the user or an application on the On-The-Go B-device. HNP may only be implemented through the Micro-AB receptacle on a device.

Since On-The-Go devices have a Micro-AB receptacle, an On-The-Go device can default to being either Host or Peripheral, depending upon which type of plug (Micro-A plug for Host, Micro-B plug for Peripheral) is inserted. By utilizing the Host Negotiation Protocol (HNP), an On-The-Go B-Device, which is the default Peripheral, may make a request to be Host. The process for this exchange of the role of Host is described in this section. This protocol eliminates the need for the user to swap the cable connection in order to change the roles of the connected devices.

When USBHS is in OTG A-Device host mode and it wants to give up its host role, it may first set PSP bit in USBHS_HPCS register to make the USB bus enter suspend status. Then, the B-device will enter suspend state after 3ms. If the B-Device wants to changes to host, software needs to set HNPREQ bit in USBHS_GOTGCS register and the USBHS will begin to perform HNP protocol on bus, and at last, the result of HNP is reported in HNPS bit in USBHS_GOTGCS register. Besides, software is always able to get the current role (host or peripheral) from COPM bit in USBHS_GINTF register.

SRP

The Session Request Protocol (SRP) allows a B-Device to request the A-Device to turn on VBUS and start a session. This protocol allows the A-Device, which may be battery powered, to conserve power by turning VBUS off when there is no bus activity while still providing a means for the B-Device to initiate bus activity. As described in OTG protocol, an OTG device must compare VBUS voltage with several threshold values and the compare result is reported in ASV and BSV bits in USBHS_GOTGCS register.

Software may set SRPREQ bit in USBHS_GOTGCS register to start a SRP request when USBHS is in B-Device OTG mode and USBHS will generate a success flag SRPS in USBHS_GOTGCS register if the SRP request successes.

When USBHS is in OTG A-Device mode and it detects an SRP request from a B-Device, it sets a SESIF flag in USBHS_GINTF register. The software should prepare to switch on the 5V power supply for VBUS pin after it gets this flag.

ADP

Attach Detection Protocol (ADP) is a protocol that allows a local device to detect when a remote device has been attached or detached. The remote device can be any USB device. ADP operates by detecting the change in VBUS capacitance that occurs when two devices are attached or detached. The capacitance is detected by first discharging the VBUS line, and then measuring the time it takes VBUS to charge to a known voltage with a known current

source. A change in capacitance is detected by looking for a change in the charge time.

Software may set bit ADPMEN, ADPEN and ENAPRB to perform ADP probe, and should perform at least one ADP probe cycle in order to obtain an initial value for TADP_RISE when an ADP-capable A-device or B-device is first powered up. For B-device, ADP sense can be performed by setting bit ENASNS. If RITM in USBHS_ADPCTL register changes, it shows that a remote device has been attached or detached.

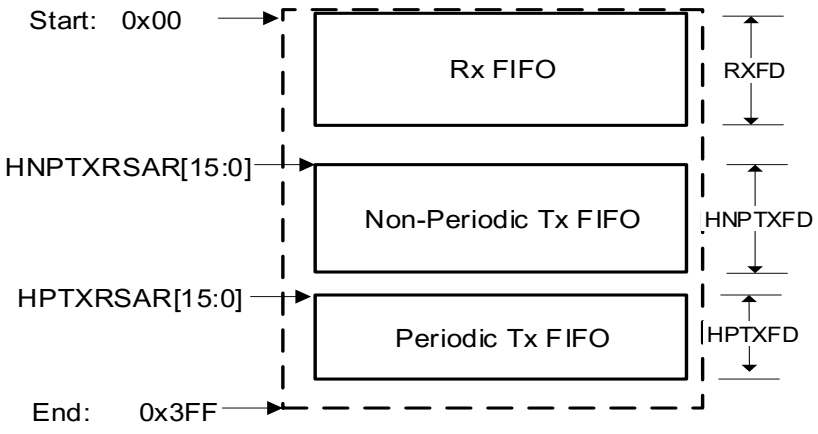
53.5.5. Data FIFO

The USBHS include a 4K bytes data FIFO to store packet data. The data FIFO is implemented by using an internal SRAM, in case that the FIFOAEN bit of USBHS_GOTGCS register is 0.

Host Mode

In host mode the data FIFO space is divided into 3 parts: Rx FIFO for received packet, Non-Periodic Tx FIFO for non-period transmission packet and Periodic Tx FIFO for periodic transmission packet. All IN channels shares the Rx FIFO for receiving packets. All the periodic OUT channels share the periodic Tx FIFO to transmit packets. All the non-periodic OUT channels share the non-Periodic FIFO for transmit packets. Software should configure the size and start offset of these data FIFOs by use these registers: USBHS_GRFLEN, USBHS_HNPTFLEN and USBHS_HPTFLEN. ***Figure 53-6. Host mode FIFO space in SRAM*** describes the structure of these FIFOs in SRAM. The values in the figure are in term of 32-bit words.

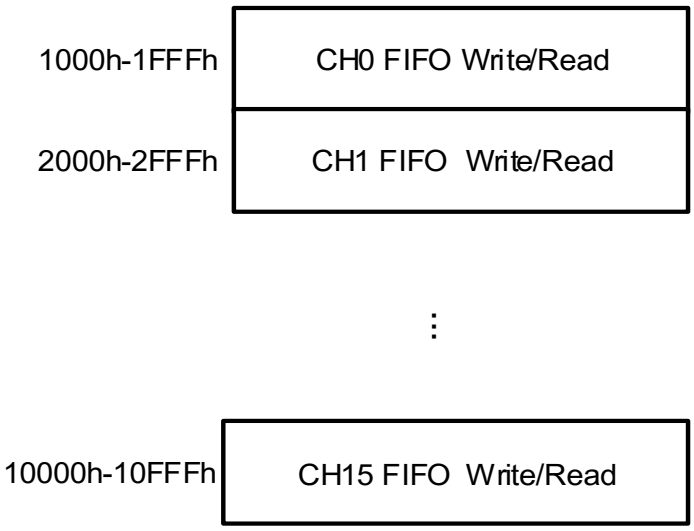
Figure 53-6. Host mode FIFO space in SRAM



In DMA mode, DMA engine is responsible for packet data copy between system memory and the internal data FIFOs. In non-DMA mode the application needs to manually write packet data into or read packet from the data FIFOs. USBHS provides a special register area for software to write and read the internal data FIFO. ***Figure 53-7. Host mode FIFO access register map*** describes the register memory area for data FIFO access. The addresses in the figure are in term of byte. Each channel has its own FIFO access register space, although all Non-periodic channels share the same FIFO and all the Periodic channels share the same

FIFO. This is important for USBHS to know the current pushed packet belongs to which channel. Rx FIFO is also able to be accessed by using USBHS_GRSTATR/USBHS_GRSTATP register.

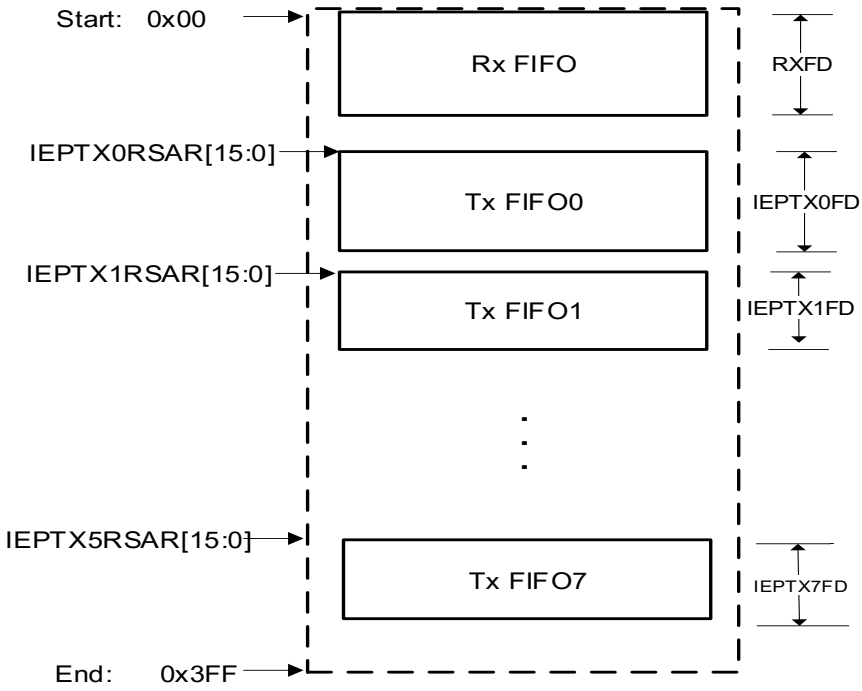
Figure 53-7. Host mode FIFO access register map



Device mode

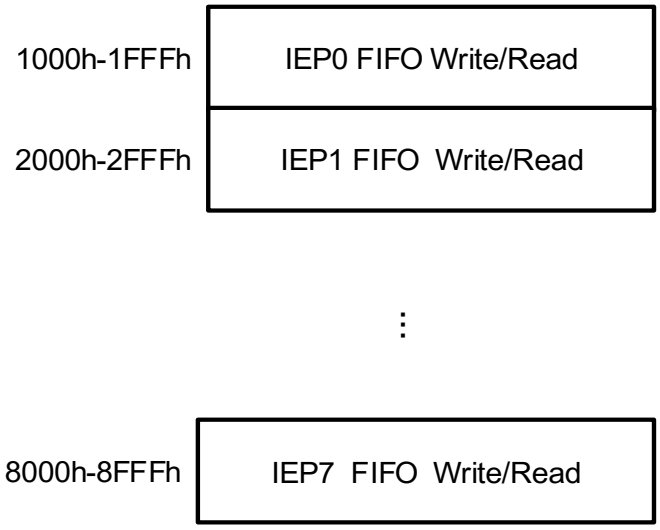
In device mode, the data FIFO is divided into several parts: one Rx FIFO, and 8 Tx FIFOs (one for each IN endpoint). All the OUT endpoints share the Rx FIFO for receiving packets. Software should configure the size and start offset of these data FIFOs by using USBHS_GRFLEN and USBHS_DIEPxTFLEN (x=0…7) registers. ***Figure 53-8. Device mode FIFO space in SRAM*** describes the structure of these FIFOs in SRAM. The values in the figure are in term of 32-bit words.

Figure 53-8. Device mode FIFO space in SRAM



In DMA mode, DMA engine is responsible for packet data copy between system memory and the internal data FIFOs. In non-DMA mode the application needs to manually write packet data into or read packet from the data FIFOs. USBHS provides a special register area for software to write and read the internal data FIFO. ***Figure 53-9. Device mode FIFO access register map*** describes the register memory area for data FIFO access. The addresses in the figure are in term of byte. Each endpoint has its own FIFO access register space. Rx FIFO is also able to be accessed by using USBHS_GRSTATR/USBHS_GRSTATP register.

Figure 53-9. Device mode FIFO access register map



53.5.6. DMA function

This section describes the DMA scheduler and DMA engine in USBHS.

DMA Requests and Scheduler

DMA function is enabled by setting DMAEN bit in USBHS_GAHBCS register. When an IN/OUT channel or IN endpoint is properly configured and enabled, or the Rx FIFO is not empty, USBHS will generate DMA request. There is a DMA scheduler in USBHS responsible for responding to theses DMA requests.

There may be several requests simultaneously and the DMA scheduler arbitrates among these requests. These requests are sorted into 3 kinds: Rx FIFO DMA request, periodic transfer DMA requests and non-periodic transfer DMA requests. Rx FIFO DMA request takes the highest priority, and periodic transfer DMA requests take the medium priority, and non-periodic transfer DMA requests take the lowest priority when arbitration. DMA scheduler performs round-robin arbitration method within the periodic or non-periodic transfer DMA requests.

As is described above, DMA will automatically handle the Rx FIFO not empty event, so software should ignore the RXFNEIF flag in USBHS_GINTF register in DMA mode.

DMA Engine

Receive:

In host or device mode, once Rx FIFO DMA request gets arbitration, DMA engine begins to read a packet or a status entry from Rx FIFO. For data packet, DMA write the data into the specified system address configured in the HCHxDMAADDR register or DIEPxDMAADDR/DOEPxDMAADDR register. For status entry, DMA will generate the specified flags or interrupts on related channels or endpoints.

Host Transfer:

When a periodic or non-periodic IN channel DMA request gets arbitration, DMA writes IN request entries into the periodic or non-periodic request queue. After the desired IN transfers completes, or an AHB/USB bus error occurs, DMA halts the specified channel and generate TF and CH flags in USBHS_HCHxINTF register. The received packet during IN transfers copied into system memory after the Rx FIFO DMA request is generated, as described above.

When an OUT periodic or non-periodic channel DMA request gets arbitration, DMA reads packet data from system memory and writes to internal Tx FIFO. DMA always writes an OUT request entry into the request queue when it finishes a packet data copying. After the desired OUT transfers completes, or an AHB/USB bus error occurs, DMA halt the specified channel and generate TF and CH flags in USBHS_HCHxINTF register.

Device Transfer:

In device mode, when an IN endpoint DMA request gets arbitration, DMA reads packet data from system memory and writes to the endpoint's Tx FIFO. When USBHS gets an IN token

on an IN endpoint, it transmits the packet copied by DMA engine.

53.5.7. Operation guide

This section describes the advised operation guide for USBHS.

Host mode

Global register initialization sequence

1. Program USBHS_GAHBCS register according to application's demand, such as: whether to enable DMA, burst type of DMA transfer and the TxFIFO's empty threshold, etc. GINTEN bit should be kept cleared at this time.
2. Program USBHS_GUSBCS register according to application's demand, such as: the operation mode (host, device or OTG) and some parameters of OTG, ULPI and USB protocols.
3. Program USBHS_GCCFG register according to application's demand.
4. Program USBHS_GRFLEN, USBHS_HNPTFLEN_DIEP0TFLEN and USBHS_HPTFLEN registers to configure the data FIFOs according to application's demand.
5. Program USBHS_GINTEN register to enable Mode Fault and Host Port interrupt and set GINTEN bit in USBHS_GAHBCS register to enable global interrupt.
6. Program SPDFSLS bit in USBHS_HCTL register to select whether to limit the device speed to Full-Speed.
7. Program USBHS_HPCS register and set PP bit.
8. Wait for a device's connection, and once a device is connected, the connection interrupt PCD in USBHS_HPCS register will be triggered. Then set PRST bit to perform a port reset. Wait for at least 10ms and then clear PRST bit.
9. Wait PEDC interrupt in USBHS_HPCS register and then read PE bit to ensure that the port is successfully enabled. Read PS[1:0] bits to get the connected device's speed and then program USBHS_HFT register if software want to change the SOF interval.

Channel initialization and enable sequence

1. Program USBHS_HCHxCTL register with desired transfer type, direction, packet size, etc. Ensure that CEN and CDIS bits keep cleared during configuration.
2. Program USBHS_HCHxINTEN register. Set the desired interrupt enable bits.
3. If DMA is enabled, program USBHS_HCHxDMAADDR register.
4. Program USBHS_HCHxLEN register. PCNT is the number of packets in a transfer and TLEN is the total bytes number of all the transmitted or received packets in a transfer.

For OUT channel: If PCNT=1, the single packet's size is equal to TLEN. If PCNT>1, the former PCNT-1 packets are considered as max-packet-length packets whose size are defined by MPL field in USBHS_HCHxCTL register, and the last packet's size is calculated based on PCNT, TLEN and MPL. If software wants to send out a zero-length packet, it should program TLEN=0, PCNT=1.

For IN channel: Because the application doesn't know the actual received data size before the IN transaction finishes, software may program TLEN as a maximum possible value supported by Rx FIFO.

5. Set CEN bit in USBHS_HCHxCTL register to enable the channel.

Channel disable sequence

Software can disable the channel by setting both CEN and CDIS bits at the same time. USBHS will generate a channel disable request entry in request queue after the register setting operation. When the request entry reaches to the top of request queue, it is processed by USBHS immediately:

For OUT channel, the specified channel will be disabled immediately. Then, a CH flag will be generated and the CEN and CDIS bits will be cleared by USBHS.

For IN channels, USBHS pushes a channel disable status entry into Rx FIFO. Software should then handle the Rx FIFO not empty event: read and pop this status entry, then, a CH flag will be generated and the CEN and CDIS bits will be cleared.

IN transfers operation sequence with DMA disabled

1. Initialize USBHS global registers.
2. Initialize the channel.
3. Enable the channel.
4. After the IN channel is enabled by software, USBHS generates a Rx request entry in the corresponding request queue.
5. When the Rx request entry reaches to the top of the request queue, USBHS begins to process this request entry. If bus time for the IN transaction indicated by the request entry is enough, USBHS starts the IN transaction on USB bus.
6. If the IN transaction finishes successfully (ACK handshake received), USBHS pushes the received data packet into the Rx FIFO and triggers ACK flag. Otherwise, the status flag (NAK) report the transaction result.
7. If the IN transaction described in step 5 is successful and PCNT is larger than 1 in step2, software should return to step 3 and continues to receive the remaining packets. If the IN transaction described in step 5 is not successful, software should return to step 3 to re-receive the packet again.
8. After all the transactions in a transfer are successful received on USB bus, USBHS push

a TF status entry into the Rx FIFO on top of the last packet data. After software reads and pops all the received data packet, and at last, the TF status entry, USBHS generates TF flag to indicate that the transfer successfully finishes.

9. Disable the channel. Now the channel is in IDLE state and is ready for other transfers.

IN transfers operation sequence with DMA enabled

1. Initialize USBHS global registers.
2. Initialize and enable the channel.
3. After the IN channel is enabled by software, USBHS begins to generate Rx request entry in the corresponding request queue.
4. USBHS processes the request entries in request queue one by one and perform the indicated IN transactions on USB bus.
5. When a IN transaction gets a NAK handshake, the DMA is able to re-send IN tokens automatically until that USBHS get the desired number of packets.
6. After USBHS gets the desired number of packets specified by PCNT in USBHS_HCHxLEN register, USBHS generates TF and CH flags to indicate that the transfer successfully finishes and the channel is disabled. If USB bus error or DMA write error occurs during these transactions, DMA will trigger related error flags, stops the processing for this channel, disable this channel and at last, trigger the CH flag.

Note: In DMA mode, software should not enable or process the RXFNEIF interrupt because the DMA will automatically process the Rx FIFO.

OUT transfers operation sequence with DMA disabled

1. Initialize USBHS global registers.
2. Initialize and enable the channel.
3. Write a packet into the channel's Tx FIFO (Periodic Tx FIFO or non-periodic Tx FIFO). After the whole packet data is written into the FIFO, USBHS generates a Tx request entry in the corresponding request queue and decrease the TLEN field in USBHS_HCHxLEN register with the written packet's size.
4. When the request entry reaches to the top of the request queue, USBHS begins to process this request entry. If bus time for the transaction indicated by the request entry is enough, USBHS starts the OUT transaction on USB bus.
5. When the OUT transaction indicated by the request entry finishes on USB bus, PCNT in USBHS_HCHxLEN register is decreased by 1. If the transaction finishes successfully (ACK handshake received), the ACK flag is triggered. Otherwise, the status flag (NAK) report the transaction result.
6. If the OUT transaction described in step 5 is successful and PCNT is larger than 1 in step2, software should return to step 3 and continues to send the remaining packets. If

the OUT transaction described in step 5 is not successful, software should return to step 3 to resend the packet again.

7. After all the transactions in a transfer are successful sent on USB bus, USBHS generates TF flag to indicate that the transfer successfully finishes.
8. Disable the channel. Now the channel is in IDLE state and is ready for other transfers.

OUT transfers operation sequence with DMA enabled

1. Initialize USBHS global registers.
2. Initialize and enable the channel.
3. DMA in USBHS begins to fetch packets from the address specified by DMAADDR in USBHS_HCHxDMAADDR register and write them into the channel's Tx FIFO (Periodic Tx FIFO or non-periodic Tx FIFO). Each time a whole packet data is written into the FIFO, USBHS generates a Tx request entry in the corresponding request queue and decrease the TLEN field in USBHS_HCHxLEN register with the written packet's size.
4. USBHS processes the request entries in request queue one by one and sends out the indicated transactions on USB bus.
5. When a transaction gets a NAK or NYET handshake, the DMA is able to re-fetch and re-send the packet as well as perform PING protocol automatically.
6. If all the transactions are successful sent on USB bus, USBHS generates TF and CH flags to indicate that the transfer successfully finishes and the channel is disabled. If USB bus error or DMA fetch error occurs during these transactions, DMA will trigger related error flags, stops the processing for this channel, disable this channel and at last, trigger the CH flag.

Note: In DMA mode, software should not enable or process the RXFNEIF interrupt because the DMA will automatically process the Rx FIFO.

Device mode

Global register initialization sequence

1. Program USBHS_GAHBCS register according to application's demand, such as: whether to enable DMA, burst type of DMA transfer and the TxFIFO's empty threshold, etc. GINTEN bit should be kept cleared at this time.
2. Program USBHS_GUSBCS register according to application's demand, such as: the operation mode (host, device or OTG) and some parameters of OTG, ULPI and USB protocols.
3. Program USBHS_GCCFG register according to application's demand.
4. Program USBHS_GRFLEN, USBHS_HNPTFLEN_DIEP0TFLEN and USBHS_DIEPxTFLEN registers to configure the data FIFOs according to application's

demand.

5. Program USBHS_GINTEN register to enable Mode Fault, Suspend, SOF, Enumeration Done and USB Reset interrupt and then, set GINTEN bit in USBHS_GAHBCS register to enable global interrupt.
6. Program USBHS_DCFG register according to application's demand, such as the device speed and device address, etc.
7. After the device is connected to a host, the host will perform port reset on USB bus and this will trigger the RST interrupt in USBHS_GINTF register.
8. Wait for ENUMF interrupt in USBHS_GINTF register and then read ES[1:0] bits in USBHS_DSTAT register to get the current enumerated device speed.

Endpoint initialization and enable sequence

1. Program USBHS_DIEPxCTL or USBHS_DOEPxCTL register with desired transfer type, packet size, etc.
2. Program USBHS_DIEPINTEN or USBHS_DOEPINTEN register. Set the desired interrupt enable bits.
3. If DMA is enabled, program USBHS_DIEPxDMAADDR or USBHS_DOEPxDMAADDR register.
4. Program USBHS_DIEPxLEN or USBHS_DOEPxLEN register. PCNT is the number of packets in a transfer and TLEN is the total bytes number of all the transmitted or received packets in a transfer.

   For IN endpoint：If PCNT=1, the single packet's size is equal to TLEN. If PCNT>1, the former PCNT-1 packets are considered as max-packet-length packets whose size are defined by MPL field in USBHS_DIEPxCTL register, and the last packet's size is calculated based on PCNT, TLEN and MPL. If software wants to send out a zero-length packet, it should program TLEN=0, PCNT=1.

   For OUT endpoint：Because the application doesn't know the actual received data size before the OUT transaction finishes, software may program TLEN as a maximum possible value supported by Rx FIFO.
5. Set EPEN bit in USBHS_DIEPxCTL or USBHS_DOEPxCTL register to enable the endpoint.

Endpoint disable sequence

Software can disable the endpoint anytime when clearing the EPEN bit in USBHS_DIEPxCTL or USBHS_DOEPxCTL register.

IN transfers operation sequence with DMA disabled

1. Initialize USBHS global registers.

2. Initialize and enable the IN endpoint.
3. Write packets into the endpoint's Tx FIFO. Each time a data packet is written into the FIFO, USBHS decreases the TLEN field in USBHS_DIEPxLEN register with the written packet's size.
4. When an IN token is received, USBHS transmit the data packet, and after the transaction finishes on USB bus, PCNT in USBHS_DIEPxLEN register is decreased by 1. If the transaction finishes successfully (ACK handshake received), the ACK flag is triggered. Otherwise, the status flags report the transaction result.
5. After all the data packets in a transfer are successful sent on USB bus, USBHS generates TF flag to indicate that the transfer successfully finishes and disable the IN endpoint.

IN transfers operation sequence with DMA enabled

1. Initialize USBHS global registers.
2. Initialize and enable the IN endpoint.
3. DMA in USBHS begins to fetch packets from the address specified by DMAADDR in USBHS_DIEPxDMAADDR register and write them into the IN endpoint's Tx FIFO. Each time a whole packet data is written into the FIFO, USBHS decreases the TLEN field in USBHS_DIEPxLEN register with the written packet's size.
4. When an IN token is received, USBHS transmit the data packet, and after the transaction finishes on USB bus, PCNT in USBHS_DIEPxLEN register is decreased by 1. If the transaction finishes successfully (ACK handshake received), the ACK flag is triggered. Otherwise, the status flags report the transaction result.
5. If all the transactions are successful sent on USB bus, USBHS generates TF and EPDIS flags to indicate that the transfer successfully finishes and the endpoint is disabled. If USB bus error or DMA fetch error occurs during these transactions, DMA will trigger related error flags.

Note: In DMA mode, software should not enable or process the RXFNEIF interrupt because the DMA will automatically process the Rx FIFO.

OUT transfers operation sequence with DMA disabled

1. Initialize USBHS global registers.
2. Initialize the endpoint and enable the endpoint.
3. When an OUT token is received, USBHS receive the data packet or response with an NAK handshake based on the status of Rx FIFO and register configuration. If the transaction finishes successfully (USBHS receives and saves the data packet into Rx FIFO successfully and sends ACK handshake on USB bus), PCNT in USBHS_DOEPxLEN register is decreased by 1 and the ACK flag is triggered, otherwise, the status flags report the transaction result.

4. After all the data packets in a transfer are successful received on USB bus, USBHS push a TF status entry into the Rx FIFO on top of the last packet data. After software reads and pops all the received data packet, and at last, the TF status entry, USBHS generates TF flag to indicate that the transfer successfully finishes and disable the OUT endpoint.

OUT transfers operation sequence with DMA enabled

1. Initialize USBHS global registers.
2. Initialize and enable the OUT endpoint.
3. When an OUT token received, USBHS receive the data packet or response with an NAK handshake based on the status of Rx FIFO and register configuration. If the transaction finishes successfully (USBHS receives and saves the data packet into Rx FIFO successfully and sends ACK handshake on USB bus), PCNT in USBHS_DOEPxLEN register is decreased by 1 and the ACK flag is triggered, otherwise, the status flags report the transaction result.
4. If all the transactions are successful received on USB bus, USBHS generates TF and EPDIS flags to indicate that the transfer successfully finishes and the endpoint is disabled. If USB bus error or DMA write error occurs during these transactions, DMA will trigger related error flags.

Note: In DMA mode, software should not enable or process the RXFNEIF interrupt because the DMA will automatically process the Rx FIFO.

53.6. Interrupts

USBHS has four interrupts: global interrupt, wake-up interrupt, endpoint1 IN interrupt and endpoint1 OUT interrupt.

Global interrupt is the main interrupt software should process, the source flags of the global interrupt are readable in USBHS_GINTF register and listed in the following ***Table 53-3. USBHS global interrupt.***

Table 53-3. USBHS global interrupt

Interrupt Flag	Description	Operation Mode
SESIF	Session interrupt	Host or device mode
DISCIF	Disconnect interrupt flag	Host Mode
IDPSC	ID pin status change	Host or device mode
LPMIF	LPM interrupt flag	Host or device mode
PTXFEIF	Periodic Tx FIFO empty interrupt flag	Host Mode
HCIF	Host channels interrupt flag	Host Mode
HPIF	Host port interrupt flag	Host Mode
ISOONCIF/PXNCIF	Periodic transfer Not Complete Interrupt flag / Isochronous OUT transfer Not Complete	Host or device mode

Interrupt Flag	Description	Operation Mode
	Interrupt Flag	
ISOINCIF	Isochronous IN transfer Not Complete Interrupt Flag	Device mode
OEPIF	OUT endpoint interrupt flag	Device mode
IEPIF	IN endpoint interrupt flag	Device mode
EOPFIF	End of periodic frame interrupt flag	Device mode
ISOOPDIF	Isochronous OUT packet dropped interrupt flag	Device mode
ENUMF	Enumeration finished	Device mode
RST	USB reset	Device mode
SP	USB suspend	Device mode
ESP	Early suspend	Device mode
GONAK	Global OUT NAK effective	Device mode
GNPINAK	Global IN Non-Periodic NAK effective	Device mode
NPTXFEIF	Non-Periodic Tx FIFO empty interrupt flag	Host Mode
RXFNEIF	Rx FIFO non-empty interrupt flag	Host or device mode
SOF	Start of frame	Host or device mode
OTGIF	OTG interrupt flag	Host or device mode
MFIF	Mode fault interrupt flag	Host or device mode

Wake up interrupt is able to be triggered when USBHS is in suspend state, even when the USBHS's clocks are stopped. The source of the wake up interrupt is WKUPIF bit in USBHS_GINTF register.

Endpoint 1 IN/OUT interrupts are two special interrupts for endpoint 1. Application can use these two interrupts to make a quick response to the events on endpoint 1. The two interrupts are individually enabled by USBHS_DEP1INT register. And the source of these two interrupts also come from USBHS_DIEP1INTF and USBHS_DOEP1INTF registers, but the enable bits for these flags to generate Endpoint 1 IN/OUT interrupts are in USBHS_DIEP1INTEN and USBHS_DOEP1INTEN registers.

53.7. Register definition

USBHS0 base address: 0x4004 0000

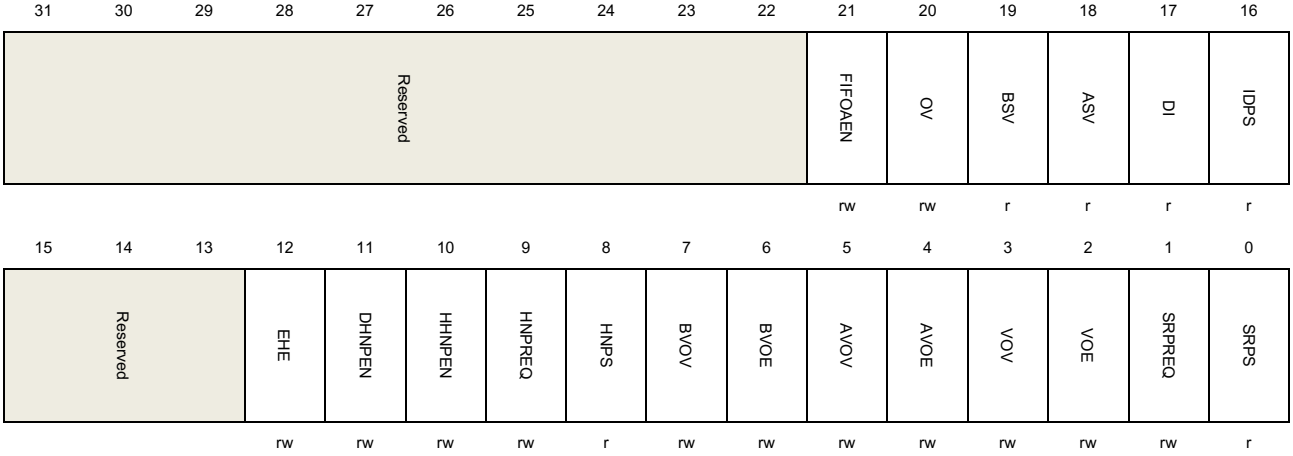
USBHS1 base address: 0x4008 0000

53.7.1. Global registers

Global OTG control and status register (USBHS_GOTGCS)

Address offset: 0x0000
Reset value: 0x0000 0800

This register has to be accessed by word (32-bit)



Bits	Fields	Descriptions
31:22	Reserved	Must be kept at reset value.
21	FIFOAEN	AHB bus access 4KB FIFO enable 0: 4KB FIFO is internally accessed by USBHS 1: 4KB FIFO is externally accessed by AHB bus
20	OV	Select OTG version 0: Version 1.3 is selected. Data line pulsing and VBUS pulsing are supported in SRP 1: Version 2.0 is selected. Only data line pulsing is supported in SRP
19	BSV	B-Session Valid (described in OTG protocol). 0: Vbus voltage level of a OTG B-device is below VBSESSVLD 1: Vbus voltage level of a OTG B-Device is not below VBSESSVLD **Note:** Only accessible in OTG B-Device mode.
18	ASV	10.1.21 session valid 10.1.22 host mode transceiver status. 0: Vbus voltage level of a OTG A-device is below VASESSVLD 1: Vbus voltage level of a OTG A-device is not below VASESSVLD

		The A-device is as host default at the start of a session. **Note:** Only accessible in OTG A-Device mode.
17	DI	Debounce interval Debounce interval of a detected connection. 0: Indicates the long debounce interval, when a plug-on and connection occur on USB bus 1: Indicates the short debounce interval, when a soft connection is used in HNP protocol. **Note:** Only accessible in host mode.
16	IDPS	ID pin status Voltage level of connector ID pin 0: USBHS is in A-device mode 1: USBHS is in B-device mode **Note:** Accessible in both device and host modes.
15:13	Reserved	Must be kept at reset value
12	EHE	Embedded host enable 0: OTG A-device state is selected 1: Embedded host state is selected
11	DHNPEN	Device HNP enable Enable the HNP function of a B-device. If this bit is cleared, USBHS doesn't start HNP protocol when application set HNPREQ bit in USBHS_GOTGCS register. 0: HNP function is not enabled. 1: HNP function is enabled **Note:** Only accessible in device mode.
10	HHNPEN	Host HNP enable Enable the HNP function of an A-device. If this bit is cleared, USBHS doesn't response to the HNP request from B-device. 0: HNP function is not enabled. 1: HNP function is enabled **Note:** Only accessible in host mode.
9	HNPREQ	HNP request This bit is set by software to start a HNP on the USB. Software can clear this bit when HNPEND bit in USBHS_GOTGINTF register is set, by writing zero to it, or clearing the HNPEND bit in USBHS_GOTGINTF register. 0: Don't send HNP request 1: Send HNP request **Note:** Only accessible in device mode.
8	HNPS	HNP successes This bit is set by the core when HNP successes and cleared when HNPREQ bit is set.

		0: HNP fails 1: HNP successes **Note:** Only accessible in device mode.
7	BVOV	Override value of B-peripheral session valid 0: B-peripheral session valid value is 0 when BVOE = 1 1: B-peripheral session valid value is 1 when BVOE = 1 **Note:** Only accessible in device mode.
6	BVOE	Override enable of B-peripheral session valid 0: Override is disable. Internally B-peripheral session valid received from PHY is selected 1: Override is enable. Internally B-peripheral session valid received from PHY is overridden with BVOV **Note:** Only accessible in device mode.
5	AVOV	Override value of A-peripheral session valid 0: A-peripheral session valid value is 0 when AVOE = 1 1: A-peripheral session valid value is 1 when AVOE = 1 **Note:** Only accessible in host mode.
4	AVOE	Override enable of A-peripheral session valid 0: Override is disable. Internally A-peripheral session valid received from PHY is selected 1: Override is enable. Internally A-peripheral session valid received from PHY is overridden with AVOV **Note:** Only accessible in host mode.
3	VOV	Override value of VBUS valid 0: VBUS valid value is 0 when VOE = 1 1: VBUS valid value is 1 when VOE = 1 **Note:** Only accessible in host mode.
2	VOE	Override enable of VBUS valid 0: Override is disable. Internally VBUS valid received from PHY is selected 1: Override is enable. Internally VBUS valid received from PHY is overridden with VOV **Note:** Only accessible in host mode.
1	SRPREQ	SRP request This bit is set by software to start a SRP on the USB. Software can clear this bit when SRPEND bit in USBHS_GOTGINTF register is set, by writing zero to it, or clearing the SRPEND bit in USBHS_GOTGINTF register. 0: No session request 1: Session request **Note:** Only accessible in device mode.
0	SRPS	SRP success

This bit is set by the core when SRP successes and cleared when SRPREQ bit is set.

0: SRP fails

1: SRP successes

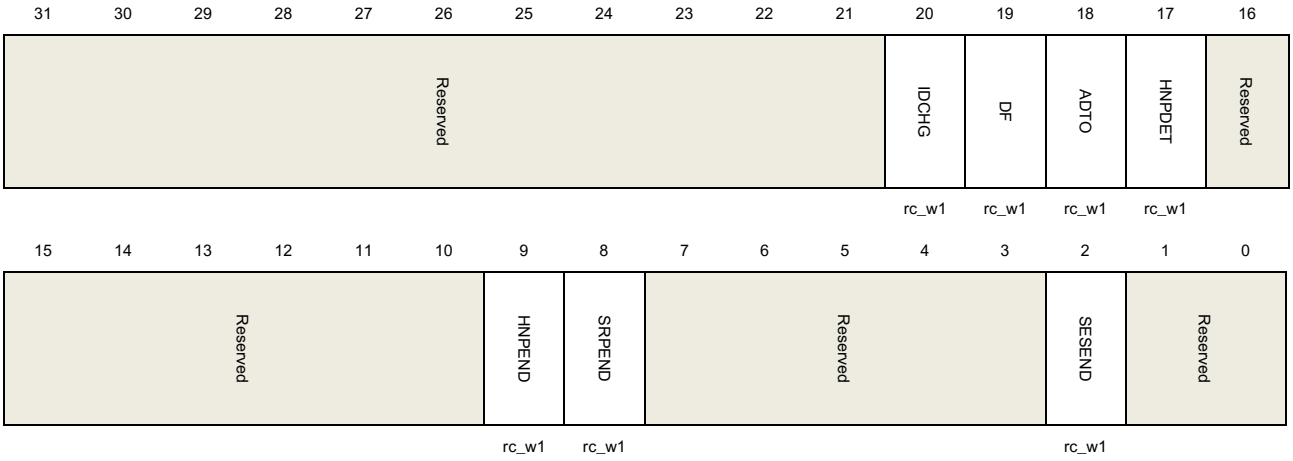
Note: Only accessible in device mode.

Global OTG interrupt flag register (USBHS_GOTGINTF)

Address offset: 0x0004

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)



Bits	Fields	Descriptions
31:21	Reserved	Must be kept at reset value.
20	IDCHG	There is a change in the value of ID input
19	DF	Debounce finish Set by USBHS when the debounce during device connection is done. **Note:** Only accessible in host mode.
18	ADTO	10.1.23 device timeout Set by USBHS when the A-device's waiting for a B-device' connection has timed out. **Note:** Accessible in both device and host modes.
17	HNPDET	Host negotiation request detected Set by USBHS when A-device detects a HNP request. **Note:** Accessible in both device and host modes.
16:10	Reserved	Must be kept at reset value.
9	HNPEND	HNP end Set by the core when a HNP ends. Software should read the HNPS in USBHS_GOTGCS register to get the result of HNP.

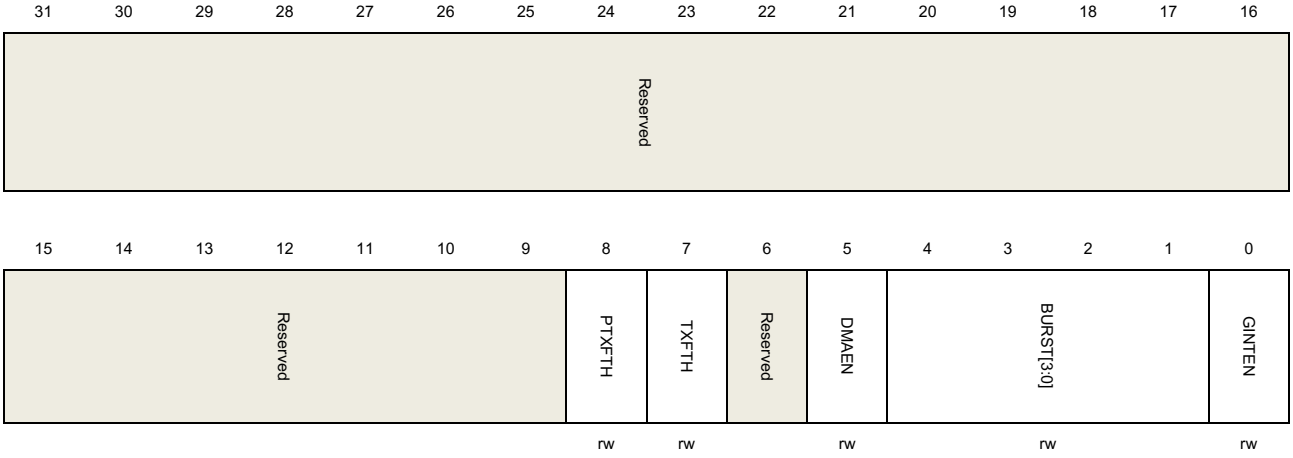
		Note: Accessible in both device and host modes.
8	SRPEND	SRPEND Set by the core when a SRP ends. Software should read the SRPS in USBHS_GOTGCS register to get the result of SRP. **Note:** Accessible in both device and host modes.
7:3	Reserved	Must be kept at reset value.
2	SESEND	Session end Set by the core when VBUS voltage is below Vb_ses_vld.
1:0	Reserved	Must be kept at reset value.

Global AHB control and status register (USBHS_GAHBCS)

Address offset: 0x0008

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)



Bits	Fields	Descriptions
31:9	Reserved	Must be kept at reset value.
8	PTXFTH	Periodic Tx FIFO threshold 0: PTXFEIF will be triggered when the periodic transmit FIFO is half empty 1: PTXFEIF will be triggered when the periodic transmit FIFO is completely empty **Note:** Only accessible in host mode.
7	TXFTH	Tx FIFO threshold **Device mode:** 0: TXFEIF will be triggered when the IN endpoint transmit FIFO is half empty 1: TXFEIF will be triggered when the IN endpoint transmit FIFO is completely empty **Host mode:** 0: NPTXFEIF will be triggered when the non-periodic transmit FIFO is half empty 1: NPTXFEIF will be triggered when the non-periodic transmit FIFO is completely

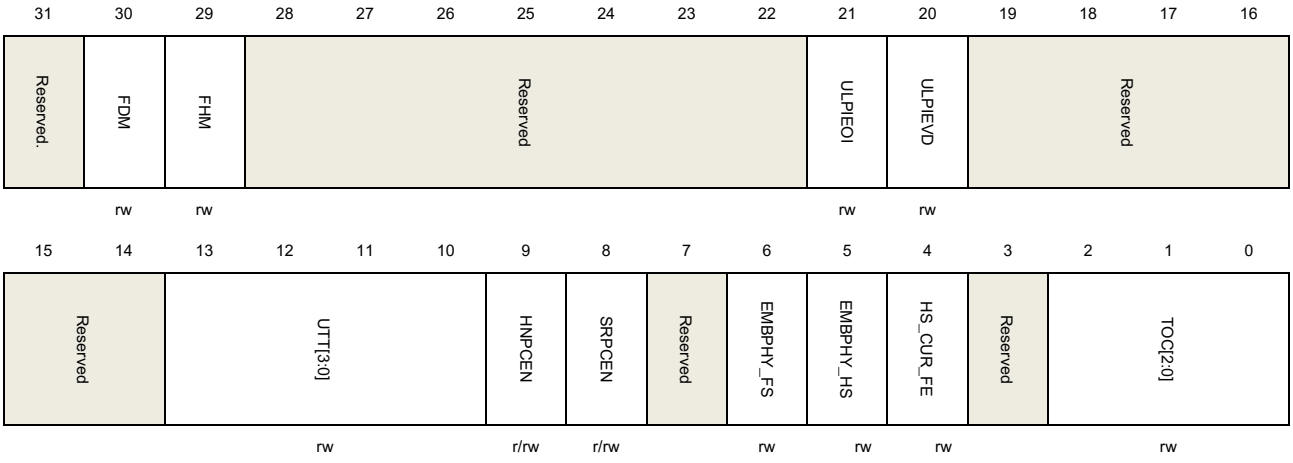
		empty
6	Reserved	Must be kept at reset value.
5	DMAEN	DMA function Enable 0: DMA function is disabled 1: DMA function is enabled
4:1	BURST[3:0]	The AHB burst type used by DMA 0000: Single 0001: INCR 0011: INCR4 0101: INCR8 0111: INCR16
0	GINTEN	Global interrupt enable 0: Global interrupt is not enabled. 1: Global interrupt is enabled. **Note:** Accessible in both device and host modes.

Global USB control and status register (USBHS_GUSBCS)

Address offset: 0x000C

Reset value: 0x0000 1400

This register has to be accessed by word (32-bit)



Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30	FDM	Force device mode Setting this bit will force the core to device mode irrespective of the USBHS ID input pin. 0: Normal mode 1: Device mode

		The application must wait at least 25 ms for the change taking effect after setting the force bit. **Note:** Accessible in both device and host modes.
29	FHM	Force host mode Setting this bit will force the core to host mode irrespective of the USBHS ID input pin. 0: Normal mode 1: Host mode The application must wait at least 25 ms for the change taking effect after setting the force bit. **Note:** Accessible in both device and host modes.
28:22	Reserved	Must be kept at reset value.
21	ULPIEOI	ULPI external over-current indicator ULPI PHY uses this bit to decide whether to use internal or external over-current indicator. This bit only takes effect when external ULPI PHY is used (EMBPHY_FS and EMBPHY_HS bits in this register are both 0). 0: ULPI PHY uses internal over-current indicator 1: ULPI PHY uses external over-current indicator
20	ULPIEVD	ULPI external VBUS driver ULPI PHY uses this bit to decide whether VBUS is driven by ULPI PHY or by external power supply. This bit only takes effect when external ULPI PHY is used (EMBPHY_FS and EMBPHY_HS bits in this register are both 0). 0: VBUS is driven by ULPI PHY 1: VBUS is driven by external power supply
19:14	Reserved	Must be kept at reset value.
13:10	UTT[3:0]	USB turnaround time Turnaround time in PHY clocks. **Note:** Only accessible in device mode.
9	HNPCEN	HNP capability enable Controls whether the HNP capability is enabled 0: HNP capability is disabled 1: HNP capability is enabled **Note:** Accessible in both device and host modes.
8	SRPCEN	SRP capability enable Controls whether the SRP capability is enabled 0: SRP capability is disabled 1: SRP capability is enabled **Note:** Accessible in both device and host modes.
7	Reserved	Must be kept at reset value.

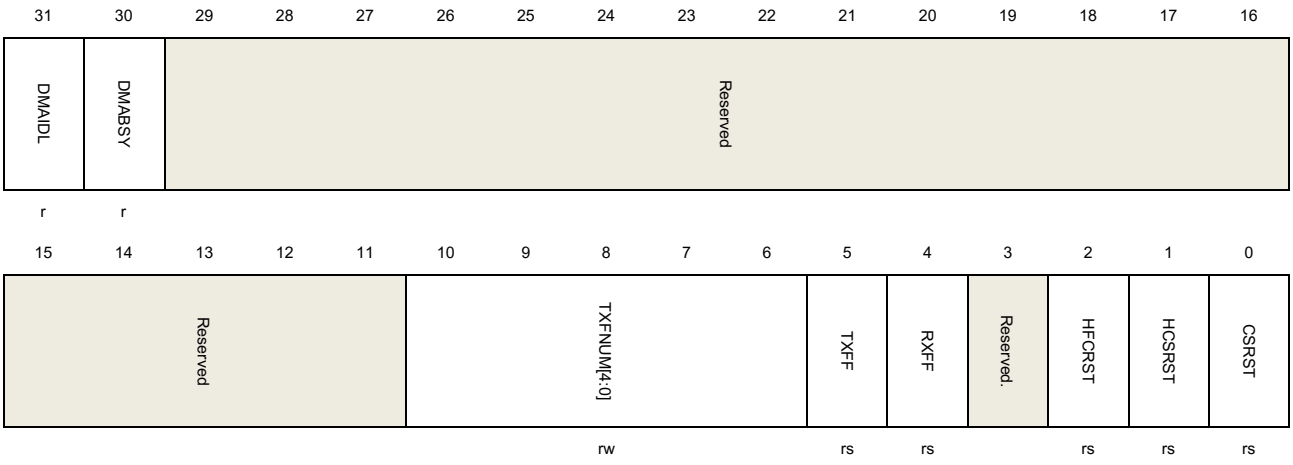
6	EMBPHY_FS	Embedded FS PHY selected 0: Embedded FS PHY is disabled 1: Embedded FS PHY is enabled **Note:** This bit can only be set when EMBPHY_HS is set to 0. Accessible in both device and host modes.
5	EMBPHY_HS	Embedded HS PHY selected 0: Embedded HS PHY is disabled 1: Embedded HS PHY is enabled **Note:** This bit can only be set when EMBPHY_FS is set to 0. Accessible in both device and host modes.
4	HS_CUR_FE	HS current software enable 0: Release the HS mode TX current enable 1: Force HS mode TX current enable
3	Reserved	Must be kept at reset value.
2:0	TOC[2:0]	Timeout calibration USBHS always uses time-out value required in USB 2.0 when waiting for a packet. Application may use TOC[2:0] to add the value is in terms of PHY clock. (The frequency of PHY clock is decided by which PHY is used: 48MHZ with internal embedded PHY and 60MHz with external ULPI PHY.)

Global reset control register (USBHS_GRSTCTL)

Address offset: 0x0010
Reset value: 0x8000 0000

The application uses this register to reset various hardware features inside the core.

This register has to be accessed by word (32-bit)



Bits	Fields	Descriptions
31	DMAIDL	DMA Idle state

		This bit reports that whether DMA is in IDLE state or not. 0: DMA is not IDLE 1: DMA is IDLE **Note:** Accessible in both device and host modes.
30	DMABSY	DMA Busy This bit reports that whether DMA is busy. 0: DMA is not busy 1: DMA is busy **Note:** Accessible in both device and host modes.
29:11	Reserved	Must be kept at reset value.
10:6	TXFNUM[4:0]	Tx FIFO number Indicates which Tx FIFO will be flushed when TXFF bit in the same register is set. Host Mode: 00000: Only non-periodic Tx FIFO is flushed 00001: Only periodic Tx FIFO is flushed 1XXXX: Both periodic and non-periodic Tx FIFOs are flushed Other: Non data FIFO is flushed Device Mode: 00000: Only Tx FIFO0 is flushed 00001: Only Tx FIFO1 is flushed … 00111: Only Tx FIFO7 is flushed 1XXXX: All Tx FIFOs are flushed Other: Non data FIFO is flushed
5	TXFF	Tx FIFO flush Application sets this bit to flush data Tx FIFOs and TXFNUM[4:0] bits decide the FIFO number to be flushed. Hardware automatically clears this bit after the flush process completes. After setting this bit, application should wait until this bit is cleared before any other operation on USBHS. **Note:** Accessible in both device and host modes.
4	RXFF	Rx FIFO flush Application sets this bit to flush data Rx FIFO. Hardware automatically clears this bit after the flush process completes. After setting this bit, application should wait until this bit is cleared before any other operation on USBHS. **Note:** Accessible in both device and host modes.
3	Reserved	Must be kept at reset value.
2	HFCRST	Host frame counter reset Set by the application to reset the frame number counter in USBHS. After this bit is set, the frame number of the following SOF returns to 0. Hardware automatically clears this bit after the reset process completes. After setting this bit, application

		should wait until this bit is cleared before any other operation on USBHS. **Note:** Only accessible in host mode.
1	HCSRST	HCLK soft reset Set by the application to reset AHB clock domain circuit. Hardware automatically clears this bit after the reset process completes. After setting this bit, application should wait until this bit is cleared before any other operation on USBHS. **Note:** Accessible in both device and host modes.
0	CSRST	Core soft reset Resets the AHB and USB clock domains circuits, as well as most of the registers.

Global interrupt flag register (USBHS_GINTF)

Address offset: 0x0014

Reset value: 0x0400 0021

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WKUPIF	SESIF	DISCIF	IDPSC	LPMIF	PTXFEIF	HCIF	HPIF	Reserved		PXNCIF/ ISOONCIF	ISOINCIF	OEPIF	IEPIF	Reserved	
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	r	r	r			rc_w1	rc_w1	r	r		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EOPFIF	ISOOPDIF	ENUMF	RST	SP	ESP	Reserved		GONAK	GNPINAK	NPTXFEIF	RXFNEIF	SOF	OTGIF	MFIF	COPM
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1			r	r	r	r	rc_w1	r	rc_w1	r

Bits	Fields	Descriptions
31	WKUPIF	Wakeup interrupt flag This interrupt is triggered when a resume signal (in device mode) or a remote wakeup signal (in host mode) is detected on the USB. **Note:** Accessible in both device and host modes.
30	SESIF	Session interrupt flag This interrupt is triggered when a SRP is detected (in A-Device mode) or V_{BUS} becomes valid for a B- device (in B-Device mode). **Note:** Accessible in both device and host modes.
29	DISCIF	Disconnect interrupt flag This interrupt is triggered after a device disconnection. **Note:** Only accessible in host mode.

28	IDPSC	ID pin status change Set by the core when ID status changes. **Note:** Accessible in both device and host modes.
27	LPMIF	LPM interrupt flag In host mode, when device responds to LPM transaction with ACK, NYET or STALL, or when host has sent RECNT (USBHS_LPMCFG register) times LPM transaction, the interrupt is triggered. In device mode, when device has received LPM transaction and responds with ACK, NYET or STALL, the interrupt is triggered.
26	PTXFEIF	Periodic Tx FIFO empty interrupt flag This interrupt is triggered when the periodic transmit FIFO is either half or completely empty. The threshold is determined by the periodic Tx FIFO empty level bit (PTXFTH) in the USBHS_GAHBCS register. **Note:** Only accessible in host mode.
25	HCIF	Host channels interrupt flag Set by USBHS when one of the channels in host mode has raised an interrupt. Software should first read USBHS_HACHINT register to get the channel number, and then read the corresponding USBHS_HCHxINTF register to get the flags of the channel that cause the interrupt. This bit will be automatically cleared after the respective channel's flags which cause channel interrupt are cleared. **Note:** Only accessible in host mode.
24	HPIF	Host port interrupt flag Set by the core when USBHS detects that port status changes in host mode. Software should read USBHS_HPCS register to get the source of this interrupt. This bit will be automatically cleared after the flags that causing a port interrupt are cleared. **Note:** Only accessible in host mode.
23:22	Reserved	Must be kept at reset value.
21	PXNCIF	Periodic transfer Not Complete Interrupt flag USBHS sets this bit when there are periodic transactions for current frame not completed at the end of frame. (Host mode)
	ISOONCIF	Isochronous OUT transfer Not Complete Interrupt Flag At the end of a periodic frame (defined by EOPFT bit in USBHS_DCFG), USBHS will set this bit if there are still isochronous OUT endpoints that not completed transactions. (Device Mode)
20	ISOINCIF	Isochronous IN transfer Not Complete Interrupt Flag At the end of a periodic frame (defined by EOPFT[1:0] bits in USBHS_DCFG), USBHS will set this bit if there are still isochronous IN endpoints that not completed transactions. (Device Mode) **Note:** Only accessible in device mode.

19	OEPIF	OUT endpoint interrupt flag Set by USBHS when one of the OUT endpoints in device mode has raised an interrupt. Software should first read USBHS_DAEPINT register to get the device number, and then read the corresponding USBHS_DOEPxINTF register to get the flags of the endpoint that cause the interrupt. This bit will be automatically cleared after the respective endpoint's flags which cause this interrupt are cleared. **Note:** Only accessible in device mode.
18	IEPIF	IN endpoint interrupt flag Set by USBHS when one of the IN endpoints in device mode has raised an interrupt. Software should first read USBHS_DAEPINT register to get the device number, and then read the corresponding USBHS_DIEPxINTF register to get the flags of the endpoint that cause the interrupt. This bit will be automatically cleared after the respective endpoint's flags which cause this interrupt are cleared. **Note:** Only accessible in device mode.
17:16	Reserved	Must be kept at reset value.
15	EOPFIF	End of periodic frame interrupt flag When USB bus time in a frame has reaches the value defined by EOPFT[1:0] bits in USBHS_DCFG register, USBHS sets this flag. **Note:** Only accessible in device mode.
14	ISOOPDIF	Isochronous OUT packet dropped interrupt flag USBHS set this bit if it receives an isochronous OUT packet but cannot save it into Rx FIFO because the FIFO doesn't have enough space. **Note:** Only accessible in device mode.
13	ENUMF	Enumeration finished USBHS sets this bit after the speed enumeration finishes. Software is able to read USBHS_DSTAT register to get the current device speed. **Note:** Only accessible in device mode.
12	RST	USB reset USBHS sets this bit when it detects a USB reset signal on bus. **Note:** Only accessible in device mode.
11	SP	USB suspend USBHS sets this bit when it detects that the USB bus is idle for 3 ms and enters suspend state. **Note:** Only accessible in device mode.
10	ESP	Early suspend USBHS sets this bit when it detects that the USB bus is idle for 3 ms. **Note:** Only accessible in device mode.
9:8	Reserved	Must be kept at reset value.
7	GONAK	Global OUT NAK effective

		Software is able to write 1 to SGONAK bit in the USBHS_DCTL register and USBHS will set GONAK flag after writing to SGONAK takes effect. And this bit can be cleared by writing 1 to CGONAK bit in the USBHS_DCTL register. **Note:** Only accessible in device mode.
6	GNPINAK	Global IN Non-Periodic NAK effective Software is able to write 1 to SGINAK bit in the USBHS_DCTL register and USBHS will set GNPINAK flag after writing to SGINAK takes effect. And this bit can be cleared by writing 1 to CGINAK bit in the USBHS_DCTL register. **Note:** Only accessible in device mode.
5	NPTXFEIF	Non-Periodic Tx FIFO empty interrupt flag This interrupt is triggered when the non-periodic transmit FIFO is either half or completely empty. The threshold is determined by the non-periodic Tx FIFO empty level bit (TXFTH) in the USBHS_GAHBCS register. **Note:** Only accessible in host mode.
4	RXFNEIF	Rx FIFO non-empty interrupt flag USBHS sets this bit when there is at least one packet or status entry in the Rx FIFO. **Note:** Accessible in both host and device modes.
3	SOF	Start of frame Host Mode: USBHS sets this bit when it prepares to transmit a SOF or Keep-Alive on USB bus. Software can clear this bit by writing 1. Device Mode: USBHS sets this bit to after it receives a SOF token. The application can read the Device Status register to get the current frame number. Software can clear this bit by writing 1. **Note:** Accessible in both host and device modes.
2	OTGIF	OTG interrupt flag USBHS sets this bit when the flags in USBHS_GOTGINTF register generate a interrupt. Software should read USBHS_GOTGINTF register to get the source of this interrupt. This bit is cleared after the flags in USBHS_GOTGINTF causing this interrupt are cleared. **Note:** Accessible in both host and device modes.
1	MFIF	Mode fault interrupt flag USBHS sets this bit if software operates host-only register in device mode, or operates device-mode in host mode. These fault operations won't take effect. **Note:** Accessible in both host and device modes.
0	COPM	Current operation mode 0: Device mode 1: Host mode **Note:** Accessible in both host and device modes.

Global interrupt enable register (USBHS_GINTEN)

Address offset: 0x0018

Reset value: 0x0000 0000

This register works with the global interrupt flag register (USBHS_GINTF) to interrupt the application. When an interrupt enable bit is disabled, the interrupt associated with that bit is not generated. However, the global Interrupt flag register bit corresponding to that interrupt is still set.

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WKUPIE	SESIE	DISCIE	IDPSCIE	LPMIE	PTXFEIE	HCIE	HPIE	Reserved		PXNCIE/ ISOONCIE	ISOINCIE	OEPIE	IEPIE	Reserved	
rw	rw	rw	rw	rw	rw	rw	r			rw	rw	rw	rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EOPFIE	ISOOPDIE	ENUMFIE	RSTIE	SPIE	ESPIE	Reserved		GONAKIE	GNPINAKIE	NPTXFEIE	RXFNEIE	SOFIE	OTGIE	MFIE	Reserved
rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw	rw	

Bits	Fields	Descriptions
31	WKUPIE	Wakeup interrupt enable 0: Disable wakeup interrupt 1: Enable wakeup interrupt **Note:** Accessible in both host and device modes.
30	SESIE	Session interrupt enable 0: Disable session interrupt 1: Enable session interrupt **Note:** Accessible in both host and device modes.
29	DISCIE	Disconnect interrupt enable 0: Disable disconnect interrupt 1: Enable disconnect interrupt **Note:** Only accessible in device mode.
28	IDPSCIE	ID pin status change interrupt enable 0: Disable connector ID pin status interrupt 1: Enable connector ID pin status interrupt **Note:** Accessible in both host and device modes.
27	LPMIE	LPM interrupt enable 0: disable LPM interrupt 1: enable LPM interrupt

		Note: Accessible in both host and device modes.
26	PTXFEIE	Periodic Tx FIFO empty interrupt enable 0: Disable periodic Tx FIFO empty interrupt 1: Enable periodic Tx FIFO empty interrupt **Note:** Only accessible in host mode.
25	HCIE	Host channels interrupt enable 0: Disable host channels interrupt 1: Enable host channels interrupt **Note:** Only accessible in host mode.
24	HPIE	Host port interrupt enable 0: Disable host port interrupt 1: Enable host port interrupt **Note:** Only accessible in host mode.
23:22	Reserved	Must be kept at reset value.
21	PXNCIE	Periodic transfer not complete interrupt enable 0: Disable Periodic transfer not complete interrupt 1: Enable Periodic transfer not complete interrupt **Note:** Only accessible in host mode.
	ISOONCIE	Isochronous OUT transfer not complete interrupt enable 0: Disable Isochronous OUT transfer not complete interrupt 1: Enable Isochronous OUT transfer not complete interrupt **Note:** Only accessible in device mode.
20	ISOINCIE	Isochronous IN transfer not complete interrupt enable 0: Disable Isochronous IN transfer not complete interrupt 1: Enable Isochronous IN transfer not complete interrupt **Note:** Only accessible in device mode.
19	OEPIE	OUT endpoints interrupt enable 0: Disable OUT endpoints interrupt 1: Enable OUT endpoints interrupt **Note:** Only accessible in device mode.
18	IEPIE	IN endpoints interrupt enable 0: Disable IN endpoints interrupt 1: Enable IN endpoints interrupt **Note:** Only accessible in device mode.
17:16	Reserved	Must be kept at reset value.
15	EOPFIE	End of periodic frame interrupt enable 0: Disable end of periodic frame interrupt 1: Enable end of periodic frame interrupt

		Note: Only accessible in device mode.
14	ISOOPDIE	Isochronous OUT packet dropped interrupt enable 0: Disable isochronous OUT packet dropped interrupt 1: Enable isochronous OUT packet dropped interrupt **Note:** Only accessible in device mode.
13	ENUMFIE	Enumeration finish enable 0: Disable enumeration finish interrupt 1: Enable enumeration finish interrupt **Note:** Only accessible in device mode.
12	RSTIE	USB reset interrupt enable 0: Disable USB reset interrupt 1: Enable USB reset interrupt **Note:** Only accessible in device mode.
11	SPIE	USB suspend interrupt enable 0: Disable USB suspend interrupt 1: Enable USB suspend interrupt **Note:** Only accessible in device mode.
10	ESPIE	Early suspend interrupt enable 0: Disable early suspend interrupt 1: Enable early suspend interrupt **Note:** Only accessible in device mode.
9:8	Reserved	Must be kept at reset value.
7	GONAKIE	Global OUT NAK effective interrupt enable 0: Disable global OUT NAK interrupt 1: Enable global OUT NAK interrupt **Note:** Only accessible in device mode.
6	GNPINAKIE	Global non-periodic IN NAK effective interrupt enable 0: Disable global non-periodic IN NAK effective interrupt 1: Enable global non-periodic IN NAK effective interrupt **Note:** Only accessible in device mode.
5	NPTXFEIE	Non-periodic Tx FIFO empty interrupt enable 0: Disable non-periodic Tx FIFO empty interrupt 1: Enable non-periodic Tx FIFO empty interrupt **Note:** Only accessible in Host mode.
4	RXFNEIE	Receive FIFO non-empty interrupt enable 0: Disable receive FIFO non-empty interrupt 1: Enable receive FIFO non-empty interrupt **Note:** Accessible in both device and host modes.

3	SOFIE	Start of frame interrupt enable 0: Disable start of frame interrupt 1: Enable start of frame interrupt **Note:** Accessible in both device and host modes.
2	OTGIE	OTG interrupt enable 0: Disable OTG interrupt 1: Enable OTG interrupt **Note:** Accessible in both device and host modes.
1	MFIE	Mode fault interrupt enable 0: Disable mode fault interrupt 1: Enable mode fault interrupt **Note:** Accessible in both device and host modes.
0	Reserved	Must be kept at reset value.

Global receive status read/receive status read and pop registers (USBHS_GRSTATR/USBHS_GRSTATP)

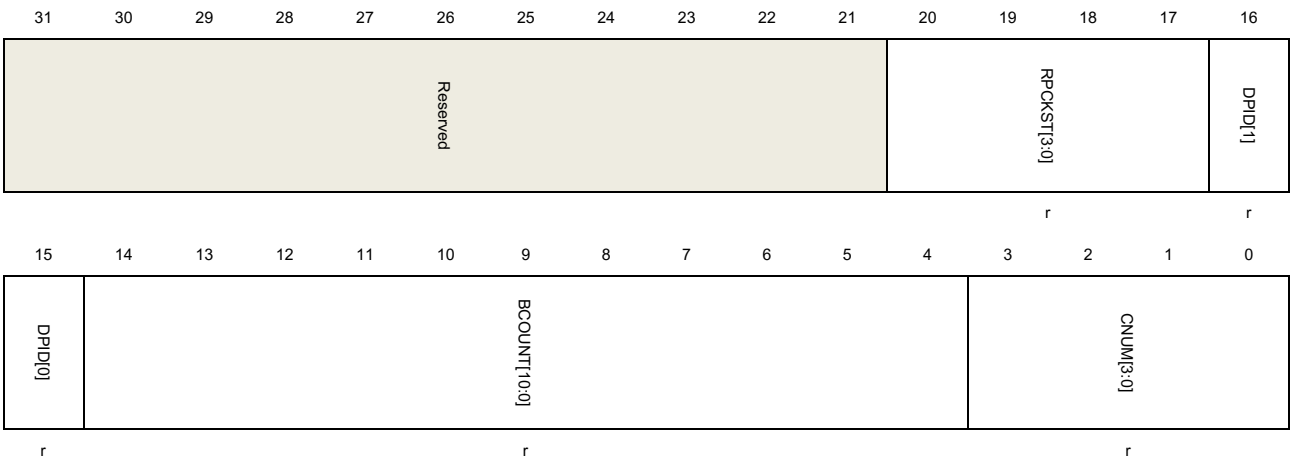
Address offset for Read: 0x001C
Address offset for Pop: 0x0020
Reset value: 0x0000 0000

A read to the receive status read register returns the entry of the top of the Rx FIFO. A read to the Receive status read and pop register additionally pops the top entry out of the Rx FIFO.

The entries in RxFIFO have different meanings in host and device modes. Software should only read this register after when Receive FIFO non-empty interrupt flag bit of the global interrupt flag register (RXFNEIF bit in USBHS_GINTF) is triggered.

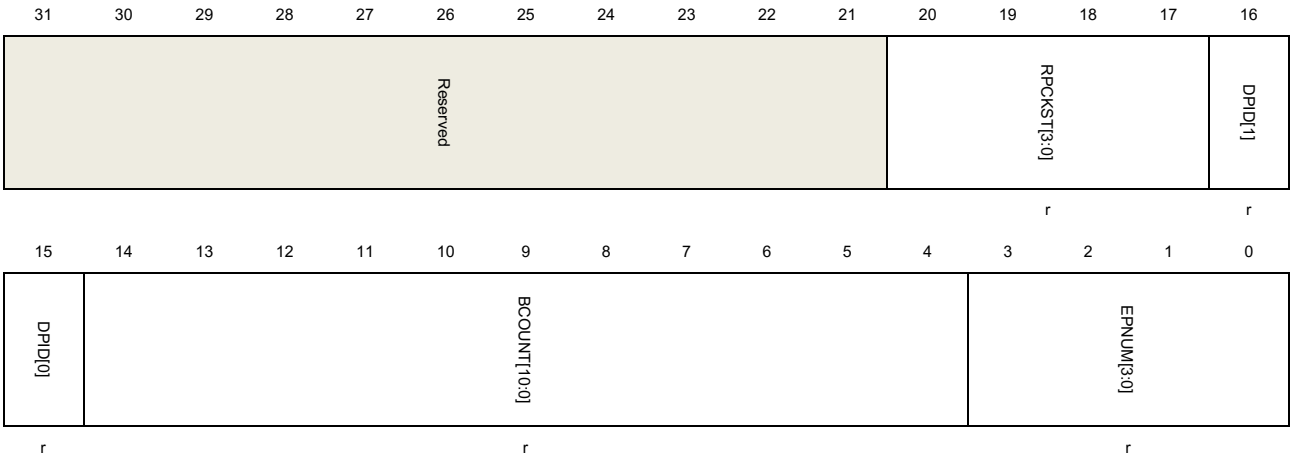
This register has to be accessed by word (32-bit)

Host mode:



Bits	Fields	Descriptions
31:21	Reserved	Must be kept at reset value.
20:17	RPCKST[3:0]	Received packet status 0010: IN data packet received 0011: IN transfer completed (generates an interrupt if poped) 0101: Data toggle error (generates an interrupt if poped) 0111: Channel halted (generates an interrupt if poped) Others: Reserved
16:15	DPID[1:0]	Data PID The Data PID of the received packet 00: DATA0 10: DATA1 01: DATA2 11: MDATA
14:4	BCOUNT[10:0]	Byte count The byte count of the received IN data packet.
3:0	CNUM[3:0]	Channel number The channel number to which the current received packet belongs.

Device mode:



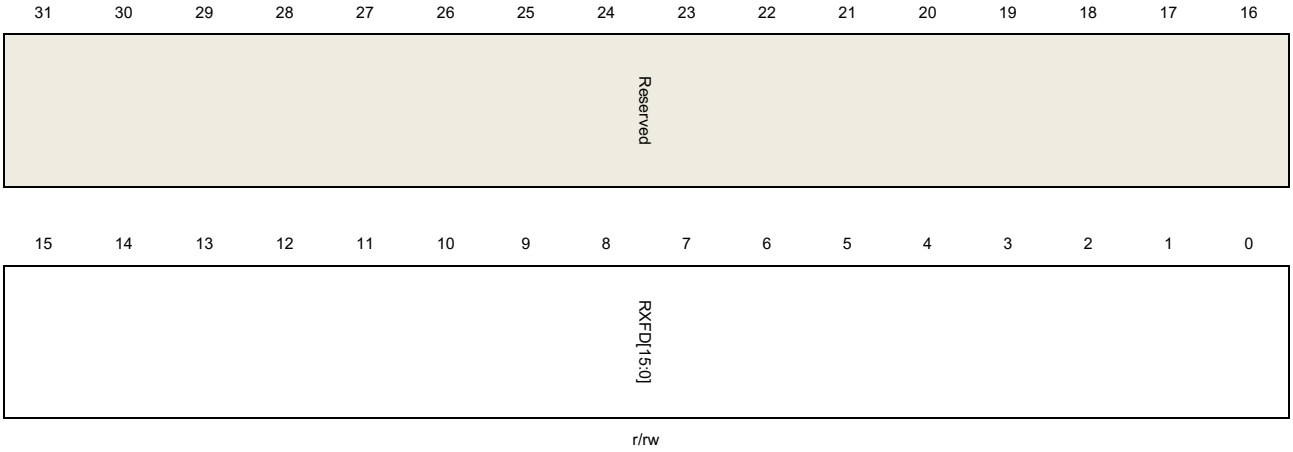
Bits	Fields	Descriptions
31:21	Reserved	Must be kept at reset value.
20:17	RPCKST[3:0]	Received packet status 0001: Global OUT NAK (generates an interrupt) 0010: OUT data packet received 0011: OUT transfer completed (generates an interrupt) 0100: SETUP transaction completed (generates an interrupt)

		0110: SETUP data packet received Others: Reserved
16:15	DPID[1:0]	Data PID The Data PID of the received OUT data packet 00: DATA0 10: DATA1 01: DATA2 11: MDATA
14:4	BCOUNT[10:0]	Byte count The byte count of the received data packet.
3:0	EPNUM[3:0]	Endpoint number The endpoint number to which the current received packet belongs.

Global receive FIFO length register (USBHS_GRFLEN)

Address offset: 0x024
Reset value: 0x0000 0200

This register has to be accessed by word (32-bit)

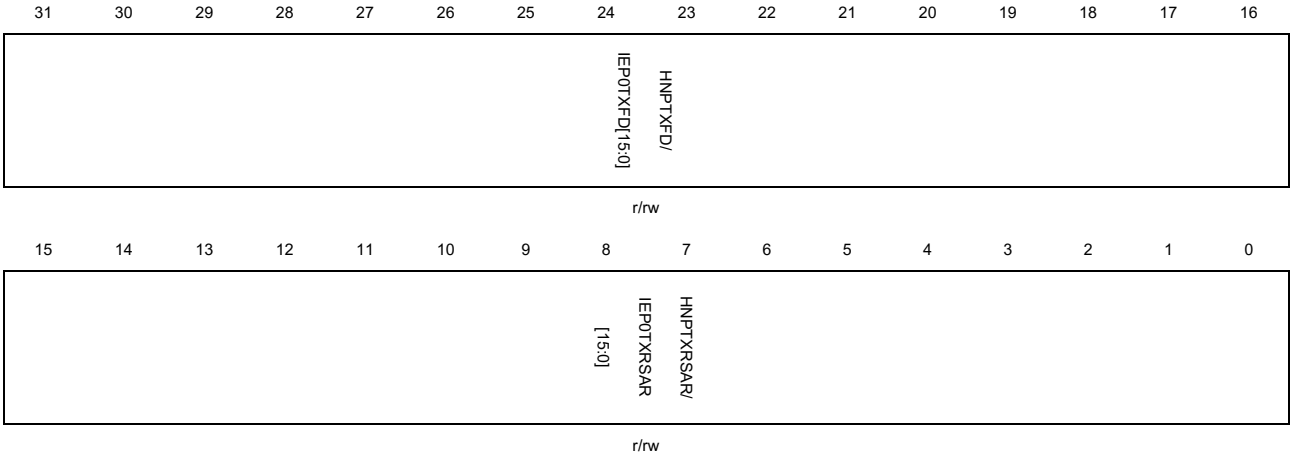


Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	RXFD[15:0]	Rx FIFO depth In terms of 32-bit word. 1<=RXFD<=1024

Host non-periodic transmit FIFO length register /Device IN endpoint 0 transmit FIFO length (USBHS_HNPTFLEN/USBHS_DIEP0TFLEN)

Address offset: 0x028
Reset value: 0x0200 0200

This register has to be accessed by word (32-bit)



Host Mode:

Bits	Fields	Descriptions
31:16	HNPTXFD[15:0]	Non-periodic Tx FIFO depth In terms of 32-bit word. 1<=HNPTXFD<=1024
15:0	HNPTXRSAR[15:0]	Non-periodic Tx RAM start address The start address for non-periodic transmit FIFO RAM in terms of 32-bit word.

Device Mode:

Bits	Fields	Descriptions
31:16	IEP0TXFD[15:0]	IN Endpoint 0 Tx FIFO depth In terms of 32-bit words. 16<=IEP0TXFD<=140
15:0	IEP0TXRSAR[15:0]	IN Endpoint 0 TX RAM start address The start address for endpoint0 transmit FIFO RAM in terms of 32-bit word.

Host non-periodic transmit FIFO/queue status register (USBHS_HNPTFQSTAT)

Address offset: 0x002C

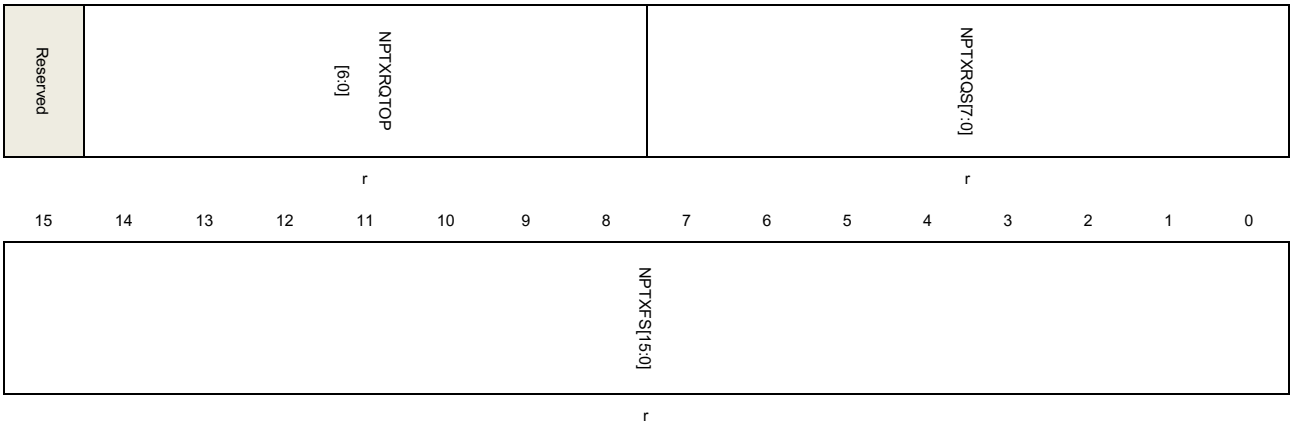
Reset value: 0x0008 0200

This register has to be accessed by word (32-bit)

This register reports the current status of the non-periodic Tx FIFO and request queue. The request queue holds IN, OUT or other request entries in host mode.

Note: In Device mode, this register is not valid.

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16



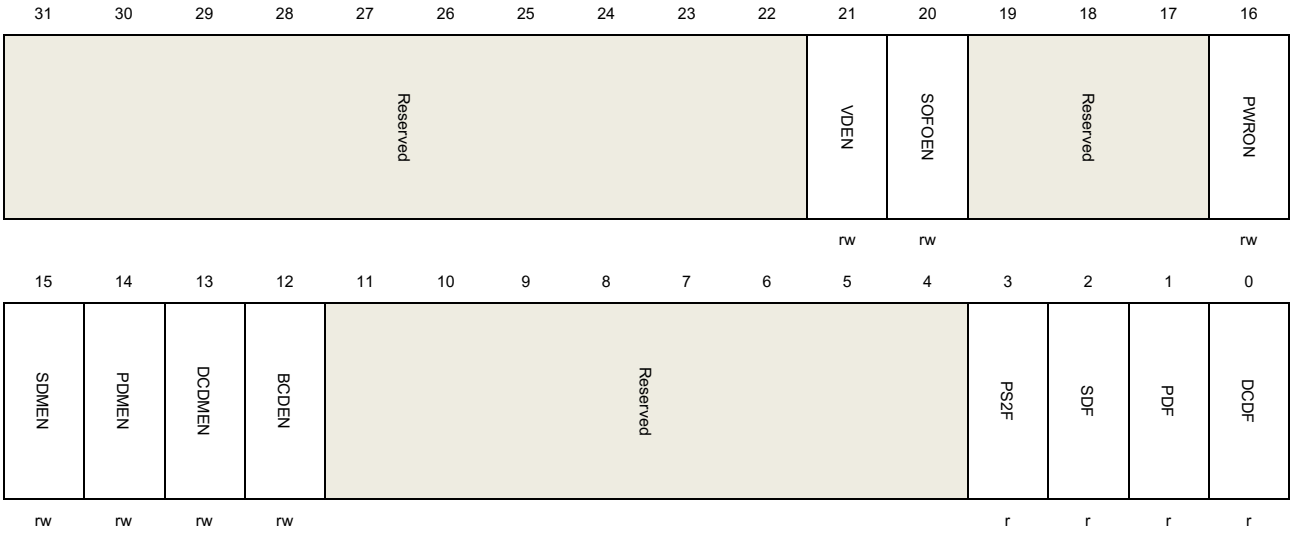
Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30:24	NPTXRQTOP[6:0]	Top entry of the non-periodic Tx request queue Entry in the non-periodic transmit request queue. Bits 30:27: Channel number Bits 26:25: - 00: IN/OUT token - 01: Zero-length OUT packet - 11: Channel halt request Bit 24: Terminate Flag, indicating last entry for selected channel.
23:16	NPTXRQS[7:0]	Non-periodic Tx request queue space The remaining space of the non-periodic transmit request queue. 0: Request queue is Full 1: 1 entry 2: 2 entries … n: n entries (0<=n<=8) Others: Reserved
15:0	NPTXFS[15:0]	Non-periodic Tx FIFO space The remaining space of the non-periodic transmit FIFO. In terms of 32-bit words. 0: Non-periodic Tx FIFO is full 1: 1 words 2: 2 words … n: n words (0<=n<=NPTXFD) Others: Reserved

Global core configuration register (USBHS_GCCFG)

Address offset: 0x0038

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	VDEN	Enable of VBUS sensing comparator to detect VBUS valid. VBUS comparator is enabled automatically if HNP or SRP is supported. 0: VBUS detection is disabled 1: VBUS detection is enabled
20	SOFOEN	SOF output enable 0: SOF pulse output disabled. 1: SOF pulse output enabled.
19:17	Reserved	Must be kept at reset value.
16	PWRON	Power on This bit is the power switch for the internal embedded PHY. 0: Embedded PHY power off. 1: Embedded PHY power on.
15	SDMEN	Secondary detection mode enable 0: Secondary detection disable 1: Secondary detection enable
14	PDMEN	Primary detection mode enable 0: Primary detection disable 1: Primary detection enable
13	DCDMEN	Data connect detection mode enable 0: Data connect detection disable 1: Data connect detection enable

12	BCDEN	Battery charging detection enable 0: Battery charging detection disable 1: Battery charging detection enable
11:4	Reserved	Must be kept at reset value.
3	PS2F	PS2 detection status, it is active only in Primary detection mode 0: Normal port is detected 1: PS2 port is detected
2	SDF	Secondary detection status 0: CDP is detected 1: DCP is detected
1	PDF	Primary detection status 0: no BCD supported is detected 1: BCD supported is detected
0	DCDF	Data connect detection status 0: Data line connect is not detected 1: Data line connect is detected

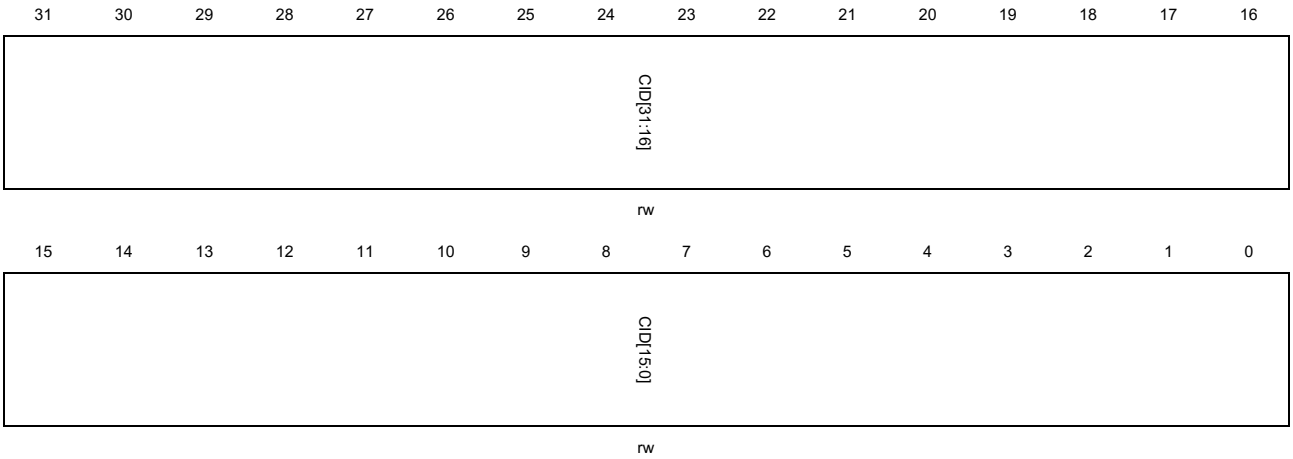
Core ID register (USBHS_CID)

Address offset: 0x003C

Reset value: 0x0000 1000

This register contains the Product ID.

This register has to be accessed by word (32-bit)



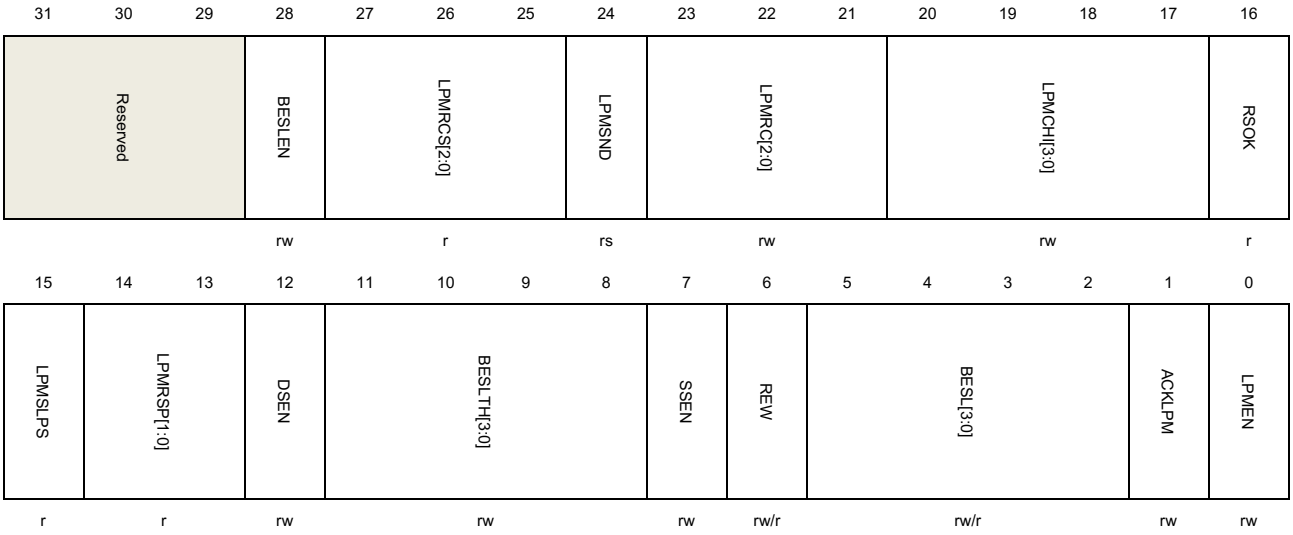
Bits	Fields	Descriptions
31:0	CID[31:0]	Core ID Software can write or read this field and uses this field as a unique ID for its application.

Global core LPM configuration register (USBHS_GLPMCFG)

Address offset: 0x0054

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	BESLEN	LPM Errata selection enable 0: USB 2.0 Link Power Management Addendum Engineering Change Notice to the USB 2.0 specification is selected 1: Errata for USB 2.0 ECN: Link Power Management (LPM) is selected
27:25	LPMRCS[2:0]	LPM retry count status **Note:** Only accessible in host mode
24	LPMSND	Send LPM transaction When ACK, STALL or NYET response is received, or all the LPM of retry count have been sent, the hardware clears this bit. **Note:** Only accessible in host mode
23:21	LPMRC[2:0]	LPM retry count It is the number of retry count when an ERROR response is received, until ACK, STALL or NYET response is received **Note:** Only accessible in host mode
20:17	LPMCHI[3:0]	Channel number index when send LPM transaction **Note:** Only accessible in host mode
16	RSOK	Resume can be sent after sleep state Host or device can send resume from sleep state after 50us (TL1Residency). When LPMSLPS is 0, it is reset.

		1: Resume can be started from sleep state 0: Resume cannot be started from sleep state
15	LPMSLPS	Sleep status **Host mode:** The host transitions to sleep status after receiving ACK response. **Device mode:** The device enters into sleep status after sending ACK response and the TL1TokenRetry timer has been expired. 1: Core is in sleep status 0: Core is not in sleep status
14:13	LPMRSP[1:0]	Response of LPM 11: ACK 10: NYET 01: STALL 00: ERROR (no response)
12	DSEN	Deep sleep enable Enable suspending the PHY in deep sleep mode
11:8	BESLTH[3:0]	BESL threshold **Device mode:** When BESL is greater than or equal to the BESLTH value, device enters into deep low power mode. **Host mode:** BESLTH indicates the duration of resume signal (TL1HubDrvResume2) when it detects device initialed resume. 0000: 75us 0001: 100us 0010: 150us 0011: 250us 0101: 450us 0110: 950us
7	SSEN	Shallow sleep enable Enable suspending the PHY in shallow sleep mode
6	REW	bRemoteWake value **Host mode:** The remote wake up value to be sent in LPM transaction **Device mode (read-only):** When ACK, STALL or NYET is sent, it is updated with bRemoteWake value in received LPM transaction
5:2	BESL[3:0]	Best effort service latency **Host mode:**

		BESL value to be sent in LPM transaction. It is also the duration of resume (TL1HubDrvResume1) when host initialed resume. **Device mode:** When ACK, STALL or NYET is sent, it is updated with BESL value in received LPM transaction 0000: 125us 0001: 150us 0010: 200us 0011: 300us 0100: 400us 0101: 500us 0110: 1000us 0111: 2000us 1000: 3000us 1001: 4000us 1010: 5000us 1011: 6000us 1100: 7000us 1101: 8000us 1110: 9000us 1111: 10000us
1	ACKLPM	ACK in LPM transaction enable 1: ACK The device response with ACK only on successful LPM transaction. - No ERROR in LPM transaction - No data pending error - bLinkState = 0001 in received LPM transaction 0: NYET The device response with NYET - The received bLinkState value is not 0001 - There is an error in received LPM transaction **Note:** Only accessible in device mode
0	LPMEN	LPM enable 1: Enable LPM 0: Disable LPM

Power down register (USBHS_PWRD)

Address offset: 0x0058

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16



Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23	ADPF	ADP event interrupt flag
22:1	Reserved	Must be kept at reset value.
0	ADPMEN	ADP module enable 1: ADP module is enable 0: ADP module is disable

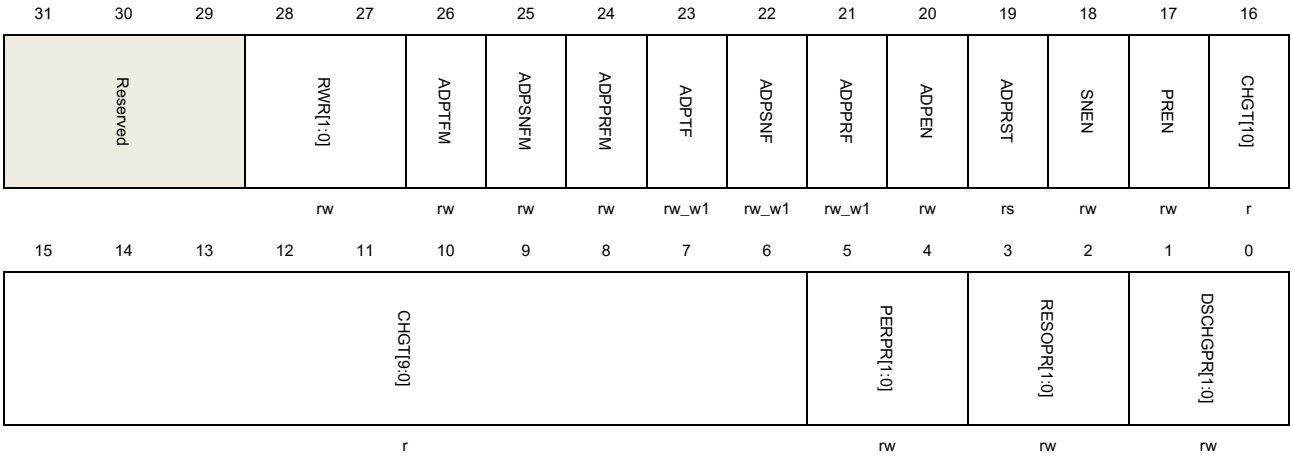
ADP control and status register (USBHS_ADPCTL)

Address offset: 0x0060
Reset value: 0x0000 0000

In order to write in the register, program RWR with 10 and keep polling until RWR = 00.

In order to read from the register, wait any ADP flag is set or write RWR with 01 and keep polling until RWR = 00.

This register has to be accessed by word (32-bit)



Bits	Fields	Descriptions

31:29	Reserved	Must be kept at reset value.
28:27	RWR[1:0]	Read and write request 00: Read or write valid (updated by core) 01: Read request 10: Write request
26	ADPTFM	The mask of ADP timeout interrupt flag
25	ADPSNFM	The mask of ADP sense interrupt flag
24	ADPPRFM	The mask of ADP probe interrupt flag
23	ADPTF	ADP timeout interrupt flag
22	ADPSNF	ADP sense interrupt flag
21	ADPPRF	ADP probe interrupt flag
20	ADPEN	ADP enable 1: ADP is enable 0: ADP is disable
19	ADPRST	ADP reset This is cleared automatically by core after reset procedure
18	SNEN	ADP sense enable 1: Sense is enable 0: Sense is disable
17	PREN	ADP probe enable 1: Probe is enable 0: Probe is disable
16:6	CHGT[10:0]	The latest time that VBUS ramps from VADPSINK to VADPPRB. These bits are defined in units of 32 kHz clock cycle. 000: 1 cycle 001: 2 cycles 002: 3 cycles 003: 4 cycles … 7ff: 2048 cycles
5:4	PERPR[1:0]	Period of probe 00: 0.625 to 0.925 second 01: 1.25 to 1.85 second 10: 1.9 to 2.6 second
3:2	RESOPR[1:0]	The resolution of CHGT value. These bits are defined in units of 32 kHz clock cycle. If 10 is chosen, the CHGT increments for every three 32 kHz clock cycle. 00: 1 cycle

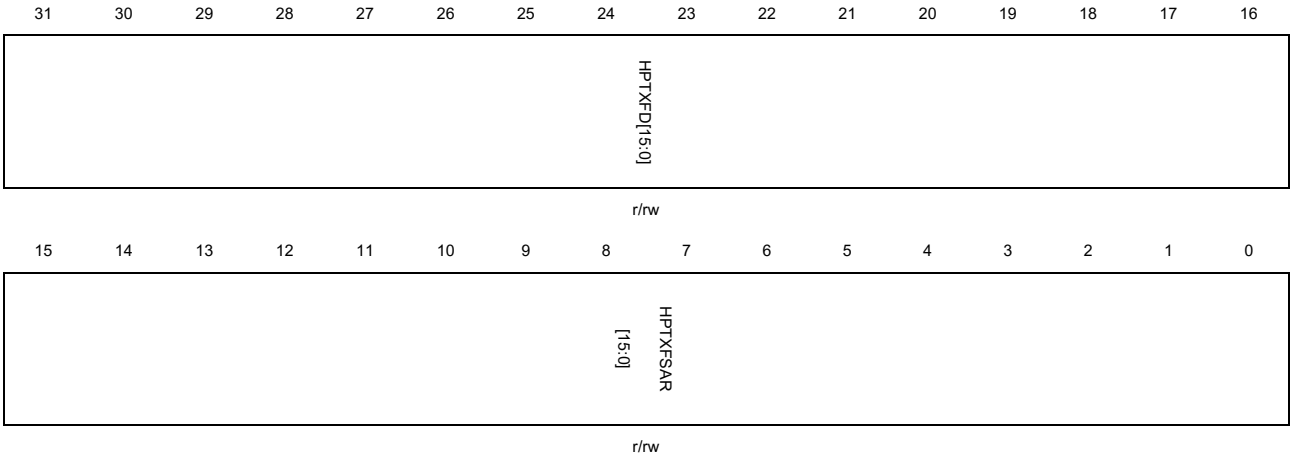
		01: 2 cycles 10: 3 cycles 11: 4 cycles
1:0	DSCHGPR[1:0]	Time of probe discharge 00: 4 ms 01: 8 ms 10: 16 ms 11: 32 ms

Host periodic transmit FIFO length register (USBHS_HPTFLEN)

Address offset: 0x0100

Reset value: 0x0200 0600

This register has to be accessed by word (32-bit)



Bits	Fields	Descriptions
31:16	HPTXFD[15:0]	Host Periodic Tx FIFO depth In terms of 32-bit word. 1<=HPTXFD<=1024
15:0	HPTXFSAR[15:0]	Host periodic Tx RAM start address The start address for host periodic transmit FIFO RAM in terms of 32-bit word.

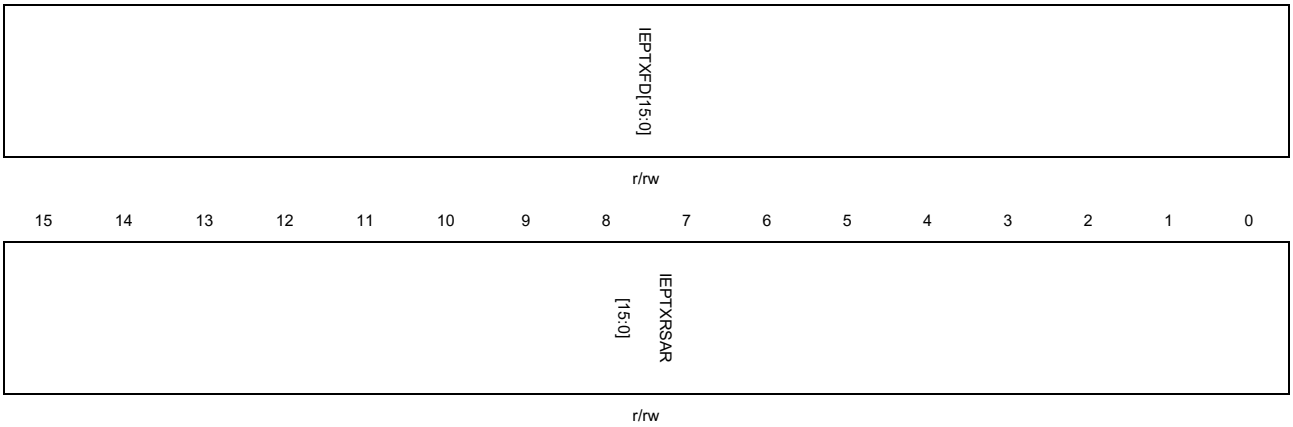
Device IN endpoint transmit FIFO length register (USBHS_DIEPxTFLEN) (x = 1..7, where x is the FIFO_number)

Address offset: 0x0104 + (FIFO_number - 1) × 0x04

Reset value: 0x0200 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



Bits	Fields	Descriptions
31:16	IEPTXFD[15:0]	IN endpoint Tx FIFO x depth In terms of 32-bit word. 1<=IEPTXFD<=1024
15:0	IEPTXRSAR[15:0]	IN endpoint FIFOx Tx x RAM start address The start address for IN endpoint transmit FIFO x in terms of 32-bit word.

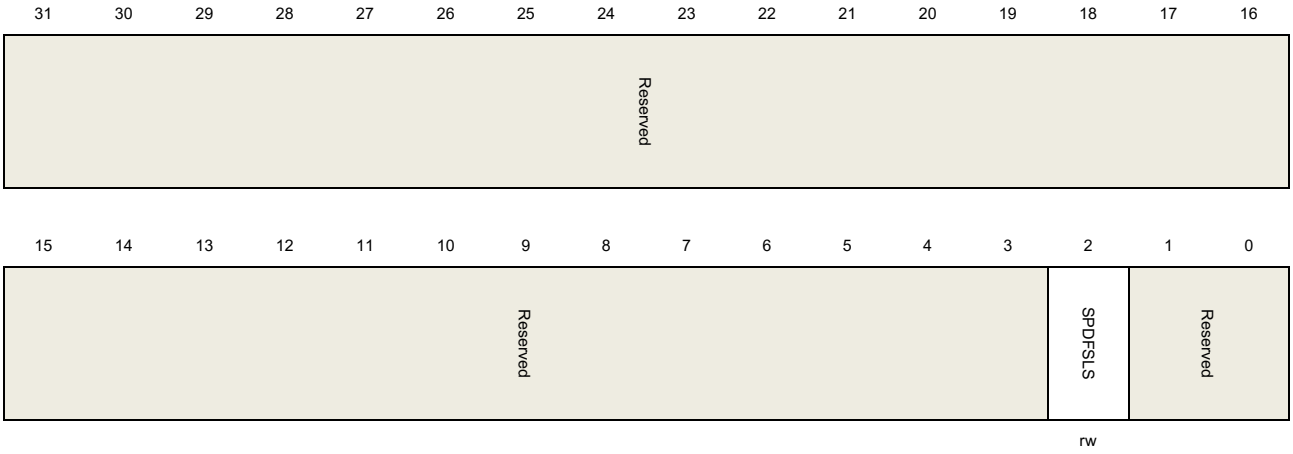
53.7.2. Host control and status registers

Host control register (USBHS_HCTL)

Address offset: 0x0400
Reset value: 0x0000 0000

This register configures the core after power-on in host mode. Do not modify it after host initialization.

This register has to be accessed by word (32-bit)



Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value.

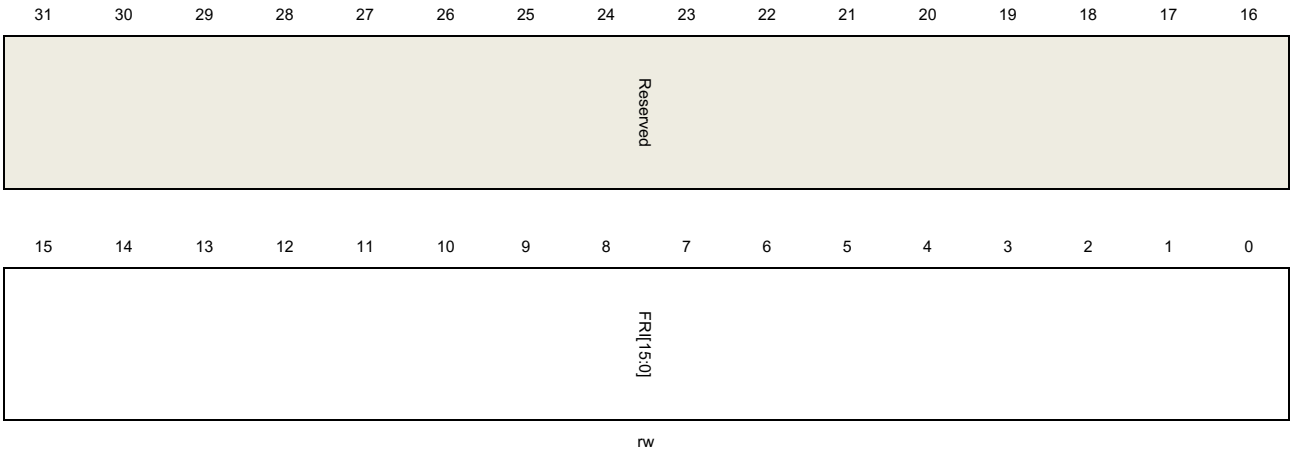
2	SPDFSLS	Speed limited to FS and LS Software may use this bit to limit USBHS's enumeration speed to FS/LS and make USBHS not perform High-Speed enumeration during reset. 0: Speed not limited. 1: Speed limited in FS/LS only.
1:0	Reserved	Must be kept at reset value.

Host frame interval register (USBHS_HFT)

Address offset: 0x0404
Reset value: 0x0000 EA60

This register sets the frame interval for the current enumerating speed when USBHS controller is enumerating.

This register has to be accessed by word (32-bit)

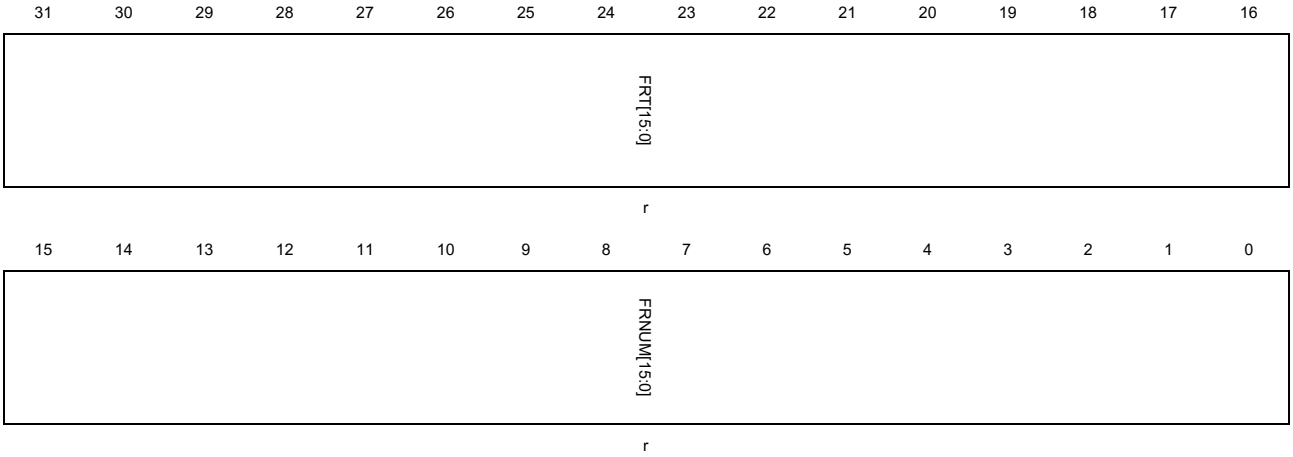


Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	FRI[15:0]	Frame interval This value describes the frame time in terms of PHY clock. Port is enabled after a port reset operation, USBHS uses a proper value according to the current speed, and software can write to this field to change the value. This value should be calculated using the frequency described below: Internal Embedded PHY: High-Speed: 60MHz Full-Speed / Low-Speed: 48MHz External ULPI PHY: 60MHz

Host frame information remaining register (USBHS_HFINFR)

Address offset: 0x408
Reset value: 0xEA60 0000

This register has to be accessed by word (32-bit)



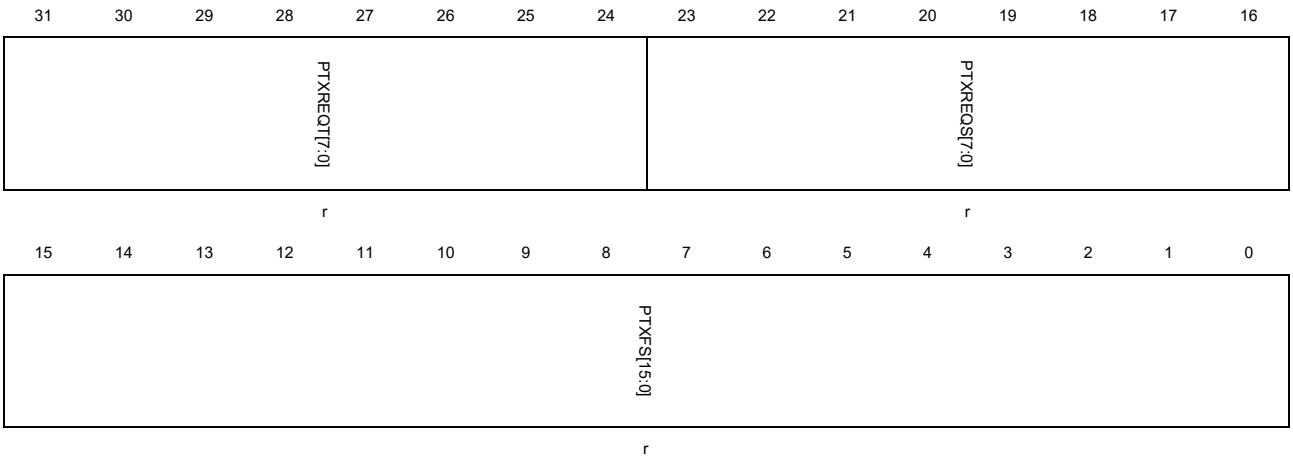
Bits	Fields	Descriptions
31:16	FRT[15:0]	Frame remaining time This field reports the remaining time of current frame in terms of PHY clock.
15:0	FRNUM[15:0]	Frame number This field reports the frame number of current frame and returns to 0 after it reaches 0x3FFF.

Host periodic transmit FIFO/queue status register (USBHS_HPTFQSTAT)

Address offset: 0x0410
Reset value: 0x0008 0200

This register reports the current status of the host periodic Tx FIFO and request queue. The request queue holds IN, OUT or other request entries in host mode.

This register has to be accessed by word (32-bit)



Bits	Fields	Descriptions
31:24	PTXREQT[7:0]	Top entry of the periodic Tx request queue

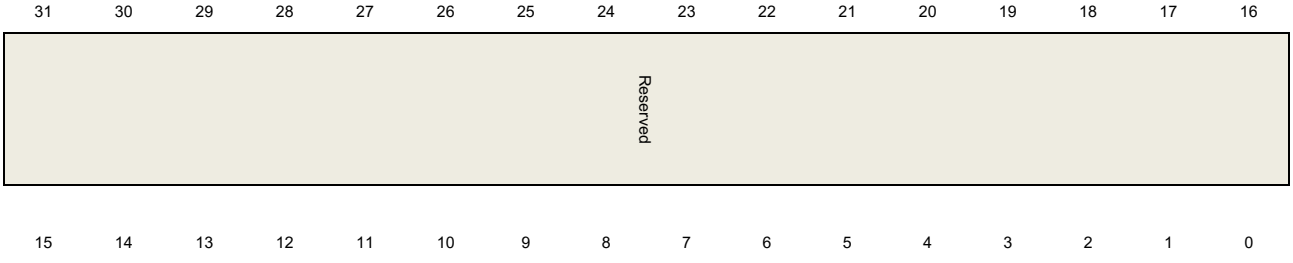
		Entry in the periodic transmit request queue. Bit 31: Odd/Even frame - 0: send in even frame - 1: send in odd frame Bits 30:27: Channel Number Bits 26:25: - 00: IN/OUT token - 01: Zero-length OUT packet - 11: Channel halt request Bit 24: Terminate Flag, indicating last entry for selected channel.
23:16	PTXREQS[7:0]	Periodic Tx request queue space The remaining space of the periodic transmit request queue. 0: Request queue is Full 1: 1 entry 2: 2 entries … n: n entries (0<=n<=8) Others: Reserved
15:0	PTXFS[15:0]	Periodic Tx FIFO space The remaining space of the periodic transmit FIFO. In terms of 32-bit word. 0: periodic Tx FIFO is full 1: 1 word 2: 2 words … n: n words (0<=n<=PTXFD) Others: Reserved

Host all channels interrupt register (USBHS_HACHINT)

Address offset: 0x0414
Reset value: 0x0000 0000

When a channel interrupt is triggered, USBHS sets corresponding bit in this register and software should read this register to know which channel is asserting interrupt.

This register has to be accessed by word (32-bit)





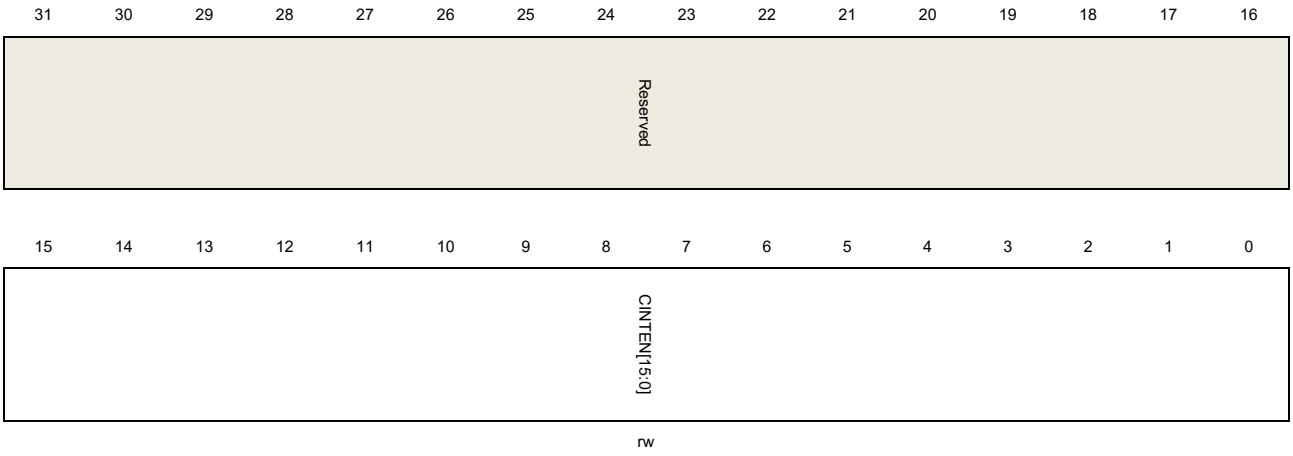
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	HACHINT[15:0]	Host all channel interrupts Each bit represents a channel: Bit 0 for channel 0, bit 15 for channel 15.

Host all channels interrupt enable register (USBHS_HACHINTEN)

Address offset: 0x0418
Reset value: 0x0000 0000

This register can be used by software to enable or disable a channel's interrupt. Only the channel whose corresponding bit in this register is set is able to cause the channel interrupt flag HCIF in USBHS_GINTF register.

This register has to be accessed by word (32-bit)



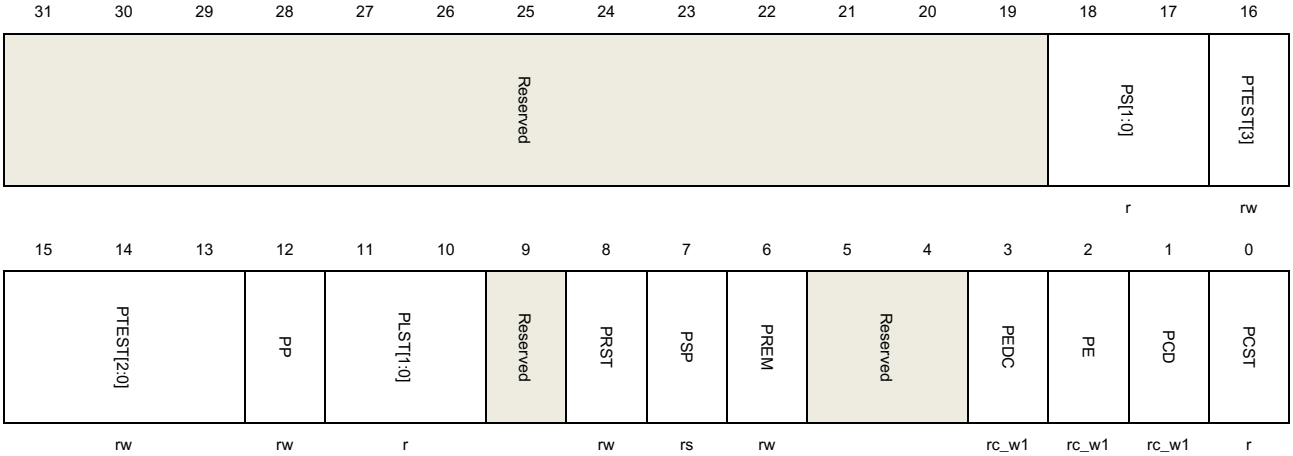
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CINTEN[15:0]	Channel interrupt enable 0: Disable channel-n interrupt 1: Enable channel-n interrupt Each bit represents a channel: Bit 0 for channel 0, bit 15 for channel 15.

Host port control and status register (USBHS_HPCS)

Address offset: 0x0440
Reset value: 0x0002 0000

This register controls the port's behavior and also has some flags which report the status of the port. The HPIF flag in USBHS_GINTF register will be triggered if one of these flags in this register is set by USBHS: PRST, PEDC and PCD.

This register has to be accessed by word (32-bit)



Bits	Fields	Descriptions
31:19	Reserved	Must be kept at reset value.
18:17	PS[1:0]	Port speed Report the enumerated speed of the device attached to this port. 00: High-Speed 01: Full-Speed 10: Low-Speed Others: Reserved
16:13	PTEST[3:0]	Port Test control The software writes a nonzero value to this field to put the port into a Test mode, and the corresponding pattern is sended on the port. When test mode is used, The HS_CUR_FE bit in USBHS_GUSBCS register should also be set. 0000: Test mode disabled 0001: Test_J mode 0010: Test_K mode 0011: Test_SE0_NAK mode 0100: Test_Packet mode 0101: Test_Force_Enable Others: Reserved
12	PP	Port power This bit should be set before a port is used. Because USBHS doesn't have power supply ability, it only uses this bit to know whether the port is in powered state. Software should ensure the true power supply on VBUS before setting this bit. 0: Port is powered off

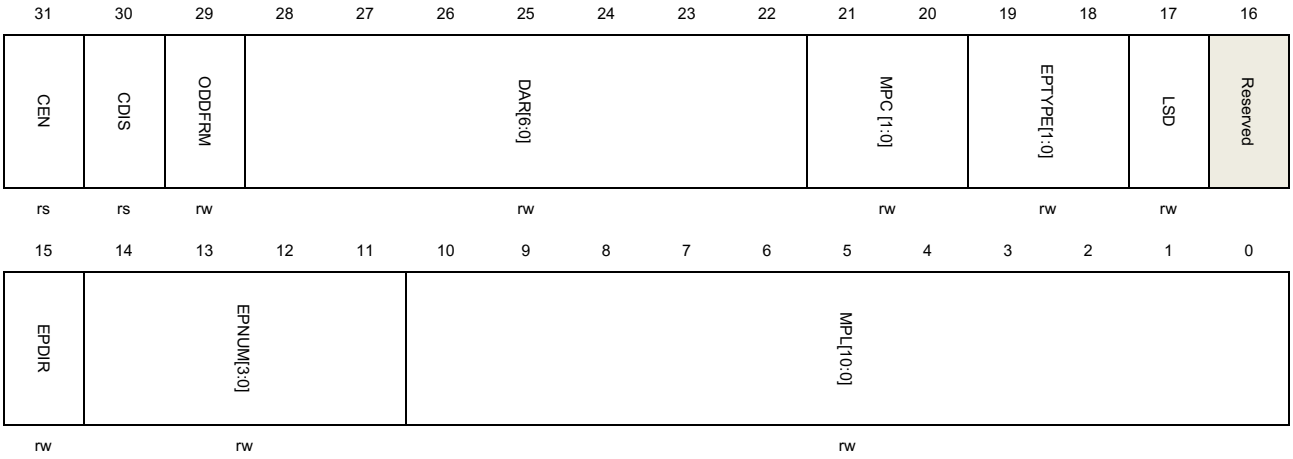
		1: Port is powered on
11:10	PLST[1:0]	Port line status Report the current state of USB data lines Bit 10: State of DP line Bit 11: State of DM line
9	Reserved	Must be kept at reset value.
8	PRST	Port reset Application sets this bit to start a reset signal on USB port. Application should clear this bit when it wants to stop the reset signal. 0: Port is not in reset state 1: Port is in reset state
7	PSP	Port suspend Application sets this bit to put port into suspend state. When this bit is set the port stops sending SOF tokens. This bit can only be cleared by the following operations: – PRST bit in this register is set by application – PREM bit in this register is set – A remote wakeup signal is detected – A device disconnection is detected 0: Port is not in suspend state 1: Port is in suspend state
6	PREM	Port resume Application sets this bit to start a resume signal on USB port. Application should clear this bit when it wants to stop the resume signal. 0: No resume driven 1: Resume driven When the application sets PREM in sleep status, the core continues to drive the resume signal until the timer specified with BESLTH has expired. When the core detects a USB remote wakeup, it starts driving the resume signal and clears it automatically at the end of resume.
5:4	Reserved	Must be kept at reset value.
3	PEDC	Port enable/disable change Set by the core when the status of the Port enable bit 2 in this register changes.
2	PE	Port Enable This bit is automatically set by USBHS after a USB reset signal finishes and cannot be set by software. This bit is cleared by the following events: – A disconnection condition – Software clears this bit

		0: Port disabled 1: Port enabled
1	PCD	Port connect detected Set by USBHS when a device connection is detected. This bit can be cleared by writing 1 to this bit.
0	PCST	Port connect status 0: Device is not connected to the port 1: Device is connected to the port

Host channel-x control register (USBHS_HCHxCTL) (x = 0..15, where x = channel_number)

Address offset: 0x0500 + (channel_number × 0x20)
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)



Bits	Fields	Descriptions
31	CEN	Channel enable Set by the application and cleared by USBHS. 0: Channel disabled 1: Channel enabled Software should follow the operation guide to disable or enable a channel.
30	CDIS	Channel disable Software can set this bit to disable the channel from processing transactions. Software should follow the operation guide to disable or enable a channel.
29	ODDFRM	Odd frame For periodic transfers (interrupt or isochronous transfer), this bit controls that whether in an odd frame or even frame this channel's transaction is desired to be processed.

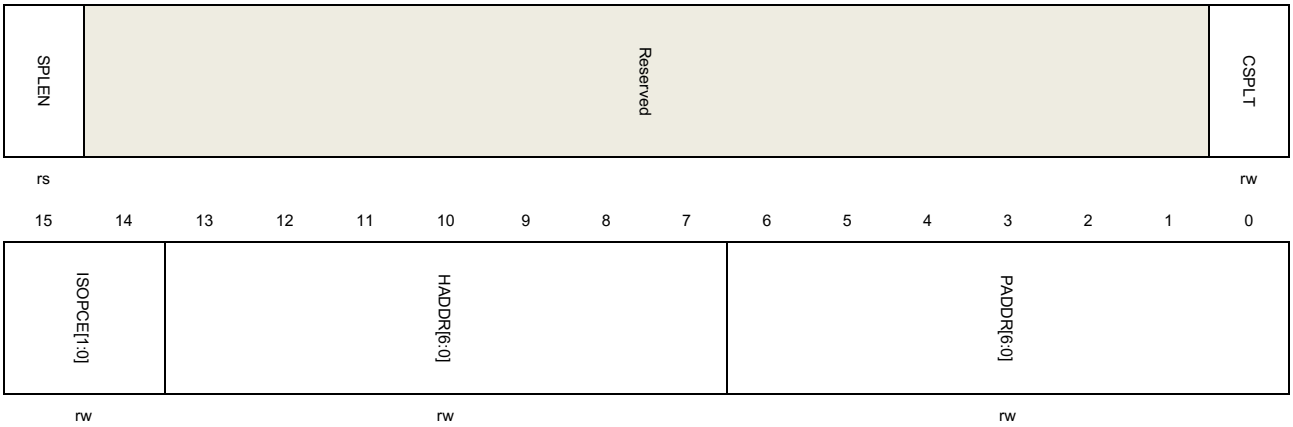
		0: Even frame 1: Odd frame
28:22	DAR[6:0]	Device address The address of the USB device that this channel wants to communicate with.
21:20	MPC[1:0]	Multiple Packet Count For periodic transfers, this field indicates to the number of transactions that must be issued per micro-frame. For nonperiodic transfers, it defines how many packets the DMA should fetch or write for this channel before the internal DMA engine changes arbitration. 00: Reserved 01: 1 transaction to be issued per micro-frame 10: 2 transactions to be issued per micro-frame 11: 3 transactions to be issued per micro-frame
19:18	EPTYPE[1:0]	Endpoint type The transfer type of the endpoint that this channel wants to communicate with. 00: Control 01: Isochronous 10: Bulk 11: Interrupt
17	LSD	Low-Speed device The device that this channel wants to communicate with is a Low-Speed Device.
16	Reserved	Must be kept at reset value.
15	EPDIR	Endpoint direction The transfer direction of the endpoint that this channel wants to communicate with. 0: OUT 1: IN
14:11	EPNUM[3:0]	Endpoint number The number of the endpoint that this channel wants to communicate with.
10:0	MPL[10:0]	Maximum packet length The target endpoint's maximum packet length.

Host channel-x split transaction control register (USBHS_HCHxSTCTL) (x = 0..15, where x = channel_number)

Address offset: 0x0504 + (channel_number × 0x20)
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16



Bits	Fields	Descriptions
31	SPLEN	Enable High-Speed split transaction Software can set this bit to enable the High-Speed split transaction on this channel. Split transaction is used to initiate a Full-/Low-Speed transaction via the hub and some Full-/Low-Speed device endpoint.
30:17	Reserved	Must be kept at reset value.
16	CSPLT	Complete-split enable Software can set this bit to make USBHS perform complete-split transaction, otherwise, USBHS performs start-split transaction.
15:14	ISOPCE[1:0]	Isochronous OUT Payload Continuation Encoding For Full-Speed isochronous OUT start-split, this field specifies how the High-Speed data payload corresponds to data for a Full-Speed data packet 00: High-Speed data is the middle of the Full-Speed data payload 01: High-Speed data is the end of the Full-Speed data payload 10: High-Speed data is the beginning of the Full-Speed data payload 11: High-Speed data is all of the Full-Speed data payload
13:7	HADDR[6:0]	HUB address This field contains the USB device address of the hub supporting the specified Full-/Low-Speed device for this Full-/Low-Speed transaction.
6:0	PADDR[6:0]	Port address This field contains the port number of the target hub for this Full-/Low-Speed transaction.

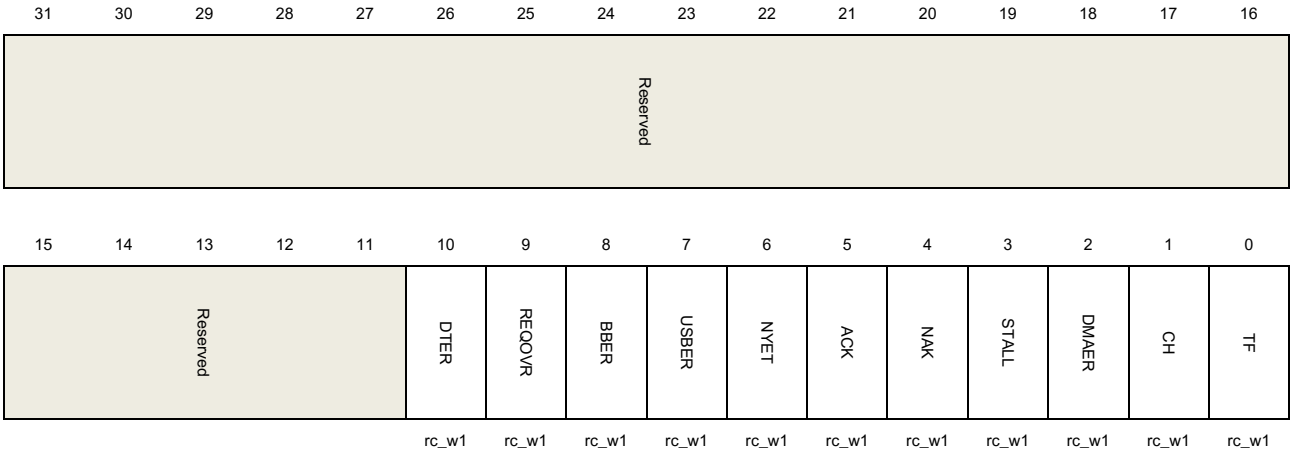
Host channel-x interrupt flag register (USBHS_HCHxINTF) (x = 0..15, where x = channel number)

Address offset: 0x0508 + (channel_number × 0x20)
Reset value: 0x0000 0000

This register contains the status and events of a channel, when software gets a channel

interrupt, it should read this register for the respective channel to know the source of the interrupt. The flag bits in this register are all set by hardware and cleared by writing 1.

This register has to be accessed by word (32-bit)



Bits	Fields	Descriptions
31:11	Reserved	Must be kept at reset value.
10	DTER	Data toggle error The IN transaction gets a data packet but the PID of this packet doesn't match DPID[1:0] bits in USBHS_HCHxLEN register.
9	REQOVR	Request queue overrun The periodic request queue is full when software starts new transfer.
8	BBER	Babble error A babble condition occurs on USB bus. A typical reason for babble condition is that a device sends a data packet and the packet length exceeds to the endpoint's maximum packet length.
7	USBER	USB Bus Error The USB error flag is set when the following conditions occur during receiving a packet: A received packet has a wrong CRC field A stuff error detected on USB bus Timeout when waiting for a response packet
6	NYET	NYET A NYET response packet received (in High-Speed).
5	ACK	ACK An ACK response is received or transmitted
4	NAK	NAK A NAK response is received.
3	STALL	STALL

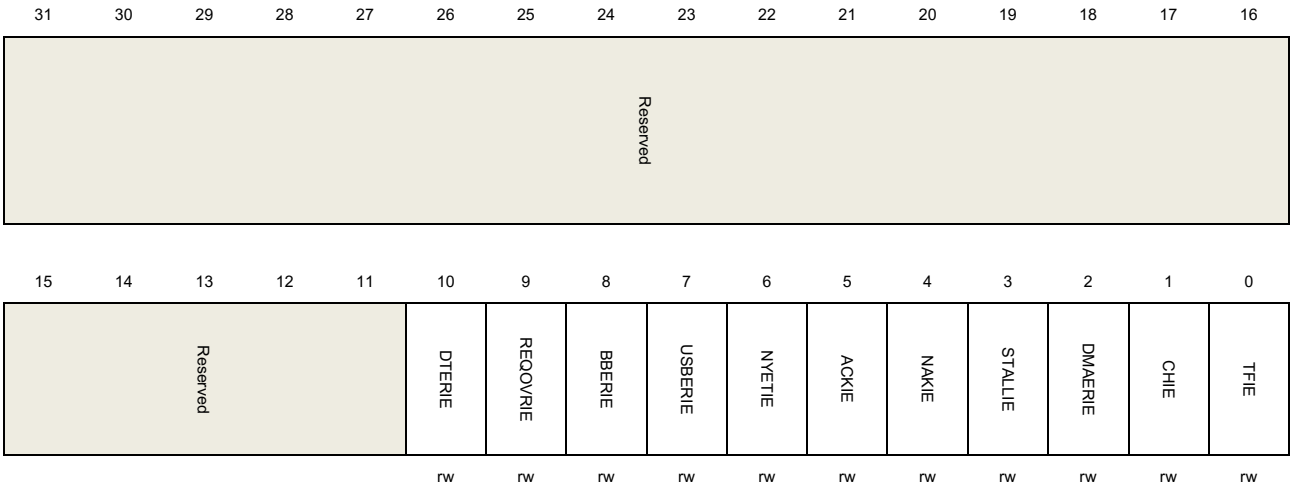
		A STALL response is received.
2	DMAER	DMA Error An error occurs when DMA tries to fetch or write packet data for this channel.
1	CH	Channel halted When DMA is not enabled: This channel is disabled by the software request. When DMA is enabled: This channel is disabled by DMA because all the transactions of this channel finish successfully or an USB error occurs.
0	TF	Transfer finished All the transactions of this channel finish successfully, and no error occurs. For IN channel, this flag will be triggered after PCNT bit in USBHS_HCHxLEN register reaches to zero. For OUT channel, this flag will be triggered when software reads and pops a TF status entry from the RxFIFO.

Host channel-x interrupt enable register (USBHS_HCHxINTEN) (x = 0..15, where x = channel number)

Address offset: 0x050C + (channel_number × 0x20)
Reset value: 0x0000 0000

This register contains the interrupt enabled bits for the flags in USBHS_HCHxINTF register. If a bit in this register is set by software, the corresponding bit in USBHS_HCHxINTF register is able to trigger a channel interrupt. The bits in this register are set and cleared by software.

This register has to be accessed by word (32-bit)



Bits	Fields	Descriptions
31:11	Reserved	Must be kept at reset value.

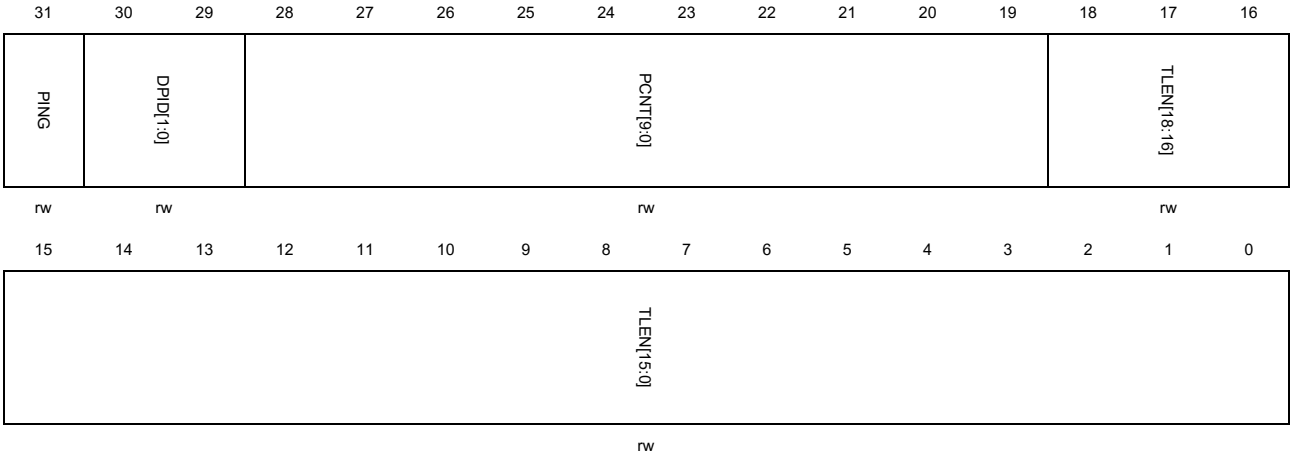
10	DTERIE	Data toggle error interrupt enable 0: Disable data toggle error interrupt 1: Enable data toggle error interrupt
9	REQOVRIE	Request queue overrun interrupt enable 0: Disable request queue overrun interrupt 1: Enable request queue overrun interrupt
8	BBERIE	Babble error interrupt enable 0: Disable babble error interrupt 1: Enable babble error interrupt
7	USBERIE	USB bus error interrupt enable 0: Disable USB bus error interrupt 1: Enable USB bus error interrupt
6	NYETIE	NYET interrupt enable 0: Disable NYET interrupt 1: Enable NYET interrupt
5	ACKIE	ACK interrupt enable 0: Disable ACK interrupt 1: Enable ACK interrupt
4	NAKIE	NAK interrupt enable 0: Disable NAK interrupt 1: Enable NAK interrupt
3	STALLIE	STALL interrupt enable 0: Disable STALL interrupt 1: Enable STALL interrupt
2	DMAERIE	DMA Error interrupt enable 0: Disable DMA Error interrupt 1: Enable DMA Error interrupt
1	CHIE	Channel halted interrupt enable 0: Disable channel halted interrupt 1: Enable channel halted interrupt
0	TFIE	Transfer finished interrupt enable 0: Disable transfer finished interrupt 1: Enable transfer finished interrupt

Host channel-x transfer length register (USBHS_HCHxLEN) (x = 0..15, where x = channel number)

Address offset: 0x0510 + (channel_number × 0x20)

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)

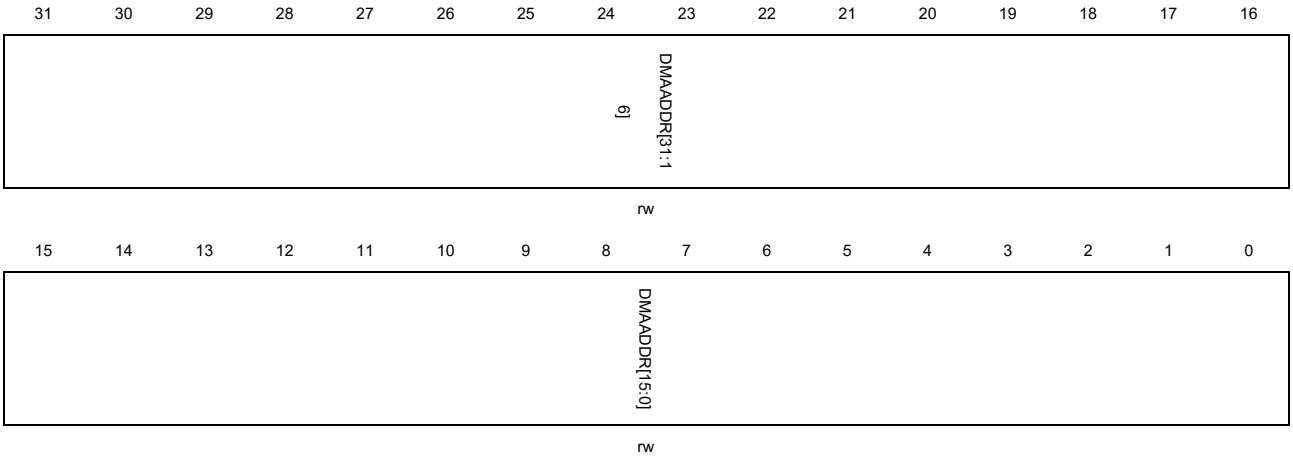


Bits	Fields	Descriptions
31	PING	PING token request For OUT transfer, USBHS will perform PING protocol if software sets this bit. USBHS will automatically set this bit when an OUT transaction receives a NAK or NYET handshake. Do not set this bit for IN transfer.
30:29	DPID[1:0]	Data PID Software should write this field before the transfer starts. For OUT transfer, this field controls the Data PID of the first transmitted packet. For IN transfer, this field controls the expected Data PID of the first received packet, and DTERR will be triggered if the Data PID doesn't match. After the transfer starts, USBHS changes and toggles this field automatically following the USB protocol. 00: DATA0 01: DATA2 10: DATA1 11: MDATA (non-control)/SETUP (control)
28:19	PCNT[9:0]	Packet count The number of data packets desired to be transmitted (OUT) or received (IN) in transfer. Software should program this field before the channel is enabled. After the transfer starts, this field is decreased automatically by USBHS after each successful data packet transmission.
18:0	TLEN[18:0]	Transfer length The total data bytes number of a transfer. For OUT transfer, this field is the total data bytes of all the data packets desired to be transmitted in an OUT transfer. Software should program this field before the channel is enabled. When software or DMA successfully writes a packet into the channel's data TxFIFO, this field is decreased by the byte size of the packet. For IN transfer each time software or DMA reads out a packet from the RxFIFO, this field is decreased by the byte size of the packet.

Host channel-x DMA address register (USBHS_HCHxDMAADDR) (x = 0..15, where x = channel number)

Address offset: 0x0514 + (channel_number × 0x20)
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)



Bits	Fields	Descriptions
31:0	DMAADDR[31:0]	DMA address This field defines the endpoint's DMA address. DMA uses this address to fetch or write packet data for this channel.

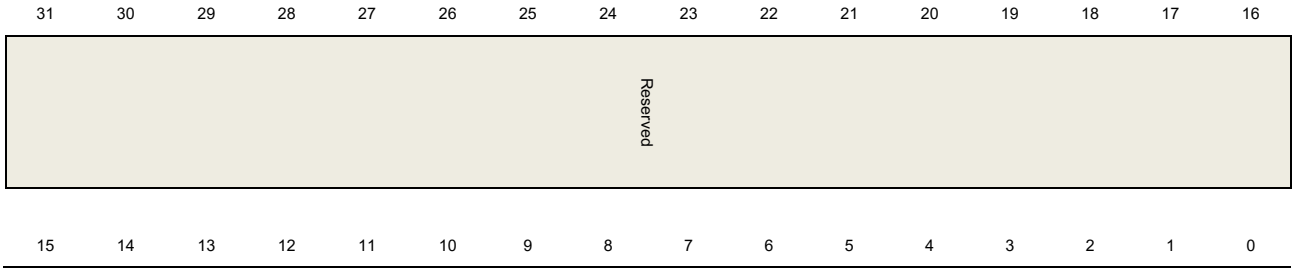
53.7.3. Device control and status registers

Device configuration register (USBHS_DCFG)

Address offset: 0x0800
Reset value: 0x0000 0000

This register configures the core in device mode after power-on or after certain control commands or enumeration. Do not change this register after device initialization.

This register has to be accessed by word (32-bit)



Reserved	EOPFT[1:0]	DAR[6:0]	Reserved	NZLSOH	DS[1:0]
	rw	rw		rw	rw

Bits	Fields	Descriptions
31:13	Reserved	Must be kept at reset value.
12:11	EOPFT[1:0]	End of periodic frame time This field defines the percentage time point in a frame when the end of periodic frame (EOPF) flag should be triggered. 00: 80% of the frame time 01: 85% of the frame time 10: 90% of the frame time 11: 95% of the frame time
10:4	DAR[6:0]	Device address This field defines the USB device's address. USBHS uses this field to match with the incoming token's device address field. Software should program this field after receiving a Set Address command from USB host.
3	Reserved	Must be kept at reset value.
2	NZLSOH	Non-zero-length status OUT handshake When a USB device receives a non-zero-length data packet during status OUT stage, this field controls that USBHS should receive this packet or reject this packet with a STALL handshake. 0: Treat this packet as a normal packet and response according to the status of NAKS and STALL bits in USBHS_DOEPxCTL register. 1: Send a STALL handshake and don't save the received OUT packet.
1:0	DS[1:0]	Device speed This field controls the device speed when the device is connected to a host. 00: High-Speed 01: Full-Speed Others: Reserved

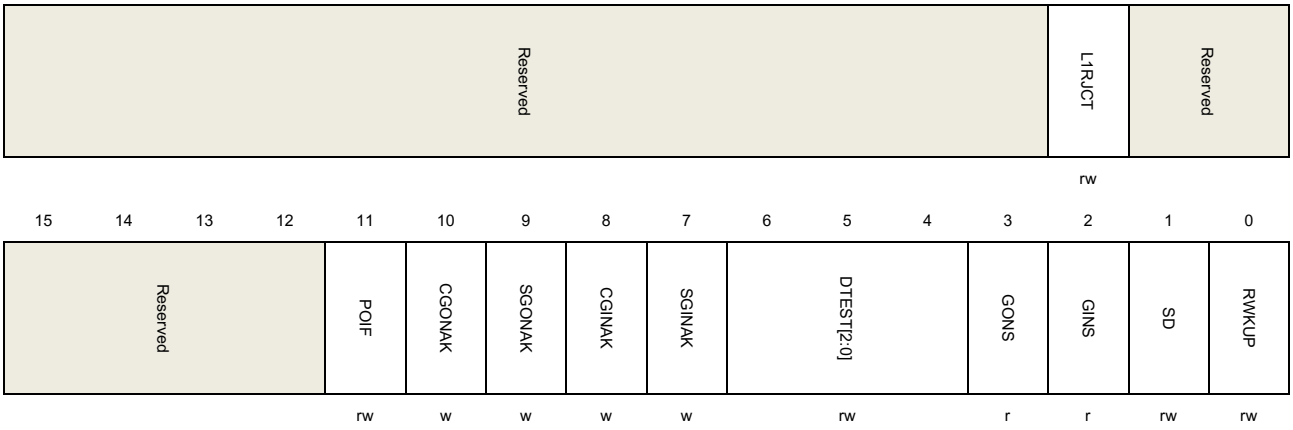
Device control register (USBHS_DCTL)

Address offset: 0x0804

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16



Bits	Fields	Descriptions
31:19	Reserved	Must be kept at reset value.
18	L1RJCT	Deep sleep reject When this bit is set, the core response NYET for LPM transaction with BESL greater than BSELTH.
17:12	Reserved	Must be kept at reset value.
11	POIF	Power-on initialization finished Software should set this bit to notify USBHS that the registers are initialized after waking up from power off state.
10	CGONAK	Clear global OUT NAK Software sets this bit to clear GONS bit in this register.
9	SGONAK	Set global OUT NAK Software sets this bit to set GONS bit in this register. When GONS bit is zero, setting this bit will also cause GONAK flag in USBHS_GINTF register triggered after a while. Software should clear the GONAK flag before writing this bit again.
8	CGINAK	Clear global IN NAK Software sets this bit to clear GINS bit in this register.
7	SGINAK	Set global IN NAK Software sets this bit to set GINS bit in this register. When GINS bit is zero, setting this bit will also cause GINAK flag in USBHS_GINTF register triggered after a while. Software should clear the GINAK flag before writing this bit again.
6:4	DTEST[2:0]	Device Test control The software writes a nonzero value to this field to put the port into a Test mode, and the corresponding pattern is sended on the port. When test mode is used, the HS_CUR_FE bit in USBHS_GUSBCS register should also be set. 000: Test mode disabled

		001: Test_J mode 010: Test_K mode 011: Test_SE0_NAK mode 100: Test_Packet mode 101: Test_Force_Enable Others: Reserved
3	GONS	Global OUT NAK status 0: The handshake that USBHS response to OUT transaction packet and whether to save the OUT data packet are decided by Rx FIFO status, endpoint's NAK and STALL bits. 1: USHBS always responses to OUT transaction with NAK handshake and doesn't save the incoming OUT data packet.
2	GINS	Global IN NAK status 0: The response to IN transaction is decided by Tx FIFO status, endpoint's NAK and STALL bits. 1: USHBS always responses to IN transaction with a NAK handshake.
1	SD	Soft disconnect Software can use this bit to generate a soft disconnect condition on USB bus. After this bit is set, USBHS first falls back to Full-Speed if currently operating at High-Speed, and then switches off the pull up resistor on DP line. This will cause the host to detect a device disconnect. 0: No soft disconnect generated. 1: Generate a soft disconnect.
0	RWKUP	Remote wakeup In suspend state, software can use this bit to generate a Remote wake up signal to inform host that it should resume the USB bus. 0: No remote wakeup signal generated. 1: Generate remote wakeup signal. When the core is LPM enabled and in sleep status, if this bit is set, the core continues to drive it and clears it automatically after 50us (TL1DevDrvResume). The application cannot set this bit when bRemoteWake value received from LPM transaction is zero.

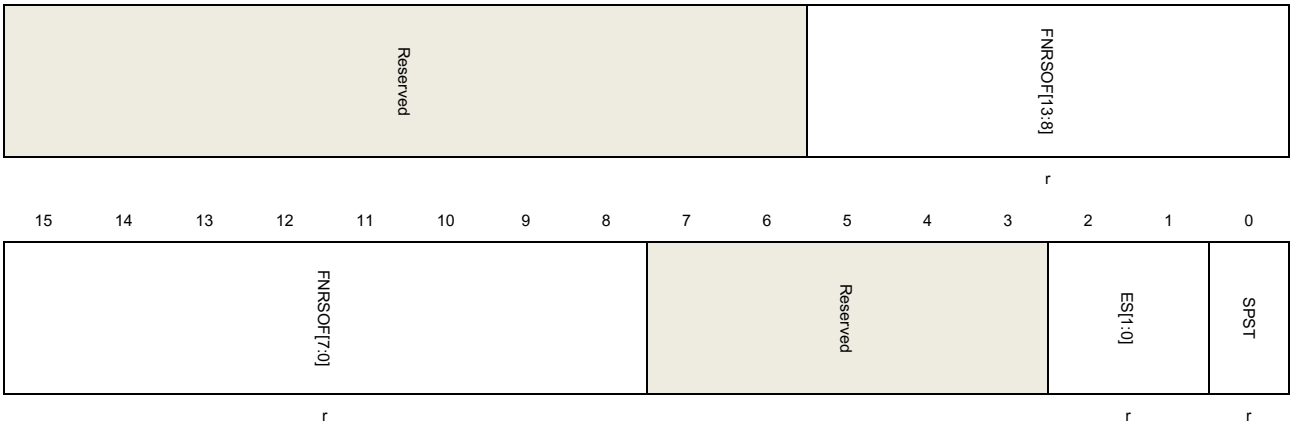
Device status register (USBHS_DSTAT)

Address offset: 0x0808
Reset value: 0x0000 0000

This register contains status and information of the USBHS in device 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



Bits	Fields	Descriptions
31:22	Reserved	Must be kept at reset value.
21:8	FNRSOF[13:0]	The frame number of the received SOF. USBHS always update this field after receiving a SOF token.
7:3	Reserved	Must be kept at reset value.
2:1	ES[1:0]	Enumerated speed This field reports the enumerated device speed. Software should read this field after the ENUMF flag in USBHS_GINTF register is triggered. 00: High-Speed 01: Full-Speed Others: reserved
0	SPST	Suspend status This bit reports whether device is in suspend state. 0: Device is not in suspend state. 1: Device is in suspend state.

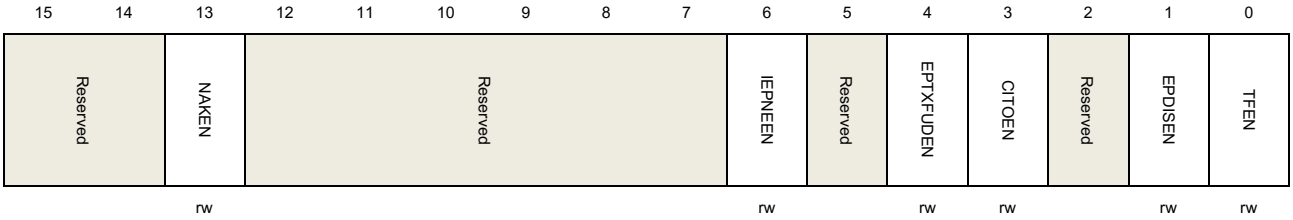
Device IN endpoint common interrupt enable register (USBHS_DIEPINTEN)

Address offset: 0x810
Reset value: 0x0000 0000

This register contains the interrupt enabled bits for the flags in USBHS_DIEPxINTF register. If a bit in this register is set by software, the corresponding bit in USBHS_DIEPxINTF register is able to trigger an endpoint interrupt in USBHS_DAEPINT register. The bits in this register are set and cleared by software.

This register has to be accessed by word (32-bit)





Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value.
13	NAKEN	NAK handshake sent by USBHS interrupt enable bit 0: Disable interrupt 1: Enable interrupt
12:7	Reserved	Must be kept at reset value.
6	IEPNEEN	IN endpoint NAK effective interrupt enable bit 0: Disable interrupt 1: Enable interrupt
5	Reserved	Must be kept at reset value.
4	EPTXFUDEN	Endpoint Tx FIFO underrun interrupt enable bit 0: Disable interrupt 1: Enable interrupt
3	CITOEN	Control In Timeout interrupt enable bit 0: Disable interrupt 1: Enable interrupt
2	Reserved	Must be kept at reset value.
1	EPDISEN	Endpoint disabled interrupt enable bit 0: Disable interrupt 1: Enable interrupt
0	TFEN	Transfer finished interrupt enable bit 0: Disable interrupt 1: Enable interrupt

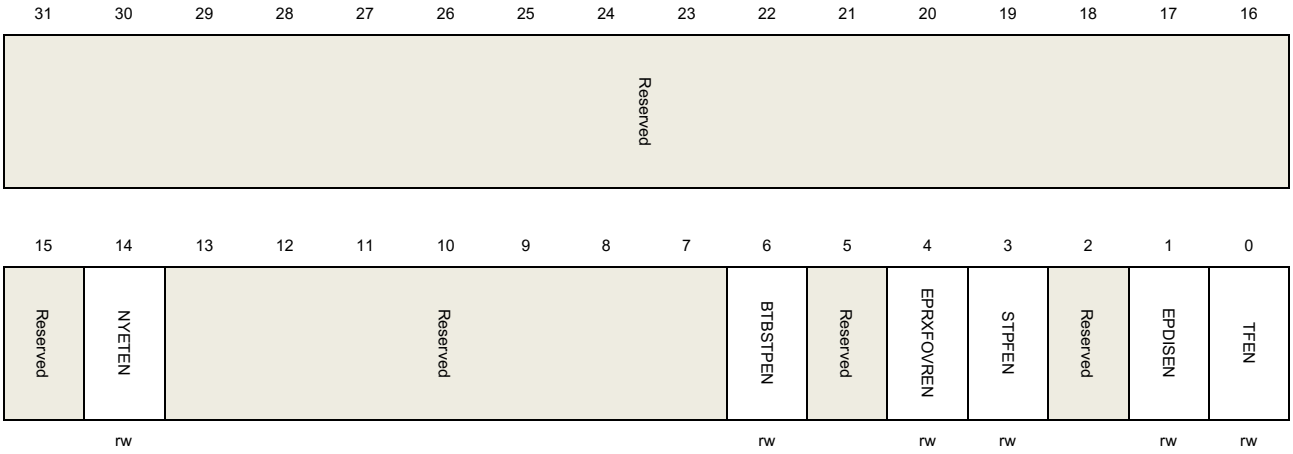
Device OUT endpoint common interrupt enable register (USBHS_DOEPINTEN)

Address offset: 0x0814
Reset value: 0x0000 0000

This register contains the interrupt enabled bits for the flags in USBHS_DOEPxINTF register. If a bit in this register is set by software, the corresponding bit in USBHS_DOEPxINTF register is able to trigger an endpoint interrupt in USBHS_DAEPINT

register. The bits in this register are set and cleared by software.

This register has to be accessed by word (32-bit)



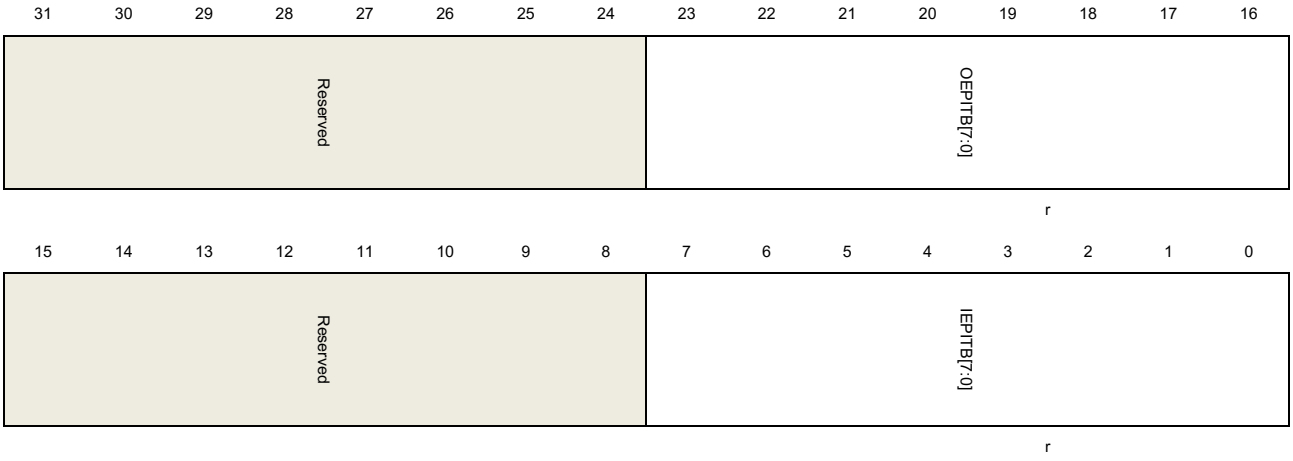
Bits	Fields	Descriptions
31:15	Reserved	Must be kept at reset value.
14	NYETEN	Send NYET handshake interrupt enable bit 0: Disable interrupt 1: Enable interrupt
13:7	Reserved	Must be kept at reset value.
6	BTBSTPEN	Back-to-back SETUP packets (Only for control OUT endpoint) interrupt enable bit 0: Disable interrupt 1: Enable interrupt
5	Reserved	Must be kept at reset value.
4	EPRXFOVREN	Endpoint Rx FIFO overrun interrupt enable bit 0: Disable interrupt 1: Enable interrupt
3	STPFEN	SETUP phase finished (Only for control OUT endpoint) interrupt enable bit 0: Disable interrupt 1: Enable interrupt
2	Reserved	Must be kept at reset value.
1	EPDISEN	Endpoint disabled interrupt enable bit 0: Disable interrupt 1: Enable interrupt
0	TFEN	Transfer finished interrupt enable bit 0: Disable interrupt 1: Enable interrupt

Device all endpoints interrupt register (USBHS_DAEPINT)

Address offset: 0x0818
Reset value: 0x0000 0000

When an endpoint interrupt is triggered, USBHS sets corresponding bit in this register and software should read this register to know which endpoint is asserting an interrupt.

This register has to be accessed by word (32-bit)



Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:16	OEPITB[7:0]	Device all OUT endpoints interrupt bits Each bit represents an OUT endpoint: Bit 16 for OUT endpoint 0, bit 23 for OUT endpoint 7.
15:8	Reserved	Must be kept at reset value.
7:0	IEPITB[7:0]	Device all IN endpoints interrupt bits Each bit represents an IN endpoint: Bit 0 for IN endpoint 0, bit 7 for IN endpoint 7.

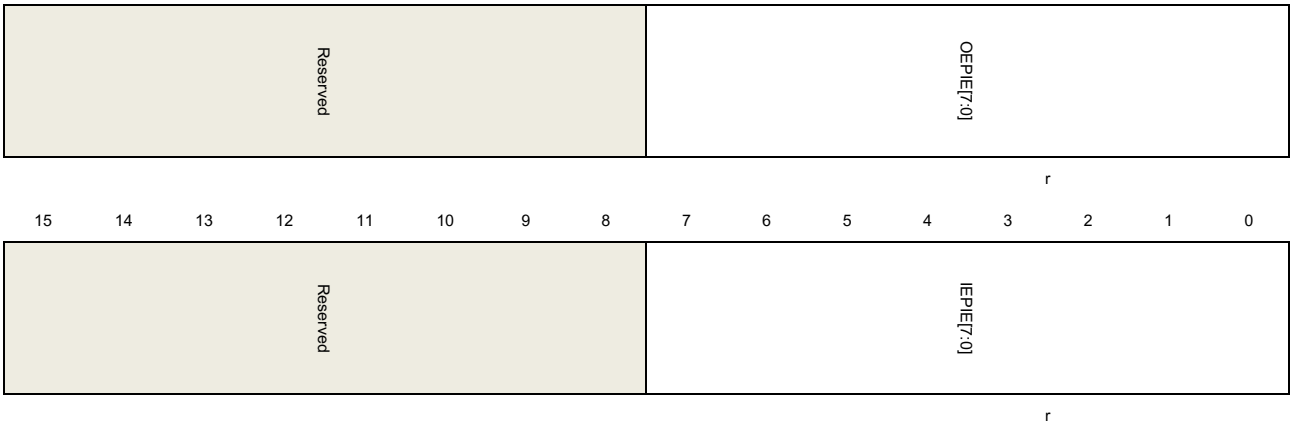
Device all endpoints interrupt enable register (USBHS_DAEPINTEN)

Address offset: 0x081C
Reset value: 0x0000 0000

This register can be used by software to enable or disable an endpoint's interrupt. Only the endpoint whose corresponding bit in this register is set is able to cause the endpoint interrupt flag OEPIF or IEPIF in USBHS_GINTF register.

This register has to be accessed by word (32-bit)





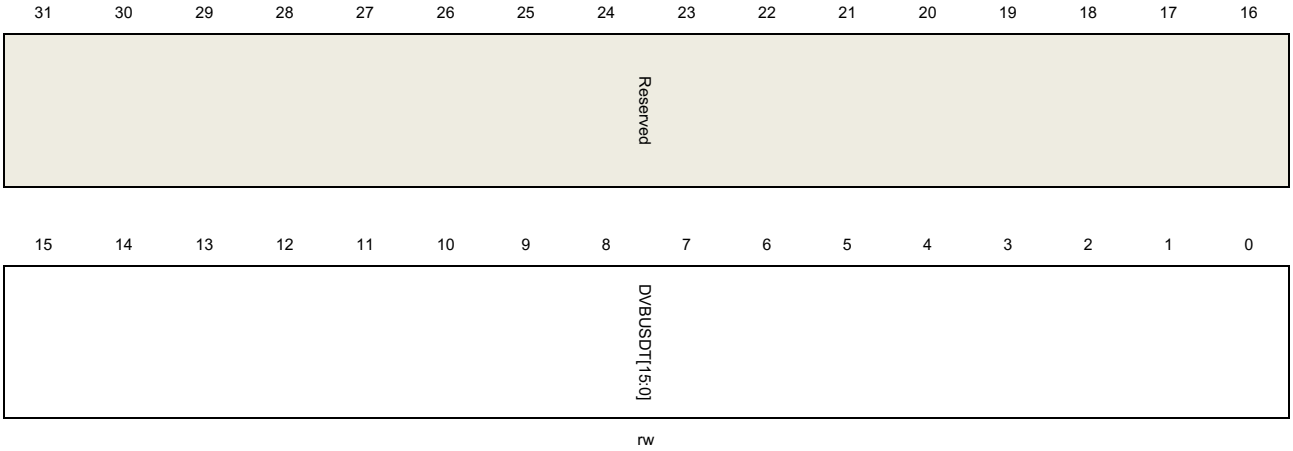
Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:16	OEPIE[7:0]	Out endpoint interrupt enable 0: Disable OUT endpoint-n interrupt 1: Enable OUT endpoint-n interrupt Each bit represents an OUT endpoint: Bit 16 for OUT endpoint 0, bit 23 for OUT endpoint 7.
15:8	Reserved	Must be kept at reset value.
7:0	IEPIE[7:0]	IN endpoint interrupt enable bits 0: Disable IN endpoint-n interrupt 1: Enable IN endpoint-n interrupt Each bit represents an IN endpoint: Bit 0 for IN endpoint 0, bit 7 for IN endpoint 7.

Device VBUS discharge time register (USBHS_DVBUSDT)

Address offset: 0x0828

Reset value: 0x0000 17D7

This register has to be accessed by word (32-bit)



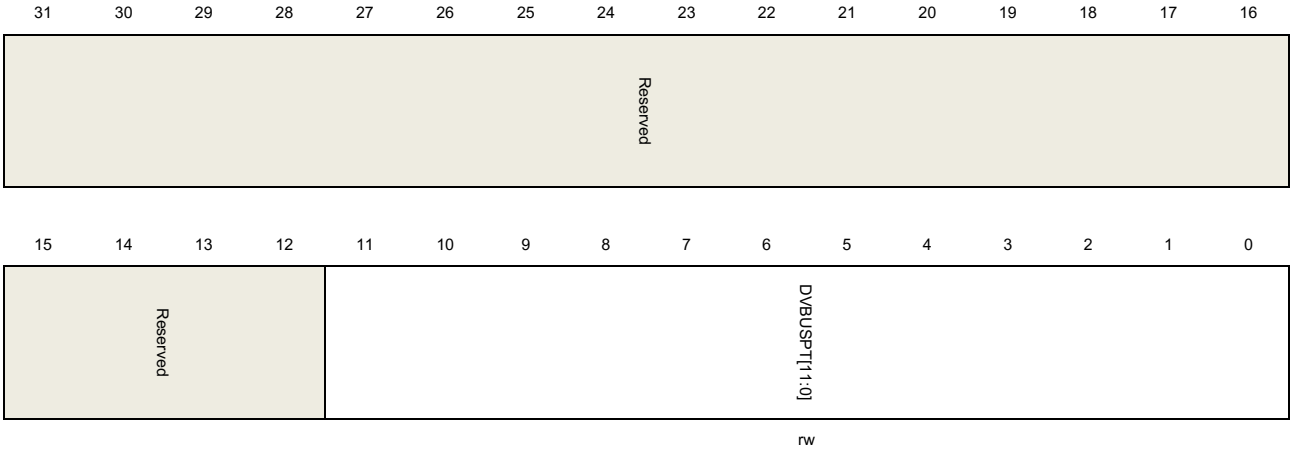
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	DVBUSDT[15:0]	Device V_{BUS} discharge time There is a discharge process after V_{BUS} pulsing in SRP protocol. This field defines the discharge time of V_{BUS}. The true discharge time is 1024*DVBUSDT[15:0] *$T_{USBCLOCK}$, where $T_{USBCLOCK}$ is the period time of USB clock.

Device VBUS pulsing time register (USBHS_DVBUSPT)

Address offset: 0x082C

Reset value: 0x0000 05B8

This register has to be accessed by word (32-bit)



Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11:0	DVBUSPT[11:0]	Device V_{BUS} pulsing time This field defines the pulsing time for V_{BUS}. The true pulsing time is 1024*DVBUSPT[11:0] *$T_{USBCLOCK}$, where $T_{USBCLOCK}$ is the period time of USB clock.

Device IN endpoint FIFO empty interrupt enable register (USBHS_DIEPFEINTEN)

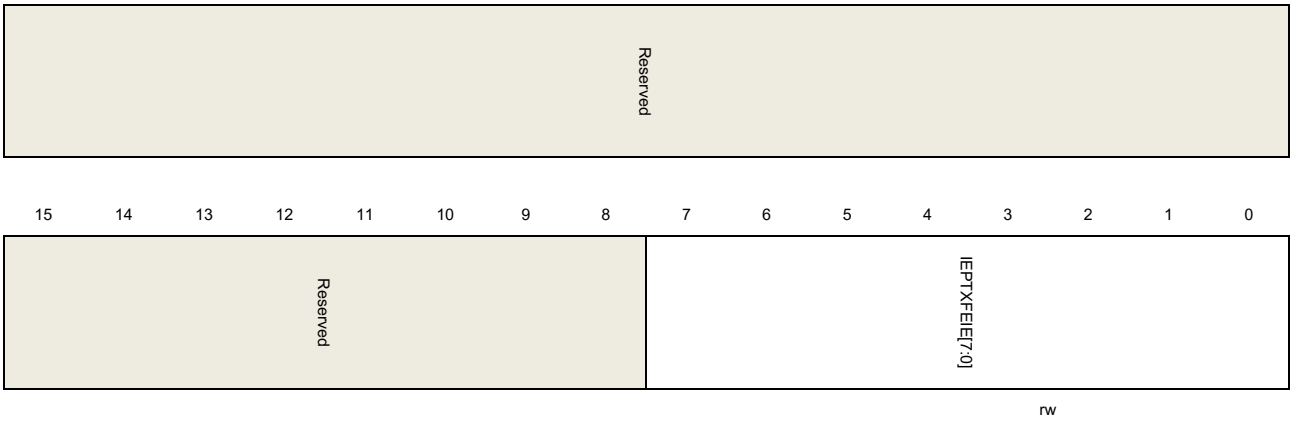
Address offset: 0x0834

Reset value: 0x0000 0000

This register contains the enabled bits for the Tx FIFO empty interrupts of IN endpoints.

This register has to be accessed by word (32-bit)

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16



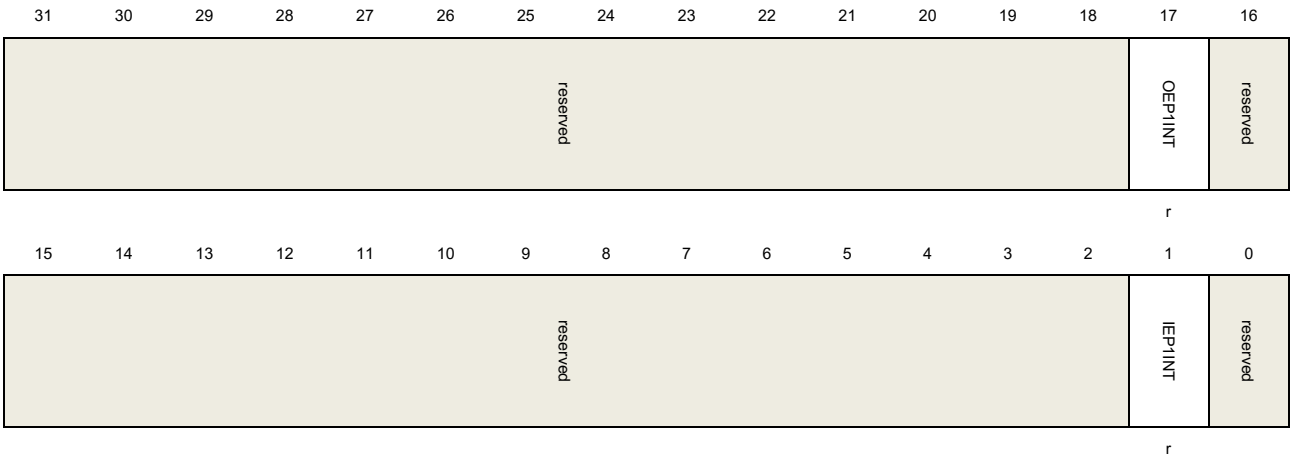
Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	IEPTXFEIE[7:0]	IN endpoint Tx FIFO empty interrupt enable bits This field controls whether the TXFE bit in USBHS_DIEPxINTF register is able to generate an endpoint interrupt bit in USBHS_DAEPINT register. Bit 0 for IN endpoint 0, bit 7 for IN endpoint 7 0: Disable FIFO empty interrupt 1: Enable FIFO empty interrupt

Device endpoint 1 interrupt register (USBHS_DEP1INT)

Address offset: 0x0838
Reset value: 0x0000 0000

When endpoint-1 out or in interrupt is triggered, USBHS sets corresponding bit in this register and software should read this register to know which endpoint is asserting the endpoint-1 interrupt.

This register has to be accessed by word (32-bit)



Bits	Fields	Descriptions

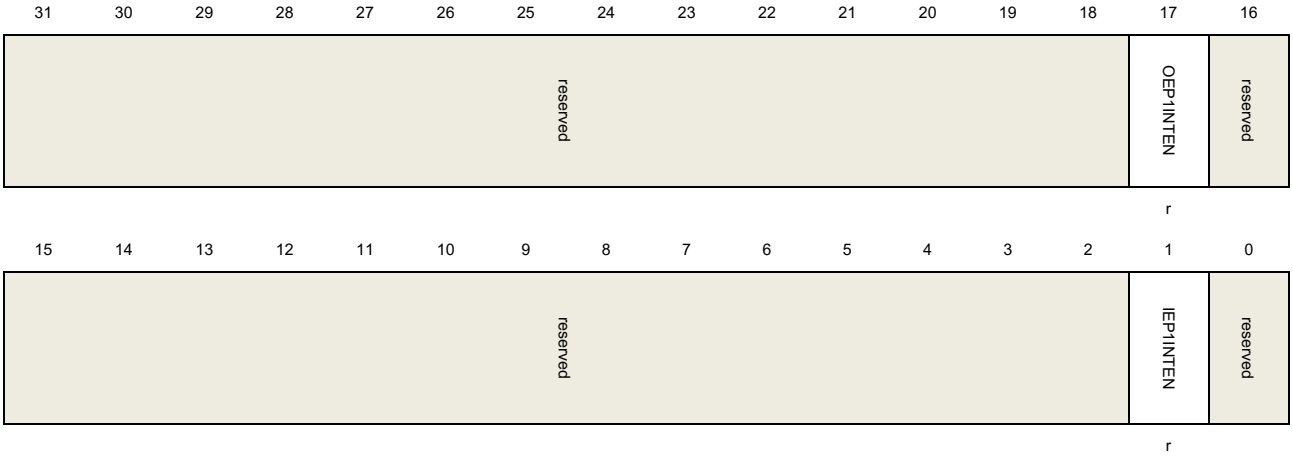
31:18	Reserved	Must be kept at reset value.
17	OEP1INT	OUT Endpoint 1 interrupt
16:2	Reserved	Must be kept at reset value.
1	IEP1INT	IN Endpoint 1 interrupt
0	Reserved	Must be kept at reset value.

Device endpoint 1 interrupt enable register (USBHS_DEP1INTEN)

Address offset: 0x083C
Reset value: 0x0000 0000

This register can be used by software to enable or disable endpoint-1's interrupt. Only the endpoint whose corresponding bit in this register is set is able to cause the endpoint-1 in or out interrupt.

This register has to be accessed by word (32-bit)



Bits	Fields	Descriptions
31:18	Reserved	Must be kept at reset value.
17	OEP1INTEN	OUT Endpoint 1 interrupt enable
16:2	Reserved	Must be kept at reset value.
1	IEP1INTEN	IN Endpoint 1 interrupt enable
0	Reserved	Must be kept at reset value.

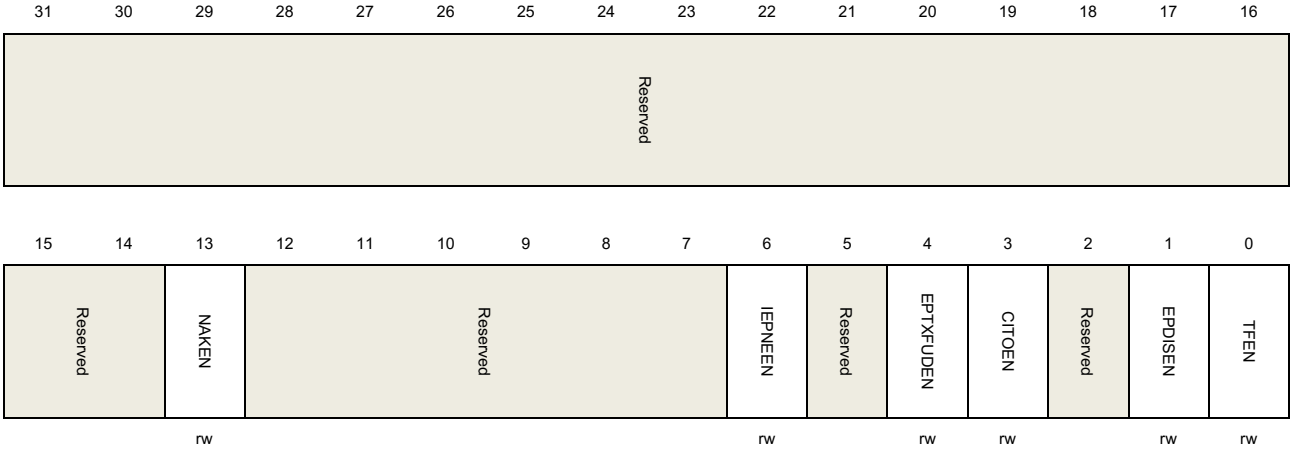
Device IN endpoint-1 interrupt enable register (USBHS_DIEP1INTEN)

Address offset: 0x844
Reset value: 0x0000 0000

This register contains the interrupt enable bits for the flags in USBHS_DIEP1INTF register.

If a bit in this register is set by software, the corresponding bit in USBHS_DIEP1INTF register is able to trigger an endpoint interrupt in USBHS_DEP1INT register. The bits in this register are set and cleared by software.

This register has to be accessed by word (32-bit)



Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value.
13	NAKEN	Interrupt enable bit of NAK handshake sent by USBHS 0: Disable interrupt 1: Enable interrupt
12:7	Reserved	Must be kept at reset value.
6	IEPNEEN	IN endpoint NAK effective interrupt enable bit 0: Disable interrupt 1: Enable interrupt
5	Reserved	Must be kept at reset value.
4	EPTXFUDEN	Endpoint Tx FIFO underrun interrupt enable bit 0: Disable interrupt 1: Enable interrupt
3	CITOEN	Control In Timeout interrupt enable bit 0: Disable interrupt 1: Enable interrupt
2	Reserved	Must be kept at reset value.
1	EPDISEN	Endpoint disabled interrupt enable bit 0: Disable interrupt 1: Enable interrupt
0	TFEN	Transfer finished interrupt enable bit 0: Disable interrupt

1: Enable interrupt

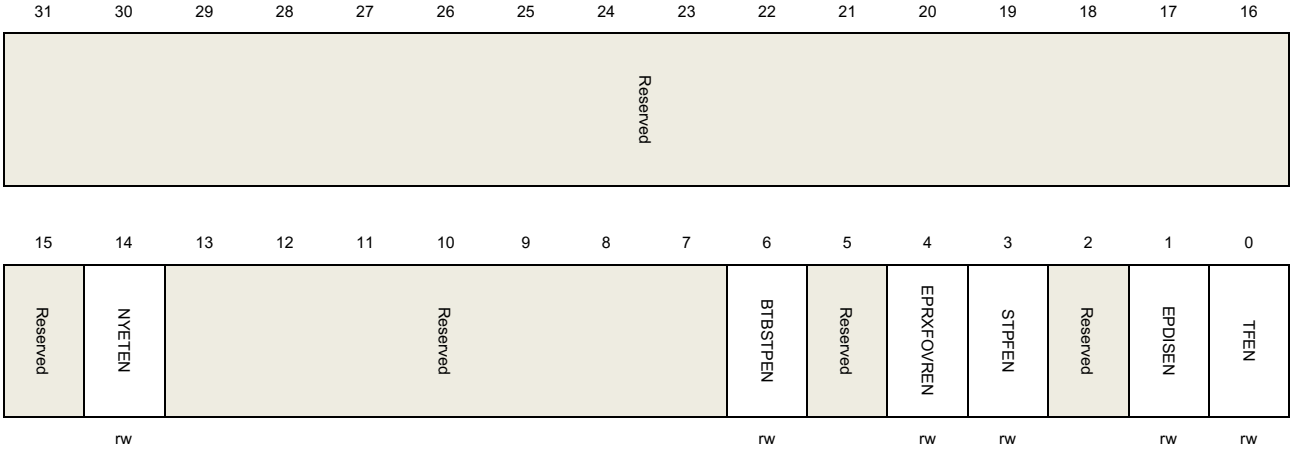
Device OUT endpoint-1 interrupt enable register (USBHS_DOEP1INTEN)

Address offset: 0x0884

Reset value: 0x0000 0000

This register contains the interrupt enabled bits for the flags in USBHS_DOEP1INTF register. If a bit in this register is set by software, the corresponding bit in USBHS_DOEP1INTF register is able to trigger an endpoint interrupt in USBHS_DEP1INT register. The bits in this register are set and cleared by software.

This register has to be accessed by word (32-bit)



Bits	Fields	Descriptions
31:15	Reserved	Must be kept at reset value.
14	NYETEN	Send NYET handshake interrupt enable bit 0: Disable interrupt 1: Enable interrupt
13:7	Reserved	Must be kept at reset value.
6	BTBSTPEN	Back-to-back SETUP packets (Only for control OUT endpoint) interrupt enable bit 0: Disable interrupt 1: Enable interrupt
5	Reserved	Must be kept at reset value.
4	EPRXFOVREN	Endpoint Rx FIFO overrun interrupt enable bit 0: Disable interrupt 1: Enable interrupt
3	STPFEN	SETUP phase finished (Only for control OUT endpoint) interrupt enable bit 0: Disable interrupt 1: Enable interrupt

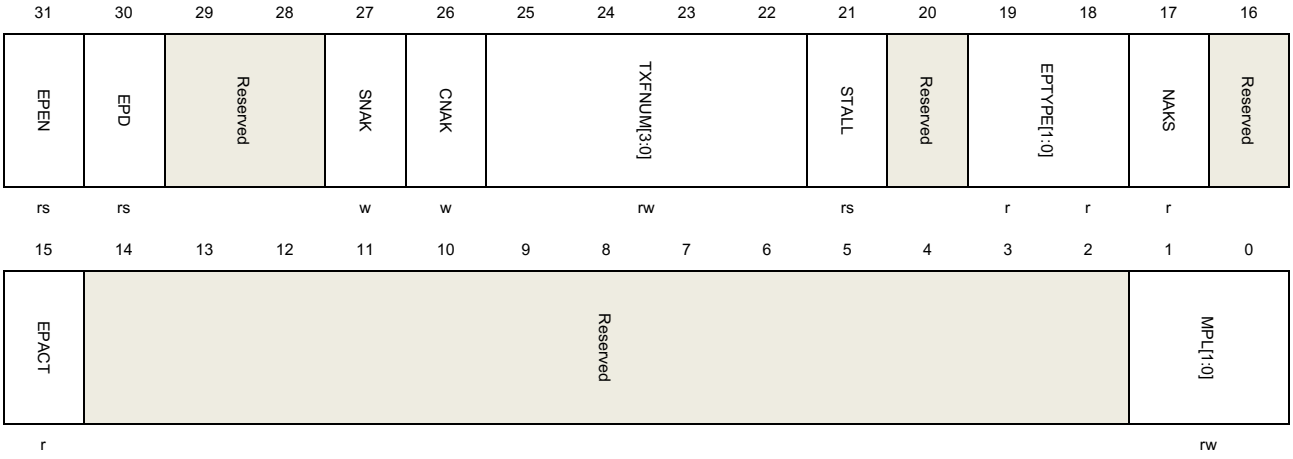
2	Reserved	Must be kept at reset value.
1	EPDISEN	Endpoint disabled interrupt enable bit 0: Disable interrupt 1: Enable interrupt
0	TFEN	Transfer finished interrupt enable bit 0: Disable interrupt 1: Enable interrupt

Device IN endpoint 0 control register (USBHS_DIEP0CTL)

Address offset: 0x0900

Reset value: 0x0000 8000

This register has to be accessed by word (32-bit)



Bits	Fields	Descriptions
31	EPEN	Endpoint enable Set by the application and cleared by USBHS. 0: Endpoint disabled 1: Endpoint enabled Software should follow the operation guide to disable or enable an endpoint.
30	EPD	Endpoint disable Software can set this bit to disable the endpoint. Software should follow the operation guide to disable or enable an endpoint.
29:28	Reserved	Must be kept at reset value.
27	SNAK	Set NAK Software sets this bit to set NAKS bit in this register.
26	CNAK	Clear NAK Software sets this bit to clear NAKS bit in this register.

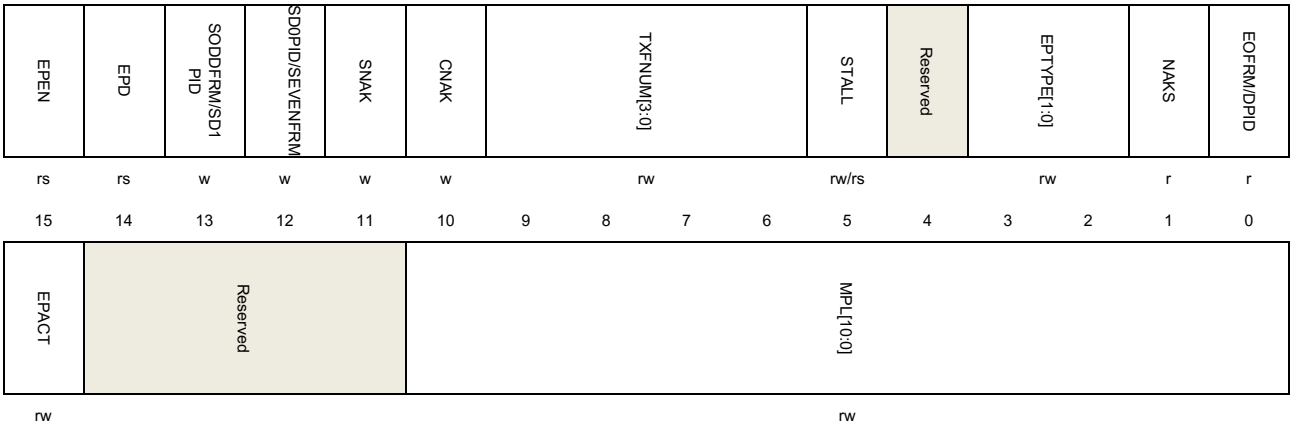
25:22	TXFNUM[3:0]	Tx FIFO number Define the Tx FIFO number of IN endpoint 0.
21	STALL	STALL handshake Software can set this bit to make USBHS send STALL handshake when receiving IN token. USBHS will clear this bit after a SETUP token is received on the corresponding OUT endpoint 0. This bit has a higher priority than NAKS bit in this register and GINS bit in USBHS_DCTL register. If both STALL and NAKS bits are set, the STALL bit takes effect.
20	Reserved	Must be kept at reset value.
19:18	EPTYPE[1:0]	Endpoint type This field is fixed to '00' for control endpoint.
17	NAKS	NAK status This bit controls the NAK status of USBHS when both STALL bit in this register and GINS bit in USBHS_DCTL register are cleared: 0: USBHS sends data or handshake packets according to the status of the endpoint's Tx FIFO. 1: USBHS always sends NAK handshake to the IN token. This bit is read-only and software should use CNAK and SNAK in this register to control this bit.
16	Reserved	Must be kept at reset value.
15	EPACT	Endpoint active This field is fixed to '1' for endpoint 0.
14:2	Reserved	Must be kept at reset value.
1:0	MPL[1:0]	Maximum packet length This field defines the maximum packet length for a control data packet. As described in USB 2.0 protocol, there are 4 kinds of length for control transfers: 00: 64 bytes 01: 32 bytes 10: 16 bytes 11: 8 bytes

Device IN endpoint-x control register (USBHS_DIEPxCTL) (x = 1..7, where x = endpoint_number)

Address offset: 0x0900 + (endpoint_number × 0x20)
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16



Bits	Fields	Descriptions
31	EPEN	Endpoint enable Set by the application and cleared by USBHS. 0: Endpoint disabled 1: Endpoint enabled Software should follow the operation guide to disable or enable an endpoint.
30	EPD	Endpoint disable Software can set this bit to disable the endpoint. Software should following the operation guide to disable or enable an endpoint.
29	SODDFRM	Set odd frame (For isochronous IN endpoints) This bit has effect only if this is an isochronous IN endpoint. Software sets this bit to set EOFRM bit in this register.
	SD1PID	Set DATA1 PID (For interrupt/bulk IN endpoints) Software sets this bit to set DPID bit in this register.
28	SEVENFRM	Set even frame (For isochronous IN endpoints) Software sets this bit to clear EOFRM bit in this register.
	SD0PID	Set DATA0 PID (For interrupt/bulk IN endpoints) Software sets this bit to clear DPID bit in this register.
27	SNAK	Set NAK Software sets this bit to set NAKS bit in this register.
26	CNAK	Clear NAK Software sets this bit to clear NAKS bit in this register.
25:22	TXFNUM[3:0]	Tx FIFO number Defines the Tx FIFO number of this IN endpoint.
21	STALL	STALL handshake Software can set this bit to make USBHS send STALL handshake when receiving IN token. This bit has a higher priority than NAKS bit in this register and GINS bit in USBHS_DCTL register. If both STALL and NAKS bits are set, the STALL bit takes effect.

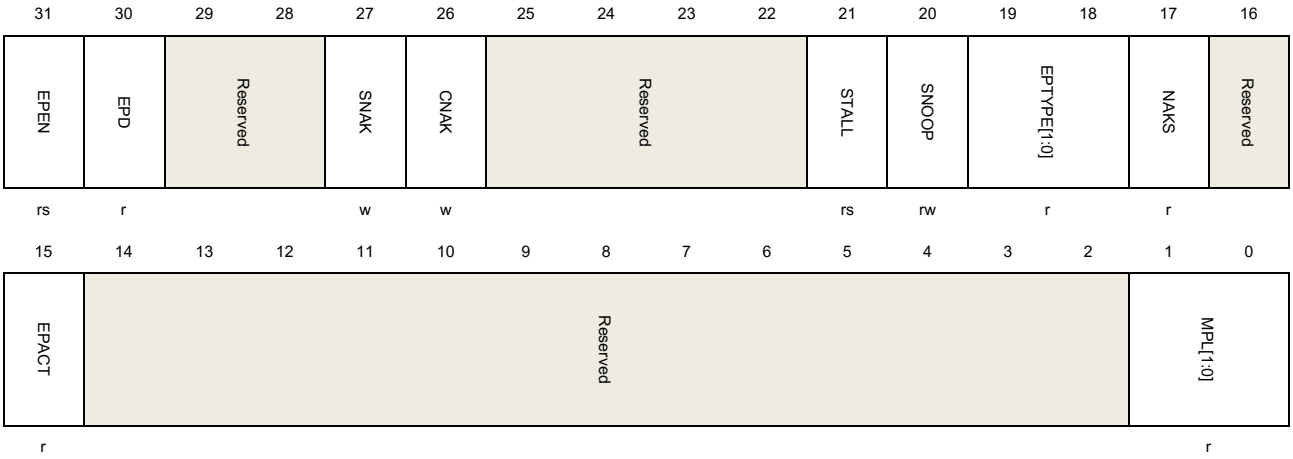
		For control IN endpoint: Only USBHS can clear this bit when a SETUP token is received on the corresponding OUT endpoint. Software is not able to clear it. For interrupt or bulk IN endpoint: Only software can clear this bit
20	Reserved	Must be kept at reset value.
19:18	EPTYPE[1:0]	Endpoint type This field defines the transfer type of this endpoint: 00: Control 01: Isochronous 10: Bulk 11: Interrupt
17	NAKS	NAK status This bit controls the NAK status of USBHS when both STALL bit in this register and GINS bit in USBHS_DCTL register are cleared: 0: USBHS sends data or handshake packets according to the status of the endpoint's Tx FIFO. 1: USBHS always sends NAK handshake to the IN token. This bit is read-only and software should use CNAK and SNAK in this register to control this bit.
16	EOFRM	Even/odd frame (For isochronous IN endpoints) For isochronous transfer, software can use this bit to control that USBHS only sends data packets for IN tokens in even or odd frames. If the current frame number's parity doesn't match with this bit, USBHS only responses with a zero-length packet. 0: Only sends data in even frames 1: Only sends data in odd frames
	DPID	Endpoint data PID (For interrupt/bulk IN endpoints) These is a data PID toggle scheme in interrupt or bulk transfer. Software should set SD0PID to set this bit before a transfer starts and USBHS maintains this bit during transfers by following the data toggle scheme described in USB protocol. 0: Data packet's PID is DATA0 1: Data packet's PID is DATA1
15	EPACT	Endpoint active This bit controls whether this endpoint is active. If an endpoint is not active, it ignores all tokens and doesn't make any response.
14:11	Reserved	Must be kept at reset value.
10:0	MPL[10:0]	This field defines the maximum packet length in bytes.

Device OUT endpoint 0 control register (USBHS_DOEP0CTL)

Address offset: 0x0B00

Reset value: 0x0000 8000

This register has to be accessed by word (32-bit)



Bits	Fields	Descriptions
31	EPEN	Endpoint enable Set by the application and cleared by USBHS. 0: Endpoint disabled 1: Endpoint enabled Software should follow the operation guide to disable or enable an endpoint.
30	EPD	Endpoint disable This bit is fixed to 0 for OUT endpoint 0.
29:28	Reserved	Must be kept at reset value.
27	SNAK	Set NAK Software sets this bit to set NAKS bit in this register.
26	CNAK	Clear NAK Software sets this bit to clear NAKS bit in this register
25:22	Reserved	Must be kept at reset value.
21	STALL	STALL handshake Software can set this bit to make USBHS send STALL handshake during an OUT transaction. USBHS will clear this bit after a SETUP token is received on OUT endpoint 0. This bit has a higher priority than NAKS bit in this register, i.e. if both STALL and NAKS bits are set, the STALL bit takes effect.
20	SNOOP	Snoop mode This bit controls the snoop mode of an OUT endpoint. In snoop mode, USBHS doesn't check the received data packet's CRC value. 0: Snoop mode disabled

		1: Snoop mode enabled
19:18	EPTYPE[1:0]	Endpoint type This field is fixed to '00' for control endpoint.
17	NAKS	NAK status This bit controls the NAK status of USBHS when both STALL bit in this register and GONS bit in USBHS_DCTL register are cleared: 0: USBHS sends data or handshake packets according to the status of the endpoint's Rx FIFO. 1: USBHS always sends NAK handshake to the OUT token. This bit is read-only and software should use CNAK and SNAK in this register to control this bit.
16	Reserved	Must be kept at reset value.
15	EPACT	Endpoint active This field is fixed to '1' for endpoint 0.
14:2	Reserved	Must be kept at reset value.
1:0	MPL[1:0]	Maximum packet length This is a read-only field, and its value comes from the MPL field of USBHS_DIEP0CTL register: 00: 64 bytes 01: 32 bytes 10: 16 bytes 11: 8 bytes

Device OUT endpoint-x control register (USBHS_DOEPxCTL) (x = 1..7, where x = endpoint_number)

Address offset: 0x0B00 + (endpoint_number × 0x20)
Reset value: 0x0000 0000

The application uses this register to control the operation of each logical OUT endpoint other than OUT endpoint 0.

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EPEN	EPD	SODDFRM/SD1PID	SEVENFRM/SD0PID	SNAK	CNAK	Reserved				STALL	SNOOP	EPTYPE[1:0]		NAKS	EOFRM/DPID
rs	rs	w	w	w	w					rw/rs	rw	rw		r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

EPACT	Reserved	MPL[10:0]
rw		rw

Bits	Fields	Descriptions
31	EPEN	Endpoint enable Set by the application and cleared by USBHS. 0: Endpoint disabled 1: Endpoint enabled Software should follow the operation guide to disable or enable an endpoint.
30	EPD	Endpoint disable Software can set this bit to disable the endpoint. Software should follow the operation guide to disable or enable an endpoint.
29	SODDFRM	Set odd frame (For isochronous OUT endpoints) This bit has effect only if this is an isochronous OUT endpoint. Software sets this bit to set EOFRM bit in this register.
	SD1PID	Set DATA1 PID (For interrupt/bulk OUT endpoints) Software sets this bit to set DPID bit in this register.
28	SEVENFRM	Set even frame (For isochronous OUT endpoints) Software sets this bit to clear EOFRM bit in this register.
	SD0PID	Set DATA0 PID (For interrupt/bulk OUT endpoints) Software sets this bit to clear DPID bit in this register.
27	SNAK	Set NAK Software sets this bit to set NAKS bit in this register.
26	CNAK	Clear NAK Software sets this bit to clear NAKS bit in this register.
25:22	Reserved	Must be kept at reset value.
21	STALL	STALL handshake Software can set this bit to make USBHS send STALL handshake during an OUT transaction. This bit has a higher priority than NAKS bit in this register and GINS in USBHS_DCTL register. If both STALL and NAKS bits are set, the STALL bit takes effect. For control OUT endpoint: Only USBHS can clear this bit when a SETUP token is received on the corresponding OUT endpoint. Software is not able to clear it. For interrupt or bulk OUT endpoint: Only software can clear this bit.
20	SNOOP	Snoop mode

		This bit controls the snoop mode of an OUT endpoint. In snoop mode, USBHS doesn't check the received data packet's CRC value. 0: Snoop mode disabled 1: Snoop mode enabled
19:18	EPTYPE[1:0]	Endpoint type This field defines the transfer type of this endpoint: 00: Control 01: Isochronous 10: Bulk 11: Interrupt
17	NAKS	NAK status This bit controls the NAK status of USBHS when both STALL bit in this register and GONS bit in USBHS_DCTL register are cleared: 0: USBHS sends handshake packets according to the status of the endpoint's Rx FIFO. 1: USBHS always sends NAK handshake to the OUT token. This bit is read-only and software should use CNAK and SNAK in this register to control this bit.
16	EOFRM	Even/odd frame (For isochronous OUT endpoints) For isochronous transfer, software can use this bit to control that USBHS only receives data packets in even or odd frames. If the current frame number's parity doesn't match with this bit, USBHS just drops the data packet. 0: Only sends data in even frames 1: Only sends data in odd frames
	DPID	Endpoint data PID (For interrupt/bulk OUT endpoints) These is a data PID toggle scheme in interrupt or bulk transfer. Software should set SD0PID to set this bit before a transfer starts and USBHS maintains this bit during transfer following the data toggle scheme described in USB protocol. 0: Data packet's PID is DATA0 1: Data packet's PID is DATA1
15	EPACT	Endpoint active This bit controls whether this endpoint is active. If an endpoint is not active, it ignores all tokens and doesn't make any response.
14:11	Reserved	Must be kept at reset value.
10:0	MPL[10:0]	This field defines the maximum packet length in bytes.

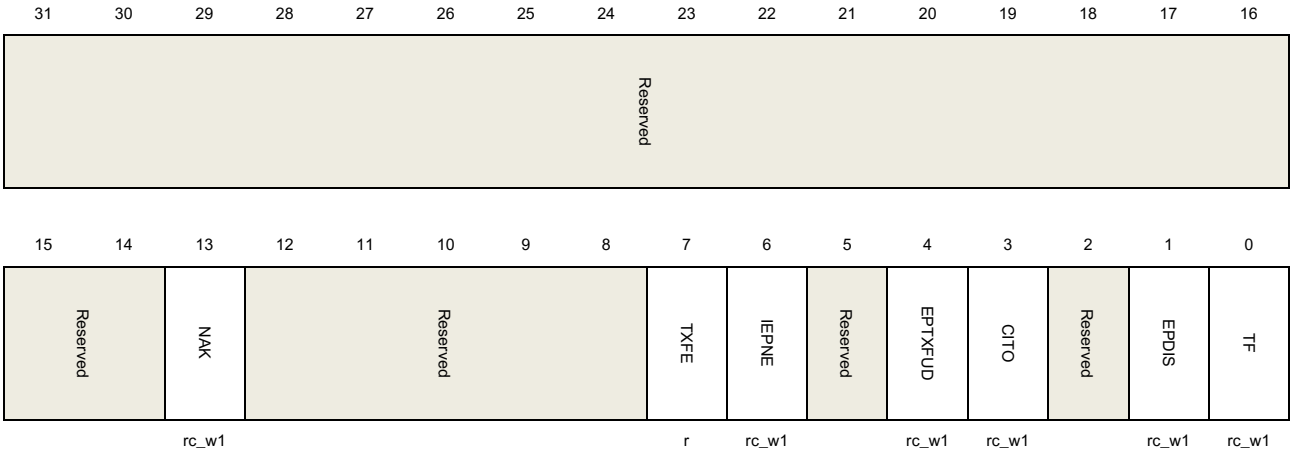
Device IN endpoint-x interrupt flag register (USBHS_DIEPxINTF) (x = 0..7, where x = endpoint_number)

Address offset: 0x0908 + (endpoint_number × 0x20)

Reset value: 0x0000 0080

This register contains the status and events of an IN endpoint, when software gets an IN endpoint interrupt, it should read this register for the respective endpoint to know the source of the interrupt. The flag bits in this register are all set by hardware and cleared by writing 1 except the read-only TXFE bit.

This register has to be accessed by word (32-bit)



Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value.
13	NAK	NAK handshake sent by USBHS USBHS sets this bit after it sends out a NAK handshake because the NAKS bit in USBHS_DIEPxCTL register is set, or there is no packet data in endpoint's Tx FIFO.
12:8	Reserved	Must be kept at reset value.
7	TXFE	Transmit FIFO empty The Tx FIFO of this IN endpoint has reached the empty threshold value defined by TXFTH field in USBHS_GAHBCS register.
6	IEPNE	IN endpoint NAK effective The setting of SNAK bit in USBHS_DIEPxCTL register takes effect. This bit can be cleared either by writing 1 to it or by setting CNAK bit in USBHS_DIEPxCTL register.
5	Reserved	Must be kept at reset value.
4	EPTXFUD	Endpoint Tx FIFO underrun This flag is triggered if the Tx FIFO has no packet data when an IN token is incoming
3	CITO	Control In Timeout interrupt This flag is triggered if the device waiting for a handshake is timeout in a control IN transaction.
2	Reserved	Must be kept at reset value.
1	EPDIS	Endpoint disabled This flag is triggered when an endpoint is disabled from the software's request.

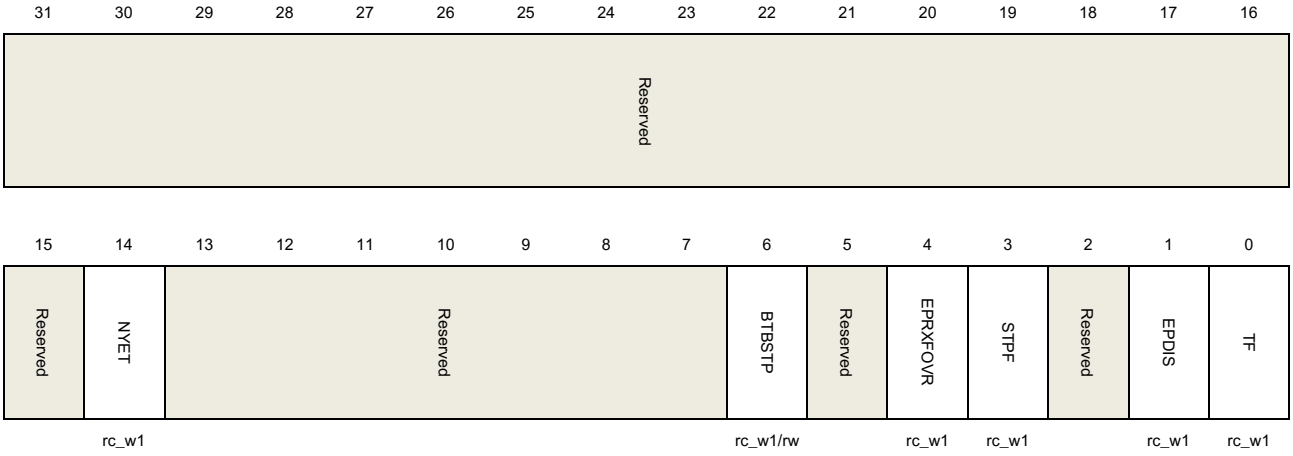
0	TF	Transfer finished This flag is triggered when all the IN transactions assigned to this endpoint have finished.

Device OUT endpoint-x interrupt flag register (USBHS_DOEPxINTF) (x = 0..7, where x = endpoint_number)

Address offset: 0x0B08 + (endpoint_number × 0x20)
Reset value: 0x0000 0000

This register contains the status and events of an OUT endpoint, when software gets an OUT endpoint interrupt, it should read this register for the respective endpoint to know the source of the interrupt. The flag bits in this register are all set by hardware and cleared by writing 1.

This register has to be accessed by word (32-bit)



Bits	Fields	Descriptions
31:15	Reserved	Must be kept at reset value.
14	NYET	NYET handshake is sent This flag is triggered if a NYET handshake is sent by USBHS.
13:7	Reserved	Must be kept at reset value.
6	BTBSTP	Back-to-back SETUP packets (Only for control OUT endpoint) This flag is triggered when a control out endpoint has received more than 3 back-to-back setup packets.
5	Reserved	Must be kept at reset value.
4	EPRXFOVR	Endpoint Rx FIFO overrun This flag is triggered if the OUT endpoint's Rx FIFO has no enough space for a packet data when an OUT token is incoming. USBHS will drop the incoming OUT data packet and send a NAK handshake in this case.

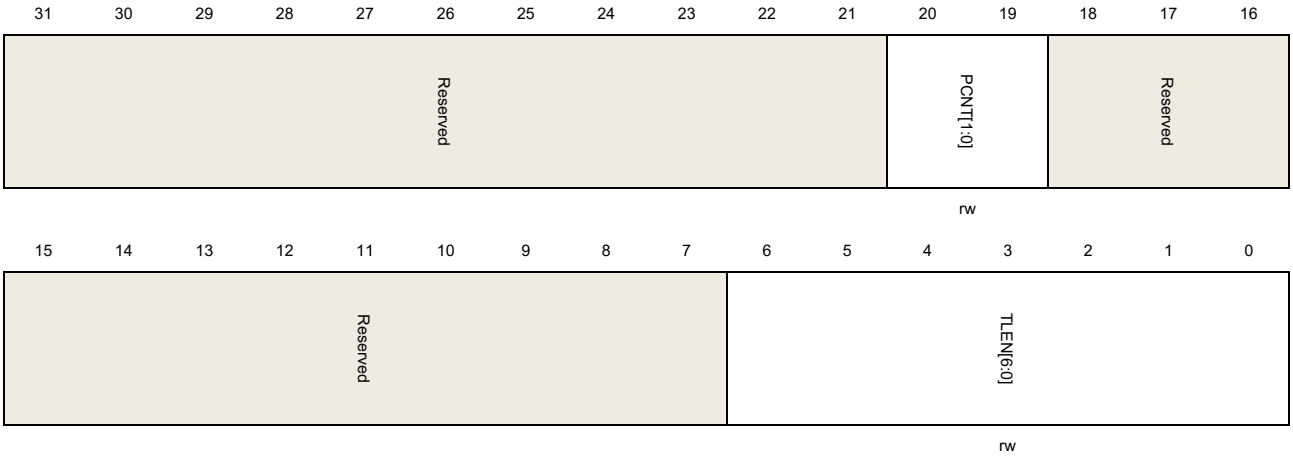
3	STPF	SETUP phase finished (Only for control OUT endpoint) This flag is triggered when a setup phase finished, i.e. USBHS receives an IN or OUT token after a setup token.
2	Reserved	Must be kept at reset value.
1	EPDIS	Endpoint disabled This flag is triggered when an endpoint is disabled from the software's request.
0	TF	Transfer finished This flag is triggered when all the OUT transactions assigned to this endpoint have finished.

Device IN endpoint 0 transfer length register (USBHS_DIEP0LEN)

Address offset: 0x0910

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)



Bits	Fields	Descriptions
31:21	Reserved	Must be kept at reset value.
20:19	PCNT[1:0]	Packet count The number of data packets desired to be transmitted in a transfer. Software should program this field before the endpoint is enabled. After the transfer starts, this field is decreased automatically by USBHS after each successful data packet transmission.
18:7	Reserved	Must be kept at reset value.
6:0	TLEN[6:0]	Transfer length The total data bytes number of a transfer. This field is the total data bytes of all the data packets desired to be transmitted in an IN transfer. Software should program this field before the endpoint is enabled. When software or DMA successfully writes a packet into the endpoint's Tx FIFO,

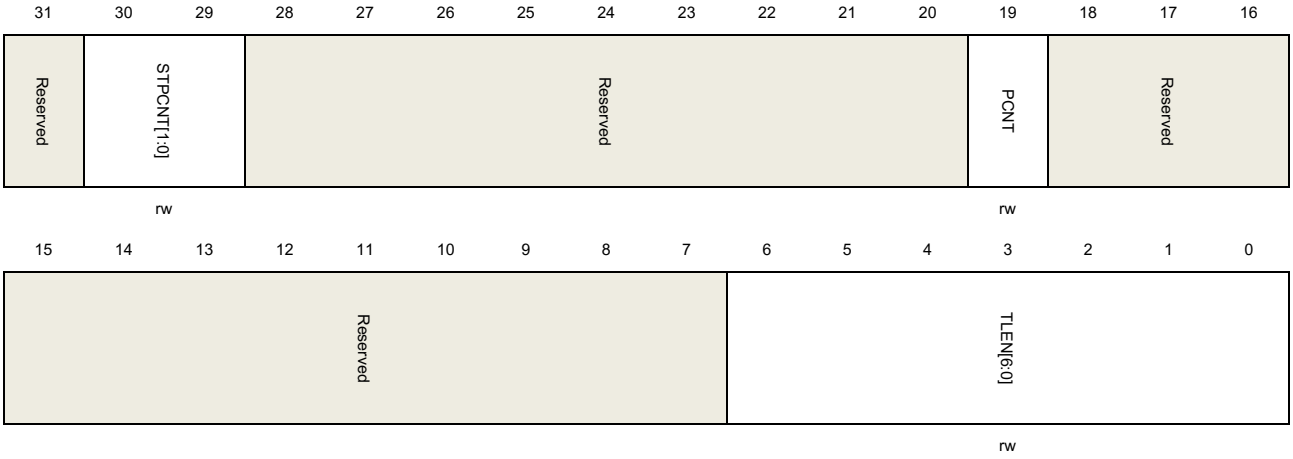
this field is decreased by the byte size of the packet.

Device OUT endpoint 0 transfer length register (USBHS_DOEP0LEN)

Address offset: 0x0B10

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)



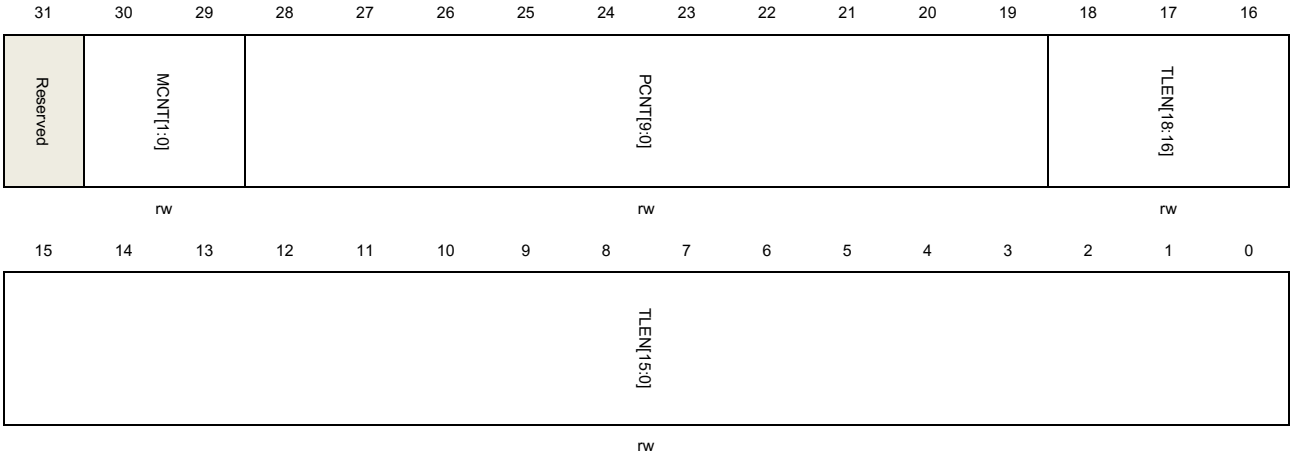
Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30:29	STPCNT[1:0]	SETUP packet count This field defines the maximum number of back-to-back SETUP packets what this endpoint can accept. Software should program this field before setup transfers. Each time a back-to-back setup packet is received, USBHS decreases this field by one. When this field reaches zero, the BTBSTP flag in USBHS_DOEP0INTF register will be triggered. 00: 0 packet 01: 1 packet 10: 2 packets 11: 3 packets
28:20	Reserved	Must be kept at reset value.
19	PCNT	Packet count The number of data packets is desired to receive in a transfer. Software should program this field before the endpoint is enabled. After the transfer starts, this field is decreased automatically by USBHS after each successful data packet reception on bus.
18:7	Reserved	Must be kept at reset value.
6:0	TLEN[6:0]	Transfer length The total data bytes number of a transfer.

This field is the total data bytes of all the data packets desired to receive in an OUT transfer. Software should program this field before the endpoint is enabled. Each time software or DMA reads out a packet from the Rx FIFO, this field is decreased by the byte size of the packet.

Device IN endpoint-x transfer length register (USBHS_DIEPxLEN) (x = 1..7, where x = endpoint_number)

Address offset: 0x910 + (endpoint_number × 0x20)
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)



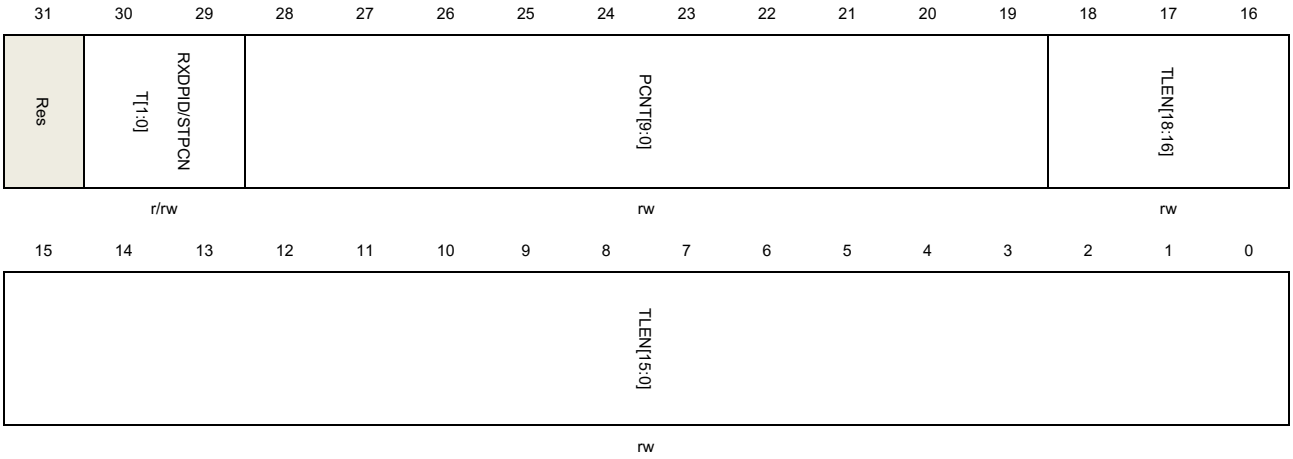
Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30:29	MCNT[1:0]	Multi count This field indicates the number of packets which should be transmitted in a frame 01: 1 packet 10: 2 packets 11: 3 packets
28:19	PCNT[9:0]	Packet count The number of data packets desired to be transmitted in a transfer. Software should program this field before the endpoint is enabled. After the transfer starts, this field is decreased automatically by USBHS after each successful data packet transmission.
18:0	TLEN[18:0]	Transfer length The total data bytes number of a transfer. This field is the total data bytes of all the data packets desired to be transmitted in an IN transfer. Software should program this field before the endpoint is enabled. When software or DMA successfully writes a packet into the endpoint's Tx FIFO,

this field is decreased by the byte size of the packet.

Device OUT endpoint-x transfer length register (USBHS_DOEPxLEN) (x = 1..7, where x = endpoint_number)

Address offset: 0x0B10 + (endpoint_number × 0x20)
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)



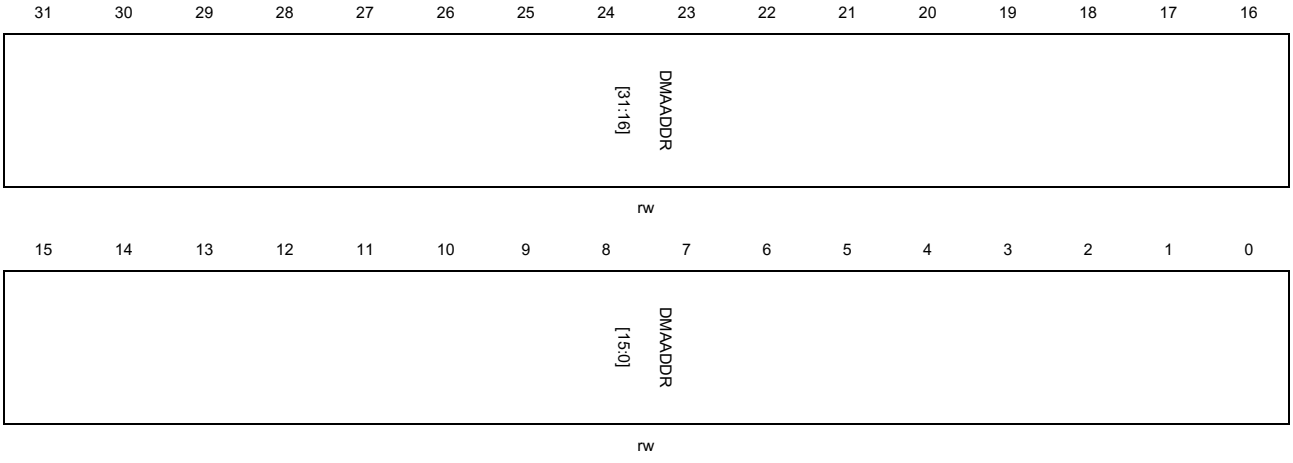
Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30:29	RXDPID[1:0]	Received data PID (For isochronous OUT endpoints) This field saves the PID of the latest received data packet on this endpoint. 00: DATA0 01: DATA2 10: DATA1 11: MDATA
	STPCNT[1:0]	SETUP packet count (For control OUT Endpoints.) This field defines the maximum number back-to-back SETUP packets this endpoint can accept. Software should program this field before setup transfers. Each time a back-to-back setup packet is received, USBHS decreases this field by one. When this field reaches zero, the BTBSTP flag in USBHS_DOEPxINTF register will be triggered. 00: 0 packet 01: 1 packet 10: 2 packets

		11: 3 packets
28:19	PCNT[9:0]	Packet count The number of data packets desired to receive in a transfer. Software should program this field before the endpoint is enabled. After the transfer starts, this field is decreased automatically by USBHS after each successful data packet reception on bus.
18:0	TLEN[18:0]	Transfer length The total data bytes number of a transfer. This field is the total data bytes of all the data packets desired to receive in an OUT transfer. Software should program this field before the endpoint is enabled. Each time software or DMA reads out a packet from the RxFIFO, this field is decreased by the byte size of the packet.

Device IN endpoint-x DMA address register (USBHS_DIEPxDMAADDR) / Device OUT endpoint-x DMA address register (USBHS_DOEPxDMAADDR) (x = 0..7, where x = endpoint_number)

Address offset:
IN endpoint: 0x0914 + (endpoint_number × 0x20)
OUT endpoint: 0x0B14 + (endpoint_number × 0x20)
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)



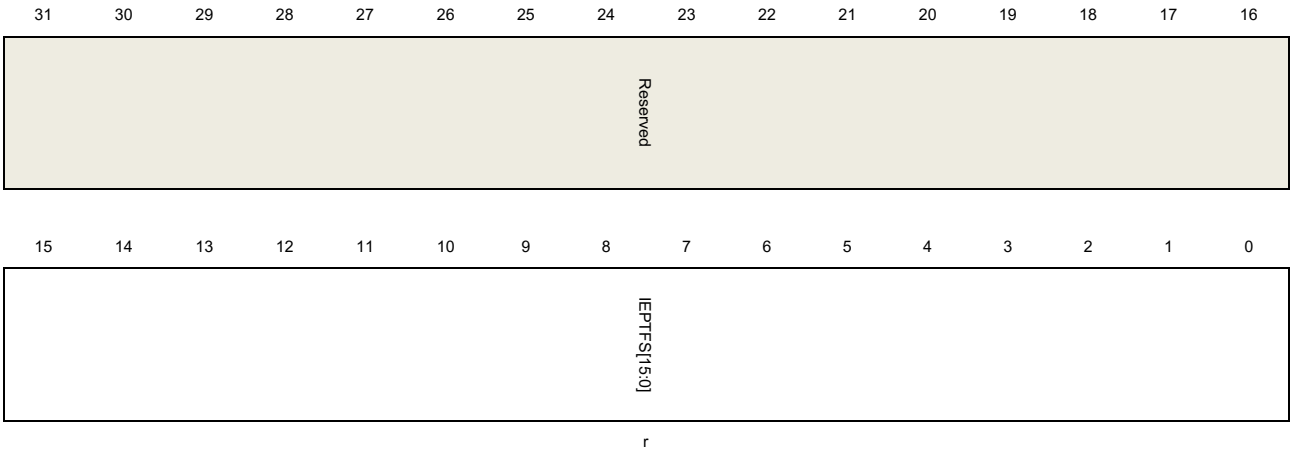
Bits	Fields	Descriptions
31:0	DMAADDR[31:0]	DMA address This field defines the endpoint's DMA address. DMA uses this address to fetch packet data for IN endpoint or write packet data for OUT endpoint.

Device IN endpoint-x transmit FIFO status register (USBHS_DIEPxTFSTAT) (x = 0..7, where x = endpoint_number)

Address offset: 0x0918 + (endpoint_number × 0x20)
Reset value: 0x0000 0200

This register contains the information of each endpoint's Tx FIFO.

This register has to be accessed by word (32-bit)

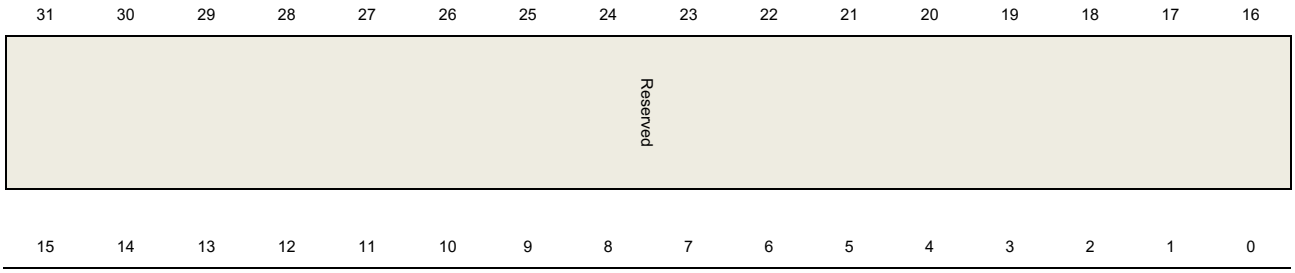


Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	IEPTFS[15:0]	IN endpoint's Tx FIFO space remaining IN endpoint's Tx FIFO space remaining in 32-bit word: 0: FIFO is full 1: 1 word available … n: n words available

53.7.4. Power and clock control register (USBHS_PWRCLKCTL)

Address offset: 0x0E00
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)



Reserved	DSLEEP	SSLEEP	SCGEN	SUSP	Reserved	SHCLK	SUCLK
	r	r	r/w	r		rw	rw

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7	DSLEEP	PHY is in deep sleep status
6	SSLEEP	PHY is in shallow sleep status
5	SCGEN	When this bit is set, the internal clock gating is enabled.
4	SUSP	PHY is in suspend status
3:2	Reserved	Must be kept at reset value.
1	SHCLK	Stop HCLK Stop the HCLK to save power. 0: HCLK is not stopped 1: HCLK is stopped
0	SUCLK	Stop the USB clock Stop the USB clock to save power. 0: USB clock is not stopped 1: UCB clock is stopped

54. Ethernet (ENET)

54.1. Overview

This chapter describes the Ethernet peripheral module. There are two media access controller (MAC) designed in Ethernet module to support 10 / 100Mbps interface speed. For more efficient datas transfer between Ethernet and memory, a DMA controller is designed in this module. The support interface protocol for Ethernet is media independent interface (MII) and reduced media independent interface (RMII). This module is mainly compliant with the following two standards: IEEE 802.3-2002 and IEEE 1588-2008.

54.2. Characteristics

MAC feature

- Support 10Mbit / s and 100Mbit / s data transfer rates.
- Support MII and RMII interface.
- Support loopback mode support for diagnosis.
- Support CSMA / CD Protocol for Half-duplex back-pressure operation.
- Support IEEE 802.3x flow control protocol. Automatic delay a pause time which is decoded from a receive pause frame after current transmitting frame complete. MAC automatically transmits pause frame or back pressure feature depending on fill level of RxFIFO in Full-duplex mode or in Half-duplex mode.
- Automatic transmission of pause frame on assertion and de-assertion of flow control input frame. Zero-quanta pause time length frame for Full-duplex operation. Support IEEE 802.3x flow control for Full-duplex operation. Support back pressure feature to the MAC core based on RxFIFO fill level (Cut-Through mode). Support IEEE 802.3x flow control for Half-duplex operation.
- Software configurable for automatic PAD / CRC generation in transmits operation.
- Software configurable for automatic PAD / CRC stripping in receives operation.
- Software configurable for frame length.
- Software configurable for inter-frame gap.
- Support different receiving filter mode.
- IEEE 802.1Q VLAN tag detection function support for reception frames.
- Support mandatory network statistics standard (RFC2819 / RFC2665).
- Two types of wakeup frame detection: LAN remote wakeup frame and AMD Magic Packet™ frames.
- Support checking checksum (Ipv4 header, TCP, UDP or ICMP encapsulated in Ipv4 or Ipv6 data format).
- Support Ethernet frame time stamping for both transmit and receive operation, which describes in IEEE 1588-2008, and 64 bits time stamps are given in each frame's status.
- Two independent FIFO for transmitting and receiving.

- Support special condition frame discards handling, e.g. late collision, excessive collisions, excessive deferral or underrun.
- In the process of frame transmission, support computation and insertion of hardware checksum under store-and-forward mode.

DMA Feature

- Two types of descriptor addressing: Ring and Chain.
- Descriptor of transmit and receive both can transfer data up to 8192 bytes.
- Software configurable normal and abnormal interrupt for many status conditions.
- Support round-robin or fixed priority to arbitrate the request of transmit and receive controller.

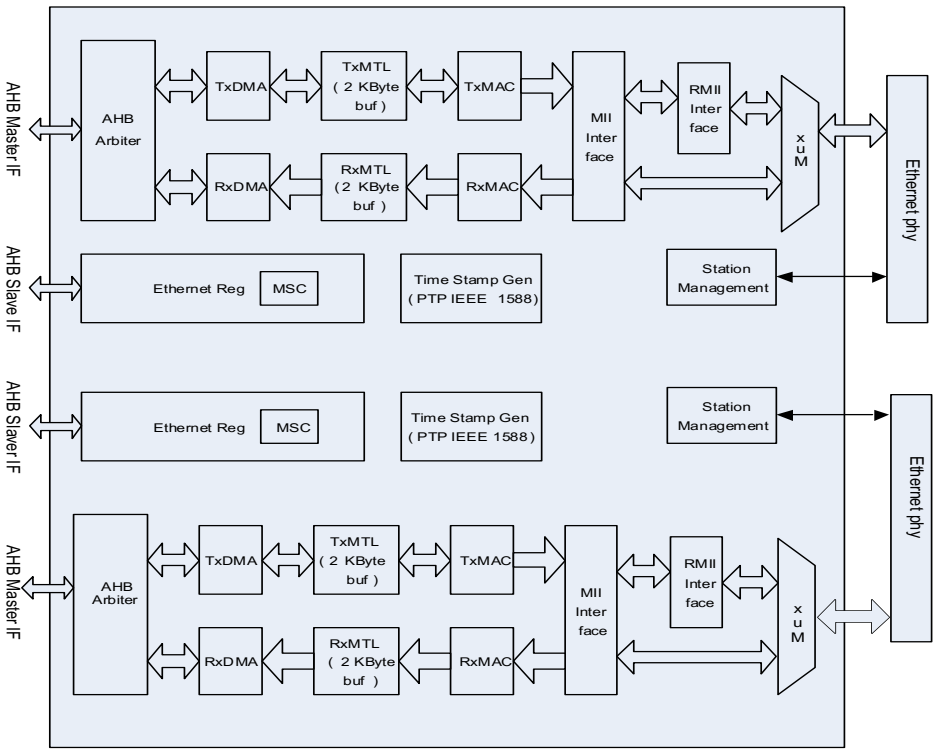
PTP Feature

- Support IEEE 1588 time synchronization function.
- Support two correction methods: Coarse or Fine.
- Support output pulse in seconds.
- Preset expected time reaching trigger and interrupt

54.2.1. Block diagram

The Ethernet module is composed of two MAC modules, two MII/RMII modules and two DMA modules by descriptor control. When using Ethernet, the user should ensure that the configured AHB clock frequency is no less than 25MHz.

Figure 54-1. ENET module block diagram



The MAC module is connected to the external PHY by MII or RMII through one selection bit (refer to SYSCFG_PMCFG0 register). The SMI (Station Management Interface) is used to configure and manage external PHY.

Transmitting data module includes:

- TxDMA controller, used to read descriptors and data from memory and writes status to memory.
- TxMTL, used to control, management and store the transmit data. TxFIFO is implemented in this module and used to cache transmitting data from memory for MAC transmission.
- The MAC transmission relative control registers, used to control frame transmit.

Receiving data module includes:

- RxDMA controller, used to read descriptors from memory and writes received frame data and status to memory.
- RxMTL, used to control, management and store reception data. RxFIFO is implemented in this module and used to temporarily store received frame data before forwarding them into the system physical memory.
- The MAC reception relative control registers, used to control frame receive and marked the receiving state. Also a receiving filter with a variety of filtering mode is implemented in MAC, used to filter out specific Ethernet frame.

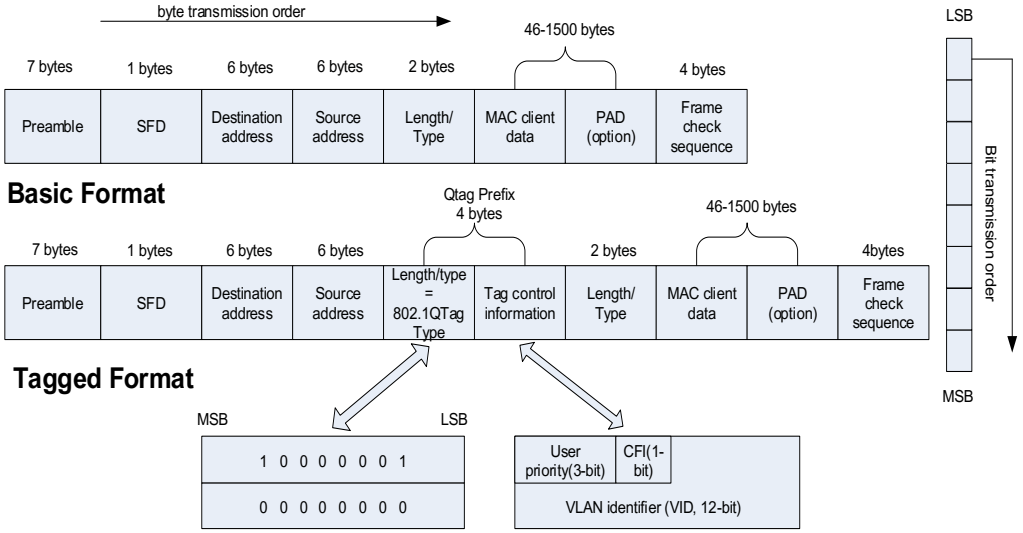
54.2.2. MAC 802.3 Ethernet packet description

Data communication of MAC can use two frame formats:

- Basic frame format.
- Tagged frame format.

Figure 54-2. MAC / Tagged MAC frame format describes the structure of the frame (Basic and Tagged) that includes the following fields:

Figure 54-2. MAC / Tagged MAC frame format



Note: The Ethernet controller transmits each byte at LSB first except FCS field.

CRC calculation data comes from all bytes in the frame except the Preamble and SFD domain. The Ethernet frame's 32-bit CRC calculation value generating polynomial is fixed 0x04C11DB7 and this polynomial is used in all 32-bit CRC calculation places in Ethernet module, as follows:

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

54.2.3. Ethernet signal description

The follow tables show the MAC module that pin is used default and remapping functions and specific configuration in MII / RMII mode. All the ENET pins below are AF output push-pull mode.

Table 54-1. Ethernet signals (MII)

Signals	Pin1(AF Number)	Pin2(AF Number)	Pin3(AF Number)	Pin4(AF Number)
ENET0_MDC	PC1 (AF11)	PF3 (AF3)		
ENET0_MDIO	PA2 (AF11)	PF2 (AF3)		
ENET0_PPS_OUT	PB5 (AF11)	PF9 (AF0)	PG8 (AF11)	
ENET0_MII_TXEN	PB11 (AF11)	PG11 (AF11)		
ENET0_MII_TXCLK	PC3 (AF11)	PF15 (AF1)	PD11 (AF6)	
ENET0_MII_TXD0	PB12 (AF11)	PG13 (AF11)	PF11 (AF1)	
ENET0_MII_TXD1	PB13 (AF11)	PG12 (AF11)	PG14 (AF11)	PF12 (AF1)
ENET0_MII_TXD2	PC2 (AF11)	PF13 (AF1)	PD11 (AF13)	PG12 (AF2)
ENET0_MII_TXD3	PB8 (AF11)	PE2 (AF11)	PF14 (AF1)	
ENET0_MII_RXDV	PA7 (AF11)	PE13 (AF17)		
ENET0_MII_RXCLK	PA1 (AF11)	PE12 (AF17)		
ENET0_MII_RXD0	PC4 (AF11)	PE8 (AF17)		

Signals	Pin1(AF Number)	Pin2(AF Number)	Pin3(AF Number)	Pin4(AF Number)
ENET0_MII_RXD1	PC5 (AF11)	PE9 (AF17)		
ENET0_MII_RXD2	PB0 (AF11)	PE10 (AF17)	PH6 (AF11)	
ENET0_MII_RXD3	PB1 (AF11)	PE11 (AF17)	PH7 (AF17)	
ENET0_MII_RXER	PB10 (AF11)	PE14 (AF17)		
ENET0_MII_CRS	PA0 (AF11)	PH2 (AF11)	PD1 (AF10)	
ENET0_MII_COL	PA3 (AF11)	PH3 (AF11)	PD0 (AF10)	
ENET1_MDC	PG6 (AF6)	PF3 (AF14)		
ENET1_MDIO	PH14 (AF6)	PC6 (AF6)	PF2 (AF14)	
ENET1_PPS_OUT	PG8 (AF6)	PF10 (AF0)		
ENET1_MII_TXEN	PG11 (AF6)	PG2 (AF9)	PC7 (AF0)	PB6 (AF8)
ENET1_MII_TXCLK	PG9 (AF6)	PB3 (AF4)	PC8 (AF6)	
ENET1_MII_TXD0	PG13 (AF6)	PG3 (AF9)	PA10 (AF3)	
ENET1_MII_TXD1	PG14 (AF6)	PG4 (AF9)	PA9 (AF3)	
ENET1_MII_TXD2	PG12 (AF6)	PG5 (AF9)	PA8 (AF5)	
ENET1_MII_TXD3	PG15 (AF6)	PC9 (AF7)	PB4 (AF10)	PG6 (AF9)
ENET1_MII_RXDV	PH11 (AF6)	PD2 (AF6)	PK2 (AF9)	
ENET1_MII_RXCLK	PH12 (AF6)	PK1 (AF9)	PC12 (AF3)	
ENET1_MII_RXD0	PH8 (AF6)	PD4 (AF9)		
ENET1_MII_RXD1	PH9 (AF6)	PD5 (AF9)		
ENET1_MII_RXD2	PH6 (AF6)	PD6 (AF9)		
ENET1_MII_RXD3	PH7 (AF6)	PD7 (AF8)		
ENET1_MII_RXER	PH10 (AF6)	PD3 (AF6)		
ENET1_MII_CRS	PH15 (AF6)	PC10 (AF2)	PG1 (AF14)	
ENET1_MII_COL	PH13 (AF6)	PG0 (AF14)	PA15 (AF3)	

Table 54-2. Ethernet signals (RMII)

Signals	Pin1(AF Number)	Pin2(AF Number)	Pin3(AF Number)	Pin4(AF Number)
ENET0_MDC	PC1 (AF11)	PF3 (AF3)		
ENET0_MDIO	PA2 (AF11)	PF2 (AF3)		
ENET0_PPS_OUT	PB5 (AF11)	PF9 (AF0)	PG8 (AF11)	
ENET0_RMII_REFCLK	PA1 (AF11)	PE12 (AF17)		
ENET0_RMII_TXEN	PB11 (AF11)	PG11 (AF11)		
ENET0_RMII_TXD0	PB12 (AF11)	PG13 (AF11)	PF11 (AF1)	
ENET0_RMII_TXD1	PB13 (AF11)	PG12 (AF11)	PG14 (AF11)	PF12 (AF1)
ENET0_RMII_CRSDV	PA7 (AF11)	PE13 (AF17)		
ENET0_RMII_RXD0	PC4 (AF11)	PE8 (AF17)		
ENET0_RMII_RXD1	PC5 (AF11)	PE9 (AF17)		
ENET1_MDC	PG6 (AF6)	PF3 (AF14)		

Signals	Pin1(AF Number)	Pin2(AF Number)	Pin3(AF Number)	Pin4(AF Number)
ENET1_MDIO	PH14 (AF6)	PC6 (AF6)	PF2 (AF14)	
ENET1_PPS_OUT	PG8 (AF6)	PF10 (AF0)		
ENET1_RMII_REFCLK	PH12 (AF6)	PC12 (AF3)	PK1 (AF9)	
ENET1_RMII_TXEN	PG11 (AF6)	PB6 (AF8)	PG2 (AF9)	PC7 (AF0)
ENET1_RMII_TXD0	PG13 (AF6)	PA10 (AF3)	PG3 (AF9)	
ENET1_RMII_TXD1	PG14 (AF6)	PG4 (AF9)	PA9 (AF3)	
ENET1_RMII_CRSDV	PH11 (AF6)	PK2 (AF9)	PD2 (AF6)	
ENET1_RMII_RXD0	PH8 (AF6)	PD4 (AF9)		
ENET1_RMII_RXD1	PH9 (AF6)	PD5 (AF9)		

54.3. Function overview

54.3.1. Interface configuration

The Ethernet block can transmit and receive Ethernet packets from an off-chip Ethernet PHY connected through the MII / RMII interface. MII or RMII mode is selected by software and carry on the PHY management through the SMI interface.

SMI: Station management interface

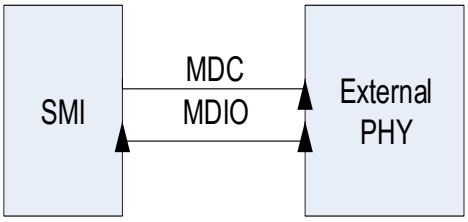
SMI is designed to access and configure PHY's configuration.

Station management interface (SMI) is performed through two wires to communicate with the external PHY: one clock line (MDC) and one data line (MDIO) . The maximum number of PHYs supported by this interface is 32. But at the same time only one register of a PHY can be accessed.

MDC and MDIO specific functions as follows:

- **MDC**: A clock of maximum frequency is 2.5 MHz. The pin remains low level when it is in idle state. The minimum high or low level lasts time of MDC must be 160ns, and the minimum period of MDC must be 400ns when it is in data transmission state.
- **MDIO**: Used to transfer data in conjunction with the MDC clock line, receiving data from external PHY or sending data to external PHY.

Figure 54-3. Station management interface signals



write operation

Applications need to write transmission data to the ENET_MAC_PHY_DATA register and operate the ENET_MAC_PHY_CTL register as follows:

1. Set the PHY device address and PHY register address, and set PW to 1, so that can select write mode.
2. Set PB bit to start transmission. In the process of transaction PB is always high until the transfer is complete. Hardware will clear PB bit automatically.

The application can be aware of whether a transaction is complete or not through checking PB bit. When PB is 1, it means the application should not change the PHY address register contents and the PHY data register contents because of operation is running. Before writing PB bit to 1, application must poll the PB bit until it is 0.

Read operation

Applications need to operate the ENET_MAC_PHY_CTL register as follows:

1. Set the PHY device address and PHY register address and set PW to 0, so that can select read mode.
2. Set PB bit to start reception. In the process of transaction PB is always high until the transfer is complete. Hardware will clear PB bit automatically.

The application can be aware of whether a transaction is complete or not through checking PB bit. When PB is 1, it means the application should not change the PHY address register contents and the PHY data register contents because of operation is running. Before writing PB bit to 1, application must poll the PB bit until it is 0.

Note: Because the PHY register address 16-31 register function is defined by each manufacturer, access different PHY device's this part should see according to the manufacturer's manual to adjust the parameters of applications. Details of catalog that firmware library currently supports the PHY device can refer to firmware library related instructions.

Clock configuration

The SMI clock is generated by dividing application clock (AHB clock). In order to guarantee the MDC clock frequency is no more than 2.5MHz, application should set appropriate division factor according to the different AHB clock frequency. *Table 54-3. Clock range* lists the frequency factor corresponding AHB clock selection.

Table 54-3. Clock range

AHB clock	MDC clock	Bits CLR[2:0] in ENET_MAC_PHY_CTL
300~375MHz	AHB clock/162	0x7
300~350MHz	AHB clock/142	0x6
250~300MHz	AHB clock/124	0x5

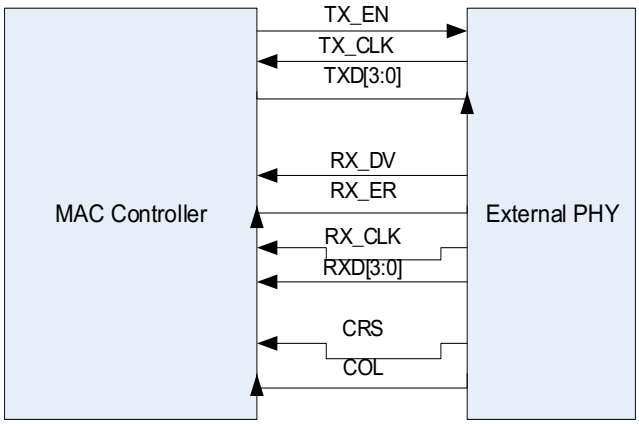
AHB clock	MDC clock	Bits CLR[2:0] in ENET_MAC_PHY_CTL
150~250MHz	AHB clock/102	0x4
35~60MHz	AHB clock/26	0x3
20~35MHz	AHB clock/16	0x2
100~150MHz	AHB clock/62	0x1
60~100MHz	AHB clock/42	0x0

MII / RMII selection

Before enable the Ethernet controller clocks or when the Ethernet controller is under the reset state, the application can select the MII or RMII mode by configuring bit 23 or bit 22 of the SYSCFG_PMCFG0 register ENET0_PHY_SEL/ENET1_PHY_SEL. The MII mode is set by default.

MII: Media independent interface

Figure 54-4. Media independent interface signals



- **MII_TX_CLK**: clock signal for transmitting data. For the data transmission speed of 10Mbit / s, the clock is 2.5MHz, for the data transmission speed of 100Mbit / s, the clock is 25MHz.

- **MII_RX_CLK**: Clock signal for receiving data. For the data transmission speed of 10Mbit / s, the clock is 2.5MHz, for the data transmission speed of 100Mbit / s, the clock is 25MHz.

- **MII_TX_EN**: Transmission enable signal. This signal must be active when the first bit of the data preamble occurs. And it needs to remain active before the all bits transmission is completed.

- **MII_TXD[3:0]**: Transmit data line, each 4 bit data transfer, data is valid when the MII_TX_EN signal is effective. The PHY would ignore the transmitted data when the MII_TX_EN signal is non-effective.

- **MII_CRS**: Carrier sense signal, only working in Half-duplex mode and controlled by the PHY. This signal does not need to be synchronized with the MII_TX_CLK and MII_RX_CLK. When it is active, means that the transmit or receive medium is not in idle state. MII_CRS signal

remains active until the transmit and receive medium are both in idle state.

- **MII_COL**: Collision detection signal, only working in Half-duplex mode, controlled by the PHY. This signal does not need to be synchronized with the MII_TX_CLK and MII_RX_CLK. It is active when a collision on the medium is detected and it will remain active while the collision condition continues.

- **MII_RXD[3:0]**: Receive data line, each 4 bit data transfer; data are valid when the MII_RX_DV signal is effective. Depending on the state of MII_RX_DV and MII_RX_ER, the MII_RXD[3:0] value can be used to convey some specific information (see ***Table 54-4. Rx interface signal encoding***).

- **MII_RX_DV**: Receive data valid signal, controlled by the PHY. This signal must be active when the first 4-bits of the frame data occurs. And it needs to remain active before the all bits transmission is completed. It must be inactive prior to the first clock cycle that follows the final 4-bit. MII_RX_DV signals should be effective before the SFD field appearing to ensure that receive the correct frame.

- **MII_RX_ER**: Receive error signal. In order to indicate that MAC detected an error in the receiving process, the MII_RX_ER signal must remain effective for one or more clock cycles (MII_RX_CLK). The specific error reason needs to cooperate with the state of the MII_RX_DV and the MII_RXD[3:0] data value (see ***Table 54-4. Rx interface signal encoding***).

Table 54-4. Rx interface signal encoding

Signal	Normal inter-frame		Normal reception frame data	False carrier indication	Data reception with errors
MII_RX_ER	0	1	0	1	1
MII_RX_DV	0	0	1	0	1
MII_RXD[3:0]	0000 to 1111	0000	0000 to 1111	1110	0000 to 1111

MII clock sources

The user needs to provide an external 25MHz clock to the external PHY to generate both TX_CLK and RX_CLK clock. This 25MHz clock does not require the same one with MAC clock. It can use the external 25MHz crystal or the output clock of microcontroller's CK_OUT0 pin. If the clock source is selected from CK_OUT0 pin, the MCU needs to configure the appropriate PLL to ensure the output frequency of CK_OUT0 pin is 25MHz.

RMII: Reduced media independent interface

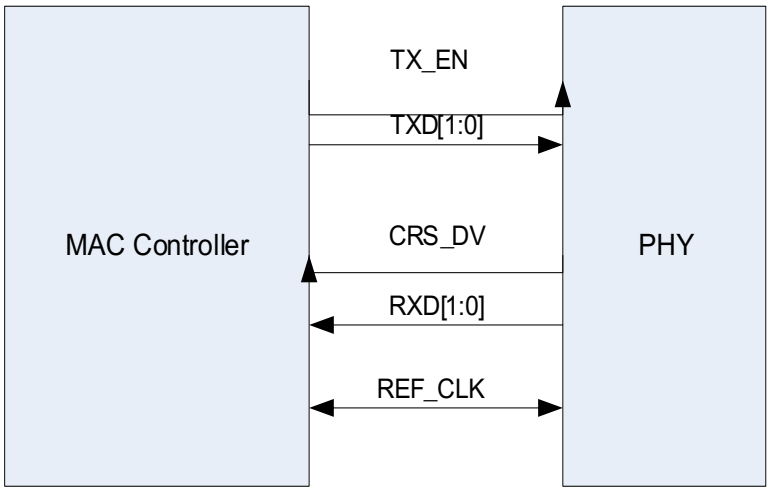
The reduced media-independent interface (RMII) specification reduces the pin count during Ethernet communication. The MII specification defines 16 pins for data and control, according to the IEEE 802.3 standard. The RMII specification is dedicated to reduce the pin count to 7.

RMII characteristics:

- The clock signal needs to be increased to 50MHz and only one clock signal.
- MAC and external PHY use the same clock source.

- Using the 2-bit wide data transceiver.

Figure 54-5. Reduced media-independent interface signals



RMII clock sources

To ensure the synchronization of the clock source, the same clock source is selected for the MAC and external PHY which is called REF_CLK. The REF_CLK input clock can be connected to the external 50MHz crystal or microcontroller CK_OUT0 pin. If the clock source is from CK_OUT0 pin, then the MCU needs to configure the appropriate PLL to ensure the output frequency of CK_OUT0 pin is 50MHz.

MII / RMII bit transmission order

No matter which interface (MII or RMII) is selected, the bit order of transmit / receive is LSB first.

The deference between MII and RMII is bit number and sending times. MII is low 4bits first and then high 4bits, but RMII is the lowest 2bits, low 2bits, high 2bits and the highest 2bits.

For example: a byte value is: 10011101b (left to right order: high to low).
Transmission order for MII use 2 cycles: 1101 -> 1001 (left to right order: high to low, 1101 corresponding to MII_T/RXD[3] to MII_T/RXD[0]).
Transmission order for RMII use 4 cycles: 01 -> 11 -> 01 -> 10 (left to right order: high to low, 01 corresponding to RMII_T/RXD[1] to RMII_T/RXD[0]).

54.3.2. MAC function overview

The MAC module can work in two modes (Half-duplex mode and Full-duplex mode). The Half-duplex mode, with the CSMA/CD algorithm to contend for using of the physical medium, at the same time only one transmission direction is active between two stations is active. The Full-duplex mode, simultaneous transmission and reception without any conflict mode, if all of the following conditions are satisfied: 1) PHY supports the feature of transmission and

reception operations at the same time. 2) Only two devices connect to the LAN and the two devices are both configured for Full-duplex mode.

MAC module can achieve the follows functions: 1) The data packaging (transmission and reception), that includes detecting / decoding frame and delimitating frame boundary; handling source address and destination address; detecting error conditions. 2) The Medium access management in Half-duplex mode, that includes allocating medium in order to prevent conflicts; deal with conflicts.

Transmission process of MAC

All transactions are controlled by the dedicated DMA controller and MAC in Ethernet. After received the sending instruction, the TxDMA fetches the transmit frames from system memory and pushes them into the TxFIFO, then the data in TxFIFO are poped to MAC for sending on MII / RMII interface. The method of popping is according to the selected TxFIFO mode (Cut-Through mode or Store-and-Forward mode, the specific definition sees the next paragraph). For convenient, application can configure automatically hardware calculated CRC and insert it to the FCS domain of Ethernet frame function. The entire transmission process complete when the MAC received the frame termination signal from transmit FIFO. When transmission completed, the transmission status information will be composed of MAC and write return to the DMA controller, the application can query through the DMA current transmit descriptor.

Operation for popping data from FIFO to the MAC has two modes:

- In Cut-Through mode, the data in FIFO is ready to be popped to MAC once the number of bytes in the FIFO exceeds or equals the configured threshold level or when the end-of-frame flag in descriptor is written. User can configure the threshold level through the TTHC[2:0] in ENET_DMA_CTL
- In Store-and-forward mode, the data is ready to be popped to the MAC core only after complete frame is stored in the FIFO. But there is another condition where the frame is not completely written into the FIFO, and FIFO will also take out data. This is when the transmitted Ethernet frame is bigger than FIFO size, the frame is popped towards the MAC before the transmit FIFO becomes full.

Handle special cases

In the transmission process, due to the insufficient TxDMA descriptor or misuse of FTF bit in ENET_DMA_CTL register (when this bit is set, it will clear FIFO data and reset the FIFO pointer, after clear operation is completed, it will be reset), there will be a transmit data underflow fault occurs because of insufficient data in FIFO. At the same time MAC will identify such data underflow state and write relevant status flag.

If one transmit frame uses two TxDMA descriptors for sending data, then the first segment (FSG) and the last segment (LSG) of the first descriptor should be 10b and the second ones should be 01b.If both the FSG bit of the first and the second descriptor are set and the LSG bit in the first descriptor is reset, then the FSB bit of the second descriptor will be ignored and these two descriptors are considered to sending the only one frame.

If the byte length of one transmission MAC frame's data field is less than 46 (for Tagged MAC frame is less than 42), application can configure the MAC for automatically adding a load of content of '0' bit to make the byte length of frame's data field in accordance with the relevant domain of definition of IEEE802.3 specification. At the same time, if automatically adding zeros function is performed, the MAC will certainly calculate CRC value of the frame and append it to the frame's FCS field domain no matter what configuration of DCRC bit in the descriptor is.

Transmission management of MAC

Jabber timer

In case of one station occupies the PHY for a long time, there is a jabber timer designed for cutting off the frame whose length is more than 2048 bytes. By default, jabber timer is enabled so when application is transmitting a frames whose byte length is more then 2048, the MAC will only transmit 2048 bytes and drop the last ones.

Collision condition solve mechanism - Re-transmission

When the MAC is running under Half-duplex mode, collision may happen when MAC is transmitting frame data on interface. When no more than 96 bytes data popped from FIFO towards MAC and collision condition occurs, the re-transmission function is active. In this case, MAC will stop current transmitting and then read frame data from FIFO again and send them on interface again. When more than 96 bytes data popped from FIFO towards MAC and collision condition occurs, MAC will abort transmitting current frame data and not re-transmit it. Also MAC will set late collision flag in descriptor to inform application.

Transmit FIFO flush operation

Application can clear TxFIFO and reset the FIFO data pointer by setting FTF bit of ENET_DMA_CTL register. The flush operation will be executed at once no matter whether TxFIFO is popping data to MAC. This results in an underflow event in the MAC transmitter, and the makes frame transmission abort. At the same time, MAC returns state information of frame and transmit status words transferred to the application. The status of such a frame is marked with both underflow and frame flush events (FRMF and UFE bits in Transmit Descriptor0). When the transmit data in TxFIFO is flushed, the transmit status word will be written back to descriptor. After the status is written, the flush operation is complete. When a flush operation is received, all the following data which should be popped from TxFIFO into MAC will be dropped unless a new FSG bit of descriptor is received. After operation completed, the FTF bit of ENET_DMA_CTL register is then automatically cleared.

Transmit inter-frame gap management

MAC can manage the interval time between two frames. This interval time is called frame gap time. For Full-duplex mode, after complete sending a frame or MAC entered idle state, the gap time counter starts counting. If another transmit frame presents before this counter has not reach the configured IGBS bit time in ENET_MAC_CFG register, this transmit frame will

be pended unless the counter reach the gap time. But if the second transmit frame presents after the gap time counter has reached the configured gap time, this frame will send immediately. For Half-duplex mode, the gap time counter follows the Truncated Binary Exponential Backoff algorithm. Briefly speaking, the gap time counter starts after the previous frame has completed transmitting on interface or the MAC entered idle state, and there are three conditions may occur during the gap time:

- The carrier sense signal active in the first 2 / 3 gap period. In this case, the counter will reload and restart.
- The carrier sense signal active in the last 1 / 3 gap period. In this case, the counter will not reload but continue counting, and when reaches gap time, the MAC sends the second frame.
- The carrier sense signal not active during the whole gap period. In this case, the counter stops after reaches the configured gap time and sends frame if the second frame has pended.

Address receive module

The MAC filter is divided into error filtering (such as too short frame, CRC error and other bad frame filtering) and address filtering. This section mainly describes the address filtering. Address filtering use the static physical address (MAC address) filter and hash list filter for implementing the function. If the FAR bit in the ENET_MAC_FRMF register is '0' (by default), only the frame passed the filter will be received. This function is configured according to the parameters of the application (frame filter register) to filter the destination or / and source address of unicast or multicast frame (The difference between an individual address and a group address is determined by I / G bit in the destination address field) and report the result of the corresponding address filtering. The frame not pass through the filter will be discarded.

Note: If the FAR bit in the ENET_MAC_FRMF register is set to 1, frames are all thought passed the filter. In this case, even the filter result will also be updated in receive descriptor but the result will not affect whether current frame passes the filter or not.

Unicast frame destination address filter

For a unicast frame, application has two modes for filtering: the one is using static physical address (by setting HUF bit to '0'), the other is using hash list (by setting HUF bit to '1').

- Static physical address (SPA) filtering.

In the filter mode, MAC supports using four MAC addresses for unicast frame filtering. In this way, the MAC compares all 6 bytes of the received unicast address to the programmed MAC address. MAC address 0 is always used and MAC address 1 to address 3 can be configured to use or not. Each byte of MAC address 1 to MAC address 3 register can be masked for comparison with the corresponding destination address byte of received frame by setting the corresponding mask byte bits (MB) in the corresponding register.

- Hash list filtering

In this filter mode, MAC uses a HASH mechanism. MAC uses a 64-bit hash list to filter the received unicast frame. This filter mode obeys the followings two filtering steps:

1. The MAC calculates the CRC value of the received frame's destination address.
2. Using the high 6 bits of the calculated CRC value as the index to retrieve the hash list. If the corresponding value of hash list is 1, the received frame passes through the filter, conversely, fail the filter.

The advantage of this type of filter is that it can cover any possible address just using a small table. But the disadvantage is that the filter is imperfect and sometimes the frames should be dropped are also be received by mistake.

Unicast frame source address filter

Enable MAC address 1 to MAC address 3 register and set the corresponding SAF bit in the MAC address high register, the MAC compares and filter the source address (SA) field in the received frame with the values programmed in the SA registers. MAC also supports the group filter on the source address. If the SAFLT bit in frame filter register ENET_MAC_FRMF is set, MAC drops the frame that failed the source address filtering; meanwhile the filter result will reflect by SAFF bit in Receive Descriptor0 of DMA receive descriptor. When the SAFLT bit is set, the destination address filter is also at work, so the result of the filter is simultaneous determined by DA and SA filter. This means that, as long as a frame does not pass any one of the filters (DA filter or SA filter), it will be discarded. Only a frame passing the entire filter can be forwarded to the application.

Multicast frame destination address filter

Application can enable the multicast frame MAC address filtering by cleaning the MFD bit in register ENET_MAC_FRMF. By configuring the value of HMF bit in ENET_MAC_FRMF register application can choose two ways just like unicast destination address filtering for address filtering.

Broadcast frame destination address filter

At default, the MAC unconditionally receives the broadcast frames. But when setting BFRMD bit in register ENET_MAC_FRMF, MAC discards all received broadcast frames.

Hash or perfect address filter

By setting the HPFLT bit in the ENET_MAC_FRMF register and setting the corresponding HUF (for unicast frame) or HMF (for multicast frame) bit in the ENET_MAC_FRMF register, the destination address (DA) filter can be configured to pass a frame when its DA matches either the hash list filter or the static physical address filter.

Reverse filtering operation

MAC can reverse filter-match result at the final output whether the destination address filtering or source address filtering. By setting the DAIFLT and SAIFLT bits in ENET_MAC_FRMF

register, this address filter reverse function can be enabled. DAIFLT bit is used for unicast and multicast frames' DA filtering result, SAIFLT bit is used for unicast and multicast frames SA filtering result.

The *Table 54-5. Destination address filtering table* and *Table 54-6. Source address filtering table* summarize the destination address and source address filters working condition at different configuration.

Table 54-5. Destination address filtering table

Frame Type	PM	HPFLT	HUF	DAIFLT	HMF	MFD	BFRMD	DA filter operation
Broadcast	1	-	-	-	-	-	-	Pass
	0	-	-	-	-	-	0	Pass
	0	-	-	-	-	-	1	Fail
Unicast	1	-	-	-	-	-	-	Pass all frames
	0	-	0	0	-	-	-	Pass on perfect / group filter match
	0	-	0	1	-	-	-	Fail on perfect / group filter match
	0	0	1	0	-	-	-	Pass on hash filter match
	0	0	1	1	-	-	-	Fail on hash filter match
	0	1	1	0	-	-	-	Pass on hash or perfect / group filter match
	0	1	1	1	-	-	-	Fail on hash or perfect / group filter match
Multicast	1	-	-	-	-	-	-	Pass all frames
	-	-	-	-	-	1	-	Pass all frames
	0	-	-	0	0	0	-	Pass on perfect / group filter match and drop PAUSE control frames if PCFRM = 0x
	0	0	-	0	1	0	-	Pass on hash filter match and drop PAUSE control frames if PCFRM = 0x
	0	1	-	0	1	0	-	Pass on hash or perfect / group filter match and drop PAUSE control frames if PCFRM = 0x
	0	-	-	1	0	0	-	Fail on perfect / group filter match and drop PAUSE control frames if PCFRM = 0x
	0	0	-	1	1	0	-	Fail on hash filter match and drop PAUSE control frames if PCFRM = 0x
	0	1	-	1	1	0	-	Fail on hash or perfect / group filter match and drop PAUSE control frames if PCFRM = 0x

Table 54-6. Source address filtering table

Frame type	PM	SAIFLT	SAFLT	SA filter operation
Unicast	1	-	-	Pass all frames
	0	0	0	Pass status on perfect / group filter match but do not drop frames that fail
	0	1	0	Fail status on perfect / group filter match but do not drop frame
	0	0	1	Pass on perfect / group filter match and drop frames that fail
	0	1	1	Fail on perfect / group filter match and drop frames that fail

Promiscuous mode

If the PM bit in ENET_MAC_FRMF register is set, promiscuous mode is enable. Then the address filter function is bypassed, all frames are thought passed through the filter. At the same time the receive status information DA / SA error bit is always '0'.

Pause control frame filter

When MAC received pause frame, it will detect 6 bytes DA field in the frame. If UPFDT bit in ENET_MAC_FCTL register is 0, it is determined by whether the value of the DA field conforms to the unique value (0x0180C2000001) with IEEE-802.3 specification control frames. If UPFDT bit in ENET_MAC_FCTL register is set, MAC additionally compares DA field with the programmed MAC address for bit match. If DA field match and receive flow control is enabled (RFCEN bit in ENET_MAC_FCTL register is set), the corresponding pause control frame function will be triggered. Whether this filter passed pause frame is forwarded to memory is depending on the PCFRM[1:0] bit in ENET_MAC_FRMF register.

Reception process of MAC

Received frames will be pushed to the RxFIFO. The MAC strips the preamble and SFD of the frame, and starts pushing the frame data beginning with the first byte following the SFD to the RxFIFO. If IEEE 1588 time stamp function is enabled, the MAC will record the current system time when a frame's SFD is detected. If the frame passes the address filter, this time stamp is passed on to the application by writing it to descriptor.

The MAC can automatically strip PAD and FCS field data when the length / type field of received frame is less than 1536 if APCD bit is set. MAC pushes the data of the frame into RxFIFO up to the count specified in the length / type field, then starts dropping bytes (including the FCS field). If the value of length / type field is greater than or equal to 0x600, the automatically strip FCS field function is configured by the TFCD bit regardless of APCD.

If the watchdog timer is enabled (WDD bit in ENET_MAC_CFG is reset), a frame has more than 2048 bytes will be cut off receiving when has received 2048 bytes. If the watchdog timer

is disabled, the MAC can extend the max receiving data bytes to 16384, any data beyond this number will be cut off.

When RxFIFO works at Cut-Through mode, it starts popping out data from RxFIFO when the number of FIFO is greater than threshold value (RTHC bits in ENET_DMA_CTL register). After all data of a frame pop out, receive status word is sent to DMA for writing back to descriptor. In this mode, if a frame has started to forward to application by DMA from FIFO, the forwarding will continue until the frame is end even if frame error is detected. Although the error frame is not discarded, the error status will reflect in descriptor status field.

When RxFIFO works at Store-and-Forward mode (set by RSFD bit in ENET_DMA_CTL), DMA reads frame data from RxFIFO only after RxFIFO has completed received the whole frame. In this mode, if the MAC is configured to discard all error frames, then only valid frames without any error can be read out from RxFIFO and forward to the application. Once the MAC detects an SFD signal on the interface, a receive operation is started. The MAC strips the preamble and SFD before processing the frame. The header fields are checked by filtering and the FCS field used to verify the CRC for the frame. The frame is discarded by MAC if it fails to pass the address filter.

Reception management of MAC

Receive operation on multi-frame handling

It is different from transmit operation, after receiving the last byte of a frame, the MAC can judge the status of the receiving operation, so the second received frame's forwarding is surely followed by the first received frame data and status.

Error handling

- If RxFIFO becomes full but the last received byte is not the end of frame (EOF), the RxFIFO will discard the whole frame data and return an overflow status. Also the counter of counting the overflow condition times will plus 1.
- If the RxFIFO is configured in Store-and-Forward mode, the MAC can filter and discard all error frames. But according to the configuration of FERF and FUF bit in ENET_DMA_CTL register, RxFIFO can also receive and forward such error frame and the frame that length is less than the minimum length.
- If the RxFIFO is configured in Cut-Through mode, not all the error frames can be dropped. Only when the start of frame (SOF) has not been read from RxFIFO and the receive frame has been detected error status, the RxFIFO will discard the whole error frame.

Flow control module

The MAC manages transmission frame through back pressure (in Half-duplex mode) and the pause control frame (in Full-duplex mode) for flow control.

- Half-duplex mode flow control: Back Pressure

When MAC is configured in Half-duplex mode, there are two conditions to trigger the back

pressure feature. Both of the two conditions are triggered to enable back pressure function which is implemented by sending a special pattern (called jam pattern) 0x5555 5555 once to notify conflict to all other sites. The first condition is triggered by application setting the FLCB / BKPA bit in ENET_MAC_FCTL register. The second condition occurs during receiving frame. When MAC receiver is receiving frame, the byte number of RxFIFO is more and more great. When this number is greater than the high threshold (RFA bits in ENET_MAC_FCTH), MAC will set the back pressure pending flag. If this flag is set and a new frame presents on interface, MAC will send a jam pattern to delay receiving this new frame a back pressure time. After this back pressure time is end, external PHY will send this new frame again. If the number of the RxFIFO is not less than low threshold (RFD bits in ENET_MAC_FCTH) during this back pressure time, a jam pattern is send again. If the number of the RxFIFO is less than low threshold (RFD bits in ENET_MAC_FCTH) during this back pressure time, MAC resets the back pressure pending flag and is enable to receive the new frame instead of sending jam pattern.

■ Full-duplex mode flow control: Pause Frame

The MAC uses a mechanism named "pause frame" for flow control in Full-duplex mode. Receiver can send a command to the sender for informing it to suspend transmission a period of time. If the application sets transmit flow control bit TFCEN in ENET_MAC_FCTL register, MAC will generate and transmit a pause frame when either of two conditions is satisfied in Full-duplex mode. There are two conditions to start transmit pause frames:

1. Application sets FLCB / BKPA bit in ENET_MAC_FCTL register to immediately send a pause frame. When doing this, MAC sends a pause frame right now with the pause time value PTM configured in ENET_MAC_FCTL register. If application considers the pause time is no need any more because the transmit frame can be transmitted without pause time, it can end the pause time by setting the pause time value PTM bits in ENET_MAC_FCTL register to 0 and set FLCB / BKPA bit to send this zero pause time frame.

2. MAC automatically sends pause time when the RxFIFO is in some condition. When MAC is receiving frame, RxFIFO will be fill in many receive data. At same time RxFIFO pops out data to RxDMA for forwarding to memory. If the popping frequency is lower than MAC pushing frequency, the number of bytes in RxFIFO is getting great. Once the data amount in RxFIFO is greater than the active threshold value (RFA bits in ENET_MAC_FCTH) of flow control, MAC will send a pause frame with PTM value in it. After sending pause frame, MAC will start a counter with configured reload value PLTS in ENET_MAC_FCTL register, when configured PLTS time has spent, the MAC will check RxFIFO again. If the byte number in RxFIFO is also greater than active threshold value, the MAC sends a pause time again. When the byte number of RxFIFO is lower than the de-active threshold value, MAC maybe send a pause frame with zero time value in frame's pause time field if DZQP bit in ENET_MAC_FCTL register is reset. This zero-pause time frame can inform send station that RxFIFO is almost empty and can receive new data again.

The MAC manages reception frames through follow method for flow control: In Full-duplex mode, the MAC can detect the pause control frames, and perform it by suspending a certain time which is indicated in pause time field of detected pause control frame and then to transmit data. This function can set by RFCEN bit in ENET_MAC_FCTL register. If this function is not enabled, the MAC will ignore the received pause frames. If this function is enabled, MAC can decode this frame. Type field, opcode field and pause time field in the frame are all recognized by the MAC. During the pause time period, if MAC received a new pause frame, the new pause time filed value is loaded to the pause time counter immediately. If the new pause time filed is zero, then the pause time counter stops and transmit operation recovers. Application can configure PCFRM bit in ENET_MAC_FRMF register to decide the solving method for such control frame.

Checksum offload engine

The MAC supports transmit checksum offload. This feature can calculate checksum and insert it in the transmit frame, and detect error in the receive frame.

The follows describes the operation of transmit checksum offload.

Note: This function is enabled only when the TSFD bit in the ENET_DMA_CTL register is set (TxFIFO is configured to Store-and-Forward mode) and application must ensure the TxFIFO deep enough to store the whole transmit frame. If the depth of the TxFIFO is less than the frame length, the MAC only does calculation and insertion for Ipv4 header checksum field.

Refer to IETF specifications RFC 791, RFC 793, RFC 768, RFC 792, RFC 2460 and RFC 4443 for Ipv4, TCP, UDP, ICMP, Ipv6 and ICMPv6 packet header specifications, respectively.

■ IP header checksum

If the value is 0x0800 in type field of Ethernet frame and the value is 0x4 in the IP datagram's version field, checksum offload module marks the frame as Ipv4 package and calculated value replace the checksum field in frame. Because of Ipv6 frame header does not contain checksum field, the module will not change any value of the Ipv6's header field. After IP header checksum calculation end, the result is stored in IPHE bit (In Transmit Descriptor0). The following shows the conditions under which the IPHE bit can be set:

- For Ipv4 frame type:

A) . Type field is 0x0800 but version filed in IP header is not 0x4.
B) . Ipv4 header length field value is greater than total frame byte length.
C) . The value of Ipv4 header length field is less than 0x5 (20 bytes).

- For Ipv6 frame type:

A) . Type field is 0x86dd but version field in IP header is not 0x6.
B) . Before the Ipv6 standard header or extension header has been completely received the frame is end. The length of Ipv6 standard header is 40 bytes, and the extension header contains corresponding header length field.

■ TCP / UDP / ICMP payload checksum

The checksum offload module processes the Ipv4 or Ipv6 header (including extension headers) and marks the type of frame (TCP, UDP or ICMP).

But when the following frame cases are detected, the checksum offload function will be bypassed and these frames will not be processed by the checksum offload module:

- Incomplete Ipv4 or Ipv6 frames.
- IP frames with security features (e.g. authentication header, security payload).
- IP frames without TCP / UDP / ICMPv4 / ICMPv6 payload.
- Ipv6 frames with routing headers.

The checksum offload module calculates the payload (TCP, UDP, or ICMP) and inserts the result into its corresponding field in the header. It has two modes when working, as follows:

1. The checksum calculation does not include TCP, UDP, or ICMPv6 pseudo-headers and assumes that the checksum field of the input frame already has the value. The checksum calculation includes checksum field, and the value of the original checksum field is replaced after the calculation is completed.

2. Checksum offload module clears the contents of the checksum field in the transmission frame and make calculation which includes TCP, UDP, or ICMPv6 pseudo-header data and will instead the transmission frame's original checksum field by the final calculation results.

After calculated by checksum offload module, the result can be found in IPPE bit of Transmit Descriptor0. The following shows the conditions under which the IPPE bit can be set:

1. In Store-and-Forward mode, frame has been forwarded to MAC transmitter but no EOF is written to TxFIFO.

2. Frame is ended but the byte numbers which the payload length field of the frame indicates has not been reached.

If the packet length is greater than the marked length, checksum module does not report errors, the excess data will be discarded as padding bytes. If the first condition of IPPE error is detected, the value of the checksum does not insert a TCP, UDP or ICMP header. If the second condition of IPPE error is detected, checksum calculation results will still insert the appropriate header fields.

Note: For ICMP packets over Ipv4 frame, the checksum field in the ICMP packet must always be 0x0000 in both modes due to such packets are not defined pseudo-headers. The follows describes the operation of receive checksum offload.

Receive checksum offload is enabled when IPFCO bit in ENET_MAC_CFG register is set. Receive checksum offload can calculate the Ipv4 header checksum and check whether it matches the contents of the Ipv4 header checksum field. The MAC identifies Ipv4 or Ipv6 frames by checking for the value of 0x0800 or 0x86DD respectively in the received Ethernet frame type field. This method is also used to identify frames with VLAN tags. Header checksum error bits in DMA receive descriptor (the IPHERR bit in Receive Descriptor0)

reflects the header checksum result. This bit is set if received IP header has the following errors:

- Any mismatch between the Ipv4 calculation result by checksum offload module and the value in received frame's checksum field.
- Any inconsistent between the data type of Ethernet type field and IP header version field.
- Received frame length is less than the length indicated in Ipv4 header length field, or Ipv4 or Ipv6 header is less than 20 bytes.

Receive checksum offload also identifies the data type of the IP packet is TCP, UDP or ICMP, and calculate their checksum according to TCP, UDP or ICMP specification. Calculation process can include data of TCP / UDP / ICMPv6 pseudo-header. Payload checksum error bits in DMA receive descriptor (the PCERR bit in Receive Descriptor0) reflects the payload checksum result. This bit is set if received IP payload has the following errors:

- Any mismatch between the TCP, UDP or ICMP checksum calculation result by checksum offload and the received TCP / UDP / ICMP frame's checksum field.
- Any inconsistent between the received TCP, UDP or ICMP data length and length of IP header.

The received checksum offload does not calculate the following conditions: Incomplete IP packets, IP packets with security features, packets of Ipv6 routing header and data type is not TCP, UDP or ICMP.

MAC loopback mode

Often, loopback mode is used for testing and debugging hardware and software system for application. The MAC loopback mode is enabled by setting the LBM bit in ENET_MAC_CFG register. In this mode, the MAC transmitter sends the Ethernet frame to its own receiver. This mode is disabled by default.

54.3.3. DMA controller description

Ethernet DMA controller is designed for frame transmission between FIFO and system memory which can reduce the occupation of CPU. Communication between the CPU and the DMA is achieved by the two kinds of data structures. Which are descriptor table (ring or chain type) and data buffer, and control and status register.

Applications need to provide the memory for storage of descriptor tables and data buffers. Descriptors that reside in the memory act as pointers to these buffers. Transmission has transmission descriptor and reception has reception descriptor. The base address of each table is stored in ENET_DMA_TDTADDR and ENET_DMA_RDTADDR register. Descriptors of transmission constituted by 4 descriptor word (Transmit Descriptor0-3) when DFM=0 and 8 descriptor word (Transmit Descriptor0-7) when DFM=1 (Enhanced descriptor mode). Likewise, reception descriptors constituted by 4 descriptor word (Receive Descriptor0-3) when DFM=0 and 8 descriptor word (Receive Descriptor0-7) when DFM=1. Each descriptor can point to a maximum of two buffers. The value of the buffer 2 can be programmed to the

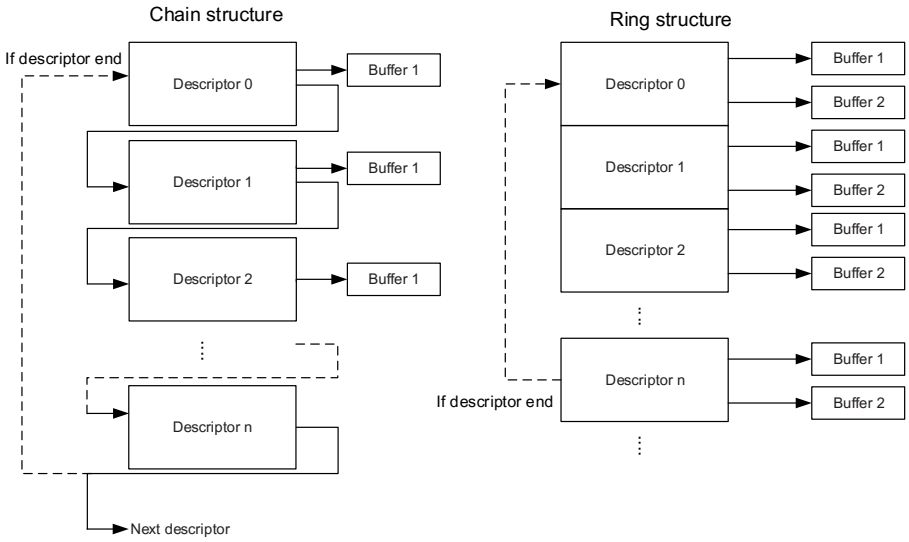
second data address or the next descriptor address which is determined by the configured descriptor table type: Ring or Chain. Buffer space only contains frame data which are located in host's physical memory space. One buffer can store only one frame data but one frame data can be stored in more than one buffer which means one buffer can only store a part of a frame. When chain structure is set, descriptor table is an explicitly one and when ring structure is set, descriptor table is an implicitly one. Explicit chaining of descriptors is accomplished by configuring the second address chained in both receive and transmit descriptors (configure RCHM bit in the Receive Descriptor1 and TCHM bit in the Transmit Descriptor0), at this time Receive Descriptor2 and Transmit Descriptor2 are stored the data buffer address, Receive Descriptor3 and Transmit Descriptor3 should be stored the next descriptor address, this connection method of descriptor table is called chain structure. Implicitly chaining of descriptors is accomplished by clearing the RCHM bit in the Receive Descriptor1 and TCHM bit in the Transmit Descriptor0, at this time Receive Descriptor2/Transmit Descriptor2 and Receive Descriptor3/Transmit Descriptor3 should be all stored the data buffer address, this connection method of descriptor table is called ring structure. When current descriptor's buffer address is used, descriptor pointer will point to the next descriptor. If chain structure is selected, the pointer points to the value of buffer 2. If ring structure is selected, the pointer points to an address calculated as below:

DFM=0: Next descriptor address = Current descriptor address + 16 + DPSL * 4

DFM=1: Next descriptor address = Current descriptor address + 32 + DPSL * 4

If current descriptor is the last one in descriptor table, application needs to set the TERM bit in Transmit Descritor0 or RERM bit in Receive Descritor1 to inform DMA the current descriptor is the last one of the table in ring structure. At this time, the next descriptor pointer points back to the first descriptor address of the descriptor table. In chain structure, can also set Receive Descriptor3 and Transmit Descriptor3 value to point back to the first descriptor address of the descriptor table. The DMA skips to the next frame buffer when the end of frame is detected.

Figure 54-6. Descriptor ring and chain structure



Alignment rule for data buffer address

The DMA controller supports all alignment types: byte alignment, half-word alignment and word alignment. This means application can configure the buffer address to any address. But during the operation of the DMA controller, access address is always word align and is different between write and read access. Follow example describes the detail:

- Buffer Reading: Assuming the transmit buffer address is 0x2000 0AB2, and 15 bytes need to be transferred. After starting operating, the DMA controller will read five word addresses which are 0x2000 0AB0, 0x2000 0AB4, 0x2000 0AB8, 0x2000 0ABC and 0x2000 0AC0. But when sending data to the FIFO, the first two bytes (0x2000 0AB0 and 0x2000 0AB1) and the last 3 bytes (0x2000 0AC1, 0x2000 0AC2 and 0x2000 0AC3) will be dropped.
- Buffer Writing: Assuming the receive buffer address is 0x2000 0CD2, and 16 bytes need to be stored. After starting operating, the DMA controller will write five times 32-bit data from address 0x2000 0CD0 to 0x2000 0CE0. But the first 2 bytes (0x2000 0CD0 and 0x2000 0CD1) and the last 2 bytes (0x2000 0CE2 and 0x2000 0CE3) will be substituted by the virtual bytes.

Note: DMA controller will not write any data out of the defined buffer range.

The effective length of the buffer

For the frame transmitting process, the effective length of the buffer is the same as the value configured by application in Transmit Descriptor1. As mentioned before, a transmitting frame can use one or more descriptors to indicate the frame information which means a frame data can be located in many buffers. When the DMA controller reads a descriptor which the FSG bit in Transmit Descriptor0 is set, it knows the current buffer is pointing to a new frame and the first byte of the frame is included. When the DMA controller reads a descriptor with FSG bit and LSG bit in Transmit Descriptor0 are both reset, it knows the current buffer is pointing to a part of current frame. When the DMA controller reads a descriptor with LSG bit in Transmit Descriptor0 is set, it knows the current buffers is pointing to the last part of the current frame. Normally one frame is stored only in one buffer (because buffer size is large enough for a normal frame), so FSG bit and LSG bit are set in the same descriptor.

For the frame receiving process, the receive buffer size must be word align. But for word-align buffer address or not word-align buffer address, the operation is different from transmitting. When the receive buffer address is word align, it's no difference with transmitting process, the effective length of the buffer is the same as the value configured by application in Receive Descriptor1. When the receive buffer address is not word align, the effective length of the buffer is less than the value configured by application in Receive Descriptor1. The effective length of the buffer should be the size value minus the low two bits value of buffer address. For example, assuming the total buffer size is 2048 bytes and buffer address is 0x2000 0001, the low two bits are 0b01, the effective length of the buffer is 2047 bytes whose address range is from 0x20000001 (for the first received frame byte) to 0x2000 07FF.

When a start of frame (SOF) is received, the FSG bit is set by DMA controller and when the end of the frame (EOF) is received, the LSG bit is set. If the receive buffer size is programmed to be large enough to store the whole frame, the FSG and the LSG bit are set in the same descriptor. The actual frame length FRML can be read from Receive Descriptor0. So application can calculate the left unused buffer space. The RxDMA always uses a new descriptor to receive the start of next frame.

Arbitration for TxDMA and RxDMA controller

There are two types of arbitration method designed for improving the efficiency of DMA controller between transmission and reception: fixed-priority and round-robin. When DAB bit in ENET_DMA_BCTL register is reset, arbiter selects round-robin method. The arbiter allocates the data bus in the ratio set by the RTPR bits in ENET_DMA_BCTL, when both of TxDMA and RxDMA controller request access simultaneously. When DAB bit in ENET_DMA_BCTL register is set, arbiter selects fixed-priority, and the RxDMA controller always has higher priority over the TxDMA.

DMA error status

During the operation of the DMA controller, when a response error presents on the bus, the DMA controller considers a fatal error occurs and stops operating at once with error flags written to the DMA status register (ENET_DMA_STAT). After such fatal error (response error) occurs, application must reset the Ethernet module and reinitialize the DMA controller.

DMA controller initialization for transmission and reception

Before using the DMA controller, the initialization must be done as follow steps:

1. Set the bus access parameters by writing the ENET_DMA_BCTL register;
2. Mask unnecessary interrupt source by configuring the ENET_DMA_INTEN register;
3. Program the Tx and Rx descriptor table start address by writing the ENET_DMA_TDTADDR register and the ENET_DMA_RDTADDR register;
4. Configure filter option by writing related registers;
5. According to the auto-negotiation result with external PHY, set the SPD bit and DPM bit for selecting the communication mode (Half-duplex / Full-duplex) and the communication speed (10Mbit / s or 100Mbit / s). Set the TEN and REN bit in ENET_MAC_CFG register to enable MAC transmit and receive operations;
6. Set STE bit and SRE bit in ENET_DMA_CTL register to enable TxDMA controller and RxDMA controller.

Note: If the HCLK frequency is too much low, application can enable RxDMA before set REN bit in ENET_MAC_CFG register to avoid RxFIFO overflow at start time.

Transmit process of DMA

As mentioned before, a frame can span over several buffers which means several descriptors.

When the FSG bit is set, the descriptor indicates the start of the frame and when the LSG bit is set, the descriptor indicates the end of the frame. All the buffers among these descriptors store the whole frame data. When the last descriptor is fetched and buffer finished reading, the transmitting status will write back to it. The other descriptors (here means the descriptor whose LSG bit is reset) of the current frame will not be changed by TxDMA controller except the DAV bit will be reset to 0. After starting transfer frame data from memory to FIFO, the transmitting has not actually start. The real start time for sending frame on interface is depended on TxDMA mode: Cut-Through mode or Store-and-Forward mode. The former mode starts sending when the byte number of FIFO is greater than configured threshold and the latter mode starts sending when the whole frame data are transferred into FIFO or when the FIFO is almost full.

Transmission management of DMA

Operate on second frame in buffer

When OSF bit in ENET_DMA_CTL is reset, the order of the transmitting is follows: the first is reading transmit descriptor, followed by reading data from memory and writing to FIFO, then sending frame data on interface through MAC and last wait frame data transmitting complete and writing back transmitting status.

Above procedure is TxDMA's standard transmitting procedure but when HCLK is much faster than TX_CLK, the efficiency of transmitting two frames will be greatly reduced.

To avoid the case mentioned above, application can set OSF to 1. If so, the second frame data can be read from the memory and push into FIFO without waiting the first frame's status writing back. OSF function is only performed between two neighboring frames.

TxDMA operation mode (A) (default mode): Non-OSF

The TxDMA controller in Non-OSF mode proceeds as follows:

1. Initialize the frame data into the buffer space and configure the descriptor (Transmit Descriptor0-3) with DAV bit of Transmit Descriptor0 sets to 1;
2. Enable TxDMA controller by setting STE bit in ENET_DMA_CTL register;
3. The TxDMA controller starts continue polling and performing transmit descriptor. When the DAV bit in Transmit Descriptor0 that TxDMA controller read is cleared, or any error condition occurs, the controller will enter suspend state and at the same time both the transmit buffer unavailable bit in ENET_DMA_STAT and normal interrupt summary bit in ENET_DMA_STAT register are set. If entered into suspend state, operation proceeds to Step 8;
4. When the DAV bit in Transmit Descriptor0 of the acquired descriptor is set, the DMA decodes the transmit frame configured and the data buffer address from the acquired descriptor;
5. DMA retrieve data from the memory and push it into the TxFIFO of MAC;
6. The TxDMA controller continues polling the descriptor table until the EOF data (LSG bit is set) is transferred. If the LSG bit of current descriptor is reset, it will be closed by

resetting the DAV bit after all buffer data pushed into TxFIFO. Then the TxDMA controller waits to write back descriptor status and IEEE 1588 timestamp value if enabled;

7. After the whole frame is transferred, the transmit status bit (TS bit in ENET_DMA_STAT register) is set only when INTC bit in Transmit Descriptor0 is set. Also an interrupt generates if the corresponding interrupt enable flag is set. The TxDMA controller returns to Step 3 for the next frame;
8. In the suspend state, application can make TxDMA returns to running state by writing any data to ENET_DMA_TPEN register and clearing the transmit underflow flag. Then the TxDMA controller process turns to Step 3.

TxDMA operation mode (B): OSF

The TxDMA controller supports transmitting two frames without waiting status write back of the first frame, this mode is called operation on second frame (OSF). When the frequency of system is much faster than the frequency of the MAC interface (10Mbit / s or 100Mbit / s), the OSF mode can improve the sending efficiency. Setting OSF bit in ENET_DMA_CTL register can enable this mode. When the TxDMA controller received EOF of the first frame, it will not enter the state of waiting status write back but to fetch the next descriptor, if the DAV bit and FSG bit of the next descriptor is set, the TxDMA controller immediately read the second frame data a push them into the MAC FIFO.

The TxDMA controller in OSF mode proceeds as follows:

1. Follow steps 1-6 operation in TxDMA default mode;
2. The TxDMA controller retrieves the next descriptor without closing the previous frame's last descriptor in which the LSG bit is set;
3. If the DAV bit of the next descriptor is set, the TxDMA controller starts reading the next frame's data from the buffer address. If the DAV bit of the next descriptor is reset, TxDMA controller enters suspend state and the next operation goes to Step 7;
4. TxDMA controller continues polling descriptor and frame data until the EOF is transferred. If a frame is described with more than one descriptor, the intermediate descriptors are all closed by TxDMA controller after fetched;
5. The TxDMA controller enters the state of waiting for the transmission status and time stamp of the previous frame (if timestamp enabled). With writing back status to descriptor, the DAV bit is also cleared by TxDMA controller;
6. After the whole frame is transferred, the transmit status bit (TS bit in ENET_DMA_STAT register) is set only when INTC bit in Transmit Descriptor0 is set. Also an interrupt generates if the corresponding interrupt enable flag is set. The TxDMA controller returns to Step 3 for the next frame if no underflow error occurred in previous frame. If underflow error of the previous frame is occurred, the TxDMA controller enters in suspend state and the next operation goes to Step 7;
7. In suspend state, when the status information and timestamp value (if the function is enable) of the transmitting frame is available, the TxDMA controller writes them back to descriptor and then close it by setting DAV=0 of descriptor;

8. In suspend state, application can make TxDMA returns to running state by writing any data to ENET_DMA_TPEN register and clearing the transmit underflow flag. Then the TxDMA controller process goes to Step 1 or Step 2.

Transmit frame format in buffer

According to IEEE 802.3 specification described before, a frame structure is made up of such fields: Preamble, SFD, DA, SA, QTAG (option), LT, DATA, PAD (option), and FCS.

The Preamble and SFD are automatically generated by the MAC, so the application only need store the DA, SA, QTAG (if needed), LT, DATA, DATA, PAD (if needed), FCS (if needed) parts. If the frame needs padding which means PAD and FCS parts are not stored in buffer, then application can configure the MAC to generate the PAD and FCS. If the frame only need FCS which means only FCS part is not stored in buffer, the application can configure the MAC to generate FCS. The DPAD bit and DCRC bit are designed to achieve the generate function of the PAD and FCS field.

Suspend during transmit polling

The DMA controller keeps querying the transmit descriptor after the transmission is started. If either of the following conditions happens, the DMA controller will enter suspend state and the transmit polling will stop. Though the DMA entered suspend state, the descriptor pointer is maintained to the descriptor following of the last closed descriptor.

- The DMA controller fetches a descriptor with DAV=0, then it enters suspend state and stops polling. In this case, the NI bit and TBU bit in ENET_DMA_STAT register are set.
- The MAC FIFO is empty during sending a frame on interface which means an error of underflow occurs. In this case, the AI bit and TU bit in ENET_DMA_STAT register are set. Also the transmit error status will write back to transmit descriptor.

Transmit DMA descriptor with IEEE 1588 timestamp format

When TTSEN bit is set, the timestamp function is enabled. The TxDMA controller writes transmit timestamp status TTMSS and timestamp back to descriptor after the frame transmission complete. The word address in descriptor for writing timestamp is depends on DFM bit in ENET_DMA_BCTL register. If the descriptor format is normal mode (DFM=0), Transmit Descriptor2 and Transmit Descriptor3 are used for timestamp recording and the old values in Transmit Descriptor2 and Transmit Descriptor3 are overwritten. If the descriptor format is enhanced mode (DFM=1), Transmit Descriptor6 and Transmit Descriptor7 are used for timestamp recording and the value in Transmit Descriptor2 and Transmit Descriptor3 are kept.

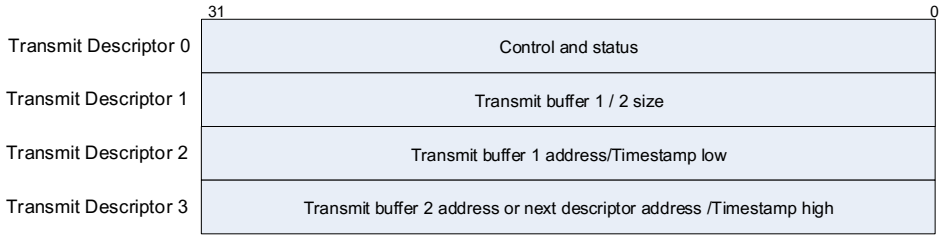
TxDMA descriptors in normal mode

The normal mode descriptor structure consists of four 32-bit words: Transmit Descriptor0 ~ Transmit Descriptor3. The descriptions of Transmit Descriptor0 ~ Transmit Descriptor3 are

given below:

Note: When a frame is described by more than one descriptor, only the control bits of the first descriptor are accept by TxDMA controller (except INTC). But the status and timestamp (if enabled) are written back to the last descriptor.

Figure 54-7. Transmit descriptor in normal mode



■ Transmit Descriptor0

Bits	Fields	Descriptions
31	DAV	DAV bit The DMA clears this bit either when it completes the frame transmission or the buffer allocated in the descriptor is read completely. This bit of the frame's first descriptor must be set after all subsequent descriptors belonging to the same frame have been set. 0: The descriptor is available for CPU not for DMA 1: The descriptor is available for DMA not for CPU
30	INTC	Interrupt on completion bit Only when the LSG bit is set, this bit is valid. 0: TS bit in ENET_DMA_STAT is not set when frame transmission complete. 1: TS bit in ENET_DMA_STAT is set when frame transmission complete.
29	LSG	Last segment bit This bit shows whether the transmit buffer contains the last segment of the frame. 0: The buffer of descriptor is not stored the last part of frame 1: The buffer of descriptor is stored the last part of frame
28	FSG	First segment bit This bit shows whether the buffer contains the first segment of a frame. 0: The buffer of descriptor is not stored the first block of frame 1: The buffer of descriptor is stored the first block of frame
27	DCRC	Disable CRC bit Only when the FSG bit is set, this bit is valid. 0: Allow MAC to insert CRC at the end of transmitted frame automatically 1: Not Allow MAC to insert CRC at the end of transmitted frame
26	DPAD	Disable adding pad bit Only when the FSG bit is set, this bit is valid.

		0: The DMA adds padding byte and CRC to transmitted frame automatically. Only the padding actually acts, the CRC is also appended. And ignore the value of DCRC bit. 1: The MAC does not add padding to a frame automatically
25	TTSEN	Transmit timestamp function enable bit. Only when the FSG bit is set, this bit is valid. 0: Disable transmit timestamp function 1: Enable IEEE 1588 hardware time stamping for the transmit frame, when TMSEN bit in the ENET_PTP_TSCTL register is set.
24	Reserved	Must be kept at reset value.
23:22	CM[1:0]	Checksum mode bits 0x0: Disable checksum insertion function 0x1: Only enable function for IP header checksum calculation and insertion 0x2: Enable IP header checksum and payload checksum calculation and insertion, hardware does not calculate checksum of pseudo-header. 0x3: Enable IP Header checksum and payload checksum calculation and insertion, hardware calculates checksum of pseudo-header.
21	TERM	Transmit end for ring mode bit This bit is used only in ring mode and has higher priority than TCHM. 0: The current descriptor is not the last descriptor in the table 1: The descriptor table reached its final descriptor. The DMA descriptor pointer returns to the start address of the table.
20	TCHM	The second address chained mode bit This bit is used only in chain mode. When TCHM bit is set, TB2S[12:0] is ignored. 0: The second address in the descriptor is the second buffer address 1:The second address in the descriptor is the next descriptor address
19:18	Reserved	Must be kept at reset value.
17	TTMSS	Transmit timestamp status bit Only when the LSG bit is set, this bit is valid. 0: Timestamp was not captured 1:A timestamp was captured for the described transmit frame and push into Transmit Descriptor2 (or Transmit Descriptor6 if DFM=1) and Transmit Descriptor 3 (or Transmit Descriptor7if DFM=1)
16	IPHE	IP header error bit IP header error occurs when any case of below happen: Ipv4 frames: 1) The header length field has a value less than 0x5. 2) The header length field value in transmitting Ipv4 frame is mismatch with the number of header bytes. 3) The version field value does not match the length / type field value.

		Ipv6 frames: 1) The main header length is not 40 bytes. 2) The version field value does not match the length / type field value. 0: The MAC transmitter did not detect error in the IP datagram header 1: The MAC transmitter detected an error in the IP datagram header
15	ES	Error summary bit Following bits are logical Ored to generate this bit: IPHE: IP header error JT: Jabber timeout FRMF: Frame flush IPPE: IP payload error LCA: Loss of carrier NCA: No carrier LCO: Late collision ECO: Excessive collision EXD: Excessive deferral UFE: Underflow error
14	JT	Jabber timeout bit Only set when the JBD bit is reset. 0: No jabber timeout occurred 1: Jabber timeout of MAC transmitter has occurred
13	FRMF	Frame flushed bit This bit is set to flush the Tx frame by software.
12	IPPE	IP payload error bit The transmitter checks the payload length received in the Ipv4 or Ipv6 header against the actual number of TCP, UDP or ICMP packet bytes received from the application and issues an error status in case of a mismatch. 0: No IP payload error occurred 1: MAC transmitter detected an error in the TCP, UDP, or ICMPIP datagram payload.
11	LCA	Loss of carrier bit When the interface signal 'CRS' lost one or more cycles and no collision happened during transmitting, the loss of carrier condition occurs. Only in Half-duplex mode this bit is valid. 0: No loss of carrier occurred 1: When the frame is transmitting, loss of carrier occurred
10	NCA	No carrier bit 0: PHY carrier sense signal is active 1: When the frame is transmitting, the carrier sense signal from the PHY was not active

9	LCO	Late collision bit If a collision occurs when 64 bytes (including preamble and SFD) has already transferred, this situation called late collision. 0: No late collision occurred 1: Late collision situation occurred **Note:** This bit is not valid if the UFE bit is set.
8	ECO	Excessive collision bit If the RTD=1 (retry function disable), this bit is set after the first collision. If the RTD=0 (retry function enable), this bit is set when failed 16 successive retry transmitting. When this bit is set, the transmission of current frame is aborted. 0: No excessive collision occurred 1: Excessive collision occurred
7	VFRM	VLAN frame bit 0: The transmitted frame was a normal frame 1: The transmitted frame was a VLAN-type frame
6:3	COCNT[3:0]	Collision count bits Only when ECO bit is cleared, this bit is valid. Before the frame was transmitted, this 4-bit counter counts the number of collisions that has occurred.
2	EXD	Excessive deferral bit Only when the DFC bit in the ENET_MAC_CFG register is set, this bit is valid. 0: No excessive deferral occurred 1: The transmission has ended because of excessive deferral time is over 3036 bytes
1	UFE	Underflow error bit This bit shows that the TxDMA comes across an empty TxFIFO while transmitting the frame before EOF which is caused by pushing data to TxFIFO late from memory. The transmission process enters the suspend state and sets both the TU (bit 5) and the TS (bit 0) in ENET_DMA_STAT. 0: No underflow error occurred 1: Underflow error occurred and the MAC aborted the frame transmitting
0	DB	Deferred bit This bit shows whether the transmitting frame is deferred because of interface signal CRS is active before MAC transmit frame. Only in Half-duplex mode this bit is valid. 0: No transmission deferred 1:The MAC is deferred before transmission

■ Transmit descriptor1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Reserved	TB2S[12:0]

rw

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Reserved	TB1S[12:0]

rw

Bits	Fields	Descriptions
31:29	Reserved	Must be kept at reset value.
28:16	TB2S[12:0]	Transmit buffer 2 size bits The second data buffer byte size. If the THCM bit in the Transmit Descriptor0 is set, this bit is ignored.
15:13	Reserved	Must be kept at reset value.
12:0	TB1S[12:0]	Transmit buffer 1 size bits The first data buffer byte size. The TxDMA ignores this buffer and uses buffer 2 (for TCHM=0) or the next descriptor (for TCHM=1) when this field is 0.

■ Transmit descriptor2

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16

TB1AP / TTSL[31:16]

rw

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

TB1AP / TTSL[15:0]

rw

Bits	Fields	Descriptions
31:0	TB1AP / TTSL[31:0]	Transmit buffer 1 address pointer / Transmit frame timestamp low 32-bit value bits Before transmitting frame, application must configure these bits for transmit buffer 1 address (TB1AP). When the transmitting frame is complete, these bits can be changed to the timestamp low 32-bit value (TTSL) for transmitting frame if DFM=0. But if DFM=1, these bits will not change and keep the value of buffer address. When these bits stand for buffer 1 address (TB1AP), the alignment is no limitation. When these bits stand for timestamp low 32-bit value, the TTSEN and LSG bit of current descriptor must be set.

■ Transmit descriptor word 3

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16

TB2AP / TTSH[31:16]

rw

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

TB2AP / TTSH[15:0]

rw

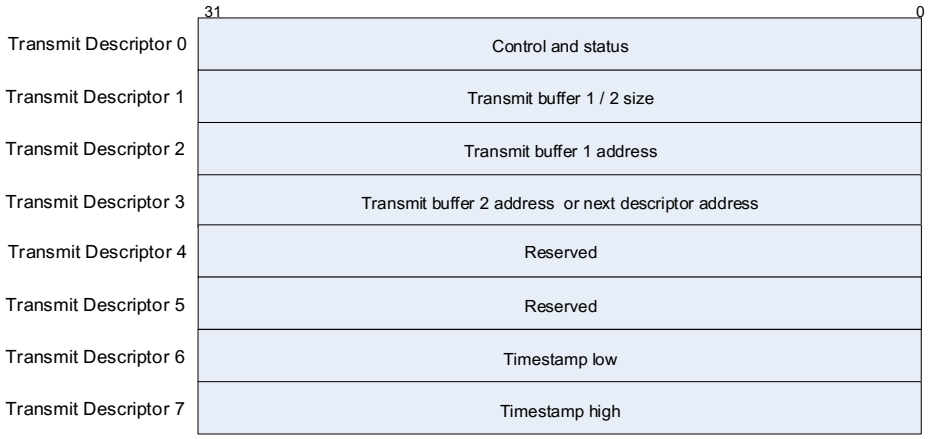
Bits	Fields	Descriptions
31:0	TB2AP / TTSH[31:0]	Transmit buffer 2 address pointer (or next descriptor address) / Transmit frame timestamp high 32-bit value bits. Before transmitting frame, application must configure these bits for transmit buffer 2 address (TB2AP) or the next descriptor address which is decided by descriptor type is ring or chain. When the transmitting frame is complete, these bits can be changed to the timestamp high 32-bit value (TTSH) for transmitting frame if DFM=0 and TTSEN =1. But if DFM=1 or TTSEN =0, these bits will not change and keep the old value. When these bits stand for buffer 2 address (TCHM=0), the alignment is no limitation. When these bits stand for the next descriptor address (TCHM=1), these bits must be word-alignment. When these bits stand for timestamp high 32-bit value, the TTSEN and LSG bit of current descriptor must be set.

TxDMA descriptors in enhanced mode

The enhanced mode descriptor structure consists of eight 32-bit words: Transmit Descriptor0 ~ Transmit Descriptor7. The descriptions of Transmit Descriptor0 ~ Transmit Descriptor3. Are the same with normal mode descriptor; Transmit Descriptor4 ~ Transmit Descriptor7. Are given below:

Note: When a frame is described by more than one descriptor, only the control bits of the first descriptor are accept by DMA controller (except INTC). But the status and timestamp (if enabled) are written back to the last descriptor.

Figure 54-8. Transmit descriptor in enhanced mode



- Transmit descriptor4
  All bits reserved.
- Transmit descriptor5
  All bits reserved.
- Transmit descriptor6

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TTSL[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TTSL[15:0]															
rw															

Bits	Fields	Descriptions
31:0	TTSL[31:0]	Transmit frame timestamp low 32-bit value bits When TTSEN =1 and LSG=1, there bits are updated by TxDMA for recording timestamp low 32-bit value of the current transmitting frame.

- Transmit descriptor 7

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TTSH[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TTSH[15:0]															
rw															

Bits	Fields	Descriptions
31:0	TTSH[31:0]	Transmit frame timestamp high 32-bit value bits When TTSEN =1 and LSG=1, these bits are updated by TxDMA for recording timestamp high 32-bit value of the current transmitting frame.

Reception process of DMA

When a frame is presented on the interface, the MAC starts to receive it. At the same time, the address filter block is running for this received frame. If the received frame fails the address filtering it will be discarded from RxFIFO in MAC and not be forwarded to buffer by RxDMA controller. If the received frame passes the address filtering, it will be forwarded to buffer when the available time comes. If the RxDMA controller is configured in Cut-Through mode, the available time means the byte number of the received frame is equal or greater than the configured threshold. If the RxDMA controller is configured in Store-and-Forward mode, the available time means the complete frame is stored in RxFIFO. During receiving frame, if any one of the below cases occurs the MAC can discard the received frame data in RxFIFO and the RxDMA controller will not forward these data:

- The received frame bytes is less than 64.
- Collision occurred during frame receiving.
- The premature termination for the receiving frame.

When the available time comes, the RxDMA controller starts transfer frame data from RxFIFO to the receive buffer. If the SOF is included in current receive buffer, the FDES bit in Receive

Descriptor0 is set when the RxDMA controller writing receive frame status to indicate this descriptor is used for storing the first part of the frame. If the EOF is included in current receive buffer, the LDES bit in Receive Descriptor0 is set when RxDMA controller writing receive frame status to indicate this descriptor is used for storing the last part of the frame. Often when the buffer size is larger than received frame, the FDES and LDES bit are set in the same descriptor. When the EOF is transferred to buffer or the receive buffer space is exhausted, the RxDMA controller fetches the next receive descriptor and closes previous descriptor by writing Receive Descriptor0 with DAV=0. If the LDES bit is set, the other status are also be updated and the RS bit in ENET_DMA_STAT register will be set (immediately when DINTC=0 or delayed when DINTC=1). If the DAV bit of the next descriptor is set, the RxDMA controller repeats above operation when received a new frame. If the DAV bit of the next descriptor is reset, the RxDMA controller enters suspend state and sets RBU bit in ENET_DMA_STAT register. The pointer value of descriptor address table is retained and be used for the starting descriptor address after exiting suspend state.

Reception management of DMA

The receiving process of the RxDMA controller is described detailed as below:

1. Applications initialize the receive descriptors with the DAV bit in the Receive Descriptor0 is set;
2. Setting the SRE bit in ENET_DMA_CTL register to make RxDMA controller entering running state. In running state, the RxDMA controller continually fetching the receive descriptors from descriptor table whose starting address is configured in ENET_DMA_RDTADDR register by application. If the DAV bit of the fetched receive descriptor is set, then this descriptor is used for receiving frame. But if the DAV bit is reset which means this receive descriptor cannot be used by RxDMA, the RxDMA controller will enter suspend state and operation goes to Step 9;
3. From the valid receive descriptor (DAV=1), the RxDMA controller marks the receiving control bit and data buffer address;
4. Processing the received frames and transfer data to the receive buffer from the RxFIFO;
5. If all frame data has completely transferred or the buffer is full, the RxDMA controller fetches the next descriptor from receive descriptor table;
6. If the current receiving frame transfer is complete, the operation of RxDMA goes to Step7. But if not complete, two conditions may occur:
   - The next descriptor's DAV bit is reset. The RxDMA controller sets descriptor error bit DERR in Receive Descriptor0 if flushing function is enabled. The RxDMA controller closes current descriptor by resetting DAV bit and sets the LSG bit (if flushing is enabled) or resets the LSG bit (if flushing is disabled). Then the operation goes to Step 8.
   - The next descriptor's DAV bit is set. The RxDMA controller closes current descriptor by resetting DAV bit and operation goes to Step 4.
7. If IEEE 1588 time stamping function is enabled, the RxDMA controller writes the time stamp value (if receiving frame meets the configured time stamping condition) to the

current descriptor's Receive Descriptor2 and Receive Descriptor3 if DFM=0 or Receive Descriptor6 and Receive Descriptor7 if DFM=1. At the same time (writing timestamp value) the RxDMA controller also writes the received frame's status word to the Receive Descriptor0 with the DAV bit cleared and the LSG bit set;

8. The latest descriptor is fetched by RxDMA controller. If the fetched descriptor bit 31 (DAV) is set, the RxDMA controller operation goes to Step 4. If the fetched descriptor bit 31 is reset, the RxDMA controller enters the suspend state and sets the RBU bit in register ENET_DMA_STAT. If flushing function is enabled, the RxDMA controller will flush the received frame data in the RxFIFO before entering suspend state;
9. In suspended state, there are two conditions to exit. The first is writing data in the ENET_DMA_RPEN register by application. The second is when a new received frame is available which means the byte number of receiving frame is greater than threshold in Cut-Through mode or when the whole frame is received in Store-and-Forward mode. Once exiting suspend mode, the RxDMA controller fetches the next descriptor and the following operation goes to Step 2.

Receive descriptor fetching regulation

Descriptor fetching occurs if any one or more of the following conditions are met:

- The time SRE bit is configured from 0 to 1 which makes the RxDMA controller entering running state.
- The total buffer size (buffer 1 for chain mode or buffer 1 plus buffer 2 for ring mode) of the current descriptor cannot hold the current receiving frame. In other word, the last byte stored in buffer space is not the EOF byte.
- After a complete frame is transferred to buffer and before current descriptor is closed.
- In suspend state, the MAC received a new frame.
- Writing any value to receive poll enable register ENET_DMA_RPEN.

Processing after a new frame received in suspend state

When a new frame is available (see available definition in the previous paragraph), the RxDMA controller fetches the descriptor. If the DAV bit in Receive Descriptor0 is set, the RxDMA controller exits suspend state and returns to running state for frame reception. But if the DAV bit in Receive Descriptor0 is reset, application can choose whether these received frame data in RxFIFO are flushed or not by configuring DAFRF bit in ENET_DMA_CTL register. If DAFRF=0, the RxDMA controller discards these received frame data and makes the missed frame counter (MSFC) increase one. If DAFRF=1, these frame data are will not be flushed and MSFC counter will not increase until the RxFIFO is full. If the DAV bit is reset in fetched descriptor, the RBU bit in ENET_DMA_STAT register will be set and the RxDMA controller will be still in suspend state.

Receive DMA descriptor with IEEE 1588 timestamp format

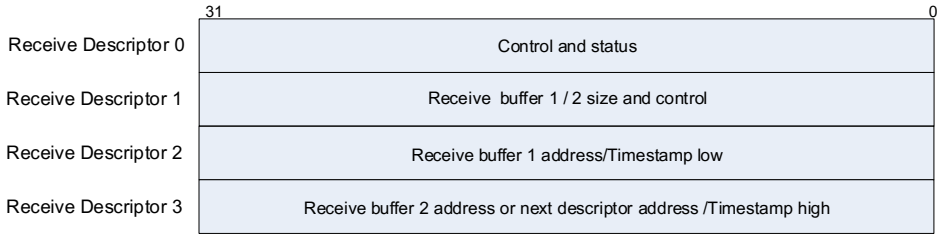
If the IEEE 1588 function enabled, the MAC writes the timestamp value to Receive

Descriptor2 and Receive Descriptor3 (DFM=0) or Receive Descriptor6 and Receive Descriptor7 (DFM=1) after a frame with timestamp reception complete and before the RxDMA controller clears the DAV bit.

RxDMA descriptors in normal mode

In normal descriptor mode, the descriptor structure consists of four 32-bit words: Receive Descriptor0 ~ Receive Descriptor3 The detailed description of Receive Descriptor0 ~ Receive Descriptor3 are given below.

Figure 54-9. Receive descriptor in normal mode



■ Receive descriptor 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DAV	DAFF	FRML[13:0]													
rw	rw	rw													

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERRS	DERR	SAFF	LERR	OERR	VTAG	FDES	LDES	IPHERR / TSV	LCO	FRMT	RWDT	RERR	DBERR	CERR	PCERR / EXSV
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31	DAV	Descriptor available bit This bit shows the DMA controller can use this descriptor. The DMA clears this bit either when it completes the frame reception or when the buffers in this descriptor are full. 0: The descriptor is owned by the CPU 1: The descriptor is owned by the DMA
30	DAFF	Destination address filter fail bit 0: A frame passed the destination address filter 1: A frame failed the destination address filter
29:16	FRML[13:0]	Frame length bits These bits show the byte length of the received frame that was transferred to the buffer (including CRC when received frame is not a type frame. If received frame is a type frame, including CRC or not is controlled by TFCD bit in ENET_MAC_CFG). Only when the bit LDES=1 and DERR=0, these bits are valid. If LDES=0 and

		ERRS=0, these bits indicate the accumulated number of bytes that have been transferred for the current frame. **Note:** The value of frame length is 0 means that for some reason (such as FIFO overflow or dynamically modify the filter value in the receiving process, resulting did not pass the filter, etc), frame data is not written to FIFO completely.
15	ERRS	Error summary bit Only when the LDES bit is set, this bit is valid. This bit is logical Ored by the following bits when DFM is equal to 0: DERR: Descriptor error OERR: Overflow error LCO: Late collision RWDT: Watchdog timeout RERR: Receive error CERR: CRC error IPHERR = 0, FRMT = 1 and PCERR = 1: payload checksum error IPHERR = 1, FRMT = 1 and PCERR = 0: header checksum error IPHERR = 1, FRMT = 1 and PCERR = 1: both header and payload checksum errors. This bit is logical Ored by the following bits when DFM is equal to 1: IPPLDERR: IP frame payload error IPHERR: IP frame header error DERR: Descriptor error OERR: Overflow error LCO: Late collision RWDT: Watchdog timeout RERR: Receive error CERR: CRC error
14	DERR	Descriptor error bit Only when the LDES bit is set, this bit is valid. When the current buffer cannot hold current received frame and the next descriptor's DAV bit is reset, the descriptor error occurs. 0: No descriptor error occurred 1: Descriptor error occurred
13	SAFF	SA filtering fail bit 0: No source address filter fail occurred 1: A received frame failed the SA filter
12	LERR	Length error bit Only when the FRMT bit is reset, this bit is valid. This bit shows whether the length field in received is mismatch the actual frame length. 0: No length error occurred

		1: Length error occurred
11	OERR	Overflow error bit When RxFIFO is overflow and the frame data has been partly forwarded to descriptor buffer, the overflow error bit sets. 0: No overflow error occurred 1: RxFIFO overflowed and frame data is not valid
10	VTAG	VLAN tag bit 0: Received frame is not a tag frame 1: Received frame is a tag frame
9	FDES	First descriptor bit This bit shows whether current descriptor contains the SOF of the received frame. 0: The current descriptor does not store the SOF of the received frame 1: The current descriptor buffer saves the SOF of the received frame
8	LDES	Last descriptor bit This bit shows whether current descriptor contains the EOF of the received frame. 0: The current descriptor buffer does not store EOF of the received frame 1: The current descriptor buffer saves the EOF of the received frame
7	IPHERR / TSV	IP frame header error bit / Timestamp valid bit When DFM=0, bit 7, 5 and 0 indicate some special cases refer to the error status table. When DFM=1, this bit indicates the timestamp value is taken and write to the Receive Descriptor6 and Receive Descriptor7. This bit is valid only when LDES is set.
6	LCO	Late collision bit This bit shows whether a collision occurs after 64 bytes have been received. This bit only valid in Half-duplex mode. 0: No late collision occurred 1: Late collision has occurred
5	FRMT	Frame type bit When DFM=0, bit 7, 5 and 0 shows some special cases refer to the error status table. When DFM=1, this bit shows the received frame is an Ethernet type frame or a tagged frame. If the received frame is runt frame, this bit is not valid for application. 0: The received frame is an IEEE802.3 frame without tagged. 1: The received frame is an Ethernet-type frame (the length / type field is greater than or equal to 0x0600, or this is a tagged frame)
4	RWDT	Receive watchdog timeout bit When WDD=0, this bit shows a frame with more than 2048 bytes was detected. When WDD=1, this bit shows a frame with more than 16384 bytes was detected.

0: No receive watchdog timeout occurred
1: Watchdog timer overflowed during receiving and current frame is only a part of frame.

3 RERR
Receive error bit
This bit shows whether the interface signal RX_ER asserted when RX_DV signal is active during frame receiving process.
0: No receive error occurred
1:Receive error occurred

2 DBERR
Dribble bit error bit
This bit shows whether there is an incomplete byte (odd cycles during reception) received. Only when in MII interface mode, this bit is valid.
0: No dribble bit error occurred
1: Dribble bit error occurred

1 CERR
CRC error bit
This bit shows whether FCS field in received frame is mismatch with the calculation result of the hardware. Only when LDES bit is set, this bit is valid.
0: No CRC error occurred
1:A CRC error occurred

0 PCERR / EXSV
Payload checksum error bit / Extended status valid bit
When DFM=0, bit 7, 5 and 0 indicate some special cases refer to the error status table.
When DFM=1, this bit indicates the descriptor Receive Descriptor4is valid for application.
This bit only valid when LDES is set.
0: Receive Descriptor4is not valid for application
1: Receive Descriptor4is valid for application

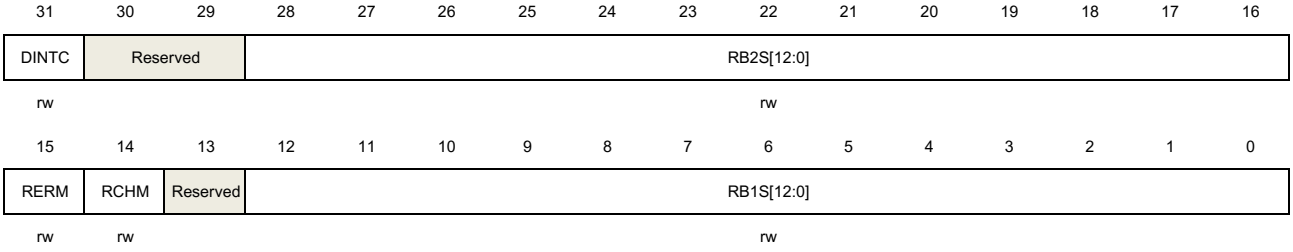
Table 54-7. Error status decoding in Receive Descriptor0, only used for normal descriptor (DFM=0) shows the combination meaning for bit 7, 5, and 0 in Receive Descriptor0:

Table 54-7. Error status decoding in Receive Descriptor0, only used for normal descriptor (DFM=0)

Bit 7: IPHERR	Bit 5: FRMT	Bit 0: PCERR	Frame status
0	0	0	IEEE 802.3 normal frame (Length field value is less than 0x0600 and not tagged)
0	0	1	Ipv4 or Ipv6 frame, no header checksum error, payload checksum is bypassed because of unsupported payload type
0	1	0	Ipv4 or Ipv6 frame, checksum checking pass
0	1	1	Ipv4 or Ipv6 frame, payload checksum error. This error may case by following condition:

			1) Calculated checksum value mismatch the checksum field 2) byte number of received payload mismatch length field
1	0	0	Reserved
1	0	1	A type (length / type field equal or greater than 0x0600) or tagged frame but neither Ipv4 nor Ipv6. Offload check engine is bypassed.
1	1	0	Ipv4 or Ipv6 frame, but a header checksum error detected This error may case by following condition: 1) Type value inconsistent with version value 2) Calculated header checksum mismatch the header checksum field 3) Expected IP header bytes is not received enough
1	1	1	Ipv4 or Ipv6 frame, both header and payload checksum detected errors

■ Receive descriptor 1



Bits	Fields	Descriptions
31	DINTC	Disable interrupt on completion bit 0: RS bit in ENET_DMA_STAT register will immediately set after receiving the completed, then if enabled the corresponding interrupt, the interrupt will trigger. 1: RS bit in ENET_DMA_STAT register will is not immediately set after receiving the completed, but will set after a configurable delay time.
30:29	Reserved	Must be kept at reset value.
28:16	RB2S[12:0]	Receive buffer 2 size bits The second buffer size in bytes. The buffer size must be a multiple of 4. This field is ignored if RCHM is set.
15	RERM	Receive end of ring mode bit This bit indicates the final descriptor in table is arrived and the next descriptor address is automatically set to the configured start descriptor address. 0: Current descriptor is not the last descriptor in table 1: Current descriptor is the last descriptor in table
14	RCHM	Receive chained mode for second address bit 0: The second address points to the second buffer address.

1: The second address points to the next descriptor address. RB2S[12:0] is ignored.

Note: If the RERM=1, the next descriptor returns to base address even this bit is set to 1.

13	Reserved	Must be kept at reset value.
12:0	RB1S[12:0]	Receive buffer 1 size bits The first buffer size in bytes. The buffer size must be a multiple of 4. If this field is 0, the RxDMA controller ignores this buffer and uses buffer 2 (RCHM=0) or the next descriptor (RCHM=1).

■ Receive descriptor 2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RB1AP / RTSL[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RB1AP / RTSL[15:0]															
rw															

Bits	Fields	Descriptions
31:0	RB1AP / RTSL[31:0]	Receive buffer 1 address pointer / Receive frame timestamp low 32-bit These bits are designed for two different functions: buffer address pointer (RB1AP) or timestamp low 32-bit value (RTSL). RB1AP: Before fetching this descriptor by RxDMA controller, these bits are configured to the buffer 1 address by application. This buffer 1 address pointer is used for RxDMA controller to store the received frame if RB1S is not 0. The buffer address alignment has no limitation. RTSL: When timestamp function is enabled and LDES is set, these bits will be changed to timestamp low 32-bit value by RxDMA controller if received frame passed the filter and satisfied the snapshot condition. If the received frame does not meet the snapshoot condition, these bits will keep RB1AP value.

■ Receive descriptor 3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RB2AP / RTSH[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RB2AP / RTSH[15:0]															
rw															

Bits	Fields	Descriptions
31:0	RB2AP / RTSH[31:0]	Receive buffer 2 address pointer (next descriptor address) / Receive frame timestamp high 32-bit value bits

These bits are designed for two different functions: buffer address pointer or next descriptor address (RB1AP) or timestamp high 32-bit value (RTSH).

RB2AP: Before fetching this descriptor by RxDMA controller, these bits are configured to the buffer 2 address (RCHM=0) or the next descriptor address (RCHM=1) by application. This buffer 2 address pointer is used for RxDMA controller to store the received frame if RB1S is not 0 when RCHM=0. If RCHM=1 and RERM=0, this address pointer is used for fetching the next descriptor. If RCHM=1 and RERM=1, these bits are ignored.

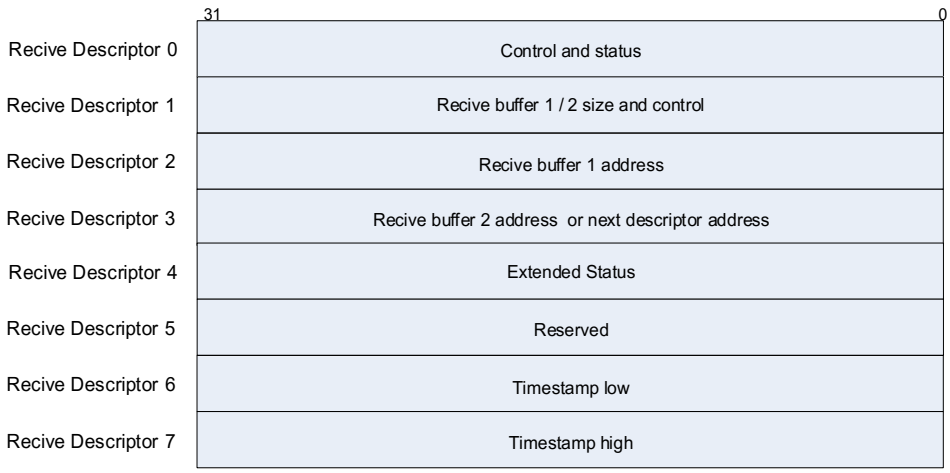
When this address is used for next descriptor address, the word alignment is needed. The other conditions have no limitation for these bits.

RTSH: When timestamp function is enabled and LDES is set, these bits will be changed to timestamp high 32-bit value by RxDMA controller if received frame passed the filter and satisfied the snapshoot condition. If the received frame does not meet the snapshot condition, these bits will keep RB2AP value.

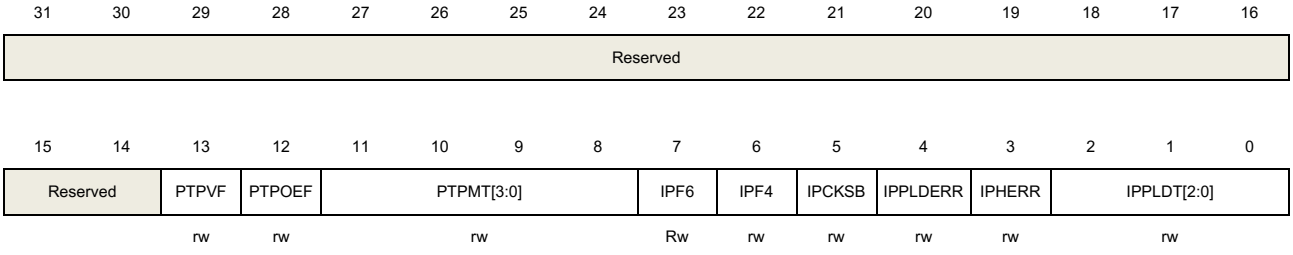
RxDMA descriptors in enhanced mode

In enhanced descriptor mode, the descriptor structure consists of eight 32-bit words: Receive Descriptor0 ~ Receive Descriptor7. The description of Receive Descriptor0 ~ Receive Descriptor3 are the same with descriptors in normal mode. The description of Receive Descriptor4 ~ Receive Descriptor7 are given below.

Figure 54-10. Receive descriptor in enhanced mode



■ Receive descriptor 4



Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value.
13	PTPVF	PTP version format bit 0: Version 1 format 1: Version 2 format
12	PTPOEF	PTP on Ethernet frame bit 0: Received PTP frame is a IP-UDP frame if PTPMT is not zero 1: Received PTP frame is a IEEE802.3 Ethernet frame
11:8	PTPMT[3:0]	PTP message type bits PTP message type is decoded to following number: 0x0: Not PTP frame received 0x1: SYNC 0x2: FOLLOW_UP 0x3: DELAY_REQ 0x4: DELAY_RESP 0x5: For peer-to-peer transparent clock: PDELAY_REQ For ordinary or boundary clock: ANNOUNCE 0x6: For peer-to-peer transparent clock: PDELAY_RESP For ordinary or boundary clock: MANAGEMENT 0x7: For peer-to-peer transparent clock: PDELAY_RESP_FOLLOW_UP For ordinary or boundary clock: SIGNALING
7	IPF6	IP frame in version 6 bit 0: Received frame is not a Ipv6 frame 1: Received frame is a Ipv6 frame
6	IPF4	IP frame in version 4 bit 0: Received frame is not a Ipv4 frame 1: Received frame is a Ipv4 frame
5	IPCKSB	IP frame checksum bypassed bit This bit is only valid when received frame is a Ipv4 or Ipv6 frame. 0: Received frame checksum checking function is not bypassed 1: Received frame checksum checking function is bypassed
4	IPPLDERR	IP frame payload error bit This bit can be set by any of below cases: 1) the calculated checksum by hardware mismatch with the TCP, UDP or ICMP checksum field in frame. 2) payload length value in IP header mismatch the received payload length. 0: Payload error not occurred in received frame 1: Payload error occurred in received frame
3	IPHERR	IP frame header error bit This bit can be set by any of below cases: 1) the calculated checksum by hardware mismatch with the IP header checksum field value. 2) Type field in IP frame is not

consistent with version field (e.g. 'type' field value is 0x0800 but 'version' field value is not 0x4, 'type' field value is 0x86dd but 'version' field value is not 0x6).
0: IP header error not occurred
1: IP header error occurred

2:0	IPPLDT[2:0]	IP frame payload type bits These bits are valid only when IPFCO=1, IPHERR=0 and LDES=1. 0x0: Unsupported payload type or IP payload bypassed 0x1: payload type is UDP 0x2: payload type is TCP 0x3: payload type is ICMP 0x4~0x7: Reserved

- Receive descriptor 5

  All bits reserved

- Receive descriptor 6

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RTSL[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTSL[15:0]															
rw															

Bits	Fields	Descriptions
31:0	RTSL[31:0]	Receive frame timestamp low 32-bit value When timestamp function is enabled and LDES is set, these bits will be written to timestamp low 32-bit value by RxDMA controller if received frame passed the filter and satisfied the snapshoot condition.

- Receive descriptor 7

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RTSH[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTSH[15:0]															
rw															

Bits	Fields	Descriptions
31:0	RTSH[31:0]	Receive frame timestamp high 32-bit value When timestamp function is enabled and LDES is set, these bits will be written to timestamp high 32-bit value by RxDMA controller if received frame passed the filter and satisfied the snapshoot condition.

54.3.4. MAC statistics counters: MSC

For knowing the statistics situation of transmitting and receiving frames, there is a group of counters designed for gathering statistics data. These MAC counters are called statistics counters (MSC).In Section ***Register definition***, there is a detailed description of the function of these registers.

- When the transmit frame does not appear the situations, such as frame underflow, no carrier, carrier lost, excessive deferral, late collision , excessive collision and jabber timeout, it can be called "good frame". MSC transmit counters will automatically update.
- When the receiving frame does not appear the situations, such as alignment error, CRC mismatch , runt frame, length error, range error and error signal valid on pin MII_RX_ER, it can be called 'good frame' and MSC reception counters will automatically update. Among them, CRC mismatch indicates that calculated CRC value is different from FSC field value, runt frame indicates that the frame length is shorter than 64 bytes, length error indicates that the length field value is different from the actual received data bytes, range error indicates that the length field value is larger than maximum size of defined in IEEE802.3 (1518 for untagged frame and 1522 for VLAN tagged frame).

Note: Only when the discarded frame is a short frame whose length is less than 6 bytes (no complete receives the DA), MSC reception counter is updated.

54.3.5. Wake up management: WUM

Ethernet (ENET) module supports two wakeup methods from Deep-sleep mode. The one is remote wakeup frame and the other is Magic Packet wakeup frame. For reduce power consuming, the host system and Ethernet can be powered down and thus the circuit driven by HCLK or transmit clock is stop working. But the circuit driven by receive clock will continues working for listening wakeup frame. If application sets the PWD bit in ENET_MAC_WUM register, the Ethernet enters into power-down state. In power-down state, MAC ignores all the frame data on the interface until the power-down state is exited. For exiting power-down state, application can choose one of or both of the two methods mentioned above. Setting WFEN bit in ENET_MAC_WUM register can make Ethernet wakeup if a remote wakeup frame received and setting MPE bit in ENET_MAC_WUM register can make Ethernet wakeup if a Magic Packet frame is received. When any type of wakeup frame is present on interface and corresponding wakeup function is enabled, Ethernet will generate a wakeup interrupt and exit power-down state at once.

Remote wakeup frame detection

Setting WFEN bit in ENET_MAC_WUM register can enable remote wakeup detection. When the MAC is in power-down state and remote wakeup function enable bit is set, MAC wakeup frame filter is active. If the received frame passes the address filter and filter CRC-16 matches the incoming examined pattern, then MAC identified the received wakeup frame, and then MAC returns to normal working state. Even if the length of the wakeup frame exceeds 512

bytes, as long as the frame has a correct CRC value, it is still considered to be effective. After received the remote wakeup frame, the WUFR bit in ENET_MAC_WUM register will be set. If remote wakeup interrupt is not masked, then a WUM interrupt is generated.

Magic packet detection

Another wakeup method is detecting Magic Packet frame (see 'Magic Packet Technology', Advanced Micro Devices). A Magic Packet frame is a special frame with formed packet solely intended for wakeup purposes. This packet can be received, analyzed and recognized by the Ethernet block and used to trigger a wakeup event. Setting MPE bit in ENET_MAC_WUM register can enable this function. This type of frame's format is as follows: starts by 6 continuous bytes of the value 0xFF (0xFFFF FFFF FFFF) in anywhere of the frame behind the destination and source address field, then there are 16 duplicate MAC addresses without any interruption and pause. If there is any discontinuity between repeating it 16 times, MAC needs to re-detect 0xFFFF FFFF FFFF in the receive frame. WUM module continuously monitors each frame received. When a Magic Packet frame passing the address filter, MAC will detect its format with Magic Packet format, once the format is matched the WUM will make MAC wakeup from power down state. Then the MAC wakes up from power-down state after receiving a Magic Packet frame. Module also accepts multicast frames as Magic Packet frame.

Example: An example of a Magic Packet with station address 0xAABB CCDD EEFF is the following (MISC indicates miscellaneous additional data bytes in the packet):

<DESTINATION><SOURCE><MISC>

…….. FF FF FF FF FF FF

AABB CCDD EEFF AABB CCDD EEFF AABB CCDD EEFF AABB CCDD EEFF

AABB CCDD EEFF AABB CCDD EEFF AABB CCDD EEFF AABB CCDD EEFF

AABB CCDD EEFF AABB CCDD EEFF AABB CCDD EEFF AABB CCDD EEFF

AABB CCDD EEFF AABB CCDD EEFF AABB CCDD EEFF AABB CCDD EEFF

<MISC><FCS>

Upon detecting a Magic Packet, the MPKR bit in ENET_MAC_WUM register will be set. If the Magic Packet interrupt is enabled, the corresponding interrupt will generate.

Precautions during system power-down state

When the MCU is in Deep-sleep mode, if external interrupt line 19 is enabled, Ethernet WUM module can still detecting frames. Because the MAC in power-down state needs detecting Magic Packet or remote wakeup frame, the REN bit in ENET_MAC_CFG register must be maintained set. The transmit function should be turned disable during the power-down state by clearing the TEN bit in the ENET_MAC_CFG register. Moreover, the Ethernet DMA block should be disabled during the power-down state, because it is not necessary that the Magic Packet or remote wakeup frame is forwarded to the application. Application can disable the

Ethernet DMA block by clearing the STE bit and the SRE bit (for the TxDMA and the RxDMA, respectively in the ENET_DMA_CTL register.

Follow steps are recommended for application to enter and exit power-down state:

1. Wait the current sending frame completes and then reset the TxDMA block by clearing STE bit in ENET_DMA_CTL register;
2. Clear the TEN and REN bit in ENET_MAC_CFG register to disable the MAC's transmit and receive function;
3. Check the RS bit in ENET_DMA_STAT register, waiting receive DMA read out all the frames in the receive FIFO and then close RxDMA;
4. Configure and enable the external interrupt line 19, so that it can generate an interrupt or event. If EXTI line 19 is configured to generate an interrupt, application still needs to modify ENET_WKUP_IRQ interrupt handling procedures to clear the pending bit of the EXTI line 19;
5. Set the MPEN or WFEN (or both) bit in ENET_MAC_WUM register to enable Magic Packet or Remote Wakeup frame (or both) detection;
6. Setting PWD bit in ENET_MAC_WUM register to enter power-down state;
7. Setting REN bit in ENET_MAC_CFG register to make MAC's receive function work;
8. Make MCU enter Deep-sleep mode;
9. After received a wakeup type frame, the Ethernet module exits the power-down state;
10. Reading the ENET_MAC_WUM register to clear the power management event flags. Enable MAC's transmit function and enable TxDMA and RxDMA;
11. Initialize the MCU system clock: enable HXTAL and configure the RCU unit.

Remote wakeup frame filter register

Wakeup frame filter register is made up of eight different registers but shared the same register offset address. So the inner pointer points the next filter register when the filter register address is accessed by writing or reading. Whatever operation, write or read, it is strongly recommended to operate eight times sequentially. This means continuously write 8 times will configure the filter registers and continuously read 8 times will get the values of filter registers.

Figure 54-11. Wakeup frame filter register

Wakeup Frame Filter Register 0	Filter 0 Byte Mask							
Wakeup Frame Filter Register 1	Filter 1 Byte Mask							
Wakeup Frame Filter Register 2	Filter 2 Byte Mask							
Wakeup Frame Filter Register 3	Filter 3 Byte Mask							
Wakeup Frame Filter Register 4	Reserve	Filter 3 Command	Reserve	Filter 2 Command	Reserve	Filter 1 Command	Reserve	Filter 0 Command
Wakeup Frame Filter Register 5	Filter 3 Offset		Filter 2 Offset		Filter 1 Offset		Filter 0 Offset	
Wakeup Frame Filter Register 6	Filter 1 CRC - 16				Filter 0 CRC - 16			
Wakeup Frame Filter Register 7	Filter 3 CRC - 16				Filter 2 CRC - 16			

■ Filter n Byte mask

This register field defines using which bytes of the frame to determine the received frame is wakeup frame or not by filter n (n=0, 1, 2, 3). Bit 31 must be set to 0. Bit 30 to bit 0 are valid byte mask. If bit m (m means byte number) is set, the filter n offset + m of the receiving frame is calculated by the CRC unit, conversely, filter n offset + m is ignored.

■ Filter n command

This four bits command controls the operation of the filter n. The bit 3 of the field is address type selection bit. If this bit is 1, the detection only detects a multicast frame and if this bit is 0, the detection only detects a unicast frame. Bit 2 and bit 1 must be set to 0. Bit 0 is the filter switch bit. Setting it to 1 means enable and 0 means disable.

■ Filter n offset

It is used in conjunction with filter n byte mask field. This register specifies offset (within the frame) of the first byte which the filter n uses to check. The minimum allowable value is 12, it represents the byte 13 in the frame (offset value 0 indicates the first byte of the frame).

■ Filter n CRC-16

This register field contains the filter comparing CRC-16 code which is used for comparing the calculated CRC-16 from frame data.

54.3.6. Precision time protocol: PTP

The majority of protocols are implemented by the UDP layer application software. The PTP module of the MAC is mainly to recording the transmitting and receiving PTP packets' precision time and returning it to application.

Specific details about the precise time protocol (PTP) please see the document "IEEE Standard 1588™".

Reference clock source

System reference time in Ethernet is maintained by a 64-bit register whose high 32-bit indicates 'second' time and low 32-bit indicates 'subsecond', this is defined in IEEE 1588 specification.

The input PTP reference clock is used to drive the system reference time (also called system time for short) and capture timestamp value for PTP frame. The frequency of this reference clock must be configured no less than the resolution of timestamp counter. The synchronization accuracy between the master node and slave node is around 0.1us.

Synchronization accuracy

The accuracy of time synchronization depends on the following factors:

- PTP reference clock input period.
- Characteristics of the oscillator (drift).
- Frequency of the synchronization procedure.

System time calibration

PTP input reference clock is used to update 64-bit PTP system time. The PTP system time is used as the source to record transmission / reception frame's timestamp. The system time initialization and calibration support two methods: coarse method and fine method. The purpose of calibration is to correct the frequency offset.

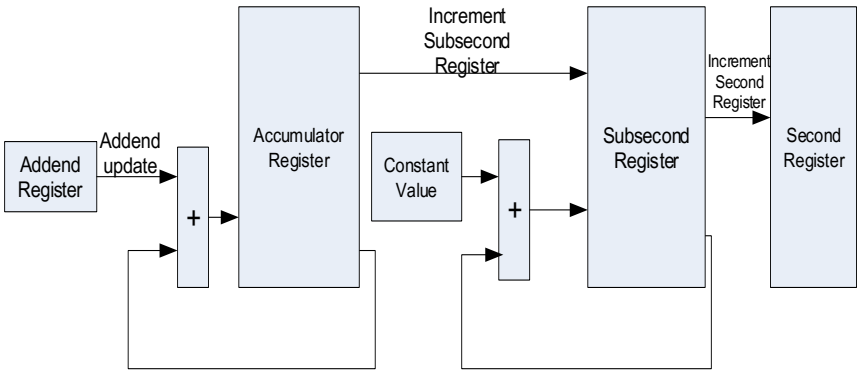
If the coarse correction method is selected, application can configure PTP timestamp update register (ENET_PTP_TSUH and ENET_PTP_TSUL) for system time initialization or correction. If TMSSTI bit is set, PTP timestamp update register is used for initialization and if TMSSTU bit is set, PTP timestamp update register is used for adjust system time by adding or subtracting.

If fine correction method is selected, operation is different. The fine correction method corrects system time not in a single clock cycle. The fine correction frequency can be configured by application to make slave clock frequency smoothly adapt master clock without unpredictability large jitter.

This method is referred to the value of ENET_PTP_TSADDEND added to the accumulator in each HCLK cycle. PTP module will produce a pulse to increase the value of ENET_PTP_TSL register when the accumulator overflowed. The increased value when this pulse occurs is in ENET_PTP_SSINC register.

Figure 54-12. System time update using the fine correction method shows the fine correction algorithm process:

Figure 54-12. System time update using the fine correction method



The following concrete example is used to descript the fine correction method how to update the system time:

Assuming the accuracy of the system time update circuit required to achieve 25ns, which means the frequency of update is 40MHz. If the reference clock of HCLK is 72MHz, the frequency ratio is calculated as 72 / 40, result is 1.8. Hence, the addend (TMSA bit in ENET_PTP_TSADDEND register) value to be set is 2^{32} / 1.8, which is equal to 0x8E38 E38E. If the reference clock frequency drifts lower, for example, down to 68MHz, the frequency ratio changes to 68 / 40 = 1.7, the value to be set in the addend register is 2^{32} / 1.7 = 0x9696 9697. If the reference clock drift higher, for example, up to 76MHz, the value should to be set in the ENET_PTP_TSADDEND register is 2^{32} / 1.9 = 0x86BC A1AF. Initially, the slave clock frequency is set to Clock Addend Value (0) in the addend register. This value is calculated as above. In addition to configuring the addend counter, application also needs to set subsecond increment register to ensure to achieve the precision of 25ns. The value of the register is to update values of timestamp low 32-bit register after accumulator register overflow. Because the STMSS[30:0] bits in the ENET_PTP_TSL register represents the subsecond value of system time, the precision is 10^9ns / 2^{31}=0.46ns. So in order to make the system time accuracy to 25ns, subsecond increment register value should be set to 25 / 0.46 = 0d54.

Note: The algorithm described below based on constant delay transferred between master and slave devices (Master-to-Slave-Delay). Synchronous frequency ratio will be confirmed by the algorithm after a few Sync cycles.

Algorithm is as follows:

- Define the master sends a SYNC message to slave time: MSYNCT (n).
  Define the slave local time: SLOCALT (n).
  Define the master local time: MLOCALT (n).
  Calculation: MLOCALT (n) = MSYNCT (n) + Master-to-Slave-Delay (n).

- Define the master clock count number between two SYNC message sent: MCLOCKC(n).
  Calculation: MCLOCKC (n) = MLOCALT (n) - MLOCALT (n-1).
  Define the slave clock count number between two received SYNC messages: SCLOCKC (n).

Calculation: SCLOCKC (n) = SLOCALT (n) - SLOCALT (n-1).

- Define the difference between these two count numbers: DIFFCC (n).
  Calculation: DIFFCC (n) = MCLOCKC (n) - SCLOCKC (n).
- Define the slave clock frequency-adjusting factor: SCFAF (n).
  Calculation: SCFAF (n) = (MCLOCKC (n) + DIFFCC (n)) / SCLOCKC (n).
- Define the Clock Addend Value for addend register: Clock Addend Value (n).
  Clock Addend Value (n) = SCFAF (n) * Clock Addend Value (n-1).

Note: During the actual operation, application may need more than once SYNC message between master and slave to lock.

System time initialization procedure

Setting TMSEN bit in ENET_PTP_TSCTL register to 1, timestamp function is enabled. Each time after this bit is set from reset, application must initialize the timestamp counter at first. Initialization steps as follow:

1. Setting TMSTIM in the ENET_MAC_INTMSK register to mask the timestamp trigger interrupt;
2. Setting TMSEN in the ENET_PTP_TSCTL register to enable timestamp function;
3. Configure the subsecond increment register according to the PTP clock frequency precision;
4. If application hopes to use fine correction method, configure the timestamp addend register and set TMSARU in the ENET_PTP_TSCTL register to 1. If application hopes to use coarse correction method, please jump directly to step 7 and step 4-6 can be ignored;
5. Poll the TMSARU in the ENET_PTP_TSCTL register until it is cleared;
6. Set TMSFCU in the ENET_PTP_TSCTL register to 1 to choose fine correction method;
7. Configure the timestamp update high and low register with the value of system time application wants to initialize;
8. Send initialization command by setting TMSSTI in the ENET_PTP_TSCTL register;
9. The timestamp counter starts counting as soon as the initialization process complete.

System time update steps

Coarse correction method

1. Program the offset (may be negative) value in the timestamp update high and low registers;
2. Set bit 3 (TMSSTU) in the ENET_PTP_TSCTL register to update the timestamp register;
3. Poll TMSSTU bit until it is cleared.

Fine correction method

2. Calculate the value of the desired system clock rate corresponding to the addend register (***System time calibration*** has explained before);

3. Program the addend register, and set the TMSARU in ENET_PTP_TSCTL register;
4. Program the target high and low register and reset the TMSTIM of the ENET_MAC_INTMSK register to allow time stamp interrupt;
5. Set TMSITEN in ENET_PTP_TSCTL register;
6. When an interrupt is generated by this event, read out the value of ENET_MAC_INTF register and clear the corresponding interrupt flag;
7. Rewrite the old value of addend register to timestamp addend register and set TMSARU in ENET_PTP_TSCTL register.

Transmission and reception of frames with the PTP feature

After enabled the IEEE 1588 (PTP) timestamp function, timestamp is recorded when the frame's SFD field is outputting from the MAC or the MAC receives a frame's SFD field. Each transmitted frame can be marked in TxDMA descriptor to indicate whether a timestamp should be captured or not, which is unrelated with whether the transmitted frame has PTP feature or not, and the timestamp of all received frames will be recorded if ARFSEN bit in ENET_PTP_TSCTL register is set. If ARFSEN is reset, the received frame which passed the address filer should be matched with the configuration in ENET_PTP_TSCTL register. In another word, only the frame matched the PTP configuration is marked a PTP frame, and timestamp will be recorded in descriptor. To be marked as a PTP frame, the received frame PTP version should be coincide with PFSV bit and then the corresponding frame type enable bit (bit 13 to bit 11 in register ENET_PTP_TSCTL) is set. Specially, the non-IP payload PTP frame (PTP over Ethernet II frame), also the DA should be the special MAC address (e.g. the DA should be 0x0e00 00c2 8001 for PDELAY_REQ / PDELAY_RESP / PDELAY_RESP_FOLLOW_UP message type, and the DA address 0x0000 0019 1B01 for other message type, detailed informations refer to Specification IEEE1588-2008). If MAFEN is set, this special MAC address can be extended to MAC address1-3 with SAF is reset.

Together with the state information of frame, the recorded timestamp value will also be stored in the corresponding transmission / reception descriptor. The 64-bit timestamp information of transmission frame is written back to the transmit descriptor and the 64-bit timestamp information of reception frame is written back to the receive descriptor. See the detailed description in "***Transmit DMA descriptor with IEEE 1588 timestamp format***" and "***Receive DMA descriptor with IEEE 1588 timestamp format***".

Internal connection trigger

MAC can provide trigger interrupt when the system time is no less than the expected time. Using an interrupt imports a known latency and an uncertainty in the command execution time. Set the TSCFG5[4:0]/TSCFG4[4:0]/TSCFG3[4:0] bit-field in SYSCFG_TIMERxCFG0(x = 1/2/30/31) register to 5'b01001 can make ENET0 signal internally connected to the ITI4 input of event mode configuration/pause mode configuration/restart mode configuration in TIMERx(x = 1/2/30/31). And set the TSCFG7[4:0]/TSCFG6[4:0] bit-field in SYSCFG_TIMERxCFG1(x = 1/2/30/31) register to 5'b01001 can make ENET1 signal internally connected to the ITI4 input of restart + event mode configuration/external clock mode

0 configuration in TIMERx(x = 1/2/30/31).. For this feature designed, no uncertainty is introduced because the clock of the TIMER1/2/30/31 and PTP reference clock (HCLK) are synchronous.

PPS output signal

Enable the ENET0 and ENET1 PPS output function by configuring ENET0_PPS_OUT pin to AF11 and ENET1_PPS_OUT pin to AF6, respectively. This function can output a signal with the pulse width of 125ms by default (other width is detailed in ***PTP PPS control register (ENET_PTP_PPSCTL)***) which can be used to check the synchronization between all nodes in the network. To test the difference between the slave clock and the master clock, both of the slave and master can output PPS(pulse-per-second) and connect them to one oscilloscope for clock measurement.

54.3.7. Example for a typical configuration flow of Ethernet

After power-on reset or system reset, the following operation flow is a typical process for application to configure and run Ethernet:

- **Enable Ethernet clock.**

  Program the RCU module to enable the HCLK and Ethernet Tx / Rx clock.

- **Setup the communication interface.**

  Configure SYSCFG_PMCFG0 to define which interface mode is selected (MII or RMII). Configure GPIO module to make selected PADs to alternate function.

- **Wait the resetting complete**

  Polling the ENET_DMA_BCTL register until the SWR bit is reset. (SWR bit is set by default after power-on reset or system reset).

- **Obtain and configure the parameters in PHY register**

  According to the frequency of HCLK, configure the SMI clock frequency and access external PHY register to obtain the information of PHY (e.g. support Half / Full duplex or not, support 10M / 100Mbit speed or not, and so on). Based on supported mode of external PHY, configure ENET_MAC_CFG register consistent with PHY register.

- **Initialize the DMA in Ethernet module for transaction**

  Configure the ENET_DMA_BCTL, ENET_DMA_RDTADDR, ENET_DMA_TDTADDR, ENET_DMA_CTL registers to initialize the DMA module. (Detailed information refer to ***DMA controller description***).

- **Initialize the physical memory space for descriptor table and data buffer**

  According to the address value in ENET_DMA_RDTADDR and ENET_DMA_TDTADDR register, program transmitting and receiving descriptors (with DAV=1) and data buffer.

■ **Enable MAC and DMA module to start transmit and receive**

Set TEN and REN bit in ENET_MAC_CFG register to make MAC work for transmit and receive. Set STE and SRE bit in ENET_DMA_CTL register to make DMA controller work for transmit and receive.

■ **If transmitting frames is needed**

1. Choose one or more programmed transmitting descriptor, write the transmit frame data into buffer address which is decided in Transmit Descriptor;
2. Set the DAV bit in these one or more transmit frame descriptor;
3. Write any value in ENET_DMA_TPEN register to make TxDMA exit suspend state and start transmitting;
4. There are two methods for application to confirm whether current transmitting frame is complete or not. The first method is that application can poll the DAV bit of current transmit descriptor until it is reset, this means the transmitting is complete. The second method can be used only when INTC=1. Application can poll the TS bit in ENET_DMA_STAT register until it is set, this means the transmitting is complete.

■ **If receiving frames is enabled**

10.2 Check the first receive descriptor in descriptor table (whose address is configured in ENET_DMA_RDTADDR register);

10.3 If DAV bit in Receive Descriptor0 is reset, then the descriptor is used and receive buffer space has stored the receive frame;

10.4 Handling this receive frame data;

10.5 Set DAV bit of this descriptor to release this descriptor for new frame receiving;

10.6 Check next descriptor in table, then goes to Step 2.

54.3.8. Ethernet interrupts

There are two interrupt vectors in Ethernet module. The first interrupt vector is made up of normal operation interrupts and the second vector is made up of WUM events for wakeup which is mapped to the EXTI line 19.

All of the MAC and DMA controller interrupt are connected to the first interrupt vector. The description for the MAC interrupt and DMA controller interrupt are showed behind.

The WUM block event is connected to the second interrupt vector. The event can be remote wakeup frame received event or / and Magic Packet wakeup frame received event. This interrupt is inner mapped on the EXTI line 19. So, if the EXTI line 19 is enabled and configured to trigger by rising edge, the Ethernet WUM event can make the system exiting Deep-sleep mode after a WUM event occurred. In addition, if the WUM interrupt is not masked, both the EXTI line 19 interrupt and Ethernet normal interrupt to CPU are both generated.

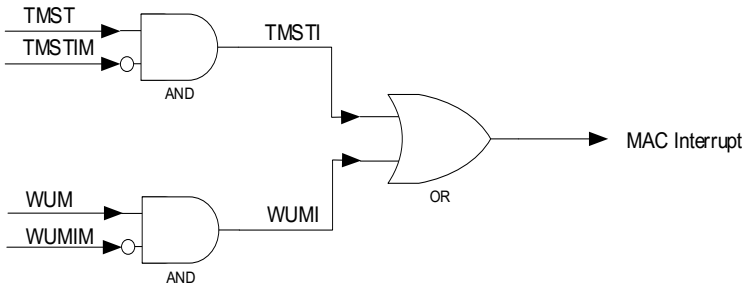
Note: Because of the WUM registers are designed in RX_CLK domain, clear these registers by reading them will need a long time delay (depends on the frequency disparity between HCLK and RX_CLK). To avoid entering the same event interrupt twice, it's strongly

recommended that application polls the WUFR and MPKR bit until they reset to zero during the interrupt service routine.

MAC interrupts

All of the MAC events can be read from ENET_MAC_INTF and each of them has a mask bit for masking corresponding interrupt. The MAC interrupt is logical Ored of all interrupts.

Figure 54-13. MAC interrupt scheme



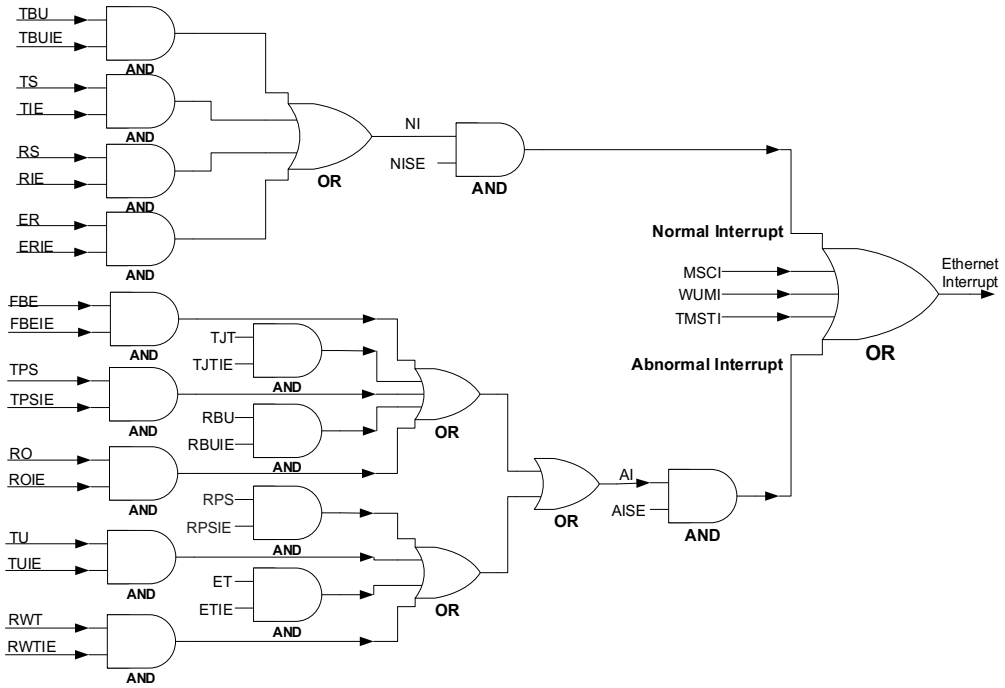
DMA controller interrupts

The DMA controller has two types of event: Normal and Abnormal.

No matter what type the event is, it has an enable bit (just like mask bit) to control the generating interrupt or not. Each event can be cleared by writing 1 to it. When all of the events are cleared or all of the event enable bits are cleared, the corresponding summary interrupt bit is cleared. If both normal and abnormal interrupts are cleared, the DMA interrupt will be cleared.

Figure 54-14. Ethernet interrupt scheme shows the Ethernet module interrupt connection:

Figure 54-14. Ethernet interrupt scheme



54.4. Register definition

ENET0 base address: 0x4002 8000
ENET1 base address: 0x4002 A000

54.4.1. MAC configuration register (ENET_MAC_CFG)

Address offset: 0x0000
Reset value: 0x0000 8000

This register can be accessed by byte (8-bit), half-word(16-bit) or word (32-bit).
This register configures the operation mode of the MAC. It also configures the MAC receiver and MAC transmitter operating mode.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						TFCD	Reserved	WDD	JBD	Reserved		IGBS[2:0]			CSD
						rw		rw	rw				rw		rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	SPD	ROD	LBM	DPM	IPFCO	RTD	Reserved	APCD	BOL[1:0]		DFC	TEN	REN	Reserved	
	rw	rw	rw	rw	rw	rw		rw	rw		rw	rw	rw		

Bits	Fields	Descriptions
31:26	Reserved	Must be kept at reset value.
25	TFCD	Type Frame CRC Dropping 0: FCS field (last 4 bytes) of frame will not be dropped before forwarding 1: FCS field (last 4 bytes) of frame will be dropped before forwarding **Note:** This bit only valid when LT field of frame greater than 0x0600.
24	Reserved	Must be kept at reset value.
23	WDD	Watchdog disable bit This bit indicates the maximum bytes for receiving, data beyond this will be cut off. 0: The received frame that Less than or equals to 2048 bytes is allowed by MAC 1: The watchdog timer that on the receiver is disabled by MAC. And the received frame up to 16384 bytes is allowed by MAC.
22	JBD	Jabber disable bit This bit indicates the maximum bytes for transmitting data, data beyond this will be cut off. 0: The maximum transmission byte is 2048 1: The maximum transmission byte can be 16384
21:20	Reserved	Must be kept at reset value.
19:17	IGBS[2:0]	Inter frame gap bit selection bits These bits can select the minimum inter frame gap bit time between two

		neighboring frames during transmission. 0x0: 96 bit times 0x1: 88 bit times 0x2: 80 bit times 0x3: 72 bit times 0x4: 64 bit times 0x5: 56 bit times (For Half-duplex, must be reserved) 0x6: 48 bit times (For Half-duplex, must be reserved) 0x7: 40 bit times (For Half-duplex, must be reserved)
16	CSD	Carrier sense disable bit 0: The carrier sense error is generated by MAC transmitter, and the transmission will be aborted. 1: The MII CRS signal is ignored by MAC transmitter while in frame transmitting. Loss of carrier error and no carrier error will not be generated.
15	Reserved	Must be kept at reset value.
14	SPD	Fast Ethernet speed bit Indicates the speed in Fast Ethernet mode: 0: 10 Mbit / s 1: 100 Mbit / s
13	ROD	Receive own disable bit When in Full-duplex mode, this bit can be ignored. 0: The packets that transmitting from PHY are all received by MAC 1: Receiving frames from PHY is disabled by MAC
12	LBM	Loopback mode bit 0: The MAC is configured in normal mode 1: The MAC is configured in loopback mode at the MII
11	DPM	Duplex mode bit 0: Half-duplex mode enable 1: Full-duplex mode enable
10	IPFCO	IP frame checksum offload bit 0: The checksum offload function in the receiver is disabled 1: IP frame checksum offload function enabled for received IP frame
9	RTD	Retry disable bit When in Full-duplex mode, this bit can be ignored. 0: Up to 16 times retries based on the settings of BOL is attempted by MAC 1: Only 1 transmission is attempted by MAC
8	Reserved	Must be kept at reset value.
7	APCD	Automatic pad / CRC drop bit This bit only valid for a non tagged frame and its length field value is equal or less

		than 1536. 0: The MAC forwards all received frames without modify it 1: The MAC strips the Pad / FCS field on received frames
6:5	BOL[1:0]	Back-off limit bits When in Full-duplex mode, these bits can be ignored. When a collision occurred, the MAC needs to retry sending current frame after delay some time. The base time unit for this delay time (dt) called slot time which means 1 slot time is equal to 512 bit times. This delay time (dt) is a random integer number calculated by following formula: $0<=dt<2^{k.}$ 0x0: k = min (n, 10) 0x1: k = min (n, 8) 0x2: k = min (n, 4) 0x3: k = min (n, 1), n = number of times for retransmission attempt
4	DFC	Deferral check bit When in Full-duplex mode, this bit can be ignored. 0: Disable the deferral check function of MAC. Until the CRS signals changed to inactive, the MAC defers sending. 1: Enable the deferral check function of MAC. If deferred more than 24288 bit times, excessive deferral error occurs and MAC abort transmitting frame. If CRS signal active during deferral time running, the deferral time will reset and restart.
3	TEN	Transmitter enable bit 0: The MAC transmit function is disabled after finish the transmission of the current frame, and no frames to be transmitted anymore. 1: The transmit function of the MAC is enabled for transmission
2	REN	Receiver enable bit 0: The MAC reception function is disabled after finish the reception of the current frame, and no frames will be received anymore. 1: The MAC reception function is enabled for receiving frames
1:0	Reserved	Must be kept at reset value.

54.4.2. MAC frame filter register (ENET_MAC_FRMF)

Address offset: 0x0004
Reset value: 0x0000 0000

This register can be accessed by byte (8-bit), half-word(16-bit) or word (32-bit).

This register configures the filtering method for receiving frames

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FAR	Reserved														
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Reserved	HPFLT	SAFLT	SAIFLT	PCFRM[1:0]	BFRMD	MFD	DAIFLT	HMF	HUF	PM
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31	FAR	Frames all received bit This bit controls the receive filter function. 0: Only the frame passed the filter can be forwarded to application 1: All received frame are forwarded to application. But filter result will also be updated to receive descriptor status.
30:11	Reserved	Must be kept at reset value.
10	HPFLT	Hash or perfect filter bit 0: If the HUF or HMF bit is set, only frames that match the hash filter are passed. 1: If the HUF or HMF bit is set, the receive filter passes frames that match either the perfect filtering or the hash filtering.
9	SAFLT	Source address filter bit Enable source address filtering function besides destination address filtering. The filter also compares the SA field value in received frames with the values configured in the enabled SA registers. If SA comparison matches, the SA match bit in the receive descriptor status is set high. 0: Source address function in filter disable 1: Source address function in filter enable
8	SAIFLT	Source address inverse filtering bit This bit makes the result of SA matching inverse. 0: Not inverse for source address filtering 1: Inverse source address filtering result. When SA matches the enabled SA registers, filter marks it as failing the SA address filter.
7:6	PCFRM[1:0]	Pass control frames bits These bits set the forwarding conditions for all control frames (including unicast and multicast pause frame). For pause control frame, the processing (not forwarding) depends only on RFCEN in ENET_MAC_FCTL[2]. 0x0: The MAC does not forward any control frames to the application 0x1: The MAC forwards any control frames except pause control frames to the application 0x2: Even if the control frames failed the address filter, the MAC forwards all of them to application 0x3: Only the control frames pass the address filter, the MAC forwards them to application
5	BFRMD	Broadcast frames disable bit 0: Ignore the address filters, and all received broadcast frames is passed.

		1: All received broadcast frames is filtered by address filters
4	MFD	Multicast filter disable bit 0: Multicast filter is enabled. The filtering mode of multicast frame is determined by HMF bit. 1: Multicast filter is disabled. All received multicast frames are passed. The first bit in the destination address field of multicast frames is '1', but not all bits in the destination are '1'.
3	DAIFLT	Destination address inverse filtering bit This bit makes the result of DA filtering inverse. 0: Not inverse DA filtering result 1: Inverse DA filtering result
2	HMF	Hash multicast filter bit 0: The filter uses perfect mode for filtering multicast frame. 1: The filter uses hash mode for filtering multicast frame
1	HUF	Hash unicast filter bit 0: The filter uses perfect mode for filtering unicast frame 1: The filter uses hash mode for filtering unicast frame
0	PM	Promiscuous mode bit This bit can make the filter bypassed which means all received frames are thought pass the filer and DA / SA filtering result status in descriptor is always '0'. 0: Promiscuous mode disabled 1: Promiscuous mode enabled

54.4.3. MAC hash list high register (ENET_MAC_HLH)

Address offset: 0x0008

Reset value: 0x0000 0000

This register can be accessed by byte (8-bit), half-word(16-bit) or word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HLH[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HLH[15:0]															
rw															

Bits	Fields	Descriptions
31:0	HLH[31:0]	Hash list high bits These bits take the high 32-bit value of hash list.

54.4.4. MAC hash list low register (ENET_MAC_HLL)

Address offset: 0x000C

Reset value: 0x0000 0000

This register can be accessed by byte (8-bit), half-word(16-bit) or word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HLL[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HLL[15:0]															
rw															

Bits	Fields	Descriptions
31:0	HLL[31:0]	Hash list low bits These bits take the low 32-bit value of hash list.

54.4.5. MAC PHY control register (ENET_MAC_PHY_CTL)

Address offset: 0x0010

Reset value: 0x0000 0000

This register can be accessed by byte (8-bit), 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
PA[4:0]					PR[4:0]					Reserved	CLR[2:0]			PW	PB
rw					rw						rw			rw	rc_w1

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:11	PA[4:0]	PHY address bits These bits choose which PHY device is to be accessed.
10:6	PR[4:0]	PHY register bits These bits choose the register address in selected PHY device.
5	Reserved	Must be kept at reset value.
4:2	CLR[2:0]	Clock range bits MDC clock divided factor select which is decided by HCLK frequency range. 0x0: HCLK/42 (HCLK range: 60-100 MHz) 0x1: HCLK/62 (HCLK range: 100-150 MHz) 0x2: HCLK/16 (HCLK range: 20-35 MHz)

		0x3: HCLK/26 (HCLK range: 35-60 MHz) 0x4: HCLK/102 (HCLK range: 150-250 MHz) 0x5: HCLK/124 (HCLK range: 250-300 MHz) 0x6: HCLK/142 (HCLK range: 300-350 MHz) 0x7: HCLK/162 (HCLK range: 300-375 MHz)
1	PW	PHY write bit This bit indicates the PHY operation mode. 0: Sending read operation to PHY 1: Sending write operation to PHY
0	PB	PHY busy bit This bit indicates the running state of operation on PHY. Application sets this bit to 1 and should wait it cleared by hardware. Application must make sure this bit is zero before writing data to ENET_MAC_PHY_CTL register and reading / writing data from / to ENET_MAC_PHY_DATA register.

54.4.6. MAC PHY data register (ENET_MAC_PHY_DATA)

Address offset: 0x0014
Reset value: 0x0000 0000

This register can be accessed by byte (8-bit), 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
PD[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	PD[15:0]	PHY data bits For reading operation, these bits contain the data from external PHY. For writing operation, these bits contain the data will be sent to external PHY.

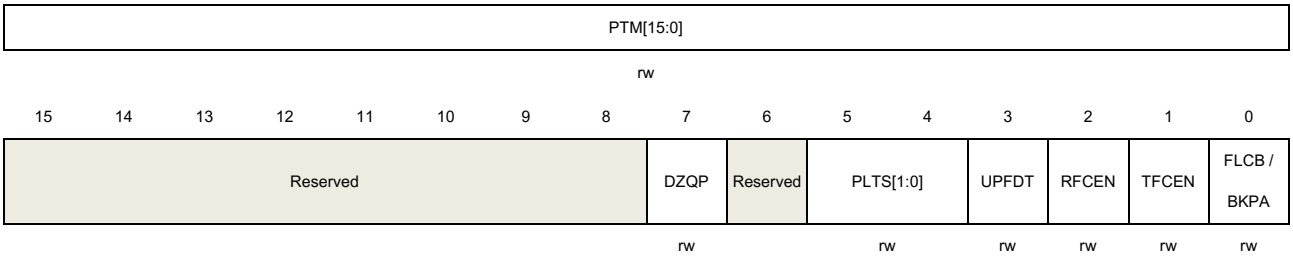
54.4.7. MAC flow control register (ENET_MAC_FCTL)

Address offset: 0x0018
Reset value: 0x0000 0000

This register can be accessed by byte (8-bit), half-word(16-bit) or word (32-bit).

This register configures the generation and reception of the control frames.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16



Bits	Fields	Descriptions
31:16	PTM[15:0]	Pause time bits These bits configured the pause time filed value in transmit pause control frame.
15:8	Reserved	Must be kept at reset value.
7	DZQP	Disable Zero-quanta pause bit 0: Enable automatic zero-quanta generation function for pause control frame 1: Disable the automatic zero-quanta generation function for pause control frame
6	Reserved	Must be kept at reset value.
5:4	PLTS[1:0]	Pause low threshold bits These bits configure the threshold of the pause timer for retransmitting frames automatically. Application must make sure the low threshold bits are greater than 0 and less than configured pause time. The low threshold calculation formula is PTM-PLTS. For example, if PTM = 0x80 (128 slot-times), and PLTS = 0x1 (28 slot-times), then the second pause frame is automatically transmitted when pause timer counted at 100 (128 - 28) slot-times after the first pause frame is transmitted. 0x0: 4 slot times is subtracted by pause time 0x1: 28 slot times is subtracted by pause time 0x2: 144 slot times is subtracted by pause time 0x3: 256 slot times is subtracted by pause time **Note:** One slot time equals the time of transmitting 512 bits on the MII interface.
3	UPFDT	Unicast pause frame detect bit 0: Only the unique multicast address for pause frame which is specified in IEEE802.3 can be detected 1: Besides the unique multicast address, MAC can also use the MAC0 address (ENET_MAC_ADDR0H and ENET_MAC_ADDR0L register) to detecting pause frame.
2	RFCEN	Receive flow control enable bit 0: Decode function for pause frame is disabled 1: Enable decoding function for the received pause frame and process it. The MAC disables its transmitter for a specified (pause time field value in received frame) time.
1	TFCEN	Transmit flow control enable bit 0: Disable the flow control operation in the MAC. Both pause frame sending in Full-duplex mode and back-pressure feature in Half-duplex mode are not performed.

1: Enable the flow control operation in the MAC. Both pause frame sending in Full-duplex mode and back-pressure feature in Half-duplex mode can be performed by transmitter.

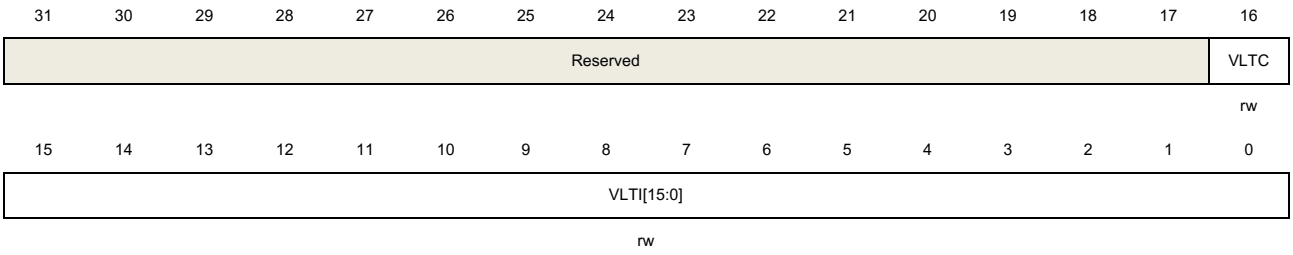
0	FLCB / BKPA	Flow control busy / back pressure activate bit This bit only valid when TFCEN is set. This bit can send a pause frame in Full-duplex mode or activate the back pressure function in Half-duplex mode by application. For Full-duplex mode, application must make sure this bit is 0 before writing ENET_MAC_FCTL register. After set by application, MAC sends a pause frame to interface and this bit will keep set until the pause frame has completed transmitting. For Half-duplex mode, MAC can enter back-pressure state by application setting this bit. When the MAC is in back-pressure state, any frame presented on interface will make the MAC send a JAM pattern to inform outside a collision occurred.

54.4.8. MAC VLAN tag register (ENET_MAC_VLT)

Address offset: 0x001C
Reset value: 0x0000 0000

This register can be accessed by byte (8-bit), half-word(16-bit) or word (32-bit).

This register configures the IEEE 802.1Q VLAN Tag to identify the VLAN frames. The MAC compares the 13th and 14th byte (length / type field) of the receiving frame with 0x8100, and the following 2 bytes (the 15th and 16th byte) are compared with the VLAN tag.



Bits	Fields	Descriptions
31:17	Reserved	Must be kept at reset value.
16	VLTC	12-bit VLAN tag comparison bit This bit selects 12 or 16 bit VLAN tag for comparison. 0: All 16 bits (the 15th and 16th byte) of the VLAN tag in received frame are used for comparison 1: Only low 12 bits of the VLAN tag in received frame are used for comparison
15:0	VLTI[15:0]	VLAN tag identifier (for receive frames) bits These bits are configured for detecting VLAN frame using 802.1Q VLAN tag format. The format shows below: VLTI[15:13]: UP (user priority)

VLTI[12]: CFI (canonical format indicator)
VLTI[11:0]: VID (VLAN identifier)
When comparison bits (VLTI[11:0] if VLTC=1 or VLTI[15:0] if VLTC=0) are all zeros, VLAN tag comparison is bypassed and every frame with type filed value of 0x8100 is considered a VLAN frame.
When comparison bits not all zeros, VLAN tag comparison use bit VLTI[11:0] (if VLTC=1) or VLTI[15:0] (if VLTC=0) for checking.

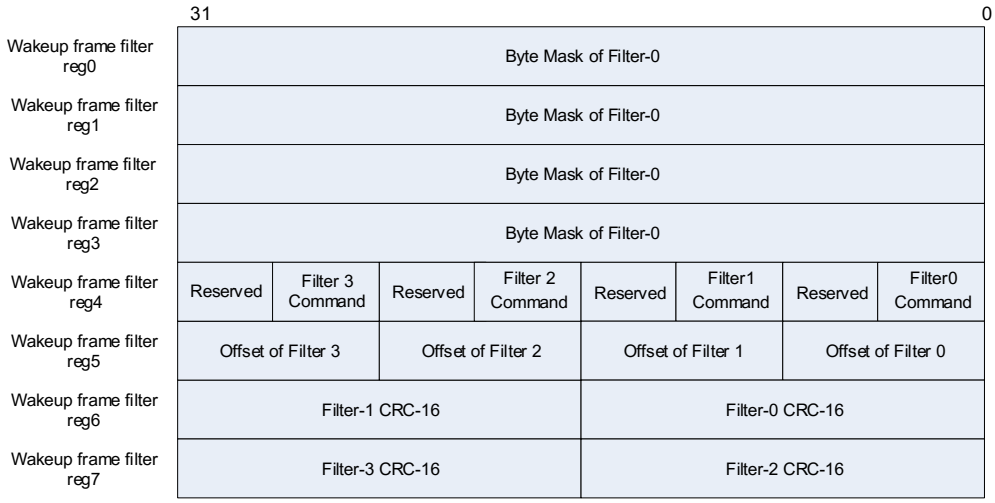
54.4.9. MAC remote wakeup frame filter register (ENET_MAC_RWFF)

Address offset: 0x0028
Reset value: 0x0000 0000

This register can be accessed by byte (8-bit), half-word(16-bit) or word (32-bit).

The MAC remote wakeup frame filter register is actually a pointer to eight (with same address offset) such wakeup frame filter registers. Eight sequential write operations to this address with the offset (0x0028) will write all wakeup frame filter registers. Eight sequential read operations from this address with the offset (0x0028) will read all wakeup frame filter registers.

Figure 54-15. Wakeup frame filter register

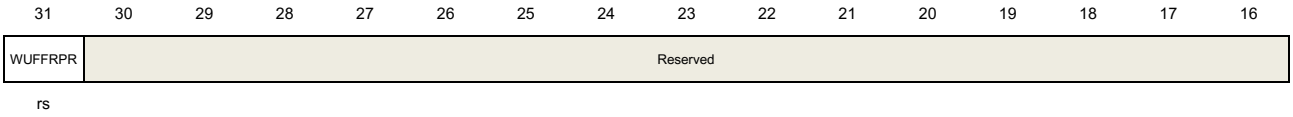


54.4.10. MAC wakeup management register (ENET_MAC_WUM)

Address offset: 0x002C
Reset value: 0x0000 0000

This register can be accessed by byte (8-bit), half-word(16-bit) or word (32-bit).

This register configures the request of wakeup events and monitors the wakeup events.



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						GU	Reserved		WUFR	MPKR	Reserved		WFEN	MPEN	PWD
						rw			rc_r	rc_r			rw	rw	rs

Bits	Fields	Descriptions
31	WUFFRPR	Wakeup frame filter register pointer reset bit This bit can reset the inner pointer of ENET_MAC_RWFF register by application set it to 1. Hardware clears it when resetting completes. 0: No effect 1: Reset the ENET_MAC_RWFF register inner pointer
30:10	Reserved	Must be kept at reset value.
9	GU	Global unicast bit 0: Not all of received unicast frame is considered to be a wakeup frame 1: Any received unicast frame passed address filtering is considered to be a wakeup frame
8:7	Reserved	Must be kept at reset value.
6	WUFR	Wakeup frame received bit This bit is cleared when this register is read. 0: Has not received the wake-up frame 1:The wakeup event was generated due to reception of a wakeup frame
5	MPKR	Magic packet received bit This bit is cleared when this register is read. 0: Has not received the Magic Packet frame 1: Received the Magic Packet frame, and generating the wakeup event
4:3	Reserved	Must be kept at reset value.
2	WFEN	Wakeup frame enable bit 0: Disable generating a wakeup event due to wakeup frame reception 1: Enable generating a wakeup event due to wakeup frame reception
1	MPEN	Magic Packet enable bit 0: Disable generating a wakeup event due to Magic Packet reception 1: Enable generating a wakeup event due to Magic Packet reception
0	PWD	Power down bit This bit is set by application and reset by hardware. When this bit is set, MAC drops all received frames. When power-down mode exit because of wakeup event occurred, hardware resets this bit.

54.4.11. MAC debug register (ENET_MAC_DBG)

Address offset: 0x0034

Reset value: 0x0000 0000

This register can be accessed by byte (8-bit), 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						TXFF	TXFNE	Reserved	TXFW	TXFRS[1:0]		PCS	SOMT[1:0]		MTNI
						ro	ro		ro	ro		ro	ro		ro

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						RXFS[1:0]		Reserved	RXFRS[1:0]		RXFW	Reserved	RXAFS[1:0]		MRNI
						ro			ro		ro		ro		ro

Bits	Fields	Descriptions
31:26	Reserved	Must be kept at reset value.
25	TXFF	TxFIFO Full flag 0: TxFIFO is not full 1: TxFIFO is full
24	TXFNE	TxFIFO not empty flag 0: TxFIFO is empty 1: TxFIFO is not empty
23	Reserved	Must be kept at reset value.
22	TXFW	TxFIFO is writing 0: TxFIFO is not doing write operation 1: TxFIFO is doing write operation
21:20	TXFRS[1:0]	TxFIFO read operation status 0x0: TxFIFO read controller is in idle state 0x1: TxFIFO read controller is in reading state 0x2: TxFIFO read controller is in waiting feedback Tx status from MAC transmitter 0x3: TxFIFO read controller is in writing the Tx descriptor status or flushing the TxFIFO
19	PCS	Pause condition status 0: MAC transmitter is not in pause condition 1: MAC transmitter is under pause condition and will delay transmitting frame
18:17	SOMT[1:0]	Status of MAC transmitter 0x0: The MAC transmitter controller is in idle state 0x1: The MAC transmitter controller is in Waiting feedback of previous frame status or the end of IFG / BACKOFF period 0x2: For Full-duplex mode, indicates pause control frame is transmitting 0x3: The MAC transmitter controller is in Reading input frame from FIFO for transmission
16	MTNI	MAC transmit state not idle 0: MAC transmitter is in idle state

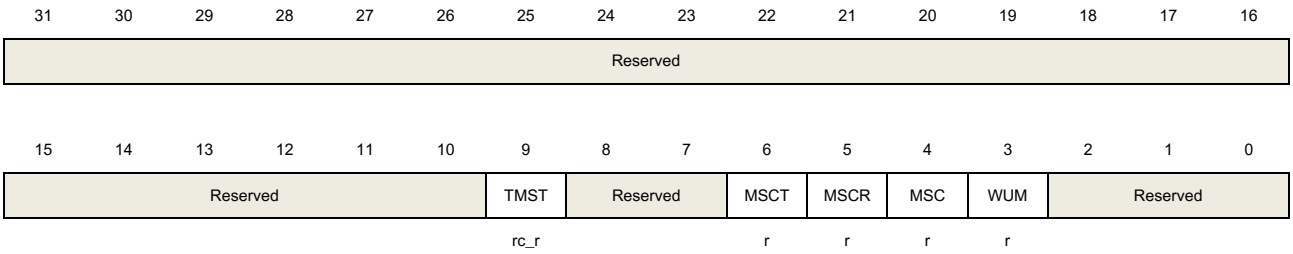
		1: MAC transmitter is not in idle state
15:10	Reserved	Must be kept at reset value.
9:8	RXFS	RxFIFO state 0x0: The RxFIFO is empty 0x1: The flow-control low threshold is greater than RxFIFO number of value 0x2: The flow-control high threshold is lower than RxFIFO number of value 0x3: The RxFIFO is full
7	Reserved	Must be kept at reset value.
6:5	RXFRS[1:0]	RxFIFO read operation status 0x0: RxFIFO read controller is in idle state 0x1: RxFIFO read controller is in reading state 0x2: RxFIFO read controller is reading frame status (including time-stamp) 0x3: RxFIFO read controller is flushing frame
4	RXFW	RxFIFO is writing 0: RxFIFO is not doing write operation 1: RxFIFO is doing write operation
3	Reserved	Must be kept at reset value.
2:1	RXAFS[1:0]	Rx asynchronous FIFO status RXAFS[1]:Rx asynchronous FIFO reading state in HCLK Clock domain RXAFS[0]:Rx asynchronous FIFO writing state in MAC RX_CLK Clock domain
0	MRNI	MAC receive state not idle 0: MAC receiver is in idle state 1: MAC receiver is not in idle state

54.4.12. MAC interrupt flag register (ENET_MAC_INTF)

Address offset: 0x0038

Reset value: 0x0000 0000

This register can be accessed by byte (8-bit), half-word(16-bit) or word (32-bit).



Bits	Fields	Descriptions
31:10	Reserved	Must be kept at reset value.

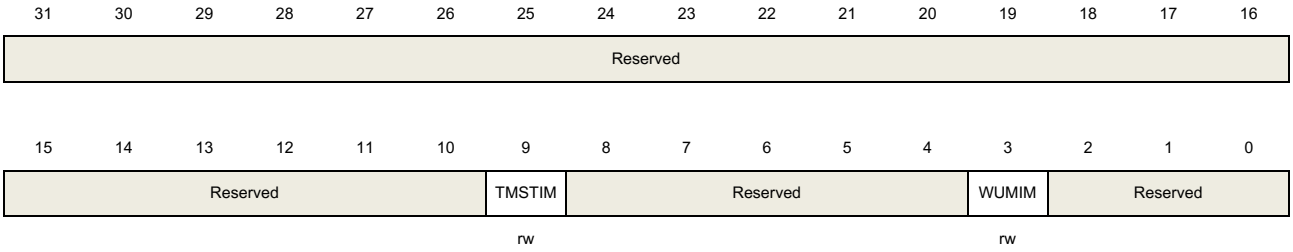
9	TMST	Time stamp trigger status bit This bit is cleared when ENET_PTP_TSF register is read. 0: The system time value is less than the value specified in the ENET_PTP_ETH and ENET_PTP_ETL registers 1: The system time value is no less than the value specified in the ENET_PTP_ETH and ENET_PTP_ETL registers
8:7	Reserved	Must be kept at reset value.
6	MSCT	MSC transmit status bit 0: All the bits in register ENET_MSC_TINTF are cleared 1: An interrupt is generated in the ENET_MSC_TINTF register
5	MSCR	MSC receive status bit 0: All the bits in register ENET_MSC_RINTF are cleared 1: An interrupt is generated in the ENET_MSC_RINTF register
4	MSC	MSC status bit This bit is logic Ored from MSCT and MSCR bit. 0: Both MSCT and MSCR bits in this register are low 1: Any of bit 6 (MSCT) or bit 5 (MSCR) is set high
3	WUM	WUM status bit This bit is logic Ored from WUFR and MPKR bit in ENET_MAC_WUM register. 0: Wakeup frame or Magic Packet frame is not received 1: A Magic packet or remote wakeup frame is received in power down Mode
2:0	Reserved	Must be kept at reset value.

54.4.13. MAC interrupt mask register (ENET_MAC_INTMSK)

Address offset: 0x003C

Reset value: 0x0000 0000

This register can be accessed by byte (8-bit), half-word(16-bit) or word (32-bit).



Bits	Fields	Descriptions
31:10	Reserved	Must be kept at reset value.
9	TMSTIM	Timestamp trigger interrupt mask bit 0: Unmask the timestamp interrupt generation

		1: Mask the timestamp interrupt generation
8:4	Reserved	Must be kept at reset value.
3	WUMIM	WUM interrupt mask bit 0: Unmask the interrupt generation due to the WUM bit in ENET_MAC_INTF register 1: Mask the interrupt generation due to the WUM bit in ENET_MAC_INTF register
2:0	Reserved	Must be kept at reset value.

54.4.14. MAC address 0 high register (ENET_MAC_ADDR0H)

Address offset: 0x0040
Reset value: 0x8000 FFFF

This register can be accessed by byte (8-bit), half-word(16-bit) or word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MO	Reserved														
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR0H[15:0]															
rw															

Bits	Fields	Descriptions
31	MO	Always read 1 and must be kept.
30:16	Reserved	Must be kept at reset value.
15:0	ADDR0H[15:0]	MAC address0 high16-bit These bits contain the high 16-bit (bit 47 to 32) of the 6-byte MAC address0. These bits are used for address filtering in frame reception and address inserting in pause frame transmitting during transmit flow control.

54.4.15. MAC address 0 low register (ENET_MAC_ADDR0L)

Address offset: 0x0044
Reset value: 0xFFFF FFFF

This register can be accessed by byte (8-bit), half-word(16-bit) or word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR0L[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR0L[15:0]															
rw															

Bits	Fields	Descriptions
31:0	ADDR0L[31:0]	MAC addresss0 low 32-bit These bits contain the low 32-bit (bit 31 to 0) of the 6-byte MAC address0. These bits are used for address filtering in frame reception and address inserting in pause frame transmitting during transmit flow control.

54.4.16. MAC address 1 high register (ENET_MAC_ADDR1H)

Address offset: 0x0048

Reset value: 0x0000 FFFF

This register can be accessed by byte (8-bit), half-word(16-bit) or word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFE	SAF	MB[5:0]						Reserved							
rw	rw	rw													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR1H[15:0]															
rw															

Bits	Fields	Descriptions
31	AFE	Address filter enable bit 0: MAC address1 is ignored by address filter for filtering 1: MAC address1 is used by address filter for perfect filtering
30	SAF	Source address filter bit 0: Comparing MAC address1 with the destination address field of the received frame 1: Comparing MAC address1 with the source address field of the received frame
29:24	MB[5:0]	Mask byte bits If these bits is set, the destination address / source address corresponding byte of the received frame is not compared with MAC address1. Each bit controls one byte mask as follows: MB[5]: ENET_MAC_ADDR1H[15:8] MB[4]: ENET_MAC_ADDR1H[7:0] MB[3]: ENET_MAC_ADDR1L[31:24] MB[2]: ENET_MAC_ADDR1L[23:16] MB[1]: ENET_MAC_ADDR1L[15:8] MB[0]: ENET_MAC_ADDR1L[7:0]
23:16	Reserved	Must be kept at reset value.
15:0	ADDR1H[15:0]	MAC address1 high[47:32] bits This field contains the high 16-bit (bit 47 to 32) of the 6-byte MAC address1.

54.4.17. MAC address 1 low register (ENET_MAC_ADDR1L)

Address offset: 0x004C

Reset value: 0xFFFF FFFF

This register can be accessed by byte (8-bit), half-word(16-bit) or word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR1L[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR1L[15:0]															
rw															

Bits	Fields	Descriptions
31:0	ADDR1L[31:0]	MAC address1 low 32-bit This field contains the low 32-bit of the 6-byte MAC address1.

54.4.18. MAC address 2 high register (ENET_MAC_ADDR2H)

Address offset: 0x0050

Reset value: 0x0000 FFFF

This register can be accessed by byte (8-bit), half-word(16-bit) or word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFE	SAF	MB[5:0]						Reserved							
rw	rw	rw													

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR2H[15:0]															
rw															

Bits	Fields	Descriptions
31	AFE	Address filter enable bit 0: MAC address2 is ignored by address filter for filtering 1: MAC address2 is used by address filter for perfect filtering
30	SAF	Source address filter bit 0: Comparing MAC address2 with the destination address field of the received frame 1: Comparing MAC address2 with the source address field of the received frame
29:24	MB[5:0]	Mask byte bits If these bits is set, the destination address / source address corresponding byte of the received frame is not compared with MAC address2. Each bit controls one byte mask as follows: MB[5]: ENET_MAC_ADDR2H[15:8]

MB[4]: ENET_MAC_ADDR2H[7:0]
MB[3]: ENET_MAC_ADDR2L[31:24]
MB[2]: ENET_MAC_ADDR2L[23:16]
MB[1]: ENET_MAC_ADDR2L[15:8]
MB[0]: ENET_MAC_ADDR2L[7:0]

23:16	Reserved	Must be kept at reset value.
15:0	ADDR2H[15:0]	MAC address2 high 16-bit This field contains the high 16-bit (bit 47 to 32) of the 6-byte MAC address2.

54.4.19. MAC address 2 low register (ENET_MAC_ADDR2L)

Address offset: 0x0054
Reset value: 0xFFFF FFFF

This register can be accessed by byte (8-bit), half-word(16-bit) or word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR2L[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR2L[15:0]															
rw															

Bits	Fields	Descriptions
31:0	ADDR2L[31:0]	MAC address2 low 32-bit This field contains the low 32-bit of the 6-byte MAC address2.

54.4.20. MAC address 3 high register (ENET_MAC_ADDR3H)

Address offset: 0x0058
Reset value: 0x0000 FFFF

This register can be accessed by byte (8-bit), half-word(16-bit) or word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFE	SAF	MB[5:0]						Reserved							
rw	rw	rw													

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR3H[15:0]															
rw															

Bits	Fields	Descriptions
31	AFE	Address filter enable bit 0: MAC address3 is ignored by address filter for filtering

		1: MAC address3 is used by address filter for perfect filtering
30	SAF	Source address filter bit 0: Comparing MAC address3 with the destination address field of the received frame 1: Comparing MAC address3 with the source address field of the received frame
29:24	MB[5:0]	Mask byte bits If these bits is set, the destination address / source address corresponding byte of the received frame is not compared with MAC address3. Each bit controls one byte mask as follows: MB[5]: ENET_MAC_ADDR3H[15:8] MB[4]: ENET_MAC_ADDR3H[7:0] MB[3]: ENET_MAC_ADDR3L[31:24] MB[2]: ENET_MAC_ADDR3L[23:16] MB[1]: ENET_MAC_ADDR3L[15:8] MB[0]: ENET_MAC_ADDR3L[7:0]
23:16	Reserved	Must be kept at reset value.
15:0	ADDR3H[15:0]	MAC address3 high 16-bit This field contains the high 16-bit (bit 47 to 32) of the 6-byte MAC address3.

54.4.21. MAC address 3 low register (ENET_MAC_ADDR3L)

Address offset: 0x005C

Reset value: 0xFFFF FFFF

This register can be accessed by byte (8-bit), half-word(16-bit) or word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR3L[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR3L[15:0]															
rw															

Bits	Fields	Descriptions
31:0	ADDR3L[31:0]	MAC address3 low 32-bit This field contains the low 32-bit of the 6-byte MAC address3.

54.4.22. MAC flow control threshold register (ENET_MAC_FCTH)

Address offset: 0x1080

Reset value: 0x0000 0015

This register can be accessed by byte (8-bit), 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									RFD[2:0]			Reserved	RFA[2:0]		
									rw				rw		

Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value.
6:4	RFD[2:0]	Threshold of deactive flow control This field configures the threshold of the deactive flow control. The value should always be less than the Threshold of active flow control value configured in bits[2:0]. When the value of the unprocessed data in RxFIFO is less than this value configured, the flow control function will deactive. 0x0: 256 bytes 0x1: 512 bytes 0x2: 768 bytes 0x3: 1024 bytes 0x4: 1280 bytes 0x5: 1536 bytes 0x6,0x7: 1792 bytes
3	Reserved	Must be kept at reset value.
2:0	RFA[2:0]	Threshold of active flow control This field configures the threshold of the active flow control. If flow control function is enabled, when the value of the unprocessed data in RxFIFO is more than this value configured, the flow control function will active. 0x0: 256 bytes 0x1: 512 bytes 0x2: 768 bytes 0x3: 1024 bytes 0x4: 1280 bytes 0x5: 1536 bytes 0x6,0x7: 1792 bytes

54.4.23. MSC control register (ENET_MSC_CTL)

Address offset: 0x0100
Reset value: 0x0000 0000

This register can be accessed by byte (8-bit), 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										AFHPM	PMC	MCFZ	RTOR	CTSR	CTR
										rw	wo	rw	rw	rw	rw

Bits	Fields	Descriptions
31:6	Reserved	Must be kept at reset value.
5	AFHPM	Almost full or half preset mode 0: Preset all MSC counters to almost-half (0x7FFF FFF0) value 1: Preset all MSC counters to almost-full (0xFFFF FFF0) value **Note:** This bit is valid only when PMC is set.
4	PMC	Preset MSC counter 0: No effect 1: Preset MSC counters to a preset value. Preset value depends on AFHPM.
3	MCFZ	MSC counter freeze bit 0: MSC counters are not frozen 1: Freezes all the MSC counters to their current value. RTOR bit can work on this frozen state.
2	RTOR	Reset on read bit 0: The MSC counters are not reset after reading MSC counter 1: The MSC counters are reset to zero after read them
1	CTSR	Counter stop rollover bit 0: The counters roll over to zero after they reached the maximum value 1: The counters do not roll over to zero after they reached the maximum value
0	CTR	Counter reset bit Cleared by hardware 1 clock after set. This bit is cleared automatically after 1 clock cycle. 0: No effect 1: Reset all counters

54.4.24. MSC receive interrupt flag register (ENET_MSC_RINTF)

Address offset: 0x0104

Reset value: 0x0000 0000

This register can be accessed by byte (8-bit), 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														RGUF	Reserved
														rc_r	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Reserved	RFAE	RFCE	Reserved
	rc_r	rc_r	

Bits	Fields	Descriptions
31:18	Reserved	Must be kept at reset value.
17	RGUF	Received good unicast frames bit 0: Good unicast frame received counter is less than half of the maximum value 1: Good unicast frame received counter reaches half of the maximum value **Note**: This flag bit needs to be cleared by reading the ENET_MSC_RGUFCNT register
16:7	Reserved	Must be kept at reset value.
6	RFAE	Received frames alignment error bit 0: Alignment error frame received counter is less than half of the maximum value 1: Alignment error frame received counter reaches half of the maximum value、 **Note**: This flag bit needs to be cleared by reading the ENET_MSC_RFAECNT register
5	RFCE	Received frames CRC error bit 0: CRC error frame received counter is less than half of the maximum value 1: CRC error frame received counter reaches half of the maximum value **Note**: This flag bit needs to be cleared by reading the ENET_MSC_RFCECNT register
4:0	Reserved	Must be kept at reset value.

54.4.25. MSC transmit interrupt flag register (ENET_MSC_TINTF)

Address offset: 0x0108
Reset value: 0x0000 0000

This register can be accessed by byte (8-bit), 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										TGF	Reserved				
										rc_r					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TGFMSC	TGFSC	Reserved													
rc_r	rc_r														

Bits	Fields	Descriptions
31:22	Reserved	Must be kept at reset value.
21	TGF	Transmitted good frames bit 0: Good frame transmitted counter is less than half of the maximum value

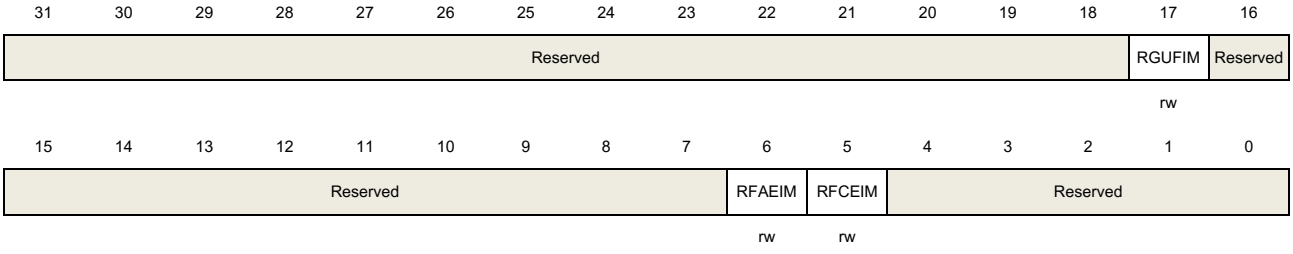
		1: Good frame transmitted counter reaches half of the maximum value **Note**: This flag bit needs to be cleared by reading the ENET_MSC_TGFCNT register
20:16	Reserved	Must be kept at reset value.
15	TGFMSC	Transmitted good frames more single collision bit 0: Good frame after more than a single collision transmitted counter is less than half of the maximum value 1:Good frame after more than a single collision transmitted counter reaches half of the maximum value **Note**: This flag bit needs to be cleared by reading the ENET_MSC_MSCCNT register
14	TGFSC	Transmitted good frames single collision bit 0: Good frame after a single collision transmitted counter is less than half of the maximum value 1: Good frame after a single collision transmitted counter reaches half of the maximum value **Note**: This flag bit needs to be cleared by reading the ENET_MSC_SCCNT register
13:0	Reserved	Must be kept at reset value.

54.4.26. MSC receive interrupt mask register (ENET_MSC_RINTMSK)

Address offset: 0x010C
Reset value: 0x0000 0000

This register can be accessed by byte (8-bit), half-word(16-bit) or word (32-bit).

The Ethernet MSC receive interrupt mask register maintains the masks for interrupts generated when receive statistic counters reach half their maximum value



Bits	Fields	Descriptions
31:18	Reserved	Must be kept at reset value.
17	RGUFIM	Received good unicast frames interrupt mask bit 0: Unmask the interrupt when the RGUF bit is set 1: Mask the interrupt when RGUF bit is set

16:7	Reserved	Must be kept at reset value.
6	RFAEIM	Received frames alignment error interrupt mask bit 0: Unmask the interrupt when the RFAE bit is set 1: Mask the interrupt when the RFAE bit is set
5	RFCEIM	Received frame CRC error interrupt mask bit 0: Unmask the interrupt when RFCE bit is set 1: Mask the interrupt when the RFCE bit is set
4:0	Reserved	Must be kept at reset value.

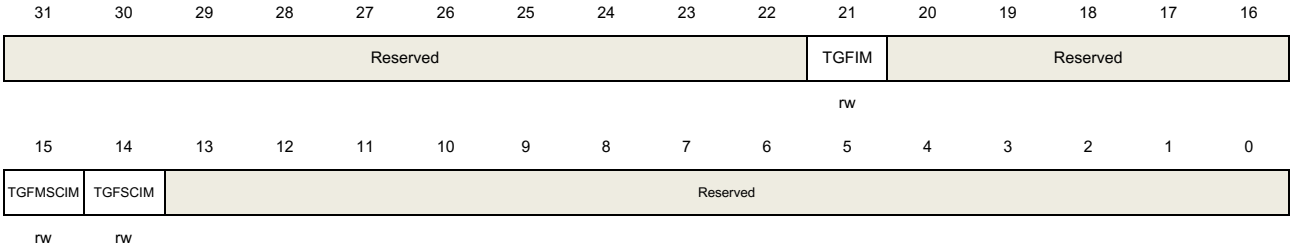
54.4.27. MSC transmit interrupt mask register (ENET_MSC_TINTMSK)

Address offset: 0x0110

Reset value: 0x0000 0000

This register can be accessed by byte (8-bit), half-word(16-bit) or word (32-bit).

The MSC transmit interrupt mask register configures the mask bits for interrupts generation



Bits	Fields	Descriptions
31:22	Reserved	Must be kept at reset value.
21	TGFIM	Transmitted good frames interrupt mask bit 0: Unmask the interrupt when the TGF bit is set 1:Mask the interrupt when the TGF bit is set
20:16	Reserved	Must be kept at reset value.
15	TGFMSCIM	Transmitted good frames more single collision interrupt mask bit 0: Unmask the interrupt when the TGFMSC bit is set 1: Mask the interrupt when the TGFMSC bit is set
14	TGFSCIM	Transmitted good frames single collision interrupt mask bit 0: Unmask the interrupt when the TFGSC bit is set 1: Mask the interrupt when the TFGSC bit is set
13:0	Reserved	Must be kept at reset value.

54.4.28. MSC transmitted good frames after a single collision counter register (ENET_MSC_SCCNT)

Address offset: 0x014C
Reset value: 0x0000 0000

This register can be accessed by byte (8-bit), half-word(16-bit) or word (32-bit).

This register counts the number of successfully transmitted frames after a single collision in Half-duplex mode.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SCC[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCC[15:0]															
r															

Bits	Fields	Descriptions
31:0	SCC[31:0]	Transmitted good frames single collision counter bits These bits count the number of a transmitted good frames after only a single collision.

54.4.29. MSC transmitted good frames after more than a single collision counter register (ENET_MSC_MSCCNT)

Address offset: 0x0150
Reset value: 0x0000 0000

This register can be accessed by byte (8-bit), half-word(16-bit) or word (32-bit).

This register counts the number of successfully transmitted frames after more than one single collision in Half-duplex mode.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MSCC[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSCC[15:0]															
r															

Bits	Fields	Descriptions
31:0	MSCC[31:0]	Transmitted good frames more one single collision counter bits These bits count the number of a transmitted good frames after more than one single collision.

54.4.30. MSC transmitted good frames counter register (ENET_MSC_TGFCNT)

Address offset: 0x0168

Reset value: 0x0000 0000

This register can be accessed by byte (8-bit), half-word(16-bit) or word (32-bit).

This register counts the number of good frames transmitted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TGF[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TGF[15:0]															
r															

Bits	Fields	Descriptions
31:0	TGF[31:0]	Transmitted good frames counter bits These bits count the number of transmitted good frames.

54.4.31. MSC received frames with CRC error counter register (ENET_MSC_RFCECNT)

Address offset: 0x0194

Reset value: 0x0000 0000

This register can be accessed by byte (8-bit), half-word(16-bit) or word (32-bit).

This register counts the number of frames received with CRC error.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RFCER[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RFCER[15:0]															
r															

Bits	Fields	Descriptions
31:0	RFCER[31:0]	Received frames with CRC error counter bits These bits count the number of receive frames with CRC error.

54.4.32. MSC received frames with alignment error counter register (ENET_MSC_RFAECNT)

Address offset: 0x0198

Reset value: 0x0000 0000

This register can be accessed by byte (8-bit), half-word(16-bit) or word (32-bit).

This register counts the number of received frames with alignment error.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RFAER[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RFAER[15:0]															
r															

Bits	Fields	Descriptions
31:0	RFAER[31:0]	Received frames alignment error counter bits These bits count the number of receive frames with alignment error.

54.4.33. MSC received good unicast frames counter register (ENET_MSC_RGUFCNT)

Address offset: 0x01C4
Reset value: 0x0000 0000

This register can be accessed by byte (8-bit), half-word(16-bit) or word (32-bit).

This register counts the number of good unicast frames received.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RGUF[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RGUF[15:0]															
r															

Bits	Fields	Descriptions
31:0	RGUF[31:0]	Received good unicast frames counter bits These bits count the number of good unicast frames received.

54.4.34. PTP time stamp control register (ENET_PTP_TSCTL)

Address offset: 0x0700
Reset value: 0x0000 2000

This register can be accessed by byte (8-bit), half-word(16-bit) or word (32-bit).

This register configures the generation and updating for timestamp.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved													MAFEN	CKNT[1:0]	
													rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MNMSEN	ETMSEN	IP4SEN	IP6SEN	ESEN	PFSV	SCROM	ARFSEN	Reserved		TMSARU	TMSITEN	TMSSTU	TMSSTI	TMSFCU	TMSEN
rw	rw	rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:19	Reserved	Must be kept at reset value.
18	MAFEN	MAC address filter enable for PTP frame 0: No effect 1: Enable MAC address1-3 to filter the PTP frame when received frame's type field is 0x88f7
17:16	CKNT[1:0]	Clock node type for time stamp 0x0: Type of ordinary clock 0x1: Type of boundary clock 0x2: Type of end-to-end transparent clock 0x3: Type of peer-to-peer transparent clock
15	MNMSEN	Received master node message snapshot enable This bit is valid only when CKNT=0x0 or 0x1. 0: Snapshot is only taken for slave node message 1: Snapshot is only take for master node message
14	ETMSEN	Received event type message snapshot enable 0: All type messages are taken snapshot except Announce, Management and Signaling message 1: Only event type messages (SYNC, DELAY_REQ, PDELAY_REQ and PDELAY_RESP) are taken snapshot
13	IP4SEN	Received PTP frame over IPv4-UDP snapshot enable 0: Do not take snapshot for PTP frame over IPv4-UDP 1: Take snapshot for PTP frame over IPv4-UDP
12	IP6SEN	Received PTP frame over IPv6-UDP snapshot enable 0: Do not take snapshot for PTP frame over IPv6-UDP 1: Take snapshot for PTP frame over IPv6-UDP
11	ESEN	Received PTP over Ethernet II packets snapshot enable 0: Do not take snapshot when received PTP over Ethernet II frame 1: Take snapshot when received PTP over Ethernet II frame
10	PFSV	PTP frame snooping version 0: Version 1 (Revision of IEEE STD. 1588-2002 / 1588-2008) 1: Version 2 (Revision of IEEE STD. 1588-2008)

9	SCROM	Subsecond counter rollover mode 0: Binary rollover mode. Subsecond rollovers when reach 0x7FFF_FFFF 1: Digital rollover mode. Subsecond rollovers when reach 0x3B9A_C9FF (0d999_999_999)
8	ARFSEN	All received frames snapshot enable 0: Not all received frames are taken snapshot 1: All received frames are taken snapshot
7:6	Reserved	Must be kept at reset value.
5	TMSARU	Time stamp addend register update bit 0: The value of ENET_PTP_TSADDEND register is not updated to the PTP block for fine correction 1: The value of ENET_PTP_TSADDEND register is updated to the PTP block for fine correction **Note**: Before user set it, the TMSARU bit must be read as 0. When update is finish, the TMSARU bit is cleared.
4	TMSITEN	Timestamp interrupt trigger enable bit 0: Disable timestamp interrupt 1: When the system time is no less than the value in ENET_PTP_ETH and ENET_PTP_ETL registers, a timestamp interrupt is generated. **Note**: After the timestamp trigger interrupt happened the TMSITEN bit is cleared.
3	TMSSTU	Timestamp system time update bit Both the TMSSTU and TMSSTI bits must be read as zero before application set this bit. 0: Not update the system time 1: Update the system time with the value in the ENET_PTP_TSUH and ENET_PTP_TSUL registers. It is cleared by hardware when the update finished.
2	TMSSTI	Timestamp system time initialize bit This bit must be read as 0 before application set it. 0: The system time is maintained without any change 1: Initializing the system time with the value in ENET_PTP_TSUH and ENET_PTP_TSUL registers. It is cleared by hardware when the initialization finished.
1	TMSFCU	Timestamp fine or coarse update bit 0: The system timestamp uses the coarse method for updating 1:The system timestamp uses the fine method for updating
0	TMSEN	Timestamp enable bit 0: Disable timestamp function 1: Enable timestamp function for transmit and receive frames **Note:** After setting this to 1, application must initialize the system time.

Table 54-8. Supported time stamp snapshot with PTP register configuration

CKNT (Bit 17:16)	0X			10		11	
MNMSEN (Bit 15)	X(*)	1	0	X			
ETMSEN (Bit 14)	0	1	1	0	1	0	1
Supported message type for snapshot	SYNC FOLLOW_UP DELAY_REQ DELAY_RESP	DELAY_REQ	SYNC	SYNC FOLLOW_UP DELAY_REQ DELAY_RESP	SYNC FOLLOW_UP	SYNC FOLLOW_UP DELAY_REQ DELAY_RESP PDELAY_REQ PDELAY_RESP	SYNC PDELAY_REQ PDELAY_RESP

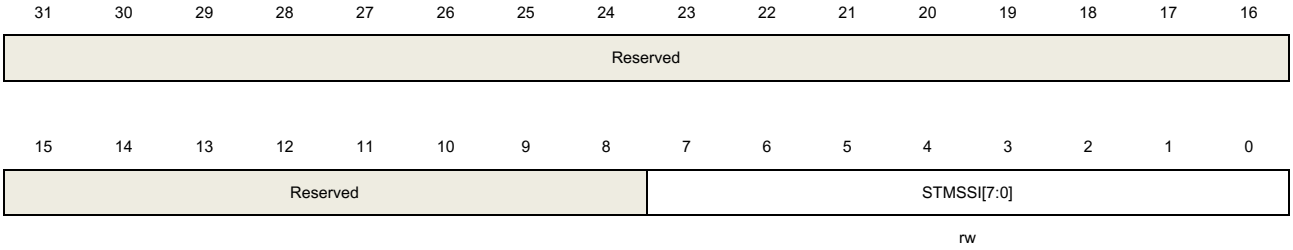
*: means do not care

54.4.35. PTP subsecond increment register (ENET_PTP_SSINC)

Address offset: 0x0704
Reset value: 0x0000 0000

This register can be accessed by byte (8-bit), half-word(16-bit) or word (32-bit).

This register configures the 8-bit value for the incrementing subsecond register. In coarse mode, this value is added to the system time every HCLK clock cycle. In fine mode, this value is added to the system time when the accumulator reaches overflow.



Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	STMSSI[7:0]	System time subsecond increment bits In every update operation, these bits are added to the subsecond value of system time.

54.4.36. PTP time stamp high register (ENET_PTP_TSH)

Address offset: 0x0708
Reset value: 0x0000 0000

This register can be accessed by byte (8-bit), half-word(16-bit) or word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
STMS[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STMS[15:0]															
r															

Bits	Fields	Descriptions
31:0	STMS[31:0]	System time second bits These bits show the current second of the system time.

54.4.37. PTP time stamp low register (ENET_PTP_TSL)

Address offset: 0x070C

Reset value: 0x0000 0000

This register can be accessed by byte (8-bit), half-word(16-bit) or word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
STS	STMSS[30:16]														
r	r														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STMSS[15:0]															
r															

Bits	Fields	Descriptions
31	STS	System time sign bit 0: Time value is positive 1: Time value is negative
30:0	STMSS[30:0]	System time subseconds bits These bits show the current subsecond of the system time with 0.46 ns accuracy

54.4.38. PTP time stamp update high register (ENET_PTP_TSUH)

Address offset: 0x0710

Reset value: 0x0000 0000

This register can be accessed by byte (8-bit), half-word(16-bit) or word (32-bit).

This register configures the high 32-bit of the time to be written to, added to, or subtracted from the system time value. The timestamp update registers (high and low) initialize or update the system time maintained by the MAC core. Application must write both of these registers before setting the TMSSTI or TMSSTU bits in the timestamp control register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TMSUS[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMSUS[15:0]															
rw															

Bits	Fields	Descriptions
31:0	TMSUS[31:0]	Time stamp update second bits These bits are used for initializing or adding / subtracting to second of the system time.

54.4.39. PTP time stamp update low register (ENET_PTP_TSUL)

Address offset: 0x0714
Reset value: 0x0000 0000

This register can be accessed by byte (8-bit), half-word(16-bit) or word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TMSUPNS	TMSUSS[30:16]														
rw	rw														
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMSUSS[15:0]															
rw															

Bits	Fields	Descriptions
31	TMSUPNS	Timestamp update positive or negative sign bit When TMSSTI is set, this bit must be 0. 0: Timestamp update value is added to system time 1: Timestamp update value is subtracted from system time
30:0	TMSUSS[30:0]	Timestamp update subsecond bits These bits are used for initializing or adding / subtracting to subsecond of the system time.

54.4.40. PTP time stamp addend register (ENET_PTP_TSADDEND)

Address offset: 0x0718
Reset value: 0x0000 0000

This register can be accessed by byte (8-bit), half-word(16-bit) or word (32-bit).

This register value is used only in fine update mode for adjusting the clock frequency. This register value is added to a 32-bit accumulator in every clock cycle and the system time

updates when the accumulator reaches overflow.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TMSA[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMSA[15:0]															
rw															

Bits	Fields	Descriptions
31:0	TMSA[31:0]	Time stamp addend bits In order to achieve time synchronization, the value of TMSA[31:0] is added to the accumulator register.

54.4.41. PTP expected time high register (ENET_PTP_ETH)

Address offset: 0x071C
Reset value: 0x0000 0000

This register can be accessed by byte (8-bit), half-word(16-bit) or word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ETSH[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETSH[15:0]															
rw															

Bits	Fields	Descriptions
31:0	ETSH[31:0]	Expected time high bits These bits store the expected target second time.

54.4.42. PTP expected time low register (ENET_PTP_ETL)

Address offset: 0x0720
Reset value: 0x0000 0000

This register can be accessed by byte (8-bit), half-word(16-bit) or word (32-bit).

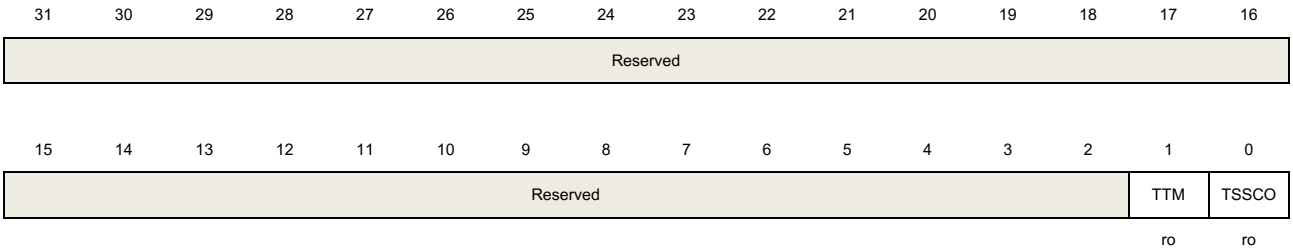
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ETSL[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETSL[15:0]															
rw															

Bits	Fields	Descriptions
31:0	ETSL[31:0]	Expected time low bits These bits store the expected target nanosecond time (signed).

54.4.43. PTP time stamp flag register (ENET_PTP_TSF)

Address offset: 0x0728
Reset value: 0x0000 0000

This register can be accessed by byte (8-bit), half-word(16-bit) or word (32-bit).

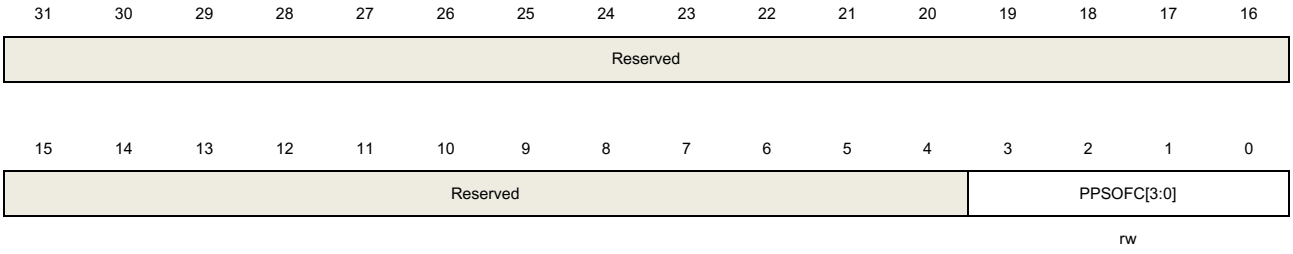


Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	TTM	Target time match bit 0: System time is not equal or greater than expected time. 1: System time is equal or greater than expected time **Note:** Reading ENET_PTP_TSF register will clear this bit.
0	TSSCO	Timestamp second counter overflow bit 0: Timestamp second counter has not overflowed 1: Timestamp second counter is greater than 0xFFFF FFFF

54.4.44. PTP PPS control register (ENET_PTP_PPSCTL)

Address offset: 0x072C
Reset value: 0x0000 0000

This register can be accessed by byte (8-bit), half-word(16-bit) or word (32-bit).



Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value.

3:0	PPSOFC	PPS output frequency configure 0x0: 1Hz (Pulse width: 125ms for binary rollover, 100ms for digital rollover) 0x1: 2Hz (Pulse width: 50% duty cycle for binary rollover) 0x2: 4Hz (Pulse width: 50% duty cycle for binary rollover) …. 0xF: 32768 (2^{15}) Hz (Pulse width: 50% duty cycle for binary rollover) **Note:** If digital rollover is selected, only PPSOFC=0 is recommended.

54.4.45. DMA bus control register (ENET_DMA_BCTL)

Address offset: 0x1000

Reset value: 0x0002 0101

This register can be accessed by byte (8-bit), 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					MB	AA	FPBL	UIP	RXDP[5:0]						FB
					rw	rw	rw	rw	rw						rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTPR[1:0]		PGBL[5:0]						DFM	DPSL[4:0]					DAB	SWR
rw		rw						rw	rw					rw	rs

Bits	Fields	Descriptions
31:27	Reserved	Must be kept at reset value.
26	MB	Mixed burst 0: AHB master interface only transfer fixed burst length with 16 and below 1: AHB master interface will transfer burst length greater than 16 with INCR **Note:** MB and FB should be and must be only one of bit is set.
25	AA	Address-aligned bit 0: Disable address-aligned 1: Enabled address-aligned. If the FB=1, all AHB interface address is aligned to the start address LS bits (bit 1 to 0). If the FB=0, the AHB interface first access address (accessing the data buffer's start address) is not aligned, but subsequent burst access addresses are aligned to the address.
24	FPBL	Four times PGBL mode bit 0: The PGBL value programmed (bits[22:17] and bits[13:8]) for the DMA data number of beats to be transferred 1: Multiple the PGBL value programmed (bits[22:17] and bits[13:8]) four times for the DMA data number of beats to be transferred
23	UIP	Use independent PGBL bit 0: The PGBL value in bits[13:8] is applicable for both TxDMA and RxDMA engines 1: The RxDMA uses the RXDP[5:0] bits as burst length while the PGBL[5:0] is used

		by TxDMA
22:17	RXDP[5:0]	RxDMA PGBL bits If UIP=0, these bits are not valid. Only when UIP=1, these bits is configured for the maximum number of beats to be transferred in one RxDMA transaction. 0x01: max beat number is 1 0x02: max beat number is 2 0x04: max beat number is 4 0x08: max beat number is 8 0x10: max beat number is 16 0x20: max beat number is 32 Other: Reserved
16	FB	Fixed burst bit 0: Both SINGLE and INCR burst transfer operations can be used by AHB 1: Only SINGLE, INCR4, INCR8 or INCR16 can be used by AHB, while in the start of normal burst transfer. **Note:** MB and FB should be and must be only one of bit is set.
15:14	RTPR[1:0]	RxDMA and TxDMA transfer priority ratio bits These bits indicate the access ratio between RxDMA and TxDMA. 0x0: RxDMA : TxDMA = 1:1 0x1: RxDMA : TxDMA = 2:1 0x2: RxDMA : TxDMA = 3:1 0x3: RxDMA : TxDMA = 4:1 **Note:** This bit is valid only when the arbitration mode is Round-robin (DAB=0).
13:8	PGBL[5:0]	Programmable burst length bits These bits indicate the maximum number of beats to be transferred in one DMA transaction. When UIP=1, the PGBL value is only used for TxDMA. When UIP=0, the PGBL value is used for both TxDMA and RxDMA. 0x01: max beat number is 1 0x02: max beat number is 2 0x04: max beat number is 4 0x08: max beat number is 8 0x10: max beat number is 16 0x20: max beat number is 32 Other: Reserved.
7	DFM	Descriptor format mode 0: Normal mode descriptor 1: Enhanced mode descriptor
6:2	DPSL[4:0]	Descriptor skip length bit These bits are valid only between two ring mode descriptors. They define the number of words (32-bit) to skip between two ring descriptors. DPSL[4:0] represents the address difference from the end of the current descriptor to the beginning of the

		next descriptor. If the value of DPSL[4:0] is 0, the DMA taking the descriptor table as contiguous.
1	DAB	DMA arbitration bit This bit indicates the arbitration mode between RxDMA and TxDMA. 0: Round-robin mode and DMA access priority is given in RTPR 1: Fixed mode. RxDMA has higher priority than TxDMA
0	SWR	Software reset bit This bit can reset all core internal registers located in CLK_TX and CLK_RX. It is cleared by hardware when the reset operation is complete in all clock domains. **Note:** Application must make sure this bit is 0 before writing any MAC core registers. 0: Core and inner register are not in reset state 1: Reset all core internal registers

54.4.46. DMA transmit poll enable register (ENET_DMA_TPEN)

Address offset: 0x1004
Reset value: 0x0000 0000

This register can be accessed by byte (8-bit), half-word(16-bit) or word (32-bit).

This register is used by the application to make the TxDMA controller poll the transmit descriptor table. The TxDMA controller can go into suspend state because of an underflow error in a transmitted frame or the descriptor unavailable (DAV=0). Application can write any value into this register for attempting to re-fetch the current descriptor.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TPE[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPE[15:0]															
rw															

Bits	Fields	Descriptions
31:0	TPE[31:0]	Transmit poll enable bits Writing to this register with any value makes DMA read the current descriptor address which is indicated in ENET_DMA_CTDADDR register. If the fetched current descriptor is available (DAV=1), DMA exits suspend state and resumes working. If the fetched current descriptor is unavailable (DAV=0), the DMA returns to suspend state again and the TBU bit in ENET_DMA_STAT register will be set.

54.4.47. DMA receive poll enable register (ENET_DMA_RPEN)

Address offset: 0x1008
Reset value: 0x0000 0000

This register can be accessed by byte (8-bit), half-word(16-bit) or word (32-bit).

This register is used by the application to make the RxDMA controller poll the receive descriptor table. Writing to this register makes the RxDMA controller exit suspend state.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RPE[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RPE[15:0]															
rw															

Bits	Fields	Descriptions
31:0	RPE[31:0]	Receive poll enable bits Writing to this register with any value makes DMA read the current descriptor address which is indicated in ENET_DMA_CRDADDR register. If the fetched current descriptor is available (DAV=1), DMA exits suspend state and resumes working. If the fetched current descriptor is unavailable (DAV=0), the DMA returns to suspend state again and the RBU bit in ENET_DMA_STAT register will be set.

54.4.48. DMA receive descriptor table address register (ENET_DMA_RDTADDR)

Address offset: 0x100C
Reset value: 0x0000 0000

This register can be accessed by byte (8-bit), half-word(16-bit) or word (32-bit).

This register points to the start of the receive descriptor table. The descriptor table is located in the physical memory space and must be word-aligned. This register can only be written when RxDMA controller is in stop state. Before starting RxDMA reception process, this register must be configured correctly.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SRT[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SRT[15:0]															
rw															

Bits	Fields	Descriptions
31:0	SRT[31:0]	Start address of receive table bits These bits indicate the start address of the receive descriptor table. SRT[1:0] are internally taken as zero so SRT[1:0] are read only.

54.4.49. DMA transmit descriptor table address register (ENET_DMA_TDTADDR)

Address offset: 0x1010
Reset value: 0x0000 0000

This register can be accessed by byte (8-bit), half-word(16-bit) or word (32-bit).

This register points to the start of the transmit descriptor table. The descriptor table is located in the physical memory space and must be word-aligned. This register can only be written when TxDMA controller is in stop state. Before starting TxDMA transmission process, this register must be configured correctly.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
STT[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STT[15:0]															
rw															

Bits	Fields	Descriptions
31:0	STT[31:0]	Start address of transmit table bits These bits indicate the start address of the transmit descriptor table. STT[1:0] are internally taken as zero so STT[1:0] are read only.

54.4.50. DMA status register (ENET_DMA_STAT)

Address offset: 0x1014
Reset value: 0x0000 0000

This register can be accessed by byte (8-bit), half-word(16-bit) or word (32-bit).

This register contains all the status bits that the DMA controller recorded. Writing 1 to meaningful bits in this register clears them but writing 0 has no effect. Each bit (bits[16:0]) can be masked by masking the corresponding bit in the ENET_DMA_INTEN register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		TST	WUM	MSC	Reserved	EB[2:0]			TP[2:0]			RP[2:0]			NI
		r	r	r		r			r			r			rc_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AI	ER	FBE	Reserved		ET	RWT	RPS	RBU	RS	TU	RO	TJT	TBU	TPS	TS
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
31:30	Reserved	Must be kept at reset value.
29	TST	Timestamp trigger status bit

		This bit indicates a timestamp event occurred. It is cleared by application through clearing TMST bit. If the corresponding interrupt mask bit is reset, an interrupt is generated. 0: Timestamp event has not occurred 1: Timestamp event has occurred
28	WUM	WUM status bit This bit indicates a WUM event occurred. It is cleared when both two source event status bits are cleared. If the corresponding interrupt mask bit is reset, an interrupt is generated. 0: WUM event has not occurred 1: WUM event has occurred
27	MSC	MSC status bit This bit indicates a MSC event occurred. It is cleared when all of event sources are cleared. If the corresponding interrupt mask bit is reset, an interrupt is generated. 0: MSC event has not occurred 1: MSC event has occurred
26	Reserved	Must be kept at reset value.
25:23	EB[2:0]	Error bits status bit When FBE=1, these bits decode the type of error that caused a bus response error on AHB bus. EB[0]: 1: Error occurs while TxDMA transfer data 0: Error occurs while RxDMA transfer data EB[1]: 1: Error occurs while read transfer 0: Error occurs while write transfer EB[2]: 1: Error occurs while access descriptor 0: Error occurs while access data buffer
22:20	TP[2:0]	Transmit process state bit These bits decode the TxDMA state. 0x0: Stopped; Issuing transmit command which is Reset or Stop. 0x1: Running; Fetching the transfer descriptor that belongs to transmit. 0x2: Running; Waiting for status 0x3: Running; Queuing it to TxFIFO after reading transmit packet data from host memory buffer. 0x4, 0x5: Reserved 0x6: Suspended; Unavailable of transmit descriptor or underflow of transmit buffer. 0x7: Running; Closing the descriptor that belongs to transmit.
19:17	RP[2:0]	Receive process state bit

		These bits decode the RxDMA state. 0x0: Stopped; Issuing receive command which is Reset or Stop. 0x1: Running; Fetching the transfer descriptor that belongs to receive. 0x2: Reserved 0x3: Running; Waiting for the packet that belongs to receive. 0x4: Suspended: Unavailable of receive descriptor 0x5: Running: Closing the descriptor that belongs to receive. 0x6: Reserved 0x7: Running; Transferring it to host memory after reading the receive packet data from RxFIFO.
16	NI	Normal interrupt summary The NI bit is logical Ored of the following if the corresponding interrupt bit is enabled in the ENET_DMA_INTEN register: TS: Interrupt of transmit TBU: Unavailable of transmit buffer RS: Interrupt of receive ER: Interrupt of early receive **Note:** Each time when this bit is set, application must cleared its source bit by writing 1 to that bit.
15	AI	Abnormal interrupt summary bit The AI bit is logical Ored of the following if the corresponding interrupt bit is enabled in the ENET_DMA_INTEN register: TPS: Halt of transmit process TJT: Timeout of transmit jabber RO: Overflow of receive FIFO TU: Underflow transmit RBU: Receive buffer RPS: Unavailable of receive process stopped RWT: Timeout of receive watchdog ET: Interrupt of early transmit FBE: Error of fatal bus error **Note:** Each time when this bit is set, application must cleared its source bit by writing 1 to that bit.
14	ER	Early receive status bit This bit is automatically cleared when the RS bit is set. 0: The first buffer has not been filled 1: The first buffer has filled with received frame
13	FBE	Fatal bus error status bit This bit indicates a response error on AHB interface is occurred and the error type can be decoded by EB[2:0] bits. 0: Bus error has not occurred

		1: A bus error occurred and the corresponding DMA stops all operations
12:11	Reserved	Must be kept at reset value.
10	ET	Early transmit status bit 0: The frame to be transmitted has not fully transferred into the TxFIFO 1: The frame to be transmitted has fully transferred into the TxFIFO
9	RWT	Receive watchdog timeout status bit 0: No received a frame with a length greater than 2048 bytes 1: A frame with a length greater than 2048 bytes is received
8	RPS	Receive process stopped status bit 0: The receive process is not in stop state 1: The receive process is in stop state
7	RBU	Receive buffer unavailable status bit 0: The DAV bit in fetched next receive descriptor is set 1: The DAV bit in fetched next receive descriptor is reset and RxDMA enters suspend state
6	RS	Receive status bit 0: Frame reception has not completed 1: Frame reception has completed
5	TU	Transmit underflow status bit 0: Underflow error has not occurred during frame transmission 1: The TxFIFO encountered an underflow error during frame transmission and entered suspend state
4	RO	Receive overflow status bit 0: Receive overflow error has not occurred during frame reception 1: The RxFIFO encountered an overflow error during frame reception. If a part of frame data has transferred to the memory, the overflow status in OERR bit is also set.
3	TJT	Transmit jabber timeout status bit 0: Transmit jabber timeout has not occurred during frame transmission 1: The transmit jabber timer expired. The TxDMA controller cancels the current transmission and enters stop state. This also causes JT bit in Transmit Descriptor0 set.
2	TBU	Transmit buffer unavailable status bit 0: The DAV bit in fetched next transmit descriptor is set 1: The DAV bit in fetched next transmit descriptor is reset and TxDMA enters suspend state
1	TPS	Transmit process stopped status bit 0: The transmission is not in stop state

1: The transmission is in stop state

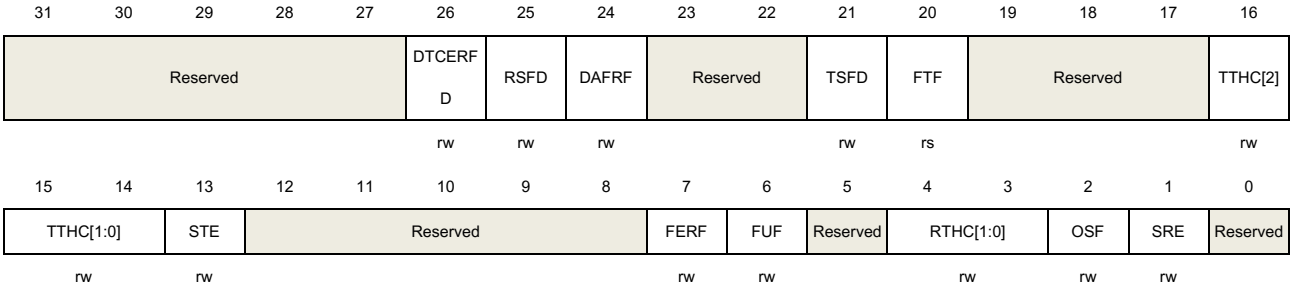
0	TS	Transmit status bit This bit can only be set when both LSG and INTC are set in Transmit Descriptor0. 0: Current frame transmission is not finished 1: Current frame transmission is finished.

54.4.51. DMA control register (ENET_DMA_CTL)

Address offset: 0x1018
Reset value: 0x0000 0000

This register can be accessed by byte (8-bit), half-word(16-bit) or word (32-bit).

This register configures both the transmitting and receiving operation modes and commands. This register should be written at last during the process of DMA initialization.



Bits	Fields	Descriptions
31:27	Reserved	Must be kept at reset value.
26	DTCERFD	Dropping of TCP / IP checksum error frames disable bit 0: All error frames will be dropped when FERF=0 1: The received frame with only payload error but no other errors will not be dropped.
25	RSFD	Receive Store-and-Forward bit 0: The RxFIFO operates in Cut-Through mode. The forwarding threshold depends on the RTHC bits 1: The RxFIFO operates in Store-and-Forward mode. The RTHC bits are don't care and the frame forwarding starts after the whole frame has pushed into RxFIFO.
24	DAFRF	Disable flushing of received frames bit 0: The RxDMA flushes all frames because of unavailable receive descriptor 1: The RxDMA does not flush any frames even though receive descriptor is unavailable
23:22	Reserved	Must be kept at reset value.
21	TSFD	Transmit Store-and-Forward bit 0: The TxFIFO operates in Cut-Through mode. The TTHC bits in ENET_DMA_CTL register defines the start popping time from TxFIFO.

		1: The TxFIFO operates in Store-and-Forward mode. Transmission on interface starts after the full frame has been pushed into the TxFIFO. The TTHC bits are don't care in this mode. **Note:** This bit can be changed when transmission is in stop state.
20	FTF	Flush transmit FIFO bit This bit can be set by application to reset TxFIFO inner control register and logic. If set, all data in TxFIFO are flushed. It is cleared by hardware after the flushing operation is finish. **Note:** Before this bit is reset, this register (ENET_DMA_CTL) must not be written.
19:17	Reserved	Must be kept at reset value.
16:14	TTHC[2:0]	Transmit threshold control bit These bits control the start transmitting byte threshold of the TxFIFO. When TSFD=1, these bits are ignored. 0x0: 64 0x1: 128 0x2: 192 0x3: 256 0x4: 40 0x5: 32 0x6: 24 0x7: 16
13	STE	Start / stop transmission enable bit 0: The TxDMA controller will enter stop state after transmitting complete if the current frame is being transmitted. After complete transmitting, the next descriptor address will become current descriptor address for the address pointer. If the TxDMA controller is in suspend state, reset this bit make the controller entering stop state. 1: The TxDMA controller will enter running state. TxDMA controller fetches current descriptor address for frame transmitting. Transmit descriptor's fetching can either from base address in ENET_DMA_TDTADDR register or from the pointer position when transmission was stopped previously. If the DAV bit of current descriptor is reset, TxDMA controller enters suspend state and the TBU bit will be set. This bit should be set after all other DMA registers have been configured otherwise the action of TxDMA is unpredictable.
12:8	Reserved	Must be kept at reset value.
7	FERF	Forward error frames bit 0: When RxFIFO is in Cut-Through mode (RSFD=0), if frame error (CRC error, collision error, checksum error, watchdog timeout, overflow error) is detected before popping RxFIFO data to memory, RxFIFO drops this error frame. But if frame error is detected after popping RxFIFO data to memory, RxFIFO will not drop this frame data. When RxFIFO is in Store-and-Forward mode, once frame error is detected

		during reception the RxFIFO drops this frame. 1: All frame received with error except runt error are forwarded to memory
6	FUF	Forward undersized good frames bit 0: The RxFIFO drops all frames whose length is less than 64 bytes. However, if this frame has already started forwarding (may due to lower value of receive threshold in Cut-Through mode), the whole frame will be forwarded. 1: The RxFIFO forwards received frame whose frame length is less than 64 bytes but without any other error
5	Reserved	Must be kept at reset value.
4:3	RTHC[1:0]	Receive threshold control bit These bits control the threshold bytes of the RxFIFO. **Note:** These bits are valid only when the RSFD=0 and are ignored when the RSFD=1. 0x0: 64 0x1: 32 0x2: 96 0x3: 128
2	OSF	Operate on second frame bit 0: The TxDMA controller process the second transmit frame after the status of the first frame is written back to descriptor 1: The TxDMA controller process the second transmit frame after pushed all first frame data into TxFIFO but before the status of the first frame is written back to descriptor
1	SRE	Start / stop receive enable bit 0: The RxDMA controller will enter stop state after transfer complete if current received frame is transmitting to memory by RxDMA. After transfer complete, the next descriptor address in the receive table will become the current descriptor address when restart the RxDMA controller. Only RxDMA controller is in running state or suspend state, this bit can be reset by application. 1: The RxDMA controller will enter running state. RxDMA controller fetches receive descriptor from receive descriptor table for receiving frames. The descriptor address can either from current address in the ENET_DMA_RDTADDR register or the address after previous frame stopped by application. If the DAV bit in fetched descriptor is reset, RxDMA controller will enter suspend state and RBU bit will be set. Setting this bit can only when RxDMA controller is in stop state or suspend state. This bit should be set after all other DMA registers have been configured otherwise the action of RxDMA is unpredictable.
0	Reserved	Must be kept at reset value.

54.4.52. DMA interrupt enable register (ENET_DMA_INTEN)

Address offset: 0x101C

Reset value: 0x0000 0000

This register can be accessed by byte (8-bit), half-word(16-bit) or word (32-bit).

This register configures the interrupts which are reflected in ENET_DMA_STAT register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															NIE
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AIE	ERIE	FBEIE	Reserved		ETIE	RWTIE	RPSIE	RBUIE	RIE	TUIE	ROIE	TJTIE	TBUIE	TPSIE	TIE
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	NIE	Normal interrupt summary enable bit 0: Disable normal interrupt 1: Enable normal interrupt This bit enables the following bits: TS: Interrupt of transmit TBU: Unavailable transmit buffer RS: Interrupt of receive ER: Interrupt of Early receive
15	AIE	Abnormal interrupt summary enable bit 0: Disable abnormal interrupt 1: Enable abnormal interrupt This bit enables the following bits: TPS: Halt of transmit process TJT: Timeout of transmit jabber RO: Overflow of receive FIFO TU: Underflow transmit RBU: Receive buffer RPS: Unavailable of receive process stopped RWT: Timeout of receive watchdog ET: Interrupt of early transmit FBE: Error of fatal bus error
14	ERIE	Early receive interrupt enable bit 0: Disable early receive interrupt 1: Enable early receive interrupt
13	FBEIE	Fatal bus error interrupt enable bit

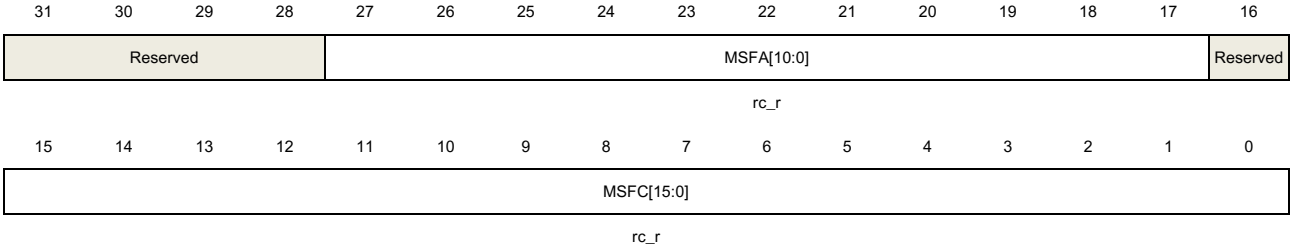
		0: Disable fatal bus error interrupt 1: Enable fatal bus error interrupt
12:11	Reserved	Must be kept at reset value.
10	ETIE	Early transmit interrupt enable bit 0: Disable early transmit interrupt 1: Enable early transmit interrupt
9	RWTIE	Receive watchdog timeout interrupt enable bit 0: Disable receive watchdog timeout interrupt 1: Enable receive watchdog timeout interrupt
8	RPSIE	Receive process stopped interrupt enable bit 0: Disable receive stopped interrupt 1: Enable receive stopped interrupt
7	RBUIE	Receive buffer unavailable interrupt enable bit 0: Disable receive buffer unavailable interrupt 1: Enable receive buffer unavailable interrupt
6	RIE	Receive interrupt enable bit 0: Disable receive interrupt 1: Enable receive interrupt
5	TUIE	Transmit underflow interrupt enable bit 0: Disable underflow interrupt 1: Enable underflow interrupt
4	ROIE	Receive overflow interrupt enable bit 0: Disable overflow interrupt 1: Enable overflow interrupt
3	TJTIE	Transmit jabber timeout interrupt enable bit 0: Disable transmit jabber timeout interrupt 1: Enable transmit jabber timeout interrupt
2	TBUIE	Transmit buffer unavailable interrupt enable bit 0: Disable transmit buffer unavailable interrupt 1: Enable transmit buffer unavailable interrupt
1	TPSIE	Transmit process stopped interrupt enable bit 0: Disable transmission stopped interrupt 1: Enable transmission stopped interrupt
0	TIE	Transmit interrupt enable bit 0: Disable transmit interrupt 1: Enable transmit interrupt

54.4.53. DMA missed frame and buffer overflow counter register (ENET_DMA_MFBOCNT)

Address offset: 0x1020
Reset value: 0x0000 0000

This register can be accessed by byte (8-bit), half-word(16-bit) or word (32-bit).

There are two counters designed in DMA controller for tracking the number of missed frames during receiving. The counter value can be read from this register for debug purpose.



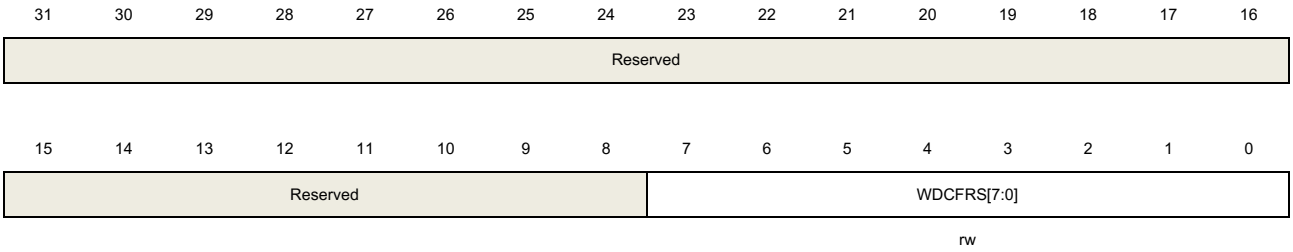
Bits	Fields	Descriptions
31:28	Reserved	Must be kept at reset value.
27:17	MSFA[10:0]	Missed frames by the application bits These bits indicate the number of frames dropped by RxFIFO.
16	Reserved	Must be kept at reset value.
15:0	MSFC[15:0]	Missed frames by the controller bits These bits indicate the number of frames missed by the RxDMA controller because of the unavailable receive buffer. Each time the RxDMA controller flushes one frame, this counter will plus 1.

54.4.54. DMA receive state watchdog counter register (ENET_DMA_RSWDC)

Address offset: 0x1024
Reset value: 0x0000 0000

This register can be accessed by byte (8-bit), half-word(16-bit) or word (32-bit).

The watchdog counter value register for RS bit (ENET_DMA_STAT register) set after delay a configured time.



Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	WDCFRS[7:0]	Watchdog counter for receive status (RS) bit These bits are only valid when DINTC (RXDES1) is set. When DINTC=1 and a frame is received, the RS bit will be set delay a time of WDCFRS*256 HCLK after receiving complete.

54.4.55. DMA current transmit descriptor address register (ENET_DMA_CTDADDR)

Address offset: 0x1048
Reset value: 0x0000 0000

This register can be accessed by byte (8-bit), half-word(16-bit) or word (32-bit).

This register points to the start descriptor address of the current transmit descriptor read by the TxDMA controller.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TDAP[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDAP[15:0]															
r															

Bits	Fields	Descriptions
31:0	TDAP[31:0]	Transmit descriptor address pointer bits These bits are automatically updated by TxDMA controller during operation.

54.4.56. DMA current receive descriptor address register (ENET_DMA_CRDADDR)

Address offset: 0x104C
Reset value: 0x0000 0000

This register can be accessed by byte (8-bit), half-word(16-bit) or word (32-bit).

This register points to the start descriptor address of the current receive descriptor read by the RxDMA controller.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RDAP[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDAP[15:0]															
r															

Bits	Fields	Descriptions
31:0	RDAP[31:0]	Receive descriptor address pointer bits These bits are automatically updated by RxDMA controller during operation.

54.4.57. DMA current transmit buffer address register (ENET_DMA_CTBADDR)

Address offset: 0x1050

Reset value: 0x0000 0000

This register can be accessed by byte (8-bit), half-word(16-bit) or word (32-bit).

This register points to the current transmit buffer address being read by the TxDMA controller.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TBAP[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TBAP[15:0]															
r															

Bits	Fields	Descriptions
31:0	TBAP[31:0]	Transmit buffer address pointer bits These bits are automatically updated by TxDMA controller during operation.

54.4.58. DMA current receive buffer address register (ENET_DMA_CRBADDR)

Address offset: 0x1054

Reset value: 0x0000 0000

This register can be accessed by byte (8-bit), half-word(16-bit) or word (32-bit).

This register points to the current receive buffer address being read by the RxDMA controller.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RBAP[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RBAP[15:0]															
r															

Bits	Fields	Descriptions
31:0	RBAP[31:0]	Receive buffer address pointer bits

	These bits are automatically updated by RxDMA controller during operation.

55. EtherCAT® SubDevice controller (ESC)

55.1. Overview

The EtherCAT® SubDevice controller (ESC) uses an EtherCAT SubDevice Controller IP Core made by Beckhoff Automation GmbH, Germany. It takes care of the EtherCAT communication as an interface between the EtherCAT fieldbus and the SubDevice application. The EtherCAT controller has 8K bytes of Process Data RAM (PDRAM) and 8 Fieldbus memory management units (FMMUs), each of which performs the task of mapping logical addresses to physical addresses. The EtherCAT SubDevice controller also includes 8 SyncManagers that allow data exchange between the EtherCAT main and the local application. The orientation and mode of operation of each SyncManager is configured by the EtherCAT MainDevice. Two working modes are available: buffer mode and mailbox mode. In buffered mode, the Cortex®-M7 MCU and EtherCAT MainDevice can write buffer simultaneously. The buffer in ESC always contains the latest data. If the new data arrives before the old data can be read, the old data will be lost. In mailbox mode, the Cortex®-M7 MCU and EtherCAT MainDevice access the buffer by shaking hands, ensuring that no data is lost.

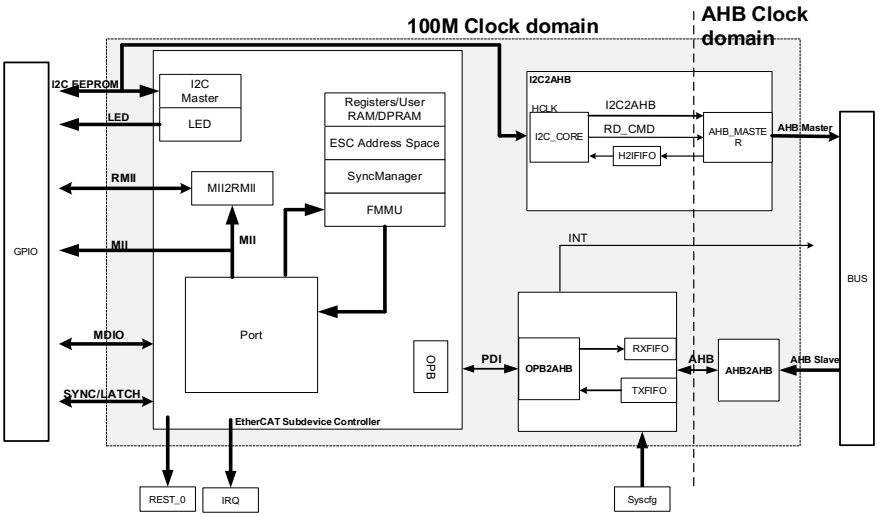
55.2. Characteristics

- Supports up to 3 MII ports, port0 and port1 support RMII ports.
- 8 Fieldbus Memory Management Units (FMMUs).
- 8KB Process Data RAM (PDRAM).
- Distributed clock 64-bit, support allows synchronization with other EtherCAT devices.
- 8 Syncmanager entities.
- DC synchronization deviation less than 1us.

55.2.1. Block diagram

The function module of the ESC is shown in ***Figure 55-1. EtherCAT system block diagram***.

Figure 55-1. EtherCAT system block diagram



55.2.2. EtherCAT SubDevice Controller Function Blocks

EtherCAT Interfaces

The EtherCAT interfaces or ports connect the ESC to other EtherCAT SubDevices and the main. The MAC layer is integral part of the ESC. The physical layer may be Ethernet. For Ethernet ports, use Ethernet PHYs connect to the MII ports of the ESC. Transmission speed for EtherCAT is fixed to 100 Mbit/s with Full Duplex communication. Link state and communication status are reported to the Monitoring device. ESC uses three ports, port 0/1/2.

Auto-Forwarder

The Auto-Forwarder receives the Ethernet frames, performs frame checking and forwards it to the Loop-back function. Time stamps of received frames are generated by the Auto-Forwarder.

Loop-back function

The Loop-back function forwards Ethernet frames to the next logical port if there is either no link at a port, or if the port is not available, or if the loop is closed for that port. The Loop-back function of port 0 forwards the frames to the EtherCAT Processing Unit. The loop settings can be controlled by the EtherCAT main.

FMMU

Fieldbus Memory Management Units are used for bitwise mapping of logical addresses to physical addresses of the ESC.

SyncManager

SyncManagers are responsible for consistent data exchange and mailbox communication between EtherCAT main and SubDevices. The communication direction can be configured

for each SyncManager. Read or write transactions may generate events for the EtherCAT main and an attached Cortex®-M7 MCU respectively. The SyncManagers are responsible for the main difference between and ESC and a dual-ported memory, because they map addresses to different buffers and block accesses depending on the SyncManager state. This is also a fundamental reason for bandwidth restrictions of the PDI.

EEPROM

One non-volatile memory is needed for EtherCAT SubDevice Information (ESI) storage, typically an I²C EEPROM.

Status / LEDs

The Status block provides ESC and application status information. It controls external LEDs like the application RUN LED/ERR LED and port Link/Activity LEDs.

55.3. Function overview

55.3.1. FMMU

Fieldbus Memory Management Units (FMMU) convert logical addresses into physical addresses by the means of internal address mapping. Thus, FMMUs allow to use logical addressing for data segments that span several SubDevice devices: one datagram addresses data within several arbitrarily distributed EtherCAT. Each FMMU channel maps one continuous logical address space to one continuous physical address space of the SubDevice. The access type supported by an FMMU is configurable to be either read, write, or read/write.

Restrictions on FMMU Settings

The FMMUs of ESCs are subject to restrictions. The logical address ranges of two FMMUs of the same direction (read or write) in one ESC must be separated by at least 3 logical bytes not configured by any FMMU of the same type, if one of the FMMUs or both use bit-wise mapping (logical start bit != 0, logical stop bit != 7, or physical start bit != 0).

Additional FMMU Characteristics

- Each logical address byte can at most be mapped either by one FMMU(read) plus one FMMU(write), or by one FMMU(read/write). If two or more FMMUs (with the same direction - read or write) are configured for the same logical byte, the FMMU with the lower number (lower configuration address space) is used, the other ones are ignored.
- One or more FMMUs may point to the same physical memory, all of them are used. Collisions cannot occur.
- It is the same to use one read/write FMMU or two FMMUs - one read, the other one write - for the same logical address.
- A read/write FMMU cannot be used together with SyncManagers, since independent read and write SyncManagers cannot be configured to use the same

(or overlapping) physical address range.

- Bit-wise reading is supported at any address. Bits which are not mapped to logical addresses are not changed in the EtherCAT datagram. E.g., this allows for mapping bits from several ESCs into the same logical byte.
- A frame/datagram addressing a logical address space which is not configured in the ESC will not change data in the ESC, and no data from the ESC is placed in the frame/datagram.

55.3.2. SyncManager

The memory of an ESC can be used for exchanging data between the EtherCAT main and a local application (on a Cortex®-M7 MCU attached to the PDI) without any restrictions. Using the memory for communication like this has some drawbacks which are addressed by the SyncManagers inside the ESCs:

- Data consistency is not guaranteed. Semaphores have to be implemented in software for exchanging data in a coordinated way.
- Data security is not guaranteed. Security mechanisms have to be implemented in software.
- Both EtherCAT main and application have to poll the memory in order to find out when the access of the other side has finished.

SyncManagers enable consistent and secure data exchange between the EtherCAT main and the local application, and they generate interrupts to inform both sides of changes.

SyncManagers are configured by the EtherCAT main. The communication direction is configurable, as well as the communication mode (Buffered Mode and Mailbox Mode). SyncManagers use a buffer located in the memory area for exchanging data. Access to this buffer is controlled by the hardware of the SyncManagers.

A buffer has to be accessed beginning with the start address, otherwise the access is denied. After accessing the start address, the whole buffer can be accessed, even the start address again, either as a whole or in several strokes. A buffer access finishes by accessing the end address, the buffer state changes afterwards and an interrupt or a watchdog trigger pulse is generated (if configured). The end address cannot be accessed twice inside a frame.

Two communication modes are supported by SyncManagers:

- Buffered Mode
  - The buffered mode allows both sides, EtherCAT main and local application, to access the communication buffer at any time. The consumer always gets the latest consistent buffer which was written by the producer, and the producer can always update the content of the buffer. If the buffer is written faster than it is read out, old data will be dropped.
  - The buffered mode is typically used for cyclic process data.
- Mailbox Mode

- The mailbox mode implements a handshake mechanism for data exchange, so that no data will be lost. Each side, EtherCAT main or local application, will get access to the buffer only after the other side has finished its access. At first, the producer writes to the buffer. Then, the buffer is locked for writing until the consumer has read it out. Afterwards, the producer has write access again, while the buffer is locked for the consumer.
- The mailbox mode is typically used for application layer protocols.

The SyncManagers accept buffer changes caused by the main only if the FCS of the frame is correct, thus, buffer changes take effect shortly after the end of the frame.

The configuration registers for SyncManagers are located beginning at register address 0x0800.

55.3.3. Distributed Clocks

The Distributed Clocks (DC) unit of EtherCAT SubDevice controllers supports the following features:

- Clock synchronization between the SubDevices (and the MainDevice).
- Generation of synchronous output signals (SyncSignals).
- Precise time stamping of input events (LatchSignals).
- Generation of synchronous interrupts.
- Synchronous Digital Output updates.
- Synchronous Digital Input sampling.

The device supports 64-bit distributed clocks as detailed in the following sub-sections.

The EtherCAT provides two input pins (SYNC and LATCH) which are used for time stamping of external events. Both rising edge and falling edge time stamps are recorded. These pins are shared with the SYNC0 and LATCH0 output pins, respectively, which are used to indicate the occurrence of time events. The functions of the SYNC/ SYNC0 and LATCH /LATCH0 pins are determined by the SYNC0/LATCH0 Configuration and SYNC/LATCH Configuration bits of the Sync/Latch PDI Configuration Register, respectively.

When set for SYNC0/LATCH0 functionality, the output type (Push-Pull vs. Open Drain/Source) and output polarity are determined by the SYNC0 Output Driver/Polarity and LATCH0 Output Driver/Polarity bits of the Sync/Latch PDI Configuration Register.

55.3.4. EtherCAT State Machine

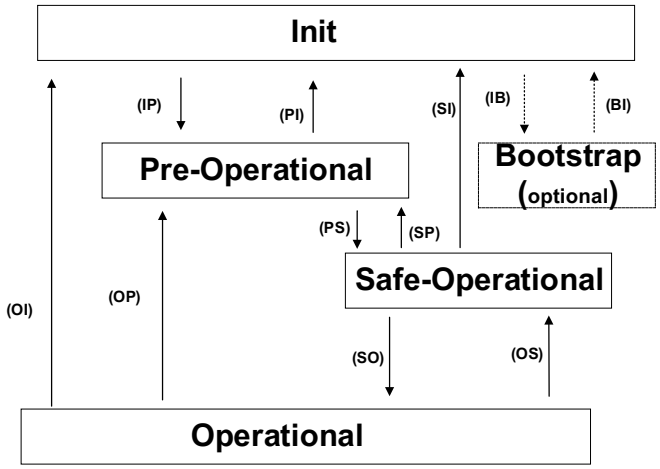
The EtherCAT State machine (ESM) is responsible for the coordination of main and SubDevice applications at start up and during operation. State changes are typically initiated by requests of the main. They are acknowledged by the local application after the associated operations have been executed. Unsolicited state changes of the local application are also possible.

There are four states an EtherCAT SubDevice shall support, plus one optional state:

- Init (state after Reset)
- Pre-Operational
- Safe-Operational
- Operational
- Bootstrap (optional)

The states and the allowed state changes are shown in *Figure 55-2. EtherCAT State Machine*:

Figure 55-2. EtherCAT State Machine



NOTE: Not all state changes are possible, the transition from 'Init' to 'Operational' requires the following sequence: Init -> Pre-Operational -> Safe-Operational -> Operational.

Each state defines required services. Before a state change is confirmed by the SubDevice all services required for the requested state have to be provided or stopped respectively.

55.3.5. EEPROM

EtherCAT SubDevice controllers use One non-volatile memory (typically a serial EEPROM with I²C interface) to store EtherCAT SubDevice Information (ESI). EEPROM sizes from 1 Kbit up to 4 Mbit are supported, depending on the ESC.

In addition, the chip also supports using internal Flash instead of an external EEPROM to store EtherCAT SubDevice Information (ESI). The size and address of the internally emulated EEPROM can be configured through the SYSCFG_ESCCFG0 register.

NOTE: When using internal EEPROM emulation, data read and write operations must be accessed on 2-byte boundaries. Single-byte access is not supported.

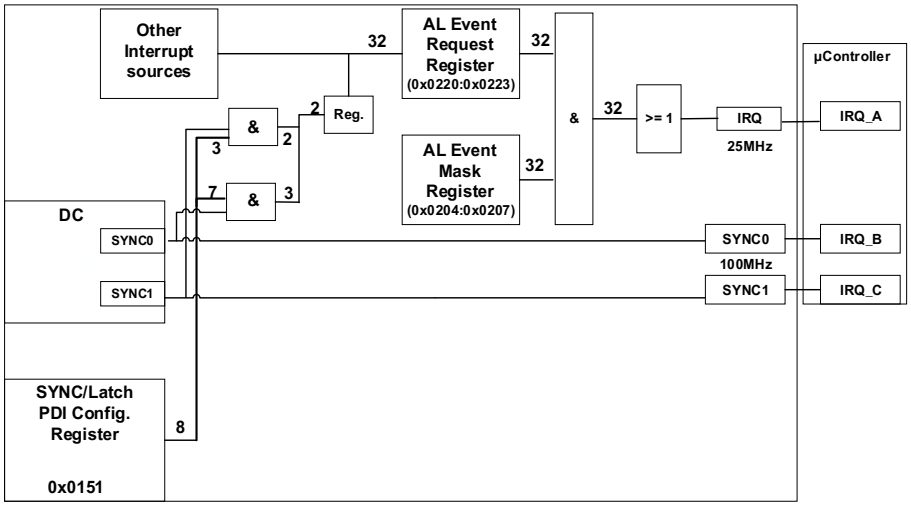
55.3.6. Interrupts

EtherCAT support two types of interrupts: AL Event Requests targeted at a Cortex®-M7 MCU, and EtherCAT event requests targeted at the EtherCAT main. Additionally, the Distributed Clocks SyncSignals can be used as interrupts for a Cortex®-M7 MCU as well.

AL Event Request (PDI Interrupt)

AL Event Requests can be signaled to a Cortex®-M7 MCU using the PDI Interrupt Request signal (IRQ, etc.). for IRQ generation, the ESC AL Event Request register (ESC_AL_EVT_REQ) is combined with the ESC PDI AL event register (ESC_PDI_AL_EVT) using a logical AND operation, then all resulting bits are combined (logical OR) into one interrupt signal. The output driver characteristics of the IRQ signal are configurable using the ESC Sync/Latch configuration register (ESC_SL_CFG). The ESC Event Mask register allows for selecting the interrupts which are relevant for the Cortex®-M7 MCU and handled by the application.

Figure 55-3. PDI Interrupt Masking and interrupt signals



The DC SyncSignals can be used for interrupt generation in two ways:

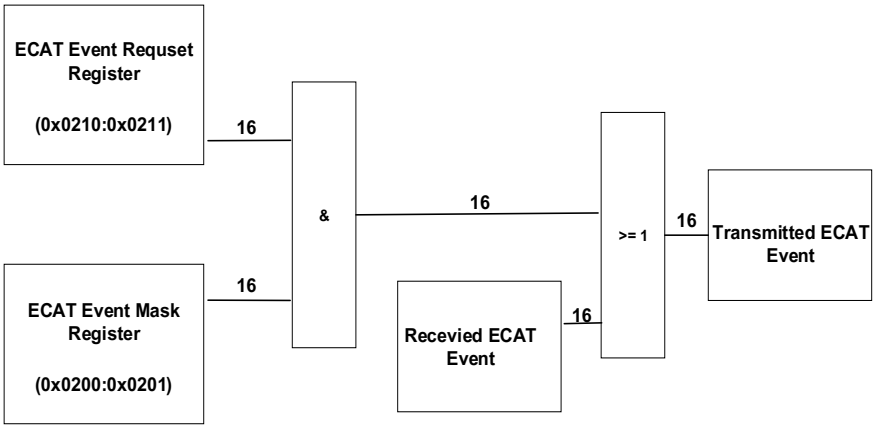
- The DC SYNC signals are mapped into the ESC AL Event Request register (configured with SYNC/LATCH PDI Configuration register 0x0151.3/7). In this case, all interrupts from the ESC to the Cortex®-M7 MCU are combined into one IRQ signal, and the Distributed Clocks LATCH0 / 1 inputs can still be used.
- The DC SyncSignals are directly connected to Cortex®-M7 MCU interrupt inputs. The Cortex®-M7 MCU can react on DC SyncSignal interrupts faster (without reading AL Request register), but it needs more interrupt inputs. The jitter of the SyncSignals is ~12 ns. The DC Latch functions are only available for one Latch input or not at all (if both DC SYNC outputs are used).

ECAT Event Request (ECAT Interrupt)

ECAT event requests are used to inform the EtherCAT main of SubDevice events. ECAT events make use of the IRQ field inside EtherCAT datagrams. The ESC Event Request register (ESC_EVT_RQST) is combined with the ESC Event Mask register (ESC_EVT_MSK) using a logical AND operation. The resulting interrupt bits are combined with the incoming ECAT IRQ field using a logical OR operation, and written into the outgoing ECAT IRQ field. The ESC Event Mask register allows for selecting the interrupts which are relevant for the EtherCAT main and handled by the main application.

NOTE: The main cannot distinguish which SubDevice (or even more than one) was the origin of an interrupt.

Figure 55-4. EtherCAT Interrupt Masking



Clearing Interrupts Accidentally

Event request registers and register actions which clear interrupts are intended to be accessed independently, i.e., with separate EtherCAT frames or separate PDI accesses. Otherwise it may happen that interrupts and/or data are missed.

55.3.7. LED

EtherCAT SubDevice controllers support LED (RUNLED) regarding link state and AL status. The LED output of an ESC is controlled by the ESC AL Status register (ESC_AL_STAT) which are automatically translated into blink codes.

The EtherCAT Core configuration provides for direct control of the run led via the ESC RUN LED Override register.

55.3.8. Station management interface(SMI)

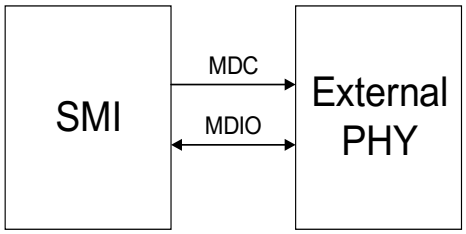
SMI is designed to access and configure PHY's configuration.
Station management interface (SMI) is performed through two wires to communicate with the external PHY: one clock line (MDC) and one data line (MDIO), it can access to the any PHY

register.

MDC and MDIO specific functions as follows:

- MDC: A clock of maximum frequency is 2.5MHz. The pin remains low level when it is in idle state. The minimum high or low level lasts time of MDC must be 160ns, and the minimum period of MDC must be 400ns when it is in data transmission state.
- MDIO: Used to transfer data in conjunction with the MDC clock line, receiving data from external PHY or sending data to external PHY.

Figure 55-5. Station management interface signals



55.3.9. AHB direct / indirect access

The system can directly / indirectly access ESC registers and core PRAM through the AHB to OPB bridge in three ways: directly accessible ESC registers, indirectly accessible ESC core registers, and indirectly accessible ESC core PRAM. The bridge provides an AHB Slave interface towards the upstream side and OPB interfaces towards the downstream side of the ESC core, where the upstream AHB side is faster than, in frequency to, the downstream ESC core, and the clocks are synchronous in phase, and have an N:1(max N=16) frequency ratio.

Direct AHB transmit access ESC register

Direct access to ESC register is used for transferring data / commands to the indirect access of ESC core registers. When the AHB bus accesses the address range of 0x0000-0x2FFF, a single register read / write operation on the ESC core is initiated based on the CCTL_RW bit in the ESC_CCTL_CMD register. At the start of a read cycle, the CCTL_BUSY bit in the ESC_CCTL_CMD register is set to 1, and at the end of the read cycle, CCTL_BUSY is cleared to 0, allowing valid data to be read on the AHB bus. At the start of a write cycle, the CCTL_BUSY bit is set to 1, valid data is written to the bus, and at the end of the write cycle, the CCTL_BUSY bit is cleared, and the valid data is written into the register.

Indirect transmit access ESC core register

ESC registers with address ranges from 0x0000 to 0x0FFF, in addition to supporting direct AHB transmit access, also support indirect transmit access methods. The system can access the ESC core registers indirectly through the ESC_CCTL_DATA and ESC_CCTL_CMD registers. When reading an ESC core register, the following steps are required: First, set the CCTL_STOP bit in the ESC_CCTL_CMD register to 1 to clear the CCTL_BUSY bit; set the CCTL_RW bit in the ESC_CCTL_CMD register to 1, write the address of the register to be accessed into the CCTL_ADDR field, and write the number of bytes to be read into the CCTL_SIZE field, then set the CCTL_BUSY bit to 1. When the CCTL_BUSY

bit is cleared, data can be read from the ESC_CCTL_DATA register.

When writing data to the ESC core registers, the following steps are required: First, set the CCTL_STOP bit in the ESC_CCTL_CMD register to 1 to clear the CCTL_BUSY bit; clear the CCTL_RW bit in the ESC_CCTL_CMD register, write the address of the register to be accessed into the CCTL_ADDR field, and write the number of bytes to be written into the CCTL_SIZE field, then set the CCTL_BUSY bit to 1. The configuration data can be written into the ESC_CCTL_CMD register at once. The completion of the write cycle is indicated by the CCTL_BUSY bit being cleared to zero.

In the above read / write operations, the valid data is always aligned with the low bits of the ESC_CCTL_DATA. The valid data can be referred to the following table.

Table 55-1. Alignment of valid data

CCTL_SIZE	ESC CCTL_ADDR[1:0]	ESC CCTL_DATA valid bytes
1	00/01/10/11	[7:0]/ [15:8]/ [23:16]/ [31:24]/
2	00/10	[15:0]/[31:16]
4	00	[31:0]

Indirect transmit access ESC core PRAM

When initiating read operation to core PRAM through AHB, after writing PRAM start address and read length to ESC_PRAM_ALR register, PRAM_BUSY_READ is set to 1, the module starts to initiate multiple OPB read operations, read data from core PRAM and write to the TX FIFO. All OPB read operations are complete. PRAM_BUSY_READ is cleared. When data is transferred from the ESC core to the TX FIFO, the PRAM read length PRAM_LEN_READ and the PRAM read address PRAM_ADDR_READ are updated to show the process. Determines the valid bytes of the first read data according to the start address. Determines the valid bytes of the last read data based on the starting address and operation length. If necessary, the read command can be stopped by setting the PRAM_STOP_READ bit in the ESC_PRAM_CR register to 1. If the OPB read period starts, the stop command takes effect after the current read operation is complete. After the stop command takes effect, data in the TX FIFO is cleared.

When writing to the PRAM through AHB, after writing the RAM start address and write length to the ESC_PRAM_ALW register, write 1 to the PRAM_BUSY_WRITE bit of the ESC_PRAM_CR register, the module initiates multiple OPB write operations. Data is read from the RX FIFO and written to the core PRAM. All OPB write operations are complete. PRAM_BUSY_WRITE is cleared Write operations support wait mechanism, PRAM_BUSY_WRITE bit is set to 1, but when RX FIFO is empty, OPB module will not immediately initiate OPB transfer operation, until there is data in RX FIFO will initiate this operation. After each OPB ACK response, it will detect the RX FIFO state. If the RX FIFO is empty, it will enter the waiting state and enter the next transmission operation until there is data. When data is transferred from the RX FIFO to the core, the PRAM write length PRAM_LEN_WRITE and the PRAM write address PRAM_ADDR_WRITE are updated to show the process. Determines the valid bytes of the first data write according to the start address. Based on the starting address and operation length, determines the valid

bytes of the last data write.

If necessary, the write command can be stopped by setting the PRAM_STOP_WRITE bit to 1. If the OPB write cycle starts, the stop command takes effect after the read operation is complete. After the stop command takes effect, data in the RX FIFO is cleared.

55.3.10. Register protection in BUSY state

If the BUSY related register bits, such as CCTL_BUSY, PRAM_BUSY_READ, and PRAM_BUSY_WRITE, are set to 1, you can set the BRP bit to protect the register from rewriting. When CCTL_BUSY is set to 1, ESC_CCTL_DATA / ESC_CCTL_CMD is protected from rewriting by AHB write operations. When PRAM_BUSY_READ is set to 1, ESC_PRAM_ALR is protected from rewriting by AHB write operations. When PRAM_BUSY_WRITE is set to 1, ESC_PRAM_ALW is protected from being overwritten by AHB write operations

When the BRP bit is set to 1, when the BUSY related register bit is set to 1, the user's AHB write operation to the corresponding register will be lost, then the WDLF flag in the ESC_CTL register is set to 1, and the interrupt will be triggered when the WDIE bit is set to 1.

When the BRP bit is set to 0, when the BUSY related register bit is set to 1, the user's AHB write operation to the corresponding register will cause the current OPB transmission error, then the WEF flag in the ESC_CTL register is set to 1, and the interrupt will be triggered when the WEIE bit is set to 1. It is advised to handle it as soon as possible to avoid more errors.

55.3.11. OPB transmission timeout function

The OPB transmission timeout function can be enabled by setting the TOEN bit in the ESC_CTL register to 1. The timeout interval can be configured by the TO_CNT bit in the ESC_CTL register. When the counter exceeds the TO_CNT programming value, the TOF flag in the ESC_CTL register is set to 1. If the TOIE bit in the ESC_CTL register is set to 1, a timeout interrupt is generated. Timeout interrupt response processing:

1. If the ESC CCTL direct read/write mode is used, the TOF flag is set after a timeout interrupt response occurs. Terminate the transmission directly to avoid the BUS being occupied.
2. If the ESC CCTL is used in indirect read/write mode, the TOF flag is set after a timeout interrupt response occurs. You can stop this operation by writing 1 to the ESC CCTL_STOP bit.
3. When the PRAM is used in indirect read/write mode, the TOF flag is set after a timeout interrupt response occurs. You can stop this operation by writing 1 to the PRAM_STOP_WRITE/ PRAM_STOP_READ bit.

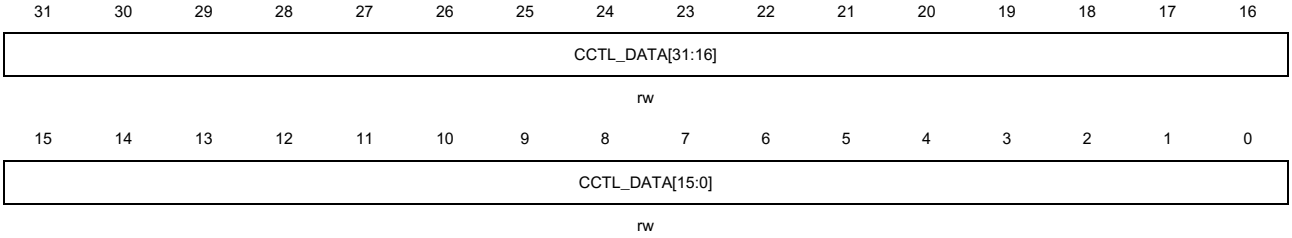
55.4. ESC Core Control Register definition

ESC core control register base address: 0x5F00 0000

55.4.1. ESC CCTL data register (ESC_CCTL_DATA)

Address Offset: 0x3300
Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)

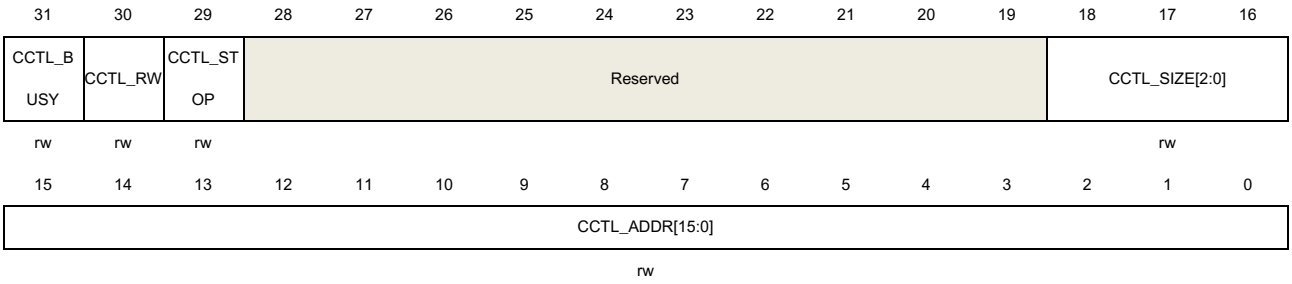


Bits	Fields	Descriptions
31:0	CCTL_DATA[31:0]	ESC CCTL data This field indicate the value read from or written to the ESC Core. Reading or writing depends on the ESC_CCTL_CMD[CCTL_RW] bit. If the CCTL_RW bit is 1, this value is the data read from the ESC Core; If the CCTL_RW bit is 0, this value is the data written to the ESC Core.

55.4.2. ESC CCTL command register (ESC_CCTL_CMD)

Address Offset: 0x3304
Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)



Bits	Fields	Descriptions
31	CCTL_BUSY	CCTL Busy. There is no effect if the bit is written to 0 0: No read / write (according to CCTL_RW) operation is being performed. 1: Read / write operation is being performed. **Note:** When the read / write operation is complete, this bit will be cleared by

		hardware. Then the valid data is available for the HOST to read from ESC_CCTL_DATA or write to ESC_CCTL_DATA register. The ESC_CCTL_CMD and ESC_CCTL_DATA registers should not be modified when this bit is 1.
30	CCTL_RW	Read or write selection 0: Write operation 1: Read operation
29	CCTL_STOP	Stop read or write operation. There is no effect if the bit is written to 0. 0: No effect 1: Stop the ESC core register reading or writing operation **Note:** When the CCTL_BUSY is cleared, the CCTL_STOP will be cleared
28:19	Reserved	Must be kept at reset value.
18:16	CCTL_SIZE[2:0]	This field specifies the size of the register that is about to be read or write. (in byte). Only 1, 2 and 4 are valid, other values are invalid.
15:0	CCTL_ADDR[15:0]	This field specifies the address of ESC registers that will be accessed.

55.4.3. ESC PRAM FIFO data read register (ESC_PRAM_FIFO_DR)

Address Offset: 0x3310
Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRAM_FIFO_DATA_READ[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRAM_FIFO_DATA_READ[15:0]															
r															

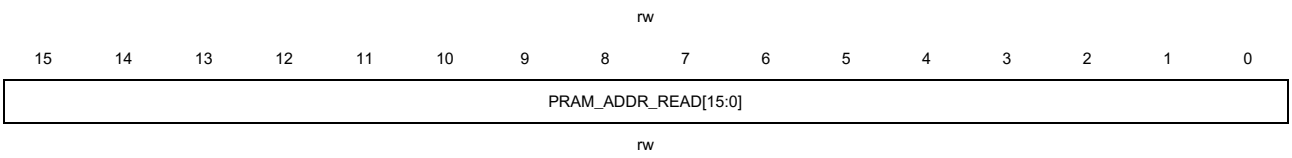
Bits	Fields	Descriptions
31:0	PRAM_FIFO_DATA_READ[31:0]	Data read from ESC PRAM. The valid value of data is determined according to the start address and the transfer length.

55.4.4. ESC PRAM address and length read register (ESC_PRAM_ALR)

Address Offset: 0x3314
Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRAM_LEN_READ[15: 0]															



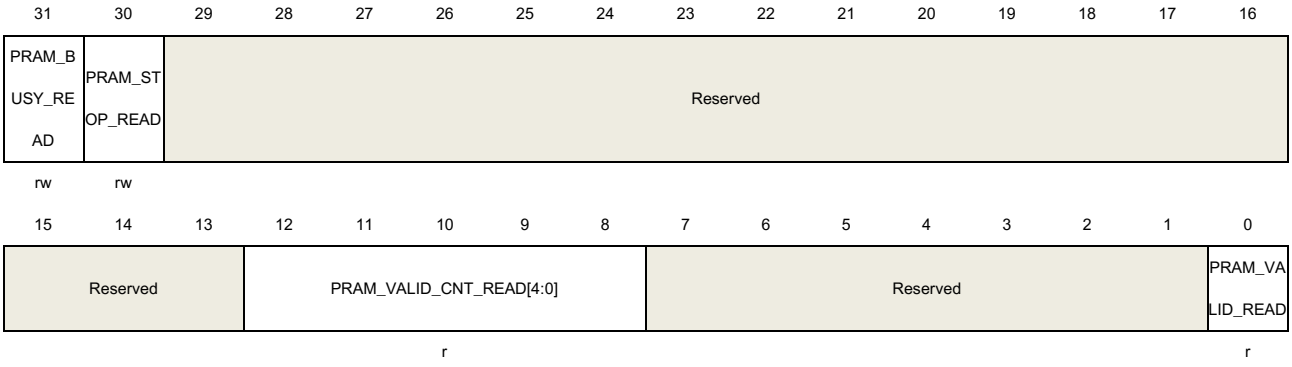
Bits	Fields	Descriptions
31:16	PRAM_LEN_READ[15: 0]	Data length read from ESC PRAM in bytes. This field is decremented as data is read into the FIFO. **Note:** When PRAM_BUSY_READ is 1, this field cannot be modified.
15:0	PRAM_ADDR_READ [15:0]	ESC PRAM data read address. This field is incremented as data is read into the FIFO. **Note:** When PRAM_BUSY_READ is 1, this field cannot be modified.

55.4.5. ESC PRAM command read register (ESC_PRAM_CR)

Address Offset: 0x3318

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)



Bits	Fields	Descriptions
31	PRAM_BUSY_READ	Whether the PRAM is being read. There is no effect if the bit is written to 0 0: No PRAM read operation 1: PRAM is being reading **Note:** This bit is cleared when the read operation is complete.
30	PRAM_STOP_READ	Stop PRAM read operation. There is no effect if the bit is written to zero 0: No effect 1: Stop the PRAM reading operation Note: After this bit is set to 1, the PRAM_BUSY_READ is cleared, and the RX FIFO is reset. Then this bit will self-clear.
29:13	Reserved	Must be kept at reset value.
12:8	PRAM_DATA_CNT_	PRAM data read valid count

	READ[4:0]	This count increases as data is read from the PRAM to the RX FIFO and decreases as data of the entire DWORD size is read from the RX FIFO
7:1	Reserved	Must be kept at reset value.
0	PRAM_VALID_DATA_READ	PRAM valid data read 0: There is no valid data to be read. 1: There is valid data to be read.

55.4.6. ESC PRAM FIFO data write register (ESC_PRAM_FIFO_DW)

Address Offset: 0x3320
Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRAM_FIFO_DATA_WRITE[31:16]															
w															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRAM_FIFO_DATA_WRITE[15:0]															
w															

Bits	Fields	Descriptions
31:0	PRAM_FIFO_DATA_WRITE[31:0]	Data write to ESC PRAM. The valid value of data is determined according to the start address and the transfer length.

55.4.7. ESC PRAM address and length write register (ESC_PRAM_ALW)

Address Offset: 0x3324
Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRAM_LEN_WRITE[15: 0]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRAM_ADDR_WRITE[15:0]															
rw															

Bits	Fields	Descriptions
31:16	PRAM_LEN_WRITE[15: 0]	Data length written to ESC PRAM in bytes. This field is decremented as data is written into the FIFO. **Note:** When PRAM_BUSY_WRITE is 1, this field cannot be modified.

15:0	PRAM_ADDR_WRITE[15:0]	ESC PRAM data write address. This field is incremented as data is read from the FIFO. **Note:** When PRAM_BUSY_WRITE is 1, this field cannot be modified.

55.4.8. ESC PRAM command write register (ESC_PRAM_CW)

Address Offset: 0x3328

Reset value: 0x0000 1001

This register has to be accessed by word(32-bit)



Bits	Fields	Descriptions
31	PRAM_BUSY_WRITE	Whether the PRAM is being written. There is no effect if the bit is written to zero 0: No PRAM write operation 1: PRAM is being writing **Note:** This bit is cleared when the write operation is complete.
30	PRAM_STOP_WRITE	Stop PRAM write operation. There is no effect if the bit is written to zero 0: No effect 1: Stop the PRAM writing operation Note: After this bit is set to 1, the PRAM_BUSY_WRITE is cleared, and the TX FIFO is reset. Then this bit will self-clear.
29:13	Reserved	Must be kept at reset value.
12:8	PRAM_VALID_CNT_WRITE[4:0]	PRAM data write valid count This count increases as data is read from the TX FIFO to PRAM and decreases as data of the entire DWORD size is written to the PRAM
7:1	Reserved	Must be kept at reset value.
0	PRAM_VALID_DATA_WRITE	PRAM valid data write 0: There is no valid data to be written 1: There is valid data to be written.

55.4.9. ESC Control register (ESC_CTL)

Address Offset: 0x3330

Reset value: 0x0000 04A0

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EERF	EERIE	WDLF	WDLIE	TOF	TOIE	WEF	WEIE	ESC_CCT LIVF	ESC_CCT LIVIE	I2CF	I2CIE	Reserved			
rc_w1	rw	rc_w1	rw	rc_w1	rw	rc_w1	rw	rc_w1	rw	rc_w1	rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					BRP	TO_CNT[8:0]									TOEN
					rw	rw									rw

Bits	Fields	Descriptions
31	EERF	ESC EEPROM reload complete flag. This flag can be cleared by writing 1. 0: ESC EEPROM reload not completed. 1: ESC EEPROM reload completed. **NOTE**: After the ESC is disabled and reset, the EERF bit is immediately set to 1. To enable the EEPROM update function, when the ESC is operating normally (i.e., PDI interface is accessible), first write 1 to this bit to clear the interrupt flag, and then write EERIE bit to set.
30	EERIE	ESC EEPROM reload interrupt enable 0: Interrupt is inhibited 1: An interrupt will occur whenever the EERF bit is set
29	WDLF	Write data lost flag. When BRP enable, AHB write operation will cause data lose when the busy bit is high, this flag will be triggered. This flag can be cleared by writing 1. 0: No data lost 1: The data in this transmission is lost
28	WDLIE	Write data lost interrupt enable 0: Interrupt is inhibited 1: An interrupt will occur whenever the WDLF bit is set
27	TOF	Time out flag, when single transmission time exceeds the programmed value, this flag will be triggered. This flag can be cleared by writing 1. 0: No ESC core timeout transmission occurred 1: An ESC core timeout transmission occurred
26	TOIE	Timer out interrupt enable 0: Interrupt is inhibited 1: An interrupt will occur whenever the TOF bit is set

25	WEF	write error flag. When BRP disable, AHB write operation will cause current OPB transmission error when the busy bit is high, this flag will be triggered. This flag can be cleared by writing 1.
24	WEIE	Write error interrupt enable. 0: Interrupt is inhibited 1: An interrupt will occur whenever the WEF bit is set
23	ESC_CCTLIVF	CCTL_SIZE & CCTL_ADDR illegal value flag. The CCTL_SIZE and CCTL_ADDR value written to the register does not meet the requirement, causing this transfer to not occur, this flag will be triggered. This flag can be cleared by writing 1.
22	ESC_CCTLIVIE	CCTL_SIZE & CCTL_ADDR Illegal value interrupt enable. 0: Interrupt is inhibited 1: An interrupt will occur whenever the ESC_CCTLIVF bit is set
21	I2CF	Internal I2C communication error flag. This bit is set when an overflow or underflow occurs. When the corresponding interrupt enable bit is set, an interrupt is issued. This flag can be cleared by writing 1.
20	I2CIE	Internal I2C overrun interrupt enable. 0：Interrupt is inhibited 1：An interrupt will occur whenever the I2CIE bit is set
19:11	Reserved	Must be kept at reset value.
10	BRP	When busy bit is high, prevent register change 1: the protection takes effect 0: the protection does not take effect
9:1	TO_CNT[8:0]	Time Out counter This field indicates the transmission timeout in bits during which no ACK response was received; The programmer data cannot be less than the minimum value recommended the manual(60).
0	TOEN	Time Out enable This field indicates that whether the function is effect 1: Timeout feature enabled 0: Timeout feature disabled.

55.4.10. ESC Sync0 register (ESC_SYNC0)

Address Offset: 0x3334
Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															SYNC0_INT
															c_w

Bits	Fields	Descriptions
31:1	Reserved	Must be kept at reset value.
0	SYNC0_INT	the Sync0 interrupt signal, This bit is not readable; writing any value by software clears it to zero.

55.4.11. ESC Sync1 register (ESC_SYNC1)

Address Offset: 0x3338
Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															SYNC1_INT
															c_w

Bits	Fields	Descriptions
31:1	Reserved	Must be kept at reset value.
0	SYNC1_INT	the Sync1 interrupt signal, This bit is not readable; writing any value by software clears it to zero.

55.5. ESC Register definition

ESC core control register base address: 0x0000 5F00

55.5.1. ESC Type register (ESC_TYPE)

Address Offset: 0x0000
Reset value: 0xBC

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
TYPE[7:0]							
r(ECAT)							

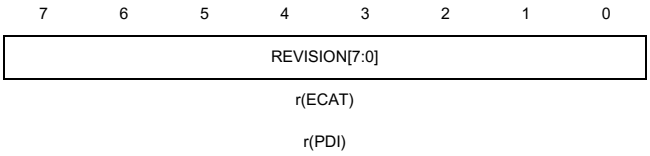
r(PDI)

Bits	Fields	Descriptions
7:0	TYPE[7:0]	Type of EtherCAT controller 0xBC: GDESC

55.5.2. ESC Revision register (ESC_REVISION)

Address Offset: 0x0001
Reset value: 0x00

This register can be accessed by byte(8-bit).

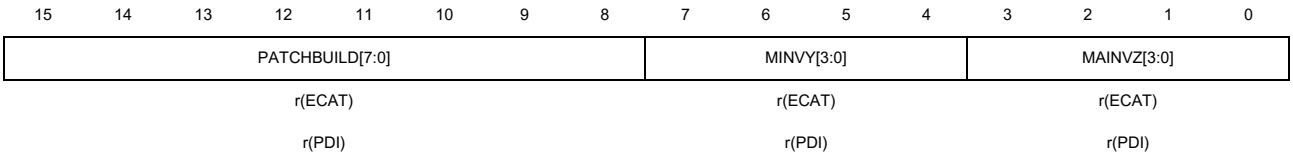


Bits	Fields	Descriptions
7:0	REVISION[7:0]	Revision of EtherCAT controller.

55.5.3. ESC Build register (ESC_BUILD)

Address Offset: 0x0002
Reset value: 0x0000

This register can be accessed by half-word(16-bit).



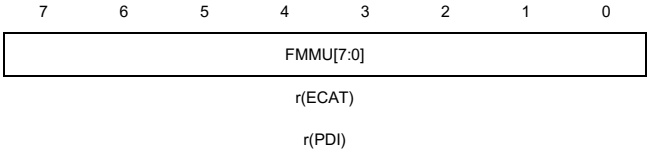
Bits	Fields	Descriptions
15:8	PATCHBUILD[7:0]	patch level / development build: 0x00: original release 0x01-0x0F: patch level of original release 0x10-0xFF: development build
7:4	MINVY[3:0]	minor version Y
3:0	MAINVZ[3:0]	maintenance version Z

55.5.4. ESC FMMU numbers register (ESC_FMMUS)

Address Offset: 0x0004

Reset value: 0x08

This register can be accessed by byte(8-bit).

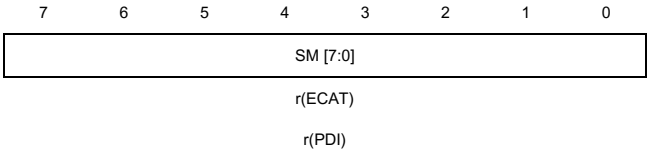


Bits	Fields	Descriptions
7:0	FMMU[7:0]	This field indicates the number of supported FMMU channels (or entities) of the EtherCAT SubDevice controller. This device provides 8.

55.5.5. ESC SyncManagers numbers register (ESC_SYNCMS)

Address Offset: 0x0005
Reset value: 0x08

This register can be accessed by byte(8-bit).

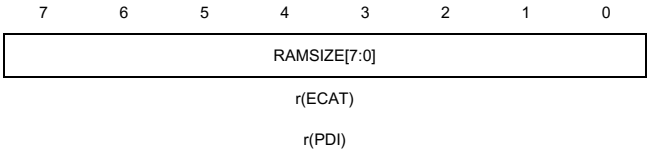


Bits	Fields	Descriptions
7:0	SM[7:0]	Number of supported SyncManager entities, The device provides 8.

55.5.6. ESC RAM size register (ESC_RAMSIZE)

Address Offset: 0x0006
Reset value: 0x08

This register can be accessed by byte(8-bit).



Bits	Fields	Descriptions
7:0	RAMSIZE[7:0]	Process Data RAM size supported in 8Kbyte

55.5.7. ESC Port descriptor register (ESC_PORT_DESC)

Address Offset: 0x0007

Reset value: 0x3F

This register can be accessed by byte(8-bit).

7 6	5 4	3 2	1 0
Port3[1:0]	Port2[1:0]	Port1[1:0]	Port0[1:0]
r(ECAT)	r(ECAT)	r(ECAT)	r(ECAT)
r(PDI)	r(PDI)	r(PDI)	r(PDI)

Bits	Fields	Descriptions
7:6	Port3[1:0]	Port 3 Configuration This field details the Port 3 configuration. 00: Not implemented 01: Not configured 10: Reserved 11: MII / RMII
5:4	Port2[1:0]	Port 2 Configuration This field details the Port 2configuration. 00: Not implemented 01: Not configured 10: Reserved 11: MII / RMII
3:2	Port1[1:0]	Port 1 Configuration This field details the Port 1 configuration. 00: Not implemented 01: Not configured 10: Reserved 11: MII / RMII
1:0	Port0[1:0]	Port 0 Configuration This field details the Port 0 configuration. 00: Not implemented 01: Not configured 10: Reserved 11: MII / RMII

55.5.8. ESC Features supporter register (ESC_FEATURE)

Address Offset: 0x0008
Reset value: 0x01CC

This register can be accessed by half-word(16-bit).

15 14 13 12	11	10	9	8	7	6	5 4	3	2	1	0
Reserved	FS11	FS10	FS9	FS8	FS7	FS6	Reserved	FS3	FS2	FS1	FS0
	r(ECAT)	r(ECAT)	r(ECAT)	r(ECAT)	r(ECAT)	r(ECAT)		r(ECAT)	r(ECAT)	r(ECAT)	r(ECAT)

r(PDI) r(PDI) r(PDI) r(PDI) r(PDI) r(PDI) r(PDI) r(PDI) r(PDI) r(PDI)

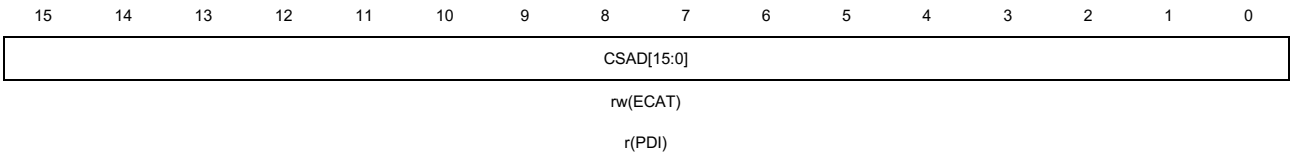
Bits	Fields	Descriptions
15:12	Reserved	Must be kept at reset value.
11	FS11	Fixed FMMU / SyncManager configuration: 0: Variable configuration 1: Fixed configuration
10	FS10	EtherCAT read / write command support (BRW, APRW, FPRW): 0: Supported 1: Not supported
9	FS9	EtherCAT LRW command support: 0: Supported 1: Not supported
8	FS8	Enhanced DC SYNC Activation: 0: Not available 1: Available NOTE: This feature refers to registers ***ESC Register activation register (ESC_REG_ACT)***[7:3]and ***ESC Activation status register (ESC_ACT_STAT)***
7	FS7	Separate Handling of FCS Errors: 0: Not supported 1: Supported, frames with wrong FCS and additional nibble will be counted separately in Forwarded RX Error Counter
6	FS6	Enhanced Link Detection MII: 0: Not available 1: Available
5:4	Reserved	Must be kept at reset value.
3	FS3	Distributed Clocks (width): 0: 32 bit 1: 64 bit
2	FS2	Distributed Clocks: 0: Not available 1: Available
1	FS1	Unused register access: 0: allowed 1: not supported
0	FS0	FMMU Operation 0: Bit oriented

1: Byte oriented

55.5.9. ESC Configured station address register (ESC_STATION_ADDR)

Address Offset: 0x0010
Reset value: 0x0000

This register can be accessed by half-word(16-bit).

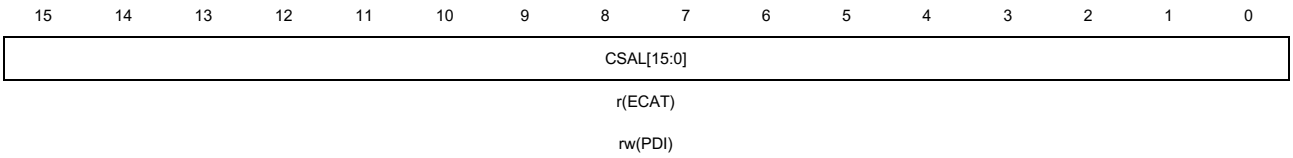


Bits	Fields	Descriptions
15:0	CSAD[15:0]	Address used for node addressing (FPRD / FPWR / FPRW / FRMW commands).

55.5.10. ESC Configured station alias register (ESC_STATION_ALS)

Address Offset: 0x0012
Reset value: 0x0000

This register can be accessed by half-word(16-bit).

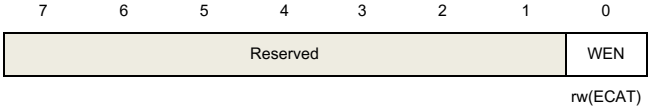


Bits	Fields	Descriptions
15:0	CSAL[15:0]	Alias Address used for node addressing (FPRD/FPWR/FPRW/FRMW commands). The use of this alias is activated by ***ESC AL control register (ESC_AL_CTL)*** Bit [24]. **NOTE**: EEPROM value is only transferred into this register at first EEPROM load after power-on or reset.

55.5.11. ESC Write enable register (ESC_REG_WEN)

Address Offset: 0x0020
Reset value: 0x00

This register can be accessed by byte(8-bit).



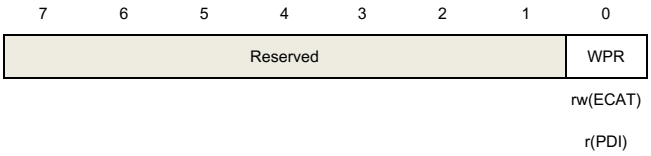
r(PDI)

Bits	Fields	Descriptions
7:1	Reserved	Must be kept at reset value.
0	WEN	If register write protection is enabled, this register has to be written in the same Ethernet frame (value does not matter) before other writes to this station are allowed. This bit is self-clearing at the beginning of the next frame (SOF), or if Register Write Protection is disabled.

55.5.12. ESC Write protection register (ESC_REG_WP)

Address Offset: 0x0021
Reset value: 0x00

This register can be accessed by byte(8-bit).

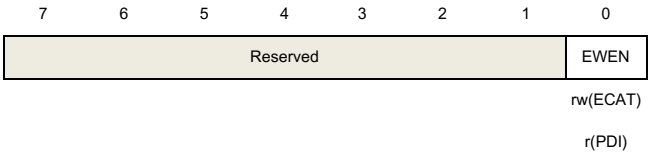


Bits	Fields	Descriptions
7:1	Reserved	Must be kept at reset value.
0	WPR	Register write protection: 0: Protection disabled 1: Protection enabled Registers 0x0000 ~ 0x0F7F are write protected, except for 0x0020 and 0x0030.

55.5.13. ESC Write enable register (ESC_WEN)

Address Offset: 0x0030
Reset value: 0x00

This register can be accessed by byte(8-bit).



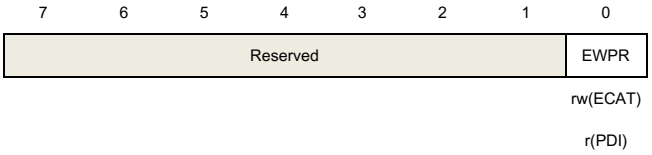
Bits	Fields	Descriptions
7:1	Reserved	Must be kept at reset value.
0	EWEN	If ESC write protection is enabled, this register has to be written in the same

Ethernet frame (value does not matter) before other writes to this station are allowed. This bit is self-clearing at the beginning of the next frame (SOF), or if ESC Write Protection is disabled.

55.5.14. ESC Write protection register (ESC_WP)

Address Offset: 0x0031
Reset value: 0x00

This register can be accessed by byte(8-bit).



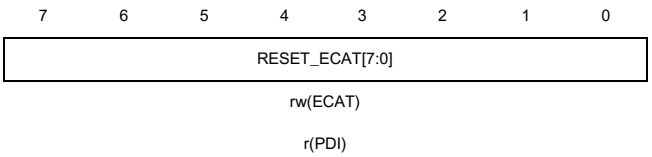
Bits	Fields	Descriptions
7:1	Reserved	Must be kept at reset value.
0	EWPR	Write protect: 0: Protection disabled 1: Protection enabled All areas are write-protected, except for 0x0030.

55.5.15. ESC Reset register (ESC_RST_ECAT)

Address Offset: 0x0040
Reset value: 0x00

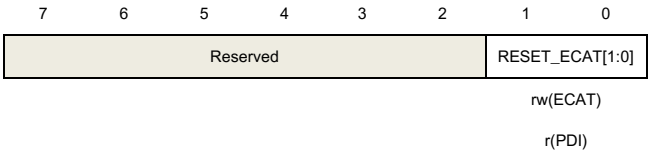
This register can be accessed by byte(8-bit).

Write:



Bits	Fields	Descriptions
7:0	RESET_ECAT[7:0]	A reset is asserted after writing the reset sequence 0x52 ('R'), 0x45 ('E') and 0x53 ('S') in this register with 3 consecutive frames. Any other frame which does not continue the sequence by writing the next expected value will cancel the reset procedure. **NOTE:** The R/E/S reset commands must be issued strictly in sequence, and no other data may be inserted between these three commands.

Read:



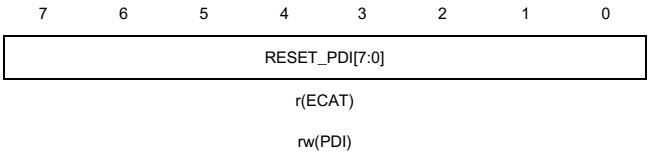
Bits	Fields	Descriptions
7:2	Reserved	Must be kept at reset value.
1:0	RESET_ECAT[1:0]	Progress of the reset procedure: 00: initial / reset state 01: after writing 0x52 ('R'), when previous state was 00 10: after writing 0x45 ('E'), when previous state was 01 11: after writing 0x53 ('S'), when previous state was 10. This value must not be observed because the ESC enters reset when this state is reached, resulting in state 00.

55.5.16. ESC Reset PDI register (ESC_RST_PDI)

Address Offset: 0x0041
Reset value: 0x00

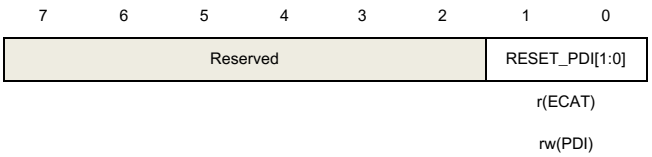
This register can be accessed by byte(8-bit).

Write:



Bits	Fields	Descriptions
7:0	RESET_PDI[7:0]	A reset is asserted after writing the reset sequence 0x52 ('R'), 0x45 ('E') and 0x53 ('S') in this register with 3 consecutive commands. Any other command which does not continue the sequence by writing the next expected value will cancel the reset procedure.

Read:



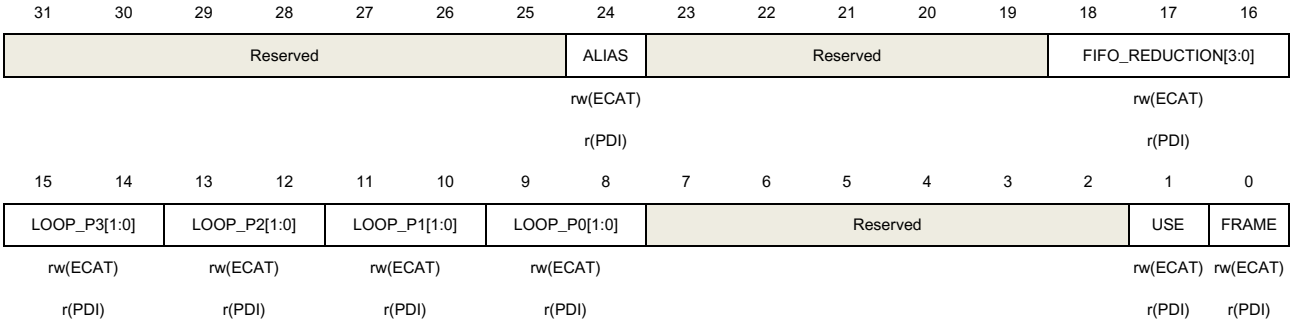
Bits	Fields	Descriptions
7:2	Reserved	Must be kept at reset value.
1:0	RESET_PDI[1:0]	Progress of the reset procedure:

00: initial / reset state
01: after writing 0x52 ('R'), when previous state was 00
10: after writing 0x45 ('E'), when previous state was 01
11: after writing 0x53 ('S'), when previous state was 10.
This value must not be observed because the ESC enters reset when this state is reached, resulting in state 00.

55.5.17. ESC DL control register (ESC_DL_CTL)

Address Offset: 0x0100
Reset value: 0x0007 C001

This register can be accessed by word(32-bit).



Bits	Fields	Descriptions
31:25	Reserved	Must be kept at reset value.
24	ALIAS	Station alias: 0: Ignore station alias 1: Alias can be used for all configured address command types (FPRD, FPWR, …)
23:19	Reserved	Must be kept at reset value.
18:16	FIFO_REDUCTION[3:0]	RX FIFO Size (ESC delays start of forwarding until FIFO is at least half full). RX FIFO Size / RX delay reduction : Value: MII: 0: -40 ns (-80 ns) 1: -40 ns (-80 ns) 2: -40 ns 3: -40 ns 4: no change 5: no change 6: no change 7: default **NOTE**: EEPROM value is only taken over at first EEPROM load after power-on or reset

15:14	LOOP_P3[1:0]	Loop Port 3: 00: Auto 01: Auto Close 10: Open 11: Closed
13:12	LOOP_P2[1:0]	Loop Port 2: 00: Auto 01: Auto Close 10: Open 11: Closed
11:10	LOOP_P1[1:0]	Loop Port 1: 00: Auto 01: Auto Close 10: Open 11: Closed
9:8	LOOP_P0[1:0]	Loop Port 0: 00: Auto 01: Auto Close 10: Open 11: Closed NOTE: Loop open means sending/receiving over this port is enabled, loop closed means sending/receiving is disabled and frames are forwarded to the next open port internally. Auto: loop closed at link down, opened at link up Auto Close: loop closed at link down, opened with writing 01 again after link up (or receiving a valid Ethernet frame at the closed port) Open: loop open regardless of link state Closed: loop closed regardless of link state
7:2	Reserved	Must be kept at reset value.
1	USE	Temporary use of settings in 0x0100:0x0103[8:15]: 0: Permanent use 1: Use for about 1 second, then revert to previous settings
0	FRAME	Forwarding rule: 0: Forward non-EtherCAT frames: EtherCAT frames are processed, non-EtherCAT frames are forwarded without processing or modification. The source MAC address is not changed for any frame. 1: Destroy non-EtherCAT frames: EtherCAT frames are processed, non-EtherCAT frames are destroyed. The source MAC address is changed by the Processing Unit for every frame (SOURCE_MAC[1] is set to 1 - locally administered address).

55.5.18. ESC Physical read / write offset register (ESC_PHYSICAL_RW_OFFSET)

Address Offset: 0x0108
Reset value: 0x0000

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RW_OFFSET[15:0]															
rw(ECAT)															
r(PDI)															

Bits	Fields	Descriptions
15:0	RW_OFFSET[15:0]	This register is used for ReadWrite commands in Device Addressing mode (FPRW, APRW, BRW). The internal read address is directly taken from the offset address field of the EtherCAT datagram header, while the internal write address is calculated by adding the Physical Read / Write Offset value to the offset address field. Internal read address = ADR, internal write address = ADR + R/W - Offset

55.5.19. ESC DL status register (ESC_DL_STAT)

Address Offset: 0x0110
Reset value: 0x0000

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DL_STAT US15	DL_STAT US14	DL_STAT US13	DL_STAT US12	DL_STAT US1`	DL_STAT US10	DL_STAT US9	DL_STAT US8	DL_STAT US7	DL_STAT US6	DL_STAT US5	DL_STAT US4	Reserved	DL_STAT US2	DL_STAT US1	DL_STAT US0
r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)		r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)

Bits	Fields	Descriptions
15	DL_STATUS15	Communication on Port 3: 0: No stable communication 1: Communication established
14	DL_STATUS14	Loop Port 3: 0: Open 1: Closed
13	DL_STATUS13	Communication on Port 2: 0: No stable communication 1: Communication established
12	DL_STATUS12	Loop Port 2: 0: Open

		1: Closed
11	DL_STATUS11	Communication on Port 1: 0: No stable communication 1: Communication established
10	DL_STATUS10	Loop Port 1: 0: Open 1: Closed
9	DL_STATUS9	Communication on Port 0: 0: No stable communication 1: Communication established
8	DL_STATUS8	Loop Port 0: 0: Open 1: Closed
7	DL_STATUS7	Physical link on Port 3 0: No link 1: Link detected
6	DL_STATUS6	Physical link on Port 2 0: No link 1: Link detected
5	DL_STATUS5	Physical link on Port 1 0: No link 1: Link detected
4	DL_STATUS4	Physical link on Port 0: 0: No link 1: Link detected
3	Reserved	Must be kept at reset value.
2	DL_STATUS2	Enhanced Link detection: 0: Deactivated for all ports 1: Activated for at least one port NOTE: EEPROM value is only transferred into this register at first EEPROM load after power-on or reset
1	DL_STATUS1	PDI Watchdog Status: 0: Watchdog expired 1: Watchdog reloaded
0	DL_STATUS0	PDI operational / EEPROM loaded correctly: 0: EEPROM not loaded, PDI not operational (no access to Process Data RAM) 1: EEPROM loaded correctly, PDI operational (access to Process Data RAM)

55.5.20. ESC AL control register (ESC_AL_CTL)

Address Offset: 0x0120

Reset value: 0x0001

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											ACK	STATE[3:0]			
											rw(ECAT)	rw(ECAT)			
											rw(PDI)	rw(PDI)			

Bits	Fields	Descriptions
15:5	Reserved	Must be kept at reset value.
4	ACK	Error Ind Ack: 0: No Ack of Error Ind in AL status register 1: Ack of Error Ind in AL status register
3:0	STATE[3:0]	Initiate State Transition of the Device State Machine: 1: Request Init State 3: Request Bootstrap State 2: Request Pre-Operational State 4: Request Safe-Operational State 8: Request Operational State Note: PDI register function acknowledge by Write command is disabled: Reading AL Control from PDI clears AL Event Request 0x0220[0]. Writing to this register from PDI is not possible. PDI register function acknowledge by Write command is enabled: Writing AL Control from PDI clears AL Event Request 0x0220[0]. Writing to this register from PDI is possible; write value is ignored (write 0).

NOTE: AL Control register behaves like a mailbox if Device Emulation is off (0x0141[0]=0): The PDI has to read / write* the AL Control register after ECAT has written it. Otherwise ECAT cannot write again to the AL Control register. After Reset, AL Control register can be written by ECAT. (Regarding mailbox functionality, both low and high byte of the AL Control register trigger read/write functions, e.g., reading 0x0121 is sufficient to make this register writable again) If Device Emulation is on, the AL Control register can always be written, its content is copied to the AL Status register.

55.5.21. ESC AL status register (ESC_AL_STAT)

Address Offset: 0x0130

Reset value: 0x0001

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Reserved	ERROR_IND	AL_STATE[3:0]
	r(ECAT) rw(PDI)	r(ECAT) rw(PDI)

Bits	Fields	Descriptions
15:5	Reserved	Must be kept at reset value.
4	ERROR_IND	Error Ind: 0: Device is in State as requested or Flag cleared by command 1: Device has not entered requested State or changed State as result of a local action
3:0	AL_STATE[3:0]	Actual State of the Device State Machine: 1: Init State 3: Bootstrap State 2: Pre-Operational State 4: Safe-Operational State 8: Operational State

NOTE: AL Status register is only writable from PDI if Device Emulation is off (0x0141[0]=0), otherwise AL Status register will reflect AL Control register values. Avoid reading AL Status register from PDI.

55.5.22. ESC AL status code register (ESC_AL_STAT_CODE)

Address Offset: 0x0134
Reset value: 0x0000

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AL_STATUS_CODE[15:0]															
r(ECAT) rw(PDI)															

Bits	Fields	Descriptions
15:0	AL_STATUS_CODE[15:0]	AL Status Code

55.5.23. ESC RUN LED override register (ESC_RUN_LED_OVRD)

Address Offset: 0x0138
Reset value: 0x00

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
Reserved			OVERRIDEN	LED_CODE[3:0]			
			rw(ECAT) rw(PDI)	rw(ECAT) rw(PDI)			

Bits	Fields	Descriptions	
7:5	Reserved	Must be kept at reset value.	
4	OVERRIDEN	Enable Override: 0: Override disabled 1: Override enabled	
3:0	LED_CODE[3:0]	LED code:	AL Status:
		0x0: Off	Init 1
		0x1-0xC: Flash 1x - 12x	SafeOp 4
		0xD: Blinking	PreOp 2
		0xE: Flickering	Bootstrap 3
		0xF: On	Operational 8

NOTE: Changes to AL Status register (0x0130) with valid values will disable RUN LED Override (0x0138[4]=0). The value read in this register always reflects current LED output.

55.5.24. ESC ERR LED override register (ESC_ERR_LED_OVRD)

Address Offset: 0x0139
Reset value: 0x00

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
Reserved			OVERRIDEN	LED_CODE[3:0]			
			rw(ECAT) rw(PDI)	rw(ECAT) rw(PDI)			

Bits	Fields	Descriptions
7:5	Reserved	Must be kept at reset value.
4	OVERRIDEN	Enable Override: 0: Override disabled 1: Override enabled
3:0	LED_CODE[3:0]	LED code:

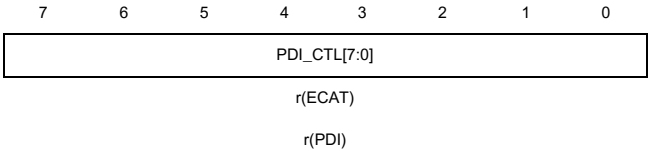
0x0: Off
0x1-0xC: Flash 1x - 12x
0xD: Blinking
0xE: Flickering
0xF: On

NOTE: New error conditions will disable ERR LED Override (0x0139[4]=0). The value read in this register always reflects current LED output.

55.5.25. ESC PDI control register (ESC_PDI_CTL)

Address Offset: 0x0140
Reset value: 0x80

This register can be accessed by byte(8-bit).

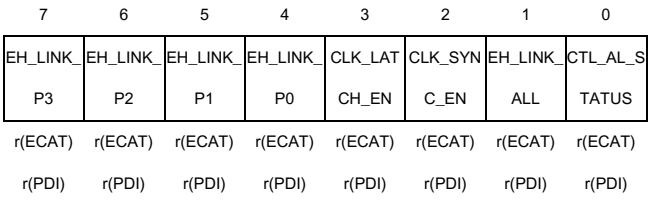


Bits	Fields	Descriptions
7:0	PDI_CTL[7:0]	Process data interface: 0x80: OPB Others: Reserved

55.5.26. ESC Configuration register (ESC_CFG)

Address Offset: 0x0141
Reset value: depends on configuration

This register can be accessed by byte(8-bit).



Bits	Fields	Descriptions
7	EH_LINK_P3	Enhanced Link port 3(default 1, later EEPROM word 0): 0: disabled (if bit 1=0) 1: enabled
6	EH_LINK_P2	Enhanced Link port 2(default 1, later EEPROM word 0): 0: disabled (if bit 1=0)

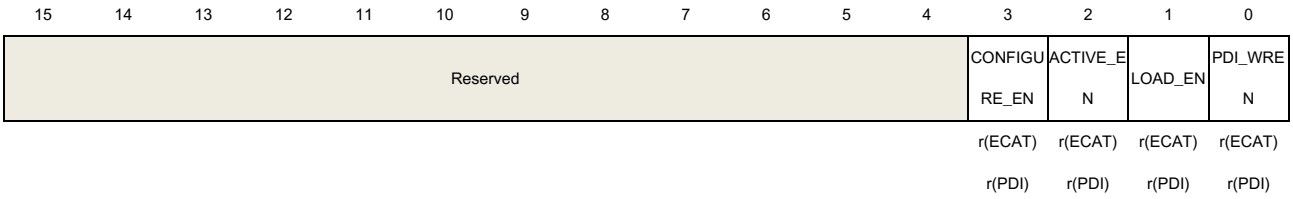
		1: enabled
5	EH_LINK_P1	Enhanced Link port 1(default 1, later EEPROM word 0): 0: disabled (if bit 1=0) 1: enabled
4	EH_LINK_P0	Enhanced Link port 0(default 1, later EEPROM word 0): 0: disabled (if bit 1=0) 1: enabled
3	CLK_LATCH_EN	Distributed Clocks Latch In Unit: 0: disabled (power saving) 1: enabled
2	CLK_SYNC_EN	Distributed Clocks SYNC Out Unit: 0: disabled (power saving) 1: enabled
1	EH_LINK_ALL	Enhanced Link detection all ports(default 1, later EEPROM word 0): 0: disabled (if bits [7:4]=0) 1: enabled at all ports (overrides bits [7:4])
0	CTL_AL_STATUS	Device emulation (control of AL status) (default 0) 0: AL status register has to be set by PDI 1: AL status register will be set to value written to AL control register **NOTE**: Reset value is 1 with Digital I/O PDI，On-chip bus Others: 0。

NOTE: EEPROM values of bits 1, 4, 5, 6, and 7 are only transferred into this register at first EEPROM load after power-on or reset.

55.5.27. ESC PDI Information register (ESC_PDI_INFO)

Address Offset: 0x014E
Reset value: depends on configuration

This register can be accessed by half-word(16-bit).



Bits	Fields	Descriptions
15:4	Reserved	Must be kept at reset value.
3	CONFIGURE_EN	PDI configuration invalid(default 0): 0: PDI configuration ok

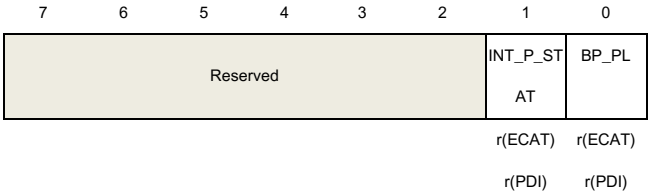
		1: PDI configuration invalid
2	ACTIVE_EN	PDI active(default 0): 0: PDI not active 1: PDI active
1	LOAD_EN	ESC configuration area loaded from EEPROM(default 0): 0: not loaded 1: loaded
0	PDI_WREN	PDI function acknowledge by write: 0: Disabled 1: Enabled Note: Reset value depends on configuration

55.5.28. ESC PDI configuration register (ESC_PDI_CFG)

Address Offset: 0x0150

Reset value: 0x00

This register can be accessed by byte(8-bit).



Bits	Fields	Descriptions
7:2	Reserved	Must be kept at reset value.
1	INT_P_STAT	Initial port state(default 0, later EEPROM word 1) 0: Always closed (0x0100[15:14]=11) 1: Auto (0x0100[15:14]=00)
0	BP_PL	Bridge port physical layer(default 0, later EEPROM word 1) 0: Reserved 1: MII

NOTE: all the bit can be configured via EEPROM.

55.5.29. ESC Sync / Latch configuration register (ESC_SL_CFG)

Address Offset: 0x0151

Reset value: depends on configuration

This register can be accessed by byte(8-bit).



SYNC1_MAP	SL1_CONFIG	SYNC1_POLARITY[0:1]	SYNC0_MAP	SL0_CONFIG	SYNC0_POLARITY[1:0]
r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)

Bits	Fields	Descriptions
7	SYNC1_MAP	SYNC1 mapped to AL Event Request Register 0x0220[3]: 0: Disabled 1: Enabled
6	SL1_CONFIG	SYNC1 / LATCH1 Configuration(default 1): 0: LATCH1 Input 1: SYNC1 Output **NOTE**: The ESC has concurrent SYNC[1:0] outputs and LATCH[1:0] inputs, independent of this configuration
5:4	SYNC1_POLARITY[1:0]	SYNC1 Output Driver / Polarity(default 10): 00: Push-Pull (Active Low) 01: Open Drain (Active Low) 10: Push-Pull (Active High) 11: Open Source (Active High)
3	SYNC0_MAP	SYNC0 mapped to AL Event Request Register 0x0220[2]: 0: Disabled 1: Enabled
2	SL0_CONFIG	SYNC0 / LATCH0 Configuration(default 1): 0: LATCH0 Input 1: SYNC0 Output NOTE: The ESC has concurrent SYNC[1:0] outputs and LATCH[1:0] inputs, independent of this configuration
1:0	SYNC1_POLARITY[1:0]	SYNC0 Output Driver/Polarity(default 10): 00: Push-Pull (Active Low) 01: Open Drain (Active Low) 10: Push-Pull (Active High) 11: Open Source (Active High)

55.5.30. ESC PDI extended configuration register (ESC_PDI_EXT_CFG)

Address Offset: 0x0152
Reset value: depends on configuration

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIR15	DIR14	DIR13	DIR12	DIR11	DIR10	DIR9	DIR8	DIR7	DIR6	DIR5	DIR4	DIR3	DIR2	DIR1	DIR0

r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)

Bits	Fields	Descriptions
15	DIR15	Direction of I/O[31:30] configured in pairs as inputs or outputs: Refer to DIR0 description
14	DIR14	Direction of I/O[29:28] configured in pairs as inputs or outputs: Refer to DIR0 description
13	DIR13	Direction of I/O[27:26] configured in pairs as inputs or outputs: Refer to DIR0 description
12	DIR12	Direction of I/O[25:24] configured in pairs as inputs or outputs: Refer to DIR0 description
11	DIR11	Direction of I/O[23:22] configured in pairs as inputs or outputs: Refer to DIR0 description
10	DIR10	Direction of I/O[21:20] configured in pairs as inputs or outputs: Refer to DIR0 description
9	DIR9	Direction of I/O[19:18] configured in pairs as inputs or outputs: Refer to DIR0 description
8	DIR8	Direction of I/O[17:16] configured in pairs as inputs or outputs: Refer to DIR0 description
7	DIR7	Direction of I/O[15:14] configured in pairs as inputs or outputs: Refer to DIR0 description
6	DIR6	Direction of I/O[13:12] configured in pairs as inputs or outputs: Refer to DIR0 description
5	DIR5	Direction of I/O[11:10] configured in pairs as inputs or outputs: Refer to DIR0 description
4	DIR4	Direction of I/O[9:8] configured in pairs as inputs or outputs: Refer to DIR0 description
3	DIR3	Direction of I/O[7:6] configured in pairs as inputs or outputs: Refer to DIR0 description
2	DIR2	Direction of I/O[5:4] configured in pairs as inputs or outputs: Refer to DIR0 description
1	DIR1	Direction of I/O[3:2] configured in pairs as inputs or outputs: Refer to DIR0 description
0	DIR0	Direction of I/O[1:0] configured in pairs as inputs or outputs: 0: Input

1: Output

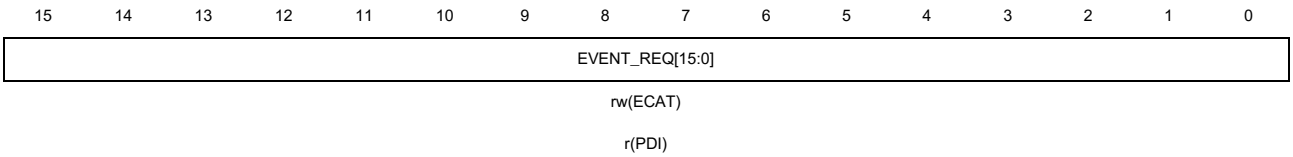
NOTE: Reserved in bidirectional mode, set to 0. Configuration bits for unavailable I/Os are reserved, set EEPROM value to 0.

55.5.31. ESC Event mask register (ESC_EVT_MSK)

Address Offset: 0x0200

Reset value: 0x0000

This register can be accessed by half-word(16-bit).



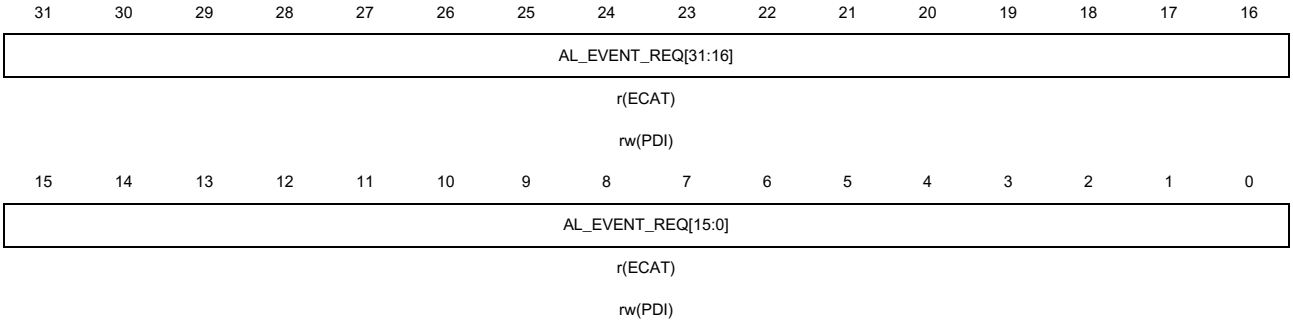
Bits	Fields	Descriptions
15:0	EVENT_REQ[15:0]	ECAT Event masking of the ECAT Event Request Events for mapping into ECAT event field of EtherCAT frames: 0: Corresponding ECAT Event Request register bit is not mapped 1: Corresponding ECAT Event Request register bit is mapped

55.5.32. ESC PDI AL event register (ESC_PDI_AL_EVT)

Address Offset: 0x0204

Reset value: 0x00FF FF0F

This register can be accessed by word(32-bit).



Bits	Fields	Descriptions
31:0	AL_EVENT_REQ[31:0]	AL Event masking of the AL Event Request register Events for mapping to PDI IRQ signal: 0: Corresponding AL Event Request register bit is not mapped 1: Corresponding AL Event Request register bit is mapped

55.5.33. ESC Event request register (ESC_EVT_REQ)

Address Offset: 0x0210

Reset value: 0x0000

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				CHANNEL7	CHANNEL6	CHANNEL5	CHANNEL4	CHANNEL3	CHANNEL2	CHANNEL1	CHANNEL0	AL_STATUS	DL_STATUS	Reserved	DC_LATCH
				r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)		r(ECAT) r(PDI)

Bits	Fields	Descriptions
15:12	Reserved	Must be kept at reset value.
11	CHANNEL7	Mirrors values of each SyncManager Status: 0: No Sync Channel 7 event 1: Sync Channel 7 event pending
10	CHANNEL6	Mirrors values of each SyncManager Status: 0: No Sync Channel 6 event 1: Sync Channel 6 event pending
9	CHANNEL5	Mirrors values of each SyncManager Status: 0: No Sync Channel 5 event 1: Sync Channel 5 event pending
8	CHANNEL4	Mirrors values of each SyncManager Status: 0: No Sync Channel 4 event 1: Sync Channel 4 event pending
7	CHANNEL3	Mirrors values of each SyncManager Status: 0: No Sync Channel 3 event 1: Sync Channel 3 event pending
6	CHANNEL2	Mirrors values of each SyncManager Status: 0: No Sync Channel 2 event 1: Sync Channel 2 event pending
5	CHANNEL1	Mirrors values of each SyncManager Status: 0: No Sync Channel 1 event 1: Sync Channel 1 event pending
4	CHANNEL0	Mirrors values of each SyncManager Status: 0: No Sync Channel 0 event 1: Sync Channel 0 event pending
3	AL_STATUS	AL Status event:

		0: No change in AL Status 1: AL Status change (Bit is cleared by reading out AL Status 0x0130:0x0131 from ECAT)
2	DL_STATUS	DL Status event: 0: No change in DL Status 1: DL Status change (Bit is cleared by reading out DL Status 0x0110:0x0111 from ECAT)
1	Reserved	Must be kept at reset value.
0	DC_LATCH	DC Latch event: 0: No change on DC Latch Inputs 1: At least one change on DC Latch Inputs (Bit is cleared by reading DC Latch event times from ECAT for ECAT-controlled Latch Units, so that Latch 0/1 Status 0x09AE:0x09AF indicates no event)

55.5.34. ESC AL event request register (ESC_AL_EVT_REQ)

Address Offset: 0x0220

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								SYNC15_INT	SYNC14_INT	SYNC13_INT	SYNC12_INT	SYNC11_INT	SYNC10_INT	SYNC9_INT	SYNC8_INT
								r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYNC7_INT	SYNC6_INT	SYNC5_INT	SYNC4_INT	SYNC3_INT	SYNC2_INT	SYNC1_INT	SYNC0_INT	Reserved	PDATA	EEPROM	SYNC_CHANGE	DC_SYNC1	DC_SYNC0	DC_LATCH	AL_CONTROL
r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)		r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23	SYNC15_INT	SyncManager interrupts (SyncManager register offset 0x5, bit [0] or [1]): 0: No SyncManager 15 interrupt 1: SyncManager 15 interrupt pending
22	SYNC14_INT	SyncManager interrupts (SyncManager register offset 0x5, bit [0] or [1]): 0: No SyncManager 14 interrupt 1: SyncManager 14 interrupt pending
21	SYNC13_INT	SyncManager interrupts (SyncManager register offset 0x5, bit [0] or [1]): 0: No SyncManager 13 interrupt

		1: SyncManager 13 interrupt pending
20	SYNC12_INT	SyncManager interrupts (SyncManager register offset 0x5, bit [0] or [1]): 0: No SyncManager 12 interrupt 1: SyncManager 12 interrupt pending
19	SYNC11_INT	SyncManager interrupts (SyncManager register offset 0x5, bit [0] or [1]): 0: No SyncManager 11 interrupt 1: SyncManager 11 interrupt pending
18	SYNC10_INT	SyncManager interrupts (SyncManager register offset 0x5, bit [0] or [1]): 0: No SyncManager 10 interrupt 1: SyncManager 10 interrupt pending
17	SYNC9_INT	SyncManager interrupts (SyncManager register offset 0x5, bit [0] or [1]): 0: No SyncManager 9 interrupt 1: SyncManager 9 interrupt pending
16	SYNC8_INT	SyncManager interrupts (SyncManager register offset 0x5, bit [0] or [1]): 0: No SyncManager 8 interrupt 1: SyncManager 8 interrupt pending
15	SYNC7_INT	SyncManager interrupts (SyncManager register offset 0x5, bit [0] or [1]): 0: No SyncManager 7 interrupt 1: SyncManager 7 interrupt pending
14	SYNC6_INT	SyncManager interrupts (SyncManager register offset 0x5, bit [0] or [1]): 0: No SyncManager 6 interrupt 1: SyncManager 6 interrupt pending
13	SYNC5_INT	SyncManager interrupts (SyncManager register offset 0x5, bit [0] or [1]): 0: No SyncManager 5 interrupt 1: SyncManager 5 interrupt pending
12	SYNC4_INT	SyncManager interrupts (SyncManager register offset 0x5, bit [0] or [1]): 0: No SyncManager 4 interrupt 1: SyncManager 4 interrupt pending
11	SYNC3_INT	SyncManager interrupts (SyncManager register offset 0x5, bit [0] or [1]): 0: No SyncManager 3 interrupt 1: SyncManager 3 interrupt pending
10	SYNC2_INT	SyncManager interrupts (SyncManager register offset 0x5, bit [0] or [1]): 0: No SyncManager 2 interrupt 1: SyncManager 2 interrupt pending
9	SYNC1_INT	SyncManager interrupts (SyncManager register offset 0x5, bit [0] or [1]): 0: No SyncManager 1 interrupt 1: SyncManager 1 interrupt pending

8	SYNC0_INT	SyncManager interrupts (SyncManager register offset 0x5, bit [0] or [1]): 0: No SyncManager 0 interrupt 1: SyncManager 0 interrupt pending
7	Reserved	Must be kept at reset value.
6	PDATA	Watchdog Process Data: 0: Has not expired 1: Has expired (Bit is cleared by reading Watchdog Status Process Data 0x0440 from PDI)
5	EEPROM	EEPROM Emulation: 0: No command pending 1: EEPROM command pending (Bit is cleared by acknowledging the command in EEPROM Control/Status register 0x0502:0x0503[10:8] from PDI)
4	SYNC_CHANGE	SyncManager activation register (SyncManager register offset 0x6) changed: 0: No change in any SyncManager 1: At least one SyncManager changed (Bit is cleared by reading SyncManager Activation registers 0x0806 etc. from PDI)
3	DC_SYNC1	State of DC SYNC1 (if register 0x0151[7]=1): (Bit is cleared by reading of SYNC1 status 0x098F from PDI, use only in Acknowledge mode)
2	DC_SYNC0	State of DC SYNC0 (if register 0x0151[3]=1): (Bit is cleared by reading SYNC0 status 0x098E from PDI, use only in Acknowledge mode)
1	DC_LATCH	DC Latch event: 0: No change on DC Latch Inputs 1: At least one change on DC Latch Inputs (Bit is cleared by reading DC Latch event times from PDI, so that Latch 0/1 Status 0x09AE:0x09AF indicates no event. Available if Latch Unit is PDI-controlled)
0	AL_CONTROL	AL Control event: 0: No AL Control Register change 1: AL Control Register has been written(AL control event is only generated if PDI emulation is turned off (ESC Configuration 0 register 0x0141[0]=0)) (Bit is cleared by reading AL Control register 0x0120:0x0121 from PDI)

55.5.35. RX error port X register (RX_PORTX_ERR) (X = 0,1,2,3)

Address Offset: 0x0300 + X * 2
Reset value: 0x0000

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EEOR_PORTX[7:0]								INVALID_PORTX[7:0]							
rwc(ECAT)								rwc(ECAT)							
r(PDI)								r(PDI)							

Bits	Fields	Descriptions
15:8	EEOR_PORTX[7:0]	Port X RX Error Counter RX Error counter of Port X (counting is stopped when 0xFF is reached).
7:0	INVALID_PORTX[7:0]	Port X Invalid Frame Counter Invalid frame counter of Port X (counting is stopped when 0xFF is reached).

NOTE: Error Counters 0x0300-0x030B are cleared if one of the implemented RX Error counters 0x0300-0x030B is written (preferably 0x0300). Write value is ignored (write 0). Errors are only counted if the loop of the port is open.

55.5.36. Forwarded RX error port X register (FWD_PORTX_ERR) (X = 0,1,2,3)

Address Offset: 0x0308 + X
Reset value: 0x00

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
F_ERROR_PORTX[7:0]							
rwc(ECAT)							
r(PDI)							

Bits	Fields	Descriptions
7:0	F_ERROR_PORTX[7:0]	Forwarded error counter of Port X (counting is stopped when 0xFF is reached).

NOTE: Error Counters 0x0300-0x030B are cleared if one of the implemented RX Error counters 0x0300-0x030B is written (preferably 0x0300). Write value is ignored (write 0). Errors are only counted if the loop of the port is open.

55.5.37. ESC processing unit error register (ESC_PU_ERR)

Address Offset: 0x030C
Reset value: 0x00

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
PU_ERROR[7:0]							
rwc(ECAT)							
r(PDI)							

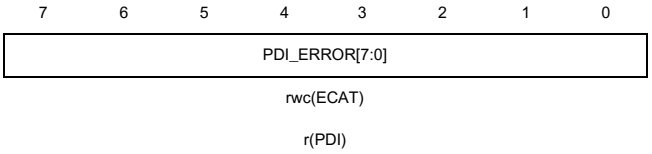
Bits	Fields	Descriptions
7:0	PU_ERROR[7:0]	ECAT Processing Unit error counter (counting is stopped when 0xFF is reached). Counts errors of frames passing the Processing Unit.

NOTE: Error Counter 0x030C is cleared if error counter 0x030C is written. Write value is ignored (write 0).

55.5.38. ESC PDI error counter register (ESC_PDI_ERR)

Address Offset: 0x030D
Reset value: 0x00

This register can be accessed by byte(8-bit).



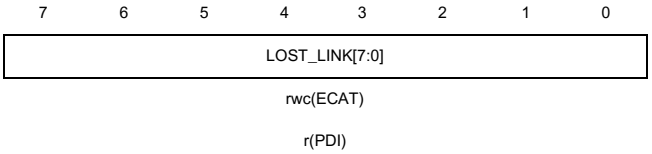
Bits	Fields	Descriptions
7:0	PDI_ERROR[7:0]	PDI Error counter (counting is stopped when 0xFF is reached). Counts if a PDI access has an interface error.

NOTE: Error Counter 0x030D and Error Code 0x030E:0x030F are cleared if error counter 0x030D is written. Write value is ignored (write 0).

55.5.39. ESC PORTx lost link register (ESC_PORTX_LOST_LINK) (x = 0,1,2,3)

Address Offset: 0x0310 + X
Reset value: 0x00

This register can be accessed by byte(8-bit).



Bits	Fields	Descriptions
7:0	LOST_LINK[7:0]	Lost Link counter of Port X (counting is stopped when 0xff is reached). Counts only if port is open and loop is Auto.

NOTE: Lost Link Counters 0x0310-0x0313 are cleared if one of the implemented Lost Link Counters 0x0310-0x0313 is written (preferably 0x0310). Write value is ignored (write 0).

55.5.40. ESC Watchdog divider register (ESC_WDG_DIV)

Address Offset: 0x0400
Reset value: 0x09C2

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WATCHDOG_DIV[15:0]															
rw(ECAT)															
r(PDI)															

Bits	Fields	Descriptions
15:0	WATCHDOG_DIV[15:0]	Watchdog divider: Number of 25 MHz tics (minus 2) that represent the basic watchdog increment. (Default value is 100µs = 2498)

55.5.41. ESC Watchdog time PDI register (ESC_WDG_TIME_PDI)

Address Offset: 0x0410
Reset value: 0x03E8

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WATCHDOG_TIME[15:0]															
rw(ECAT)															
r(PDI)															

Bits	Fields	Descriptions
15:0	WATCHDOG_TIME[15:0]	Watchdog Time PDI: number of basic watchdog increments (Default value with Watchdog divider 100µs means 100ms Watchdog)

NOTE: Watchdog is disabled if Watchdog time is set to 0x0000. Watchdog starts counting again with every PDI access.

55.5.42. ESC Watchdog time process data register (ESC_WDG_TIME_PDATA)

Address Offset: 0x0420
Reset value: 0x03E8

This register can be accessed by half-word(16-bit).

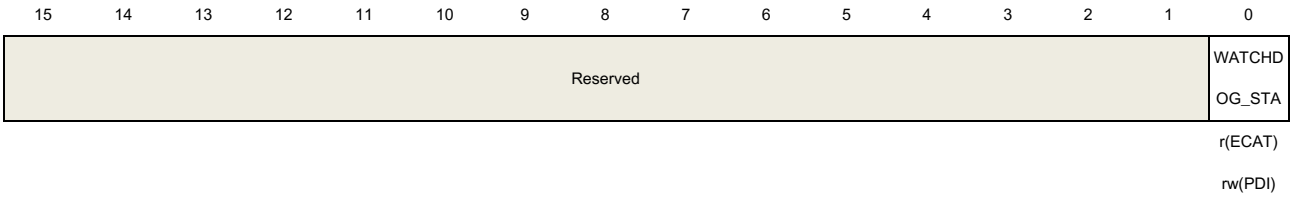
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WATCHDOG_TPD[15:0]															
rw(ECAT)															
r(PDI)															

Bits	Fields	Descriptions
15:0	WATCHDOG_TPD[15:0]	Watchdog Time Process Data: number of basic watchdog increments (Default value with Watchdog divider 100µs means 100ms Watchdog)

55.5.43. ESC Watchdog status process data register (ESC_WDG_STAT_PDATA)

Address Offset: 0x0440
Reset value: 0x0000

This register can be accessed by half-word(16-bit).

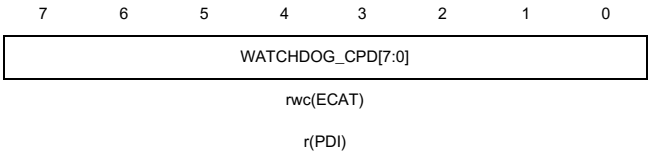


Bits	Fields	Descriptions
15:1	Reserved	Must be kept at reset value.
0	WATCHDOG_STA	Watchdog Status of Process Data (triggered by SyncManagers) 0: Watchdog Process Data expired 1: Watchdog Process Data is active or disabled NOTE: PDI register function acknowledge by Write command is disabled: Reading this register from PDI clears AL Event Request 0x0220[6]. Writing to this register from PDI is not possible. PDI register function acknowledge by Write command is enabled: Writing this register from PDI clears AL Event Request 0x0220[6]. Writing to this register from PDI is possible; write value is ignored (write 0).

55.5.44. ESC Watchdog counter process data register (ESC_WDG_CNT_PDATA)

Address Offset: 0x0442
Reset value: 0x00

This register can be accessed by byte(8-bit).

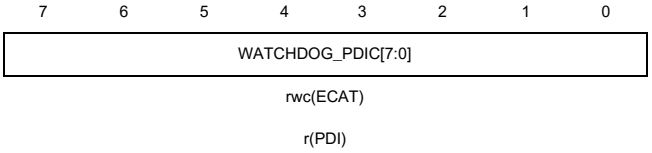


Bits	Fields	Descriptions
7:0	WATCHDOG_CPD[7:0]	Watchdog Counter Process Data (counting is stopped when 0xFF is reached). Counts if Process Data Watchdog expires.

55.5.45. ESC Watchdog counter PDI register (ESC_WDG_CNT_PDI)

Address Offset: 0x0443
Reset value: 0x00

This register can be accessed by byte(8-bit).



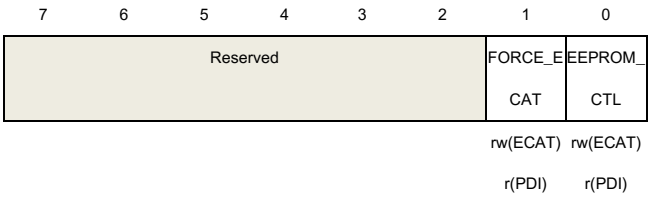
Bits	Fields	Descriptions
7:0	WATCHDOG_PDIC[7:0]	Watchdog PDI counter (counting is stopped when 0xFF is reached). Counts if PDI Watchdog expires.

NOTE: Watchdog Counters 0x0442-0x0443 are cleared if one of the Watchdog Counters 0x0442-0x0443 is written. Write value is ignored (write 0).

55.5.46. ESC EEPOM configuration register (ESC_EEPROM_CFG)

Address Offset: 0x0500
Reset value: 0x00

This register can be accessed by byte(8-bit).



Bits	Fields	Descriptions
7:2	Reserved	Must be kept at reset value.
1	FORCE_ECAT	Force ECAT access: 0: Do not change Bit 0x0501[0] 1: Reset Bit 0x0501[0] to 0
0	EEPROM_CTL	EEPROM control is offered to PDI: 0: no 1: yes (PDI has EEPROM control)

55.5.47. ESC EEPOM PDI access register (ESC_EEPROM_PDI_ACC)

Address Offset: 0x0501
Reset value: 0x00

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
Reserved							ACCESS_EEPROM
							r(ECAT) rw(PDI)

Bits	Fields	Descriptions
7:1	Reserved	Must be kept at reset value.
0	ACCESS_EEPROM	Access to EEPROM: 0: PDI releases EEPROM access 1: PDI takes EEPROM access (PDI has EEPROM control)

NOTE: write access is only possible if (0x0500[0]=1 or 0x0501[0]=1) and 0x0500[1]=0.

55.5.48. ESC EEPOM control / status register (ESC_EEPROM_CS)

Address Offset: 0x0502

Reset value: depends on configuration

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EEPROM_IF	ERROR_WEN	ERROR_ACK	EEPROM_STA	CHECKSUM	COMMAND[2:0]			SELECT	READ_BYTES	EEPROM_EMU	Reserved				ECAT_EN
r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) rw(PDI)	r(ECAT) r(PDI)	r(ECAT) rw(PDI)	rw(ECAT) rw(PDI)			r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)					rw(ECAT) r(PDI)

Bits	Fields	Descriptions
15	EEPROM_IF	Busy(default 0): 0: EEPROM Interface is idle 1: EEPROM Interface is busy
14	ERROR_WEN	Error Write Enable(default 0): 0: No error 1: Write Command without Write enable NOTE: Error bits are cleared by writing "000" (or any valid command) to Command Register Bits [10:8].
13	ERROR_ACK	Error Acknowledge/Command(default 0): 0: No error 1: Missing EEPROM acknowledge or invalid command EEPROM emulation only: PDI writes 1 if a temporary failure has occurred. NOTE: Error bits are cleared by writing "000" (or any valid command) to Command Register Bits [10:8].

12	EEPROM_STA	EEPROM loading status(default 0): 0: EEPROM loaded, device information ok 1: EEPROM not loaded, device information not available (EEPROM loading in progress or finished with a failure)
11	CHECKSUM	Checksum Error in ESC Configuration Area(default 0): 0: Checksum ok 1: Checksum error EEPROM emulation for ESC only: PDI writes 1 if a CRC failure has occurred for a reload command.
10:8	COMMAND[2:0]	Command register(default 0): Write: Initiate command. Read: Currently executed command Commands: 000: No command/EEPROM idle (clear error bits) 001: Read 010: Write 100: Reload Others: Reserved/invalid commands (do not issue) EEPROM emulation only: after execution, PDI writes command value to indicate operation is ready. NOTE: Write Enable bit 0 is self-clearing at the SOF of the next frame, Command bits [10:8] are self-clearing after the command is executed (EEPROM Busy ends). Writing "000" to the command register will also clear the error bits [14:13]. Command bits [10:8] are ignored if Error Acknowledge/Command is pending (bit 13).
7	SELECT	Selected EEPROM Algorithm: 0: 1 address byte (1Kbit - 16Kbit EEPROMs) 1: 2 address bytes (32Kbit - 4 Mbit EEPROMs)
6	READ_BYTES	Supported number of EEPROM read bytes(default 0): 0: 4 Bytes 1: 8 Bytes
5	EEPROM_EMU	EEPROM emulation: 0: Normal operation (I²C interface used) 1: PDI emulates EEPROM (I²C not used)
4:1	Reserved	Must be kept at reset value.
0	ECAT_EN	ECAT write enable(default 0): 0: Write requests are disabled 1: Write requests are enabled This bit is always 1 if PDI has EEPROM control. NOTE: Write Enable bit 0 is self-clearing at the SOF of the next frame, Command

bits [10:8] are self-clearing after the command is executed (EEPROM Busy ends). Writing "000" to the command register will also clear the error bits [14:13]. Command bits [10:8] are ignored if Error Acknowledge/Command is pending (bit 13).

55.5.49. ESC EEPROM address register (ESC_EEPROM_ADDR)

Address Offset: 0x0504
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
EEPROM_ADDR[31:0]															
rw(ECAT)															
rw(PDI)															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EEPROM_ADDR[31:0]															
rw(ECAT)															
rw(PDI)															

Bits	Fields	Descriptions
31:0	EEPROM_ADDR[31:0]	EEPROM Address 0: First word (= 16 bit) 1: Second word … Actually used EEPROM Address bits: [9:0]: EEPROM size up to 16 Kbit [17:0]: EEPROM size 32 Kbit - 4 Mbit [31:0]: EEPROM Emulation

NOTE: write access depends upon the assignment of the EEPROM interface (ECAT / PDI). Write access is blocked if EEPROM interface is busy (0x0502[15]=1).

55.5.50. ESC EEPROM data register (ESC_EEPROM_DATA)

Address Offset: 0x0508
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
EEPROM_READ[15:0]															
r(ECAT)															
rw(PDI)															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EERPOM_WRITE[15:0]															

rw(ECAT)

rw(PDI)

Bits	Fields	Descriptions
31:16	EEPROM_READ[15:0]	EEPROM Read data (data read from EEPROM, higher bytes)
15:0	EERPOM_WRITE[15:0]	EEPROM Write data (data to be written to EEPROM) or EEPROM Read data (data read from EEPROM, lower bytes)

55.5.51. ESC MII management control / status register (ESC_MII_MNG_CS)

Address Offset: 0x0510

Reset value: 0x0000

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MII_IF	COMMAND_ERROR	READ_ERROR	Reserved			COMMAND[1:0]		PHY_ADDRP0[4:0]					MII_CONFIG	PDI_CTL	WRITE_EN
r(ECAT) r(PDI)	r(ECAT) r(PDI)	rw(ECAT) rw(PDI)				rw(ECAT) rw(PDI)		r(ECAT) r(PDI)					r(ECAT) r(PDI)	r(ECAT) r(PDI)	rw(ECAT) r(PDI)

Bits	Fields	Descriptions
15	MII_IF	Busy: 0: MII Management Interface is idle 1: MII Management Interface is busy
14	COMMAND_ERROR	Command error: 0: Last Command was successful 1: Invalid command or write command without Write Enable Cleared by executing a valid command or by writing "00" to Command register bits [9:8].
13	READ_ERROR	Read error: 0: No read error 1: Read error occurred (PHY or register not available) Cleared by writing to register 0x0511.
12:10	Reserved	Must be kept at reset value.
9:8	COMMAND[1:0]	Command register: Write: Initiate command. Read: Currently executed command 00: No command/MI idle (clear error bits) 01: Read 10: Write

		Others: Reserved/invalid command (do not issue) NOTE: Write enable bit 0 is self-clearing at the SOF of the next frame, Command bits [10:8] are self-clearing after the command is executed (Busy ends). Writing "00" to the command register will also clear the error bits [14:13]. The Command bits are cleared after the command is executed.
7:3	PHY_ADDRP0[4:0]	PHY address of port 0 (this is equal to the PHY address offset, if the PHY addresses are consecutive) IP Core since V3.0.0/3.00c: Translation 0x0512[7] =0: Register 0x0510[7:3] shows PHY address of port 0 Translation 0x0512[7] =1: Register 0x0510[7:3] shows the PHY address which will be used for port 0-3 as requested by 0x0512[4:0] (valid values 0-3)
2	MII_CONFIG	MI link detection and configuration: 0: Disabled for all ports 1: Enabled for at least one MII port, refer to PHY Port Status (0x0518 ff.) for details
1	PDI_CTL	Management Interface can be controlled by PDI (registers 0x0516-0x0517): 0: Only ECAT control 1: PDI control possible
0	WRITE_EN	Write enable: 0: Write disabled 1: Write enabled This bit is always 1 if PDI has MI control. NOTE: Write enable bit 0 is self-clearing at the SOF of the next frame, Command bits [10:8] are self-clearing after the command is executed (Busy ends). Writing "00" to the command register will also clear the error bits [14:13]. The Command bits are cleared after the command is executed.

55.5.52. ESC PHY address register (ESC_PHY_ADDR)

Address Offset: 0x0512

Reset value: 0x00

This register can be accessed by byte(8-bit).



Bits	Fields	Descriptions
7	PHY_ADDR	Target PHY Address translation: 0: Enabled

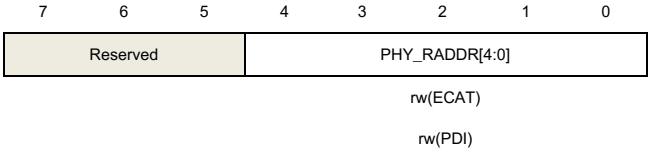
		1: Disabled Refer to 0x0512[4:0] and 0x0510[7:3] for details.
6:5	Reserved	Must be kept at reset value.
4:0	TARGET_PHY[4:0]	Target PHY Address Translation 0x0512[7]=0: 0-3: Target PHY Addresses 0-3 are used to access the PHYs at port 0-3, when the PHY addresses are properly configured 4-31: The configured PHY address of port 0 (PHY address offset) is added to the Target PHY Address values 4-31 when accessing a PHY Translation 0x0512[7]=1: 0-31: Target PHY Addresses is used when accessing a PHY without translation

NOTE: write access depends on assignment of MI (ECAT/PDI). Write access is blocked if Management interface is busy (0x0510[15]=1).

55.5.53. ESC PHY register address register (ESC_PHY_REG_ADDR)

Address Offset: 0x0513
Reset value: 0x00

This register can be accessed by byte(8-bit).

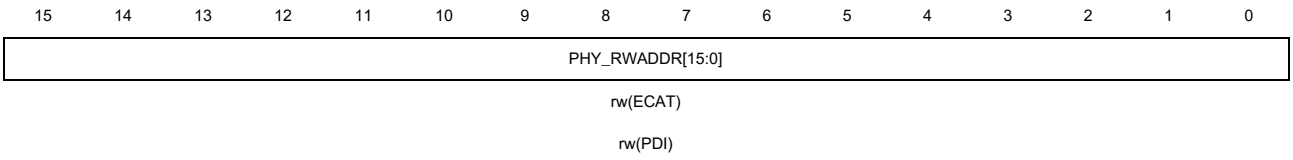


Bits	Fields	Descriptions
7:5	Reserved	Must be kept at reset value.
4:0	PHY_RADDR[4:0]	Address of PHY Register that shall be read/written

55.5.54. ESC PHY data register (ESC_PHY_DATA)

Address Offset: 0x0514
Reset value: 0x0000

This register can be accessed by half-word(16-bit).



Bits	Fields	Descriptions

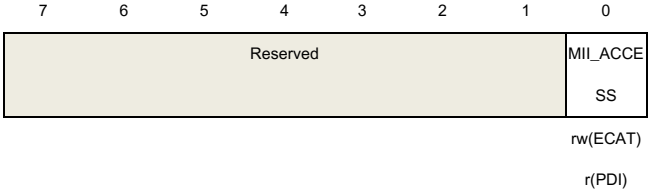
15:0	PHY_RWADDR[15:0] PHY Read/Write Data

55.5.55. MII management ECAT access state register (MIIM_ECAT_ACC_STAT)

Address Offset: 0x0516

Reset value: 0x00

This register can be accessed by byte(8-bit).



Bits	Fields	Descriptions
7:1	Reserved	Must be kept at reset value.
0	MII_ACCESS	Access to MII management: 0: ECAT enables PDI takeover of MII management interface 1: ECAT claims exclusive access to MII management interface

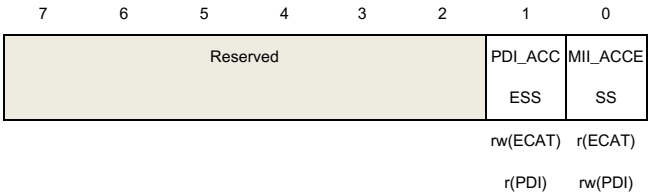
NOTE: write access is only possible if 0x0517[0]=0.

55.5.56. MII management PDI access state register (MIIM_PDI_ACC_STAT)

Address Offset: 0x0517

Reset value: 0x00

This register can be accessed by byte(8-bit).



Bits	Fields	Descriptions
7:2	Reserved	Must be kept at reset value.
1	PDI_ACCESS	Force PDI Access State: 0: Do not change Bit 0x0517[0] 1: Reset Bit 0x0517[0] to 0
0	MII_ACCESS	Access to MII management: 0: ECAT has access to MII management 1: PDI has access to MII management

55.5.57. PHY Port X status register (PHY_PORTX_STAT) (X = 0,1,2,3)

Address Offset: 0x0518 + X

Reset value: 0x00

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
Reserved		PHY_UPDATE	PARTNER_ERROR	READ_ERROR	STATUS_ERROR	LINK_STATUS	PHYLINK
		rwc(ECAT) rwc(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	rwc(ECAT) rwc(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)

Bits	Fields	Descriptions
7:6	Reserved	Must be kept at reset value.
5	PHY_UPDATE	PHY configuration updated: 0: No update 1: PHY configuration was updated Cleared by writing any value to at least one of the PHY Port X Status registers.
4	PARTNER_ERROR	Link partner error: 0: No error detected 1: Link partner error
3	READ_ERROR	Read error: 0: No read error occurred 1: A read error has occurred Cleared by writing any value to at least one of the PHY Port X Status registers.
2	STATUS_ERROR	Link status error: 0: No error 1: Link error, link inhibited
1	LINK_STATUS	Link status (100 Mbit/s, Full Duplex, Auto negotiation): 0: No link 1: Link detected
0	PHYLINK	Physical link status: 0: No physical link 1: Physical link detected

55.5.58. Logical start address FMMU X register (FMMUX_LOGIC_ADDR) (X = 0…F)

Address Offset: 0x0600 + X * 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
LOGIC_ADDR[31:16]															
rw(ECAT)															
r(PDI)															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LOGIC_ADDR[15:0]															
rw(ECAT)															
r(PDI)															

Bits	Fields	Descriptions
31:0	LOGIC_ADDR[31:0]	Logical start address within the EtherCAT Address Space.

55.5.59. Length FMMU X register (FMMUX_LENGTH) (X = 0…F)

Address Offset: 0x0604 + X * 0x10
Reset value: 0x0000

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
LENGTH_FMMU[15:0]															
rw(ECAT)															
r(PDI)															

Bits	Fields	Descriptions
15:0	LENGTH_FMMU[15:0]	Offset from the first logical FMMU byte to the last FMMU byte + 1 (e.g., if two bytes are used, then this parameter shall contain 2)

55.5.60. Start bit FMMU X in Logical address space register (FMMUX_STRA_BIT) (X = 0…F)

Address Offset: 0x0606 + X * 0x10
Reset value: 0x00

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
Reserved					START_BIT[2:0]		
					rw(ECAT)		
					r(PDI)		

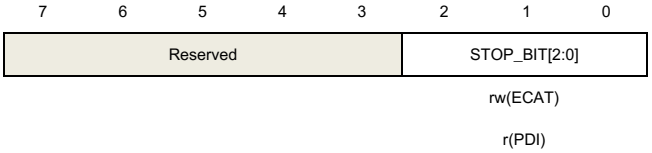
Bits	Fields	Descriptions
7:3	Reserved	Must be kept at reset value.
2:0	START_BIT[2:0]	Logical starting bit that shall be mapped (bits are counted from least significant bit

0 to most significant bit 7)

55.5.61. Stop bit FMMU X in Logical address space register (FMMUX_STOP_BIT) (X = 0…F)

Address Offset: 0x0607 + X * 0x10
Reset value: 0x00

This register can be accessed by byte(8-bit).

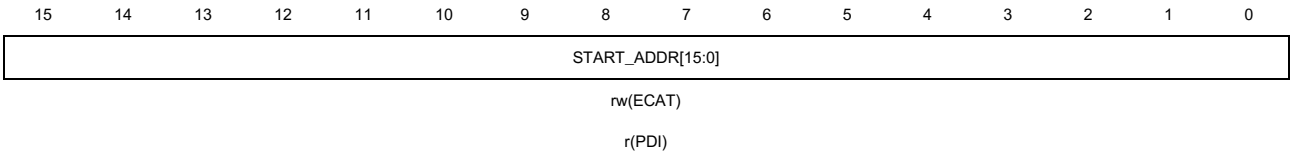


Bits	Fields	Descriptions
7:3	Reserved	Must be kept at reset value.
2:0	STOP_BIT[2:0]	Last logical bit that shall be mapped (bits are counted from least significant bit 0 to most significant bit 7)

55.5.62. Physical start address FMMU X register (FMMUX_ADDR) (X = 0…F)

Address Offset: 0x0608 + X * 0x10
Reset value: 0x0000

This register can be accessed by half-word(16-bit).

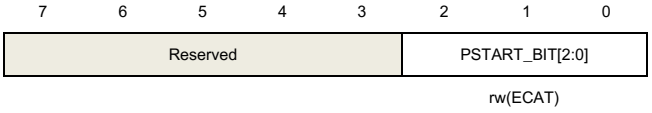


Bits	Fields	Descriptions
15:0	START_ADDR[15:0]	Physical Start Address (mapped to logical Start address)

55.5.63. Physical start bit FMMU X register (FMMUX_PSBIT) (X = 0…F)

Address Offset: 0x060A + X * 0x10
Reset value: 0x00

This register can be accessed by byte(8-bit).



r(PDI)

Bits	Fields	Descriptions
7:3	Reserved	Must be kept at reset value.
2:0	PSTART_BIT[2:0]	Physical starting bit as target of logical start bit mapping (bits are counted from least significant bit 0 to most significant bit 7)

55.5.64. Type FMMU X register (FMMUX_TYPE) (X = 0…F)

Address Offset: 0x060B + X * 0x10
Reset value: 0x00

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
Reserved						MAP_WRITE	MAP_READ
						rw(ECAT) r(PDI)	rw(ECAT) r(PDI)

Bits	Fields	Descriptions
7:2	Reserved	Must be kept at reset value.
1	MAP_WRITE	0: Ignore mapping for write accesses 1: Use mapping for write accesses
0	MAP_READ	0: Ignore mapping for read accesses 1: Use mapping for read accesses

55.5.65. Active FMMU X register (FMMUX_ACTIVE) (X = 0…F)

Address Offset: 0x060C + X * 0x10
Reset value: 0x00

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
Reserved							ACTIVE_FMMU
							rw(ECAT) r(PDI)

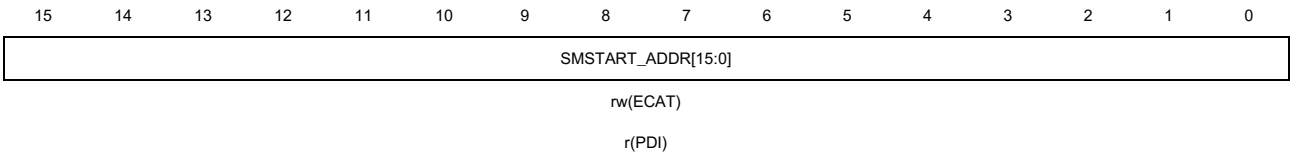
Bits	Fields	Descriptions
7:1	Reserved	Must be kept at reset value.
0	ACTIVE_FMMU	0: FMMU deactivated

1: FMMU activated. FMMU checks logically addressed blocks to be mapped according to configured mapping

55.5.66. Physical Start address SyncManager X register (SMX_START_ADDR) (X = 0…F)

Address Offset: 0x0800 + X * 8
Reset value: 0x0000

This register can be accessed by half-word(16-bit).

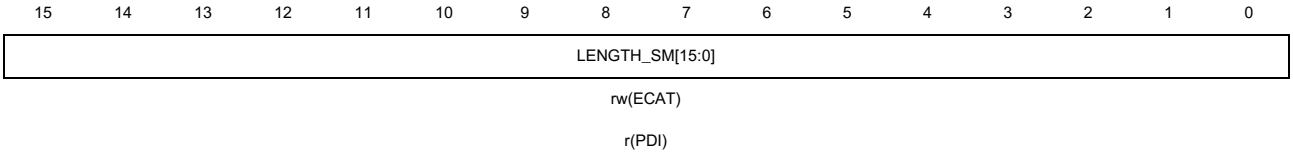


Bits	Fields	Descriptions
15:0	SMSTART_ADDR[15:0]	First byte that will be handled by SyncManager

55.5.67. Length SyncManager X register (SMX_LENGTH) (X = 0…F)

Address Offset: 0x0802 + X * 8
Reset value: 0x0000

This register can be accessed by half-word(16-bit).

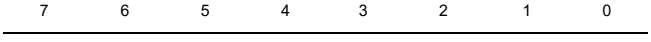


Bits	Fields	Descriptions
15:0	LENGTH_SM[15:0]	Number of bytes assigned to SyncManager (shall be greater than 1, otherwise SyncManager is not activated. If set to 1, only Watchdog Trigger is generated if configured)

55.5.68. Control register SyncManager X register (SMX_CTL) (X = 0…F)

Address Offset: 0x0804 + X * 8
Reset value: 0x00

This register can be accessed by byte(8-bit)



Reserved	TRIGGER	AL_EVENT_INT	ECAT_EVENT_INT	DIR[1:0]	OPERA_MODE[1:0]
	rw(ECAT) r(PDI)	rw(ECAT) r(PDI)	rw(ECAT) r(PDI)	rw(ECAT) r(PDI)	rw(ECAT) r(PDI)

Bits	Fields	Descriptions
7	Reserved	Must be kept at reset value.
6	TRIGGER	Watchdog Trigger Enable: 0: Disabled 1: Enabled
5	AL_EVENT_INT	Interrupt in AL Event Request Register: 0: Disabled 1: Enabled
4	ECAT_EVENT_INT	Interrupt in ECAT Event Request Register: 0: Disabled 1: Enabled
3:2	DIR[1:0]	Direction: 00: Read: ECAT read access, PDI write access. 01: Write: ECAT write access, PDI read access. 10: Reserved 11: Reserved
1:0	OPERA_MODE[1:0]	Operation Mode: 00: Buffered (3 buffer mode) 01: Reserved 10: Mailbox (Single buffer mode) 11: Reserved

55.5.69. Status register SyncManager X register (SMX_STAT) (X = 0…F)

Address Offset: 0x0805 + X * 8
Reset value: 0x30

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
WRITE_BUFFER	READ_BUFFER	BUFFER_STATUS[1:0]		MAILBOX_STATUS	Reserved	READ_INT	WRITE_INT
r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)		r(ECAT) r(PDI)		r(ECAT) r(PDI)	r(ECAT) r(PDI)

Bits	Fields	Descriptions

7	WRITE_BUFFER	Write buffer in use (opened)
6	READ_BUFFER	Read buffer in use (opened)
5:4	BUFFER_STATUS[1:0]	Buffered mode: buffer status (last written buffer): 00: 1st buffer 01: 2nd buffer 10: 3rd buffer 11: (no buffer written) Mailbox mode: reserved
3	MAILBOX_STATUS	Mailbox mode: mailbox status: 0: Mailbox empty 1: Mailbox full Buffered mode: reserved
2	Reserved	Must be kept at reset value.
1	READ_INT	Interrupt Read: 1: Interrupt after buffer was completely and successfully read 0: Interrupt cleared after first byte of buffer was written NOTE: This interrupt is signalled to the writing side if enabled in the SM Control register.
0	WRITE_INT	Interrupt Write: 1: Interrupt after buffer was completely and successfully written 0: Interrupt cleared after first byte of buffer was read NOTE: This interrupt is signalled to the reading side if enabled in the SM Control register.

55.5.70. Activate SyncManager X register (SMX_ACTIVE) (X = 0…F)

Address Offset: 0x0806 + X * 8
Reset value: 0x00

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
LATCH_PDI	LATCH_ECAT	Reserved				REQUEST	SM_EN
rw(ECAT) r(PDI)	rw(ECAT) r(PDI)					rw(ECAT) r(PDI)	r(ECAT) rw(PDI)

Bits	Fields	Descriptions
7	LATCH_PDI	Latch Event PDI: 0: No 1: Generate Latch events when PDI issues a buffer exchange or when PDI

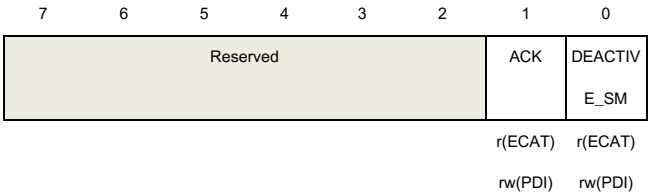
		accesses buffer start address
6	LATCH_ECAT	Latch Event ECAT: 0: No 1: Generate Latch event when EtherCAT main issues a buffer exchange
5:2	Reserved	Must be kept at reset value.
1	REQUEST	Repeat Request: A toggle of Repeat Request means that a mailbox retry is needed (primarily used in conjunction with ECAT Read Mailbox)
0	SM_EN	SyncManager Enable/Disable: 0: Disable: Access to Memory without SyncManager control 1: Enable: SyncManager is active and controls Memory area set in configuration NOTE: PDI register function acknowledge by Write command is disabled: Reading this register from PDI in all SyncManagers which have changed activation clears AL Event Request 0x0220[4]. Writing to this register from PDI is not possible.

55.5.71. PDI Control SyncManager X register (SMX_PDI_CTL) (X = 0…F)

Address Offset: 0x0807 + X * 8

Reset value: 0x00

This register can be accessed by byte(8-bit).



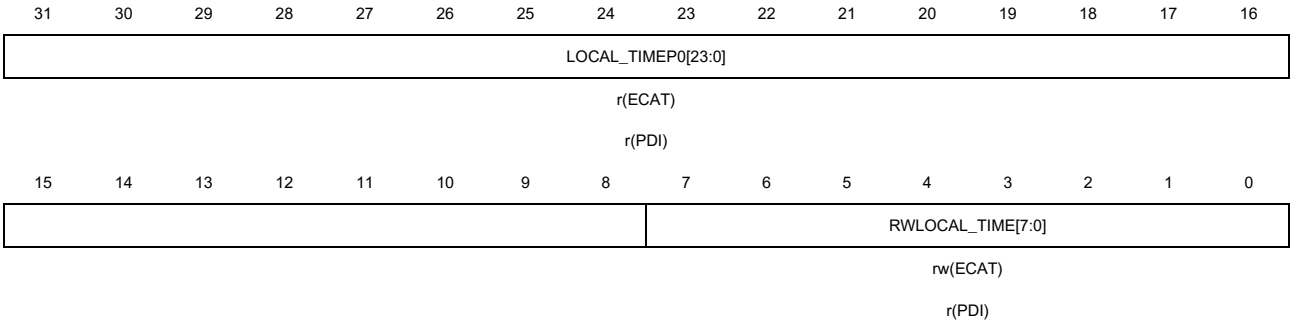
Bits	Fields	Descriptions
7:2	Reserved	Must be kept at reset value.
1	ACK	Repeat Ack: If this is set to the same value as that set by Repeat Request, the PDI acknowledges the execution of a previous set Repeat request.
0	DEACTIVE_SM	Deactivate SyncManager: Read: 0: Normal operation, SyncManager activated. 1: SyncManager deactivated and reset. SyncManager locks access to Memory area. Write: 0: Activate SyncManager 1: Request SyncManager deactivation

NOTE: Writing 1 is delayed until the end of the frame, which is currently processed.

55.5.72. ESC Receive time port 0 register (ESC_RCV_TIMEP0)

Address Offset: 0x0900
Reset value: undefined

This register can be accessed by word(32-bit).



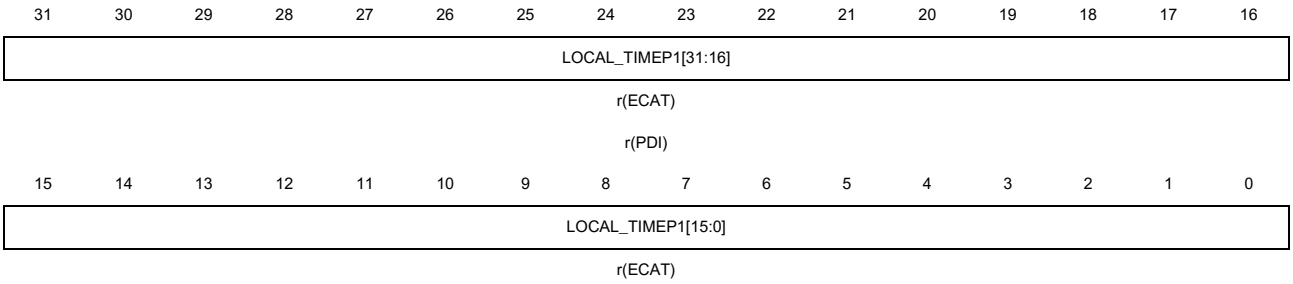
Bits	Fields	Descriptions
31:8	LOCAL_TIMEP0[23:0]	Local time at the beginning of the last receive frame containing a write access to register 0x0900.
7:0	RWLOCAL_TIME[7:0]	Write: A write access to register 0x0900 with BWR or FPWR latches the local time at the beginning of the receive frame (start first bit of preamble) at each port. Read: Local time at the beginning of the last receive frame containing a write access to this register. NOTE: FPWR requires an address match for accessing this register like any FPWR command. All write commands with address match will increment the working counter (e.g., APWR), but they will not trigger receive time latching.

NOTE: The time stamps cannot be read in the same frame in which this register was written.

55.5.73. ESC Receive time port 1 register (ESC_RCV_TIMEP1)

Address Offset: 0x0904
Reset value: undefined

This register can be accessed by word(32-bit).



r(PDI)

Bits	Fields	Descriptions
31:0	LOCAL_TIMEP1[31:0]	Local time at the beginning of a frame (start first bit of preamble) received at port 1 containing a BWR or FPWR to register 0x0900.

55.5.74. ESC Receive time port 2 register (ESC_RCV_TIMEP2)

Address Offset: 0x0908
Reset value: undefined

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
LOCAL_TIMEP2[31:16]															
r(ECAT)															
r(PDI)															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LOCAL_TIMEP2[15:0]															
r(ECAT)															
r(PDI)															

Bits	Fields	Descriptions
31:0	LOCAL_TIMEP2[31:0]	Local time at the beginning of a frame (start first bit of preamble) received at port 2 containing a BWR or FPWR to register 0x0900.

55.5.75. ESC System time register (ESC_SYS_TIME)

Address Offset: 0x0910
Reset value: 0x0000 0000 0000 0000

This register can be accessed by word(32-bit).

63	..	..	..	..	..	..	..	31	..	..	..	..	..	..	0
READ_ACESS[31:0]								WRITE_ACESS[31:0]							
r(ECAT)								w(ECAT)							
r(PDI)								w(PDI)							

Bits	Fields	Descriptions
63:0	READ_ACESS[63:0]	ECAT read access: Local copy of the System Time when the frame passed the reference clock (i.e., including System Time Delay). Time latched at beginning of the frame (Ethernet SOF delimiter) PDI read access: Local copy of the System Time. Time latched when reading first byte (0x0910)

31:0	WRITE_ACESS[31:0]	Write access: Written value will be compared with the local copy of the System time. The result is an input to the time control loop. NOTE: written value will be compared at the end of the frame with the latched (SOF) local copy of the System time if at least the first byte (0x0910) was written.

NOTE: Write access to this register depends upon ESC configuration (System Time PDI-controlled off=ECAT/ on=PDI; ECAT control is common).

55.5.76. ESC Receive time ECAT processing unit register (ESC_RCVT_ECAT_PU)

Address Offset: 0x0918
Reset value: undefined

This register can be accessed by word(32-bit).

63 .. .. 31 0
RECVE_TIME[63:0]
r(ECAT)
r(PDI)

Bits	Fields	Descriptions
63:0	RECVE_TIME[63:0]	Local time at the beginning of a frame (start first bit of preamble) received at the ECAT Processing Unit containing a write access to register 0x0900 NOTE: if port 0 is open, this register reflects the Receive Time Port 0 as a 64 Bit value. Any valid EtherCAT write access to register 0x0900 triggers latching, not only BWR/FPWR commands as with register 0x0900.

55.5.77. ESC System time offset register (ESC_SYS_TIME_OFFSET)

Address Offset: 0x0920
Reset value: 0x0000 0000 0000 0000

This register can be accessed by byte(8-bit), half-word(16-bit) and word(32-bit).

63 .. .. 31 0
OFFSET_TIME[63:0]
rw(ECAT)
rw(PDI)

Bits	Fields	Descriptions
63:0	OFFSET_TIME[63:0]	Difference between local time and System Time. Offset is added to the local time.

55.5.78. ESC System time delay register (ESC_SYS_TIME_DELAY)

Address Offset: 0x0928

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
DELAY_TIME[31:0]															
rw(ECAT)															
rw(PDI)															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DELAY_TIME[31:0]															
rw(ECAT)															
rw(PDI)															

Bits	Fields	Descriptions
31:0	DELAY_TIME[31:0]	Delay between Reference Clock and the ESC

55.5.79. ESC System time difference register (ESC_SYS_TIME_DIFF)

Address Offset: 0x092C

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
TIME_COMPARE	DIFF_TIME[30:0]														
r(ECAT) r(PDI)	r(ECAT) r(PDI)														
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIFF_TIME[30:0]															
r(ECAT)															
r(PDI)															

Bits	Fields	Descriptions
31	TIME_COMPARE	0: Local copy of System Time less than received System Time 1: Local copy of System Time greater than or equal to received System Time
30:0	DIFF_TIME[30:0]	Mean difference between local copy of System Time and received System Time values Difference = Received System Time - local copy of System Time

NOTE: Register bits [31:8] are internally latched (ECAT/PDI independently) when bits [7:0] are read, which guarantees reading a consistent value.

55.5.80. ESC Speed counter start register (ESC_SPD_CNT_START)

Address Offset: 0x0930
Reset value: 0x1000

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	BWIDTH[14:0]														
	rw(ECAT)														
	rw(PDI)														

Bits	Fields	Descriptions
15	Reserved	Must be kept at reset value.
14:0	BWIDTH[14:0]	Bandwidth for adjustment of local copy of System Time (larger values -> smaller bandwidth and smoother adjustment) A write access resets System Time Difference (0x092C:0x092F) and Speed Counter Diff (0x0932:0x0933). Valid values: 0x0080 to 0x3FFF

NOTE: Write access to this register depends upon ESC configuration (System Time PDI-controlled off=ECAT / on=PDI; ECAT control is common).

55.5.81. ESC Speed counter diff register (ESC_SPD_CNT_DIFF)

Address Offset: 0x0932
Reset value: 0x0000

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIFF_CLOCK[15:0]															
r(ECAT)															
r(PDI)															

Bits	Fields	Descriptions
15:0	DIFF_CLOCK[15:0]	Representation of the deviation between local clock period and Reference Clock's clock period (representation: two's complement) Range: +/- (Speed Counter Start - 0x7F)

NOTE: The clock deviation after System Time Difference has settled at a low value can be calculated as follows:

Deviation = Speed Counter Diff / 5(Speed Counter Start + Speed Counter Diff + 2)(Speed Counter Start - Speed Counter Diff + 2)

55.5.82. ESC System time difference filter depth register (ESC_SYS_TIME_DIFF_FD)

Address Offset: 0x0934
Reset value: 0x04

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
Reserved				FILTER_DAPTH[3:0]			
rw(ECAT)							
rw(PDI)							

Bits	Fields	Descriptions
7:4	Reserved	Must be kept at reset value.
3:0	FILTER_DAPTH[3:0]	Filter depth for averaging the received System Time deviation

NOTE: Write access to this register depends upon ESC configuration (System Time PDI-controlled off=ECAT/ on=PDI; ECAT control is common).

55.5.83. ESC Speed counter filter depth register (ESC_SPD_CNT_FD)

Address Offset: 0x0935
Reset value: 0x12

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
Reserved				FILTER_DAPTH[3:0]			
rw(ECAT)							
rw(PDI)							

Bits	Fields	Descriptions
7:4	Reserved	Must be kept at reset value.
3:0	FILTER_DAPTH[3:0]	Filter depth for averaging the clock period deviation

NOTE: Write access to this register depends upon ESC configuration (System Time PDI-controlled off=ECAT/ on=PDI; ECAT control is common).

55.5.84. ESC Cyclic unit control register (ESC_CYC_UNIT_CTL)

Address Offset: 0x0980
Reset value: 0x00

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
Reserved		LATCH_UNIT1	LATCH_UNIT0	Reserved			UNIT_CTL
		rw(ECAT) r(PDI)	rw(ECAT) r(PDI)				rw(ECAT) r(PDI)

Bits	Fields	Descriptions
7:6	Reserved	Must be kept at reset value.
5	LATCH_UNIT1	Latch In unit 1: 0: ECAT-controlled 1: PDI-controlled NOTE: Latch interrupt is routed to ECAT/PDI depending on this setting
4	LATCH_UNIT0	Latch In unit 0: 0: ECAT-controlled 1: PDI-controlled NOTE: Latch interrupt is routed to ECAT/PDI depending on this setting. Always 1 (PDI-controlled) if System Time is PDI-controlled.
3:1	Reserved	Must be kept at reset value.
0	UNIT_CTL	Cyclic Unit and SYNC0 out unit control: 0: ECAT-controlled 1: PDI-controlled

55.5.85. ESC Register activation register (ESC_REG_ACT)

Address Offset: 0x0981
Reset value: 0x00

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
SYNCSIGNAL	CONFIG	START_TIME	EXTEN	AUTO_ACTIVE	SYNC1	SYNC0	SYNC_OUT
rw(ECAT) rw(PDI)	rw(ECAT) rw(PDI)	rw(ECAT) rw(PDI)	rw(ECAT) rw(PDI)	rw(ECAT) rw(PDI)	rw(ECAT) rw(PDI)	rw(ECAT) rw(PDI)	rw(ECAT) rw(PDI)

Bits	Fields	Descriptions
7	SYNCSIGNAL	SyncSignal debug pulse (Vasily bit): 0: Deactivated 1: Immediately generate one ping only on SYNC0-1 according to 0x0981[2:1 for debugging This bit is self-clearing, always read 0. All pulses are generated at the same time, the cycle time is ignored. The configured

		pulse length is used.
6	CONFIG	Near future configuration (2442pprox..): 0: ½ DC width future (2^{31} ns or 2^{63} ns) 1: ~2.1 sec. future (2^{31} ns)
5	START_TIME	Start Time plausibility check: 0: Disabled. SyncSignal generation if Start Time is reached. 1: Immediate SyncSignal generation if Start Time is outside near future (see 0x0981[6])
4	EXTEN	Extension of Start Time Cyclic Operation (0x0990:0x0993): 0: No extension 1: Extend 32 bit written Start Time to 64 bit
3	AUTO_ACTIVE	Auto-activation by writing Start Time Cyclic Operation (0x0990:0x0997): 0: Disabled 1: Auto-activation enabled. 0x0981[0] is set automatically after Start Time is written.
2	SYNC1	SYNC1 generation: 0: Deactivated 1: SYNC1 pulse is generated
1	SYNC0	SYNC0 generation: 0: Deactivated 1: SYNC0 pulse is generated
0	SYNC_OUT	Sync Out Unit activation: 0: Deactivated 1: Activated

NOTE: Write to this register depends upon setting of 0x0980[0]

55.5.86. ESC Pulse length of sync signals register (ESC_PULSE_LEN)

Address Offset: 0x0982

Reset value: depends on configuration

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PLEN_SM[15:0]															
r(ECAT)															
r(PDI)															

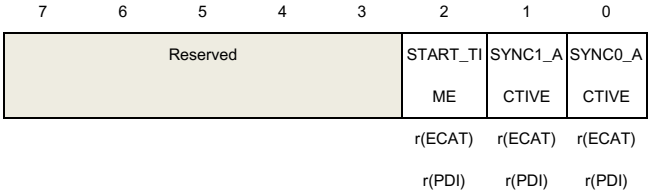
Bits	Fields	Descriptions
15:0	PLEN_SM[15:0]	Pulse length of SyncSignals (in Units of 10ns) 0: Acknowledge mode: SyncSignal will be cleared by reading SYNC[1:0] Status

register

55.5.87. ESC Activation status register (ESC_ACT_STAT)

Address Offset: 0x0984
Reset value: 0x00

This register can be accessed by byte(8-bit).

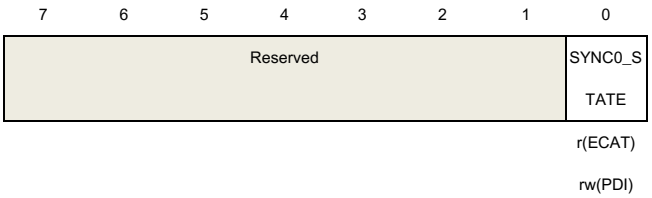


Bits	Fields	Descriptions
7:3	Reserved	Must be kept at reset value.
2	START_TIME	Start Time Cyclic Operation (0x0990:0x0997) plausibility check result when Sync Out Unit was activated: 0: Start Time was within near future 1: Start Time was out of near future (0x0981[6])
1	SYNC1_ACTIVE	SYNC1 activation state: 0: First SYNC1 pulse is not pending 1: First SYNC1 pulse is pending
0	SYNC0_ACTIVE	SYNC0 activation state: 0: First SYNC0 pulse is not pending 1: First SYNC0 pulse is pending

55.5.88. ESC SYNC0 status register (ESC_SYNC0_STAT)

Address Offset: 0x098E
Reset value: 0x00

This register can be accessed by byte(8-bit).



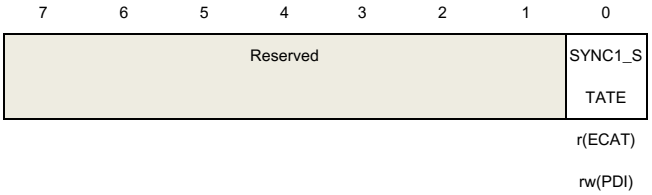
Bits	Fields	Descriptions
7:1	Reserved	Must be kept at reset value.

0	SYNC0_STATE	SYNC0 state for Acknowledge mode. SYNC0 in Acknowledge mode is cleared by reading this register from PDI, use only in Acknowledge mode NOTE: PDI register function acknowledge by Write command is disabled: Reading this register from PDI clears AL Event Request 0x0220[2]. Writing to this register from PDI is not possible. PDI register function acknowledge by Write command is enabled: Writing this register from PDI clears AL Event Request 0x0220[2]. Writing to this register from PDI is possible; write value is ignored (write 0).

55.5.89. ESC SYNC1 status register (ESC_SYNC1_STAT)

Address Offset: 0x098F
Reset value: 0x00

This register can be accessed by byte(8-bit).

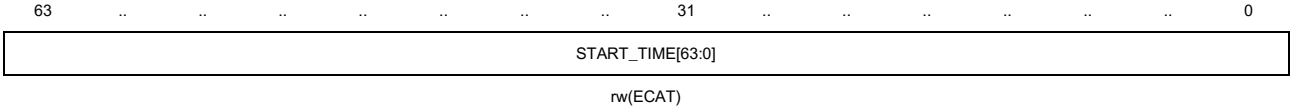


Bits	Fields	Descriptions
7:1	Reserved	Must be kept at reset value.
0	SYNC1_STATE	SYNC1 state for acknowledge mode. SYNC1 in acknowledge mode is cleared by reading this register from PDI, use only in Acknowledge mode NOTE: PDI register function acknowledge by Write command is disabled: Reading this register from PDI clears AL Event Request 0x0220[3]. Writing to this register from PDI is not possible. PDI register function acknowledge by Write command is enabled: Writing this register from PDI clears AL Event Request 0x0220[3]. Writing to this register from PDI is possible; write value is ignored (write 0).

55.5.90. ESC Start time cyclic operation register (ESC_START_TIME_CO)

Address Offset: 0x0990
Reset value: 0x0000 0000 0000 0000

This register can be accessed by word(32-bit).



rw(PDI)

Bits	Fields	Descriptions
63:0	START_TIME[63:0]	Write: Start time (System time) of cyclic operation in ns Read: System time of next SYNC0 pulse in ns

NOTE: Register bits [63:8] are internally latched (ECAT/PDI independently) when bits [7:0] are read, which guarantees reading a consistent value.

55.5.91. ESC Next SYNC1 pulse register (ESC_NEXT_SYNC1_PULSE)

Address Offset: 0x0998
Reset value: 0x0000 0000 0000 0000

This register can be accessed by word(32-bit).

63	31 0
NEXT_SYNC1[63:0]	
r(ECAT)	
r(PDI)	

Bits	Fields	Descriptions
63:0	NEXT_SYNC1[63:0]	System time of next SYNC1 pulse in ns

NOTE: Register bits [63:8] are internally latched (ECAT/PDI independently) when bits [7:0] are read, which guarantees reading a consistent value.

55.5.92. ESC SYNC0 cycle time register (ESC_SYNC0_CYC_TIME)

Address Offset: 0x09A0
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
SYNC0_CYCLE[31:16]															
rw(ECAT)															
rw(PDI)															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYNC0_CYCLE[15:0]															
rw(ECAT)															
rw(PDI)															

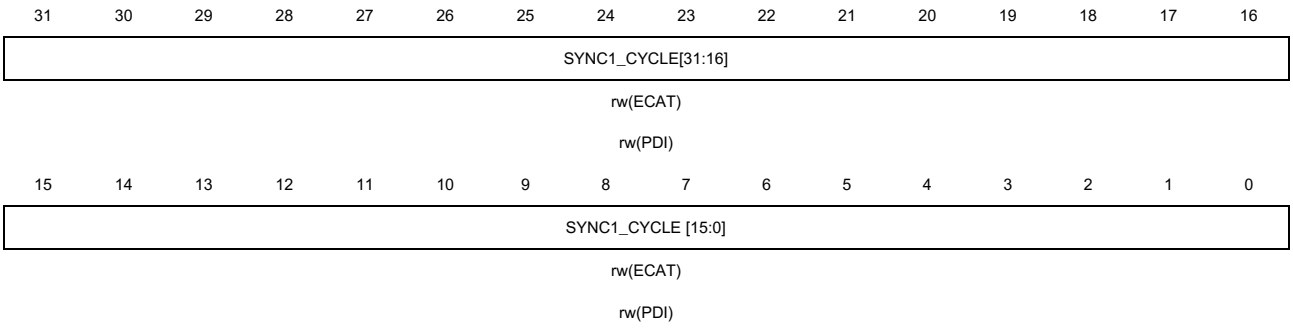
Bits	Fields	Descriptions
31:0	SYNC0_CYCLE[31:0]	Time between two consecutive SYNC0 pulses in ns. 0: Single shot mode, generate only one SYNC0 pulse. NOTE: Write to this register depends upon setting of 0x0980[0]. Minimum value

for cyclic operation: 60 [ns].

55.5.93. ESC SYNC1 cycle time register (ESC_SYNC1_CYC_TIME)

Address Offset: 0x09A4
Reset value: 0x0000 0000

This register can be accessed by word(32-bit).

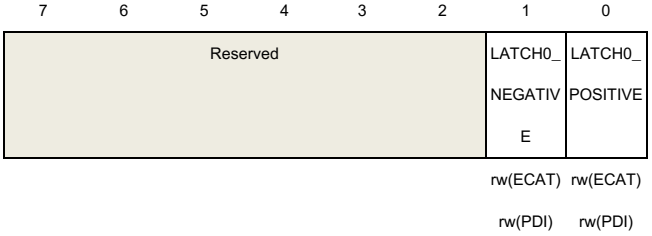


Bits	Fields	Descriptions
31:0	SYNC1_CYCLE[31:0]	Time between SYNC0 pulse and SYNC1 pulse in ns NOTE: Write to this register depends upon setting of 0x0980[0].

55.5.94. ESC Latch0 control register (ESC_LATCH0_CTL)

Address Offset: 0x09A8
Reset value: 0x00

This register can be accessed by byte(8-bit).



Bits	Fields	Descriptions
7:2	Reserved	Must be kept at reset value.
1	LATCH0_NEGATIVE	Latch0 negative edge: 0: Continuous Latch active 1: Single event (only first event active)
0	LATCH0_POSITIVE	Latch0 positive edge: 0: Continuous Latch active 1: Single event (only first event active)

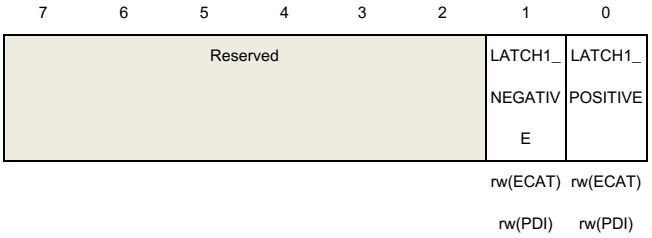
NOTE: Write access depends upon setting of 0x0980[4].

55.5.95. ESC Latch1 control register (ESC_LATCH1_CTL)

Address Offset: 0x09A9

Reset value: 0x00

This register can be accessed by byte(8-bit).



Bits	Fields	Descriptions
7:2	Reserved	Must be kept at reset value.
1	LATCH1_NEGATIVE	Latch1 negative edge: 0: Continuous Latch active 1: Single event (only first event active)
0	LATCH1_POSITIVE	Latch1 positive edge: 0: Continuous Latch active 1: Single event (only first event active)

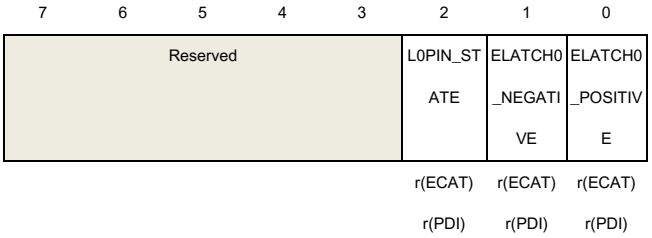
NOTE: Write access depends upon setting of 0x0980[5].

55.5.96. ESC Latch0 status register (ESC_LATCH0_STAT)

Address Offset: 0x09AE

Reset value: 0x00

This register can be accessed by byte(8-bit).



Bits	Fields	Descriptions
7:3	Reserved	Must be kept at reset value.
2	L0PIN_STATE	Latch0 pin state
1	ELATCH0_NEGATIV	Event Latch0 negative edge.

	E	0: Negative edge not detected or continuous mode 1: Negative edge detected in single event mode only. Flag cleared by reading out Latch0 Time Negative Edge.
0	ELATCH0_POSITIVE	Event Latch0 positive edge. 0: Positive edge not detected or continuous mode 1: Positive edge detected in single event mode only. Flag cleared by reading out Latch0 Time Positive Edge.

55.5.97. ESC Latch1 status register (ESC_LATCH1_STAT)

Address Offset: 0x09AF

Reset value: 0x00

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
Reserved					L1PIN_STATE	ELATCH1_NEGATIVE	ELATCH1_POSITIVE
					r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)

Bits	Fields	Descriptions
7:3	Reserved	Must be kept at reset value.
2	L1PIN_STATE	Latch0 pin state
1	ELATCH1_NEGATIV E	Event Latch1 negative edge. 0: Negative edge not detected or continuous mode 1: Negative edge detected in single event mode only. Flag cleared by reading out Latch1 Time Negative Edge.
0	ELATCH1_POSITIVE	Event Latch1 positive edge. 0: Positive edge not detected or continuous mode 1: Positive edge detected in single event mode only. Flag cleared by reading out Latch1 Time Positive Edge.

55.5.98. ESC Latch0 time positive edge register (ESC_LATCH0_TIME_PE)

Address Offset: 0x09B0

Reset value: 0x0000 0000 0000 0000

This register can be accessed by word(32-bit).

63	31 0
L0_POSITIVE[63:0]	

r(ECAT)

rw(PDI)

Bits	Fields	Descriptions
63:0	L0_POSITIVE[63:0]	System time at the positive edge of the Latch0 signal. NOTE: PDI register function acknowledge by Write command is disabled: Reading this register from PDI if 0x0980[4]=1 clears Latch0 Status 0x09AE[0]. Writing to this register from PDI is not possible. PDI register function acknowledge by Write command is enabled: Writing this register from PDI if 0x0980[4]=1 clears Latch0 Status 0x09AE[0]. Writing to this register from PDI is possible; write value is ignored (write 0).

NOTE: Register bits [63:8] are internally latched (ECAT/PDI independently) when bits [7:0] are read, which guarantees reading a consistent value. Reading this register from ECAT clears Latch0 Status 0x09AE[0] if 0x0980[4]=0. Writing to this register from ECAT is not possible.

55.5.99. ESC_ESC Latch0 time negative edge register (ESC_LATCH0_TIME_NE)

Address Offset: 0x09B8

Reset value: 0x0000 0000 0000 0000

This register can be accessed by word(32-bit).

63	31 0
L0_NEGATIVE[63:0]	

r(ECAT)

rw(PDI)

Bits	Fields	Descriptions
63:0	L0_NEGATIVE[63:0]	System time at the negative edge of the Latch0 signal. NOTE: PDI register function acknowledge by Write command is disabled: Reading this register from PDI if 0x0980[4]=1 clears Latch0 Status 0x09AE[1]. Writing to this register from PDI is not possible. PDI register function acknowledge by Write command is enabled: Writing this register from PDI if 0x0980[4]=1 clears Latch0 Status 0x09AE[1]. Writing to this register from PDI is possible; write value is ignored (write 0).

NOTE: Register bits [63:8] are internally latched (ECAT/PDI independently) when bits [7:0] are read, which guarantees reading a consistent value. Reading this register from ECAT clears Latch0 Status 0x09AE[1] if 0x0980[4]=0. Writing to this register from ECAT is not possible.

55.5.100. ESC Latch1 time positive edge register (ESC_LATCH1_TIME_PE)

Address Offset: 0x09C0
Reset value: 0x0000 0000 0000 0000

This register can be accessed by word(32-bit).

63 .. .. 31 0
L1_POSITIVE[63:0]
r(ECAT)
rw(PDI)

Bits	Fields	Descriptions
63:0	L1_POSITIVE[63:0]	System time at the positive edge of the Latch1 signal. NOTE: PDI register function acknowledge by Write command is disabled: Reading this register from PDI if 0x0980[5]=1 clears Latch0 Status 0x09AF[0]. Writing to this register from PDI is not possible. PDI register function acknowledge by Write command is enabled: Writing this register from PDI if 0x0980[5]=1 clears Latch0 Status 0x09AF[0]. Writing to this register from PDI is possible; write value is ignored (write 0).

NOTE: Register bits [63:8] are internally latched (ECAT/PDI independently) when bits [7:0] are read, which guarantees reading a consistent value. Reading this register from ECAT clears Latch0 Status 0x09AF[0] if 0x0980[5]=0. Writing to this register from ECAT is not possible.

55.5.101. ESC Latch1 time negative edge register (ESC_LATCH1_TIME_NE)

Address Offset: 0x09C8
Reset value: 0x0000 0000 0000 0000

This register can be accessed by word(32-bit).

63 .. .. 31 0
L1_NEGATIVE[63:0]
r(ECAT)
rw(PDI)

Bits	Fields	Descriptions
63:0	L1_NEGATIVE[63:0]	System time at the negative edge of the Latch1 signal. NOTE: PDI register function acknowledge by Write command is disabled: Reading this register from PDI if 0x0980[5]=1 clears Latch0 Status 0x09AF[1]. Writing to this register from PDI is not possible. PDI register function acknowledge by Write command is enabled: Writing this register from PDI if 0x0980[5]=1 clears Latch0 Status 0x09AF[1]. Writing to this register from PDI is possible; write value is ignored (write 0).

NOTE: Register bits [63:8] are internally latched (ECAT/PDI independently) when bits [7:0] are read, which guarantees reading a consistent value. Reading this register from ECAT clears Latch0 Status 0x09AF[1] if 0x0980[5]=0. Writing to this register from ECAT is not possible.

55.5.102. ESC Buffer change event time register (ESC_BUF_CET)

Address Offset: 0x09F0
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
EVENT_TIME[31:16]															
r(ECAT)															
r(PDI)															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVENT_TIME[15:0]															
r(ECAT)															
r(PDI)															

Bits	Fields	Descriptions
31:0	EVENT_TIME[31:0]	Local time at the beginning of the frame which causes at least one SyncManager to assert an ECAT event

NOTE: Register bits [31:8] are internally latched (ECAT/PDI independently) when bits [7:0] are read, which guarantees reading a consistent value.

55.5.103. ESC PDI buffer start event time register (ESC_PDI_BUF_SET)

Address Offset: 0x09F8
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
SEVENT_TIME[31:16]															
r(ECAT)															
r(PDI)															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEVENT_TIME[15:0]															
r(ECAT)															
r(PDI)															

Bits	Fields	Descriptions
31:0	PDI_SEVENT_TIME[	Local time when at least one SyncManager asserts a PDI buffer start event

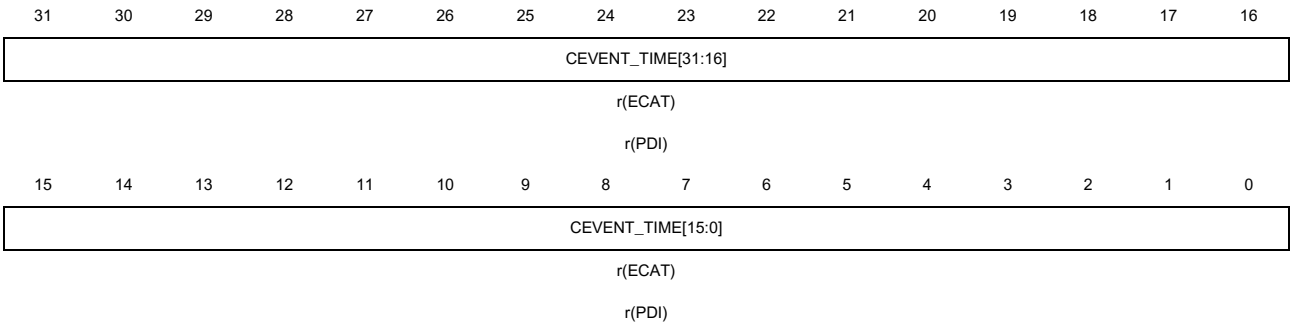
31:0]

NOTE: Register bits [31:8] are internally latched (ECAT/PDI independently) when bits [7:0] are read, which guarantees reading a consistent value.

55.5.104. ESC PDI buffer change event time register (ESC_PDI_BUF_CET)

Address Offset: 0x09FC
Reset value: 0x0000 0000

This register can be accessed by word(32-bit)



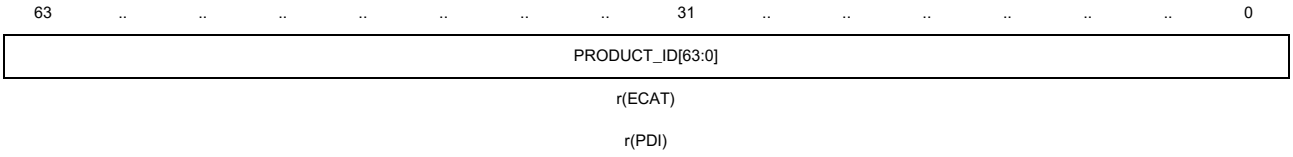
Bits	Fields	Descriptions
31:0	PDI_CEVENT_TIME[31:0]	Local time when at least one SyncManager asserts a PDI buffer change event

NOTE: Register bits [31:8] are internally latched (ECAT/PDI independently) when bits [7:0] are read, which guarantees reading a consistent value.

55.5.105. ESC Product ID register (ESC_PRODUCT_ID)

Address Offset: 0x0E00
Reset value: 0x0000 0000 0000 0000

This register can be accessed by word(32-bit).



Bits	Fields	Descriptions
63:0	PRODUCT_ID[63:0]	Product ID

55.5.106. ESC Vendor ID register (ESC_VENDOR_ID)

Address Offset: 0x0E08
Reset value: 0x0000 0000 0000 0000

This register can be accessed by word(32-bit).

63	31 0
Reserved	VENDOR_ID[31:0]
	r(ECAT)
	r(PDI)

Bits	Fields	Descriptions
63:32	Reserved	Must be kept at reset value.
31:0	VENDOR_ID[31:0]	Vendor ID: [23:0] Company [31:24] Department NOTE: Test Vendor IDs have [31:28]=0xE

55.5.107. ESC Digital I/O output data register (ESC_DIG_DATA)

Address Offset: 0x0F00
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
OUTPUT_DATA[31:0]
rw(ECAT)
r(PDI)

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
OUTPUT_DATA[31:0]
rw(ECAT)
r(PDI)

Bits	Fields	Descriptions
31:0	OUTPUT_DATA[31:0]	Output Data

NOTE: Register size depends on PDI setting and/or device configuration. This register is bit-writable (using Logical addressing).

55.5.108. ESC General purpose outputs register (ESC_GP_OUTPUT)

Address Offset: 0x0F10
Reset value: 0x0000 0000 0000 0000

This register can be accessed by byte(8-bit), half-word(16-bit) and word(32-bit)

63	31 0
GP_OUTPUT[63:0]	
rw(ECAT)	

rw(PDI)

Bits	Fields	Descriptions
63:0	GP_OUTPUT [63:0]	General Purpose Output Data

NOTE: Register size depends on PDI setting and/or device configuration

55.5.109. ESC General purpose inputs register (ESC_GP_INPUT)

Address Offset: 0x0F18

Reset value: 0x0000 0000 0000 0000

This register can be accessed by word(32-bit).

63 .. .. 31 0
GP_INPUT[63:0]

r(ECAT)

r(PDI)

Bits	Fields	Descriptions
63:0	GP_INPUT[63:0]	General Purpose Input Data

NOTE: Register size depends on PDI setting and/or device configuration

55.5.110. ESC User RAM register (ESC_USER_RAM)

Address Offset: 0x0F80

Reset value: undefined

This register can be accessed by byte(8-bit), half-word(16-bit) and word(32-bit).

1023 0
USER_RAM[1023:0]

rw(ECAT)

rw(PDI)

Bits	Fields	Descriptions
----	USER_RAM[1023:0]	Application-specific information(128 Bytes)

55.5.111. ESC Process data RAM register (ESC_PDRAM)

Address Offset: 0x1000

Reset value: undefined

This register can be accessed by byte(8-bit), half-word(16-bit) and word(32-bit)

8191 0

PDRAM[8191:0]
rw(ECAT)
rw(PDI)

Bits	Fields	Descriptions
----	PDRAM[8191:0]	Process Data RAM(8KB)

NOTE: Process Data RAM is only accessible if EEPROM was correctly loaded (register 0x0110[0] = 1).

56. Appendix

56.1. List of abbreviations used in register

Table 56-1. List of abbreviations used in register

abbreviations for registers	Descriptions
read/write (rw)	Software can read and write to this bit.
Read-only ®	Software can only read this bit.
Write-only (w)	Software can only write to this bit. Reading this bit returns the reset value.
Read/clear write 1 (rc_w1)	Software can read as well as clear this bit by writing 1. Writing 0 has no effect on the bit value.
Read/clear write 0 (rc_w0)	Software can read as well as clear this bit by writing 0. Writing 1 has no effect on the bit value.
Toggle (t)	The software can toggle this bit by writing 1. Writing 0 has no effect.
Read/set (rs)	Software can read as well as set this bit to 1. Writing '0' has no effect on the bit value.
Read/clear by read (rc_r)	Software can read this bit. Reading this bit automatically clears it to '0'. Writing '0' has no effect on the bit value.

56.2. List of terms

Table 56-2. List of terms

Glossary	Descriptions
Word	Data of 32-bit length.
Half-word	Data of 16-bit length.
Byte	Data of 8-bit length.
IAP (in-application programming)	Writing 0 has no effect IAP is the ability to re-program the Flash memory of a microcontroller while the user program is running.
ICP (in-circuit programming)	ICP is the ability to program the Flash memory of a microcontroller using the JTAG protocol, the SWD protocol or the boot loader while the device is mounted on the user application board.
Option bytes	Product configuration bits stored in the Flash memory.
AHB	Advanced high-performance bus.
APB	Advanced peripheral bus.
RAZ	Read-as-zero.
WI	Writes ignored.
RAZ/WI	Read-as-zero, writes ignored.
NVMC	Non-Volatile Memory Controller

Glossary	Descriptions
CNVM	Code Non-Volatile Memory
ECNVM	Extend Code Non-Volatile Memory
OTP	One-Time Programmable
OTA	Over-the-Air

56.3. Available peripherals

For availability of peripherals and their number across all MCU series types, refer to the corresponding device data datasheet.

57. Revision history

Table 57-1. Revision history

Revision No.	Description	Date
1.0	Initial Release	Nov. 9, 2026

Important Notice

This document is the property of GigaDevice Semiconductor Inc. and its subsidiaries (the "Company"). This document, including any product of the Company described in this document (the "Product"), is owned by the Company according to the laws of the People's Republic of China and other applicable laws. The Company reserves all rights under such laws and no Intellectual Property Rights are transferred (either wholly or partially) or licensed by the Company (either expressly or impliedly) herein. The names and brands of third party referred thereto (if any) are the property of their respective owner and referred to for identification purposes only.

To the maximum extent permitted by applicable law, the Company makes no representations or warranties of any kind, express or implied, with regard to the merchantability and the fitness for a particular purpose of the Product, nor does the Company assume any liability arising out of the application or use of any Product. Any information provided in this document is provided only for reference purposes. It is the sole responsibility of the user of this document to determine whether the Product is suitable and fit for its applications and products planned, and properly design, program, and test the functionality and safety of its applications and products planned using the Product. The Product is designed, developed, and/or manufactured for ordinary business, industrial, personal, and/or household applications only, and the Product is not designed or intended for use in (i) safety critical applications such as weapons systems, nuclear facilities, atomic energy controller, combustion controller, aeronautic or aerospace applications, traffic signal instruments, pollution control or hazardous substance management; (ii) life-support systems, other medical equipment or systems (including life support equipment and surgical implants); (iii) automotive applications or environments, including but not limited to applications for active and passive safety of automobiles (regardless of front market or aftermarket), for example, EPS, braking, ADAS (camera/fusion), EMS, TCU, BMS, BSG, TPMS, Airbag, Suspension, DMS, ICMS, Domain, ESC, DCDC, e-clutch, advanced-lighting, etc.. Automobile herein means a vehicle propelled by a self-contained motor, engine or the like, such as, without limitation, cars, trucks, motorcycles, electric cars, and other transportation devices; and/or (iv) other uses where the failure of the device or the Product can reasonably be expected to result in personal injury, death, or severe property or environmental damage (collectively "Unintended Uses"). Customers shall take any and all actions to ensure the Product meets the applicable laws and regulations. The Company is not liable for, in whole or in part, and customers shall hereby release the Company as well as its suppliers and/or distributors from, any claim, damage, or other liability arising from or related to all Unintended Uses of the Product. Customers shall indemnify and hold the Company, and its officers, employees, subsidiaries, affiliates as well as its suppliers and/or distributors harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of the Product.

Information in this document is provided solely in connection with the Product. The Company reserves the right to make changes, corrections, modifications or improvements to this document and the Product described herein at any time without notice. The Company shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. Information in this document supersedes and replaces information previously supplied in any prior versions of this document.

© 2026 GigaDevice Semiconductor Inc. - All rights reserved