ARM®-based 32-bit Cortex®-M4F MCU + FPU
with 256 to 1024 KB Flash, sLib, USB, 2 CANs,
17 timers, 3 ADCs, 21 communication interfaces MCU

HT32F49365/HT32F49395
User Manual

Revision: V1.10 Date: August 21, 2026

www.holtek.com

Table of Contents

1 System architecture 23
Feature 25
System overview 26
ARM Cortex™-M4F processor 26
Bit band 27
Interrupt and exception vectors 30
System Tick (SysTick) 32
Reset 32
List of abbreviations for registers 34
Device characteristics information 34
Flash memory size register 34
Device electronic signature 35

2 Memory resources 36
Internal memory address map 36
Flash memory 37
SRAM memory 38
Peripheral address map 39

3 Power control (PWC) 41
Introduction 41
Main Features 42
POR/LVR 42
Power voltage monitor (PVM) 42
Power domain 43
Power saving modes 44
PWC registers 46
Power control register (PWC_CTRL) 46
Power control/status register (PWC_CTRLSTS) 47

4 Clock and reset manage (CRM) 48
Clock 48
Clock sources 49
System clock 49
Peripheral clock 50
Clock fail detector 50
Auto step-by-step system clock switch 50
Internal clock output 50
Interrupts 51
Reset 51
System reset 51

Battery powered domain reset 52
CRM registers 52
Clock control register (CRM_CTRL) 52
Clock configuration register (CRM_CFG) 54
Clock interrupt register (CRM_CLKINT) 56
APB2 peripheral reset register (CRM_APB2RST) 57
APB1 peripheral reset register (CRM_APB1RST) 59
APB peripheral clock enable register (CRM_AHBEN) 61
APB2 peripheral clock enable register (CRM_AHB2EN) 62
APB1 peripheral clock enable register (CRM_AHB1EN) 63
Battery powered domain control register (CRM_BPDC) 65
Control/status register (CRM_CTRLSTS) 66
AHB peripheral reset register (CRM_AHBRST) 67
Additional register1 (CRM_MISC1) 67
Additional register2 (CRM_MISC2) 67
Additional register3 (CRM_MISC3) 68
Interrupt map register (CRM_INTMAP) 68

5 Flash memory controller (FLASH) 69
Flash memory introduction 69
Flash memory operation 75
Unlock/lock 75
Erase operation 75
Programming operation 78
Read operation 79
External memory operation 80
User system data area operation 80
Unlock/lock 80
Erase operation 80
Programming operation 81
Read operation 82
Flash memory protection 83
Access protection 83
Erase/program protection 83
Special functions 84
Security library settings 84
Flash memory registers 85
Flash performance select register (FLASH_PSR) 86
Flash unlock register (FLASH_UNLOCK) 86
Flash user system data unlock register (FLASH_USD_UNLOCK) 86
Flash status register (FLASH_STS) 86
Flash control register (FLASH_CTRL) 87
Flash address register (FLASH_ADDR) 87
User system data register (FLASH_USD) 88

Erase/program protection status register (FLASH_EPPS) 88
Flash unlock register2 (FLASH_UNLOCK2) 88
Flash status register2 (FLASH_STS2) 88
Flash control register2 (FLASH_CTRL2) 89
Flash address register2 (FLASH_ADDR2) 90
Flash unlock register3 (FLASH_UNLOCK3) 90
Flash select register (FLASH_SELECT) 90
Flash status register3 (FLASH_STS3) 90
Flash control register3 (FLASH_CTRL3) 91
Flash address register3 (FLASH_ADDR3) 91
Flash decrption address register (FLASH_DA) 92
Flash security library status register (SLIB_STS0) 92
Flash security library status register1 (SLIB_STS1) 92
Flash security library password clear register (SLIB_PWD_CLR) 93
Security library additional status register (SLIB_MISC_STS) 93
Security library password setting register (SLIB_SET_PWD) 93
Security library address setting register (SLIB_SET_RANGE) 94
Security library unlock register (SLIB_UNLOCK) 94
Flash CRC check control register (FLASH_CRC_CTRL) 94
Flash CRC check result register (FLASH_CRC_CHKR) 95

6 General-purpose I/Os (GPIOs) 96
Introduction 96
Functional descriptions 96
GPIO structure 96
GPIO reset status 97
General-purpose input configuration 97
Analog input/output configuration 98
General-purpose output configuration 98
I/O port protection 98
GPIO registers 98
GPIO configuration register low (GPIOx_CFGLR) (x = A..E) 99
GPIO configuration register high (GPIOx_CFGHR) (x = A..E) 99
GPIO input data register (GPIOx_IDT) (x = A..E) 100
GPIO output data register (GPIOx_ODT) (x = A..E) 100
GPIO set/clear register (GPIOx_SCR) (x = A..E) 100
GPIO clear register (GPIOx_CLR) (x = A..E) 100
GPIO write protection register (GPIOx_WPR) (x = A..E) 101
GPIO huge current control register (GPIOx_HDRV) (x = A..E) 101

7 Multiplex function I/Os (IOMUX) 102
Introduction 102
Functional descriptions 102
IOMUX structure 102
MUX Input configuration 103

MUX output or bidirectional MUX configuration 103
Peripheral MUX function configuration 104
IOMUX map priority 104
External interrupt/wake-up lines 105
IOMUX registers 105
Event output control register (IOMUX_EVTOUT) 106
IOMUX remap register (IOMUX_REMAP) 107
IOMUX external interrupt configuration register1 (IOMUX_EXINTC1) 110
IOMUX external interrupt configuration register2 (IOMUX_EXINTC2) 111
IOMUX external interrupt configuration register3 (IOMUX_EXINTC3) 112
IOMUX external interrupt configuration register4 (IOMUX_EXINTC4) 113
IOMUX remap register2 (IOMUX_REMAP2) 114
IOMUX remap register3 (IOMUX_REMAP3) 114
IOMUX remap register4 (IOMUX_REMAP4) 115
IOMUX remap register5 (IOMUX_REMAP5) 116
IOMUX remap register6 (IOMUX_REMAP6) 117
IOMUX remap register7 (IOMUX_REMAP7) 118
IOMUX remap register8 (IOMUX_REMAP8) 120

8 External interrupt/Event controller (EXINT) 121
EXINT introduction 121
Main features 121
Function descriptions and configuration procedure 121
EXINT registers 122
Interrupt enable register (EXINT_INTEN) 122
Event enable register (EXINT_EVTEN) 123
Polarity configuration register1 (EXINT_ POLCFG1) 123
Polarity configuration register2 (EXINT_ POLCFG2) 123
Software trigger register (EXINT_ SWTRG) 123
Interrupt status register (EXINT_ INTSTS) 124

9 DMA controller (DMA) 125
Introduction 125
Main features 125
Functional descriptions 126
DMA configuration 126
Handshake mechanism 127
Arbiter 127
Programmable data transfer width 127
Errors 129
Interrupts 129
Fixed DMA request mapping 129
Flexible DMA request mapping 130
DMA registers 131

DMA interrupt status register (DMA_STS) 132
DMA interrupt flag clear register (DMA_CLR) 134
DMA channelx configuration register (DMA_CxCTRL) (x = 1…7) 136
DMA channelx number of data register (DMA_CxDTCNT) (x = 1…7) 137
DMA channelx peripheral address register (DMA_CxPADDR) (x = 1…7) 137
DMA channelx memory address register (DMA_CxMADDR) (x = 1…7) 137
Channel source register (DMA_SRC_SEL0) 138
Channel source register1 (DMA_SRC_SEL1) 138

10 CRC calculation unit (CRC) 139
CRC introduction 139
Main features 139
CRC registers 139
Data register (CRC_DT) 139
Common data register (CRC_CDT) 140
Control register (CRC_CTRL) 140
Initialization register (CRC_IDT) 141
Polynomial register (CRC_POLY) 141

11 I²C interface 142
I²C introduction 142
Main features 142
Functional descriptions 142
I²C interface 143
I²C slave communication flow 145
I²C master communication flow 147
Utilize DMA for data transfer 155
SMBus 156
I²C interrupt requests 158
I²C debug mode 159
I²C registers 159
Control register1 (I2C_CTRL1) 159
Control register2 (I2C_CTRL2) 161
Own address register1 (I2C_OADDR1) 162
Own address register2 (I2C_OADDR2) 162
Data register (I2C_DT) 162
Status register2 (I2C_STS2) 165
Clock control register (I2C_ CLKCTRL) 166
I²C timer rise time register (I2C_TMRISE) 166

12 Universal synchronous/asynchronous receiver/transmitter (USART) 167
USART introduction 167
Main features 168
Full-duplex/half-duplex selector 169

Mode selector 170
Introduction 170
Configuration procedure 170
USART frame format and configuration 171
DMA transfer introduction 172
Transmission using DMA 172
Reception using DMA 173
Baud rate generation 173
Introduction 173
Configuration 173
Transmitter 174
Transmitter introduction 174
Transmitter configuration 174
Receiver 175
Receiver introduction 175
Receiver configuration 175
Start bit and noise detection 176
Interrupt requests 177
I/O pin control 178
USART registers 179
Status register (USART_STS) 179
Data register (USART_DT) 180
Baud rate register (USART_BAUDR) 180
Control register1 (USART_CTRL1) 181
Control register2 (USART_CTRL2) 182
Control register3 (USART_CTRL3) 183
Guard time and divider register (GDIV) 184

13 Serial peripheral interface (SPI) 185
SPI introduction 185
Functional descriptions 185
SPI description 185
Full-duplex/half-duplex selector 186
Chip select controller 188
SPI_SCK controller 189
CRC 189
DMA transfer 190
Transmitter 191
Receiver 192
Interrupt 193
IO pin control 193
Precautions 193
I²S functional description 193

Table of Contents

I²S introduction 193
I²S full-duplex 195
Operation mode selector 195
Audio protocol selector 197
I2S_CLK controller 198
DMA transfer 201
Transmitter/Receiver 201
Interrupts 203
IO pin control 203
SPI registers 203
SPI control register1 (SPI_CTRL1) (Not used in I²S mode) 204
SPI control register2 (SPI_CTRL2) 205
SPI status register (SPI_STS) 206
SPI data register (SPI_DT) 207
SPICRC register (SPI_CPOLY) (Not used in I²S mode) 207
SPIRxCRC register (SPI_RCRC) (Not used in I²S mode) 207
SPITxCRC register (SPI_TCRC) 207
SPI_I2S register (SPI_I2SCTRL) 208
SPI_I2S prescaler register (SPI_I2SCLKP) 209

14 Timer 210
Basic timer (TMR6 and TMR7) 211
TMR6 and TMR7 introduction 211
TMR6 and TMR7 main features 211
TMR6 and TMR7 functional descriptions 211
TMR6 and TMR7 registers 213
General-purpose timer (TMR2 to TMR5) 215
TMRx introduction 215
TMRx main features 215
TMRx functional descriptions 216
TMRx registers 229
General-purpose timer (TMR9 to TMR14) 242
TMRx introduction 242
TMRx main features 242
TMRx functional descriptions 243
TMR9 and TMR12 registers 251
TMR10,TMR11, TMR13 and TMR14 registers 259
Advanced-control timers (TMR1 and TMR8) 265
TMR1 and TMR8 introduction 265
TMR1 and TMR8 main features 265
TMR1 and TMR8 functional descriptions 266
TMR1 and TMR8 registers 279

15 Window watchdog timer (WWDT) 296
WWDT introduction 296

Main features 296
Functional descriptions 296
Debug mode 297
WWDT registers 297
Control register (WWDT_CTRL) 298
Configuration register (WWDT_CFG) 298
Status register (WWDT_STS) 298

16 Watchdog timer (WDT) 299
WDT introduction 299
Main features 299
Functional descriptions 299
Debug mode 300
WDT registers 300
Command register (WDT_CMD) 301
Divider register (WDT_DIV) 301
Reload register (WDT_RLD) 301
Status register (WDT_STS) 301

17 Real-time clock (RTC) 302
RTC introduction 302
Main features 302
RTC structure 302
Functional descriptions 303
Configuring RTC registers 303
Reading RTC registers 304
RTC interrupts 304
RTC registers 305
RTC control register high (RTC_CTRLH) 306
RTC control register low (RTC_CTRLL) 306
RTC divider register (RTC_ DIVH/RTC_DIVL) 307
RTC divider counter register (RTC_ DIVCNTH/RTC_DIVCNTL) 307
RTC counter value register (RTC_CNTH/RTC_CNTL) 307
RTC alarm register (RTC_TAH/RTC_TAL) 308

18 Battery powered registers (BPR) 309
BPR introduction 309
Main features 309
Functional descriptions 309
BPR registers 309
Battery powered data register x (BPR_DTx) (x = 1 … 42) 311
RTC calibration register (BPR_ RTCCAL) 311
BPR control register (BPR_ CTRL) 312

BPR control/status register (BPR_ CTRLSTS) 312

19 Analog-to-digital converter (ADC) 313

ADC introduction 313
Main features 313
ADC structure 313
Functional descriptions 315
Channel management 315
ADC operation process 316
Conversion sequence management 319
Data management 321
Voltage monitor 322
Status flag and interrupts 322
Master/Slave mode 322
Data management 323
Regular simultaneous mode 323
Alternate preempted trigger mode 324
Regular switch mode 325
ADC registers 326
ADC status register (ADC_STS) 327
ADC control register1 (ADC_CTRL1) 328
ADC control register2 (ADC_CTRL2) 330
ADC sampling time register 1 (ADC_SPT1) 332
ADC sampling time register 2 (ADC_SPT2) 334
ADC preempted channel data offset register x (ADC_PCDTOx) (x=1..4) 336
ADC voltage monitor high threshold register (ADC_VWHB) 336
ADC voltage monitor low threshold register (ADC_VWLB) 336
ADC ordinary sequence register 1 (ADC_OSQ1) 336
ADC ordinary sequence register 2 (ADC_OSQ2) 336
ADC ordinary sequence register 3 (ADC_OSQ3) 337
ADC preempted sequence register (ADC_PSQ) 337
ADC preempted data register x (ADC_PDTx) (x = 1..4) 337
ADC ordinary data register (ADC_ODT) 337

20 Digital-to-analog converter (DAC) 338

DAC introduction 338
Main features 338
Design tips 338
Functional descriptions 339
Trigger events 339
Noise/Triangular-wave generation 339
DAC data alignment 341
DAC registers 341
DAC control register (DAC_CTRL) 342

Table of Contents

DAC software trigger register (DAC_SWTRG) 344
DAC1 12-bit right-aligned data holding register (DAC_D1DTH12R) 344
DAC1 12-bit left-aligned data holding register (DAC_D1DTH12L) 345
DAC1 8-bit right-aligned data holding register (DAC_D1DTH8R) 345
DAC2 12-bit right-aligned data holding register (DAC_D2DTH12R) 345
DAC2 12-bit left-aligned data holding register (DAC_D2DTH12L) 345
DAC2 8-bit right-aligned data holding register (DAC_D2DTH8R) 345
Dual DAC 12-bit right-aligned data holding register (DAC_DDTH12R) 345
Dual DAC 12-bit left-aligned data holding register (DAC_DDTH12L) 345
Dual DAC 8-bit right-aligned data holding register (DAC_DDTH8R) 346
DAC1 data output register (DAC_D1ODT) 346
DAC2 data output register (DAC_D2ODT) 346

21 CAN 347
CAN introduction 347
Main features 347
Baud rate 347
Interrupt management 350
Interrupt management 351
Functional descriptions 351
General description 351
Operating modes 352
Test modes 353
Message filtering 353
Message transmission 356
Message reception 358
Error management 359
CAN registers 359
CAN control and status registers 361
CAN mailbox registers 373
CAN filter registers 376

22 External memory controller 379
XMC introduction 379
Main features 379
XMC architecture 380
Block diagram 380
Address mapping 381
NOR/PSRAM 382
Operation mode 382
Access mode 383
NAND 404
Operation mode 404
Access timings 405

ECC computation 406
XMC registers 407
NOR Flash and PSRAM control registers 407
NAND Flash control registers 413

23 SDIO interface 417
SDIO introduction 417
Main features 417
Functional descriptions 420
Card functional description 420
Commands and responses 426
SDIO functional description 432
SDIO I/O card-specific operations 438
SDIO registers 440
SDIO power control register (SDIO_ PWRCTRL) 440
SDIO clock control register (SDIO_ CLKCTRL) 441
SDIO argument register (SDIO_ARG) 442
SDIO command register (SDIO_CMD) 442
SDIO command response register (SDIO_RSPCMD) 443
SDIO response 1..4 register (SDIO_RSPx) 443
SDIO data timer register (SDIO_DTTMR) 444
SDIO data length register (SDIO_DTLEN) 444
SDIO data control register (SDIO_DTCTRL) 444
SDIO data counter register (SDIO_DTCNTR) 446
SDIO status register (SDIO_STS) 446
SDIO clear interrupt register (SDIO_INTCLR) 447
SDIO interrupt mask register (SDIO_INTEN) 448
SDIOBUF counter register (SDIO_BUFCNTR) 450
SDIO data BUF register (SDIO_BUF) 450

24 Universal serial bus full-seed device interface (USBFS) 451
USBFS introduction 451
USBFS clock and pin configuration 451
USB clock configuration 451
USB pin configuration 451
Functional descriptions 452
USB initialization 452
Endpoint configuration 452
USB buffer 452
Double-buffered endpoints 454
SOF output 455
Suspend/Resume 455
USBFS interrupts 455
USBFS registers 455

USBFS endpoint n register (USBFS_EPTn), n=[0..7] 456
USBFS control register (USBFS_CTRL) 457
USBFS interrupt status register (USBFS_INTSTS) 458
USBFS SOF frame number register (USBFS_SOFRNUM) 459
USBFS device address register (USBFS_DEVADDR) 460
USBFS buffer table address register (USBFS_BUFTBL) 460
USBFS CFG control register (USBFS_CFG) 460
USBFS transmission buffer first address register (USBFS_TnADDR) 460
USBFS transmission data length register (USBFS_TnLEN) 460
USBFS reception buffer first address register (USBFS_RnADDR) 461
USBFS reception data length register (USBFS_RnLEN) 461

25 HICK auto clock calibration (ACC) 462

ACC introduction 462
Main features 462
Interrupt requests 462
Functional descriptions 463
Principle 464
Register description 466
Status register (ACC_STS) 466
Control register 1 (ACC_CTRL1) 467
Control register 2 (ACC_CTRL2) 468
Compare value 1 (ACC_C1) 468
Compare value 2 (ACC_C2) 468
Compare value 3 (ACC_C3) 469

26 Debug (DEBUG) 470

Debug introduction 470
Debug and Trace 470
I/O pin control 470
DEBUG registers 471
DEBUG device ID (DEBUG_IDCODE) 471
DEBUG control register (DEBUG_CTRL) 472

List of Tables

Table 1. Bit-band address mapping in SRAM 29
Table 2. Bit-band address mapping in the peripheral area 29
Table 3. HT32F49365/HT32F49395 series vector table 30
Table 4. List of abbreviations for registers 34
Table 5. List of abbreviations for registers 34
Table 6. Flash memory organization (1024K main memory) 37
Table 7. Flash memory organization (512K main memory) 37
Table 8. Flash memory organization (256K main memory) 38
Table 9. Peripheral boundary address 39
Table 10. PW register map and reset values 46
Table 11. CRM register map and reset values 52
Table 12. Flash memory architecture(1024 K) 69
Table 13. Flash memory architecture(512 K) 70
Table 14. Flash memory architecture(256 K) 70
Table 15. Instruction set supported by external memory 72
Table 16. User system data area 73
Table 17. Flash memory access limit 83
Table 18. Flash memory interface – Register map and reset value 85
Table 19. GPIO input mode configuration 97
Table 20. GPIO analog input/output mode configuration 98
Table 21. GPIO output mode configuration 98
Table 22. GPIO register map and reset values 98
Table 23. IOMUX input configuration 103
Table 24. IOMUX output configuration 104
Table 25. Hardware preemption 104
Table 26. Debug port map 105
Table 27. IOMUX register map and reset value 105
Table 28. External interrupt/Event controller register map and reset value 122
Table 29. DMA error event 129
Table 30. DMA interrupt requests 129
Table 31. DMA1 requests for each channel 129
Table 32. DMA2 requests for each channel 129
Table 33. Flexible DMA requests for each channel 130
Table 34. DMA register map and reset value 131
Table 35. CRC register map and reset value 139
Table 36. I²C interrupt requests 158
Table 37. I²C register map and reset value 159
Table 38. Error calculation for programmed baud rate 174
Table 39. Data sampling over start bit and noise detection 177

Table 40. Data sampling over valid data and noise detection 177
Table 41. USART interrupt request 178
Table 42. USART register map and reset value 179
Table 43. Audio frequency precision using system clock 199
Table 44. SPI register map and reset value 203
Table 45. TMR functional comparison 210
Table 46. TMR6 and TMR7 – register table and reset value 213
Table 47. TMRx internal trigger connection 218
Table 48. Counting direction versus encoder signals 220
Table 49. TMRx register map and reset value 229
Table 50. Standard CxOUT channel output control bit 239
Table 51. TMRx internal trigger connection 244
Table 52. TMRx register map and reset value 251
Table 53. Standard CxOUT channel output control bit 258
Table 54. TMRx register map and reset value 259
Table 55. Standard CxOUT channel output control bit 264
Table 56. TMRx internal trigger connection 268
Table 57. Couting direction versus encoder signals 270
Table 58. TMR1 and TMR8 register map and reset value 279
Table 59. Complementary output channel CxOUT and CxCOUT control bits with break function 291
Table 60. Minimum and maximum timeout value when PCLK1=72 MHz 297
Table 61. WWDT register map and reset value 297
Table 62. WDT timeout period (LICK=40kHz) 300
Table 63. WDT register and reset value 300
Table 64. RTC register map and reset values 305
Table 65. BPR register map and reset values 309
Table 66. Trigger sources for ADC1 and ADC2 317
Table 67. Trigger sources for ADC3 318
Table 68. ADC register map and reset values 327
Table 69. Trigger source selection 339
Table 70. DAC register map and reset values 341
Table 71. Examples of filter numbering 355
Table 72. CAN register map and reset values 359
Table 73. NOR/PSRAM pins 380
Table 74. NAND pins 381
Table 75. Memory bank selection 382
Table 76. Pin signals for NOR and PSRAM 382
Table 77. Address translation between HADDR and external memory 383
Table 78. Data access width vs. external memory data width 383
Table 79. NOR/PSRAM parameter registers 384
Table 80. Mode 1 – SRAM/NOR Flash chip select control register (XMC_BK1CTRL) 384

Table 81. Mode 1 – SRAM/NOR Flash chip select timing register (XMC_ BK1TMG) 385
Table 82. Mode 2 – SRAM/NOR Flash chip select control register (XMC_BK1CTRL) 386
Table 83. Mode 2 – SRAM/NOR Flash chip select timing register (XMC_BK1TMG) 387
Table 84. Mode A – SRAM/NOR Flash chip select control register (XMC_BK1CTRL)..... 388
Table 85. Mode A – SRAM/NOR Flash chip select timing register (XMC_BK1TMG) 389
Table 86. Mode A – SRAM/NOR Flash write timing register (XMC_BK1TMGWR) 389
Table 87. Mode B – SRAM/NOR Flash chip select register (XMC_BK1CTRL)..... 391
Table 88. Mode B – SRAM/NOR Flash chip select timing register (XMC_BK1TMG) 391
Table 89. Mode B – SRAM/NOR Flash write timing register (XMC_BK1TMGWR) 392
Table 90. Mode C – SRAM/NOR Flash chip select register (XMC_BK1CTRL) 393
Table 91. Mode C – SRAM/NOR Flash chip select timing register (XMC_BK1TMG) 394
Table 92. Mode C – SRAM/NOR Flash write timing register (XMC_BK1TMGWR) 394
Table 93. Mode D – SRAM/NOR Flash chip select register (XMC_BK1CTRL) 396
Table 94. Mode D – SRAM/NOR Flash chip select timing register (XMC_BK1TMG) 396
Table 95. Mode D – SRAM/NOR Flash write timing register (XMC_BK1TMGWR)..... 397
Table 96. Multiplexed mode – SRAM/NOR Flash chip select control register (XMC_BK1CTRL) 398
Table 97. Multiplexed mode – SRAM/NOR Flash chip select timing register (XMC_BK1TMG) 399
Table 98. Synchronous mode – SRAM/NOR Flash chip select control register (XMC_ BK1CTRL) 400
Table 99. Synchronous mode – SRAM/NOR Flash chip select timing register (XMC_BK1TMG) 401
Table 100. Typical pin signals for NAND Flash..... 404
Table 101. Data access width vs. external memory data width 404
Table 102. NAND parameter registers..... 405
Table 103. lists the ECC result bits corresponding to the number of bytes 407
Table 104. XMC register address mapping..... 407
Table 105. Lock/unlock command structure 424
Table 106. Commands..... 427
Table 107. Data block read commands 428
Table 108. Data stream read/write commands 428
Table 109. Data block write commands..... 428
Table 110. Block-based write protect commands..... 429
Table 111. Erase commands..... 429
Table 112. I/O mode commands 429
Table 113. Card lock commands..... 429
Table 114. Application-specific commands 430
Table 115. R1 response 430
Table 116. R2 response 431
Table 117. R3 response 431
Table 118. R4 response 431
Table 119. R4b response 431
Table 120. R5 response..... 431
Table 121. R6 response..... 432

Table 122. SDIO pin definitions 433
Table 123. Command formats 434
Table 124. Short response format 434
Table 125. Long response format 434
Table 126. Command path status flags 434
Table 127. Data token formats 437
Table 128. A summary of the SDIO registers. 440
Table 129. Response type and SDIO_RSPx register 443
Table 130. USB pin configuration 451
Table 131. Buffer size configuration table 452
Table 132. The structure of a regular endpoint register buffer description field 453
Table 133. USBFS register map and reset values 455
Table 134. ACC interrupt requests 462
Table 135. ACC register map and reset values 466
Table 136. Trace function enable 471
Table 137. Trace function mode 471
Table 138. DEBUG register address and reset value 471
Table 139. Description of ID code 471

List of Figures

Figure 1. HT32F49365/HT32F49395 Series microcontrollers system architecture 24
Figure 2. Internal block diagram of Cortex®-M4F 27
Figure 3. Comparison between bit-band region and its alias region: image A 27
Figure 4. Comparison between bit-band region and its alias region: image B 28
Figure 5. Reset process 33
Figure 6. Example of MSP and PC initialization 33
Figure 7. HT32F49365/HT32F49395 address mapping 36
Figure 8. Block diagram of each power supply 41
Figure 9. Power-on reset/Low voltage reset waveform 42
Figure 10. PVM threshold and output 43
Figure 11. Clock tree 48
Figure 12. System reset circuit 51
Figure 13. External memory ciphertext protection 71
Figure 14. Reference circuit for external memory 71
Figure 15. Flash memory page erase process 76
Figure 16. Flash memory mass erase process 77
Figure 17. Flash memory programming process 79
Figure 18. System data area erase process 81
Figure 19. System data area programming process 82
Figure 20. GPIO basic structure 97
Figure 21. Basic structure of IOMUX basic structure 103
Figure 22. External interrupt/Event controller block diagram 121
Figure 23. DMA block diagram 126
Figure 24. Re-arbitrae after request/acknowledge 127
Figure 25. PWIDTH: byte, MWIDTH: half-word 128
Figure 26. PWIDTH: half-word, MWIDTH: word 128
Figure 27. PWIDTH: word, MWIDTH: byte 128
Figure 28. I²C bus protocol 142
Figure 29. I²C function block diagram 143
Figure 30. Transfer sequence of slave transmitter 145
Figure 31. Transfer sequence of slave receiver 146
Figure 32. Transfer sequence of master transmitter 148
Figure 33. Transfer sequence of master receiver 150
Figure 34. Transfer sequence of master receiver when N>2 151
Figure 35. Transfer sequence of master receiver when N=2 153
Figure 36. Transfer sequence of master receiver when N=1 154
Figure 37. USART block diagram 168
Figure 38. IrDA DATA(3/16) – normal mode 171
Figure 39. Word length 172

Figure 40. USART interrupt map diagram 178
Figure 41. SPI block diagram 185
Figure 42. SPI two-wire unidirectional full-duplex connection 186
Figure 43. Single-wire unidirectional receive only in SPI master mode 187
Figure 44. Single-wire unidirectional receive only in SPI slave mode 187
Figure 45. Single-wire bidirectional half-duplex mode 188
Figure 46. SPI interrupts 193
Figure 47. I²S block diagram 194
Figure 48. I²S full-duplex structure 195
Figure 49. I²S slave device transmission 195
Figure 50. I²S slave device reception 196
Figure 51. I²S master device transmission 196
Figure 52. I²S master device reception 197
Figure 53. CK & MCK source in master mode 199
Figure 54. I²S interrupt 203
Figure 55. Basic timer block diagram 211
Figure 56. Control circuit with CK_INT divided by 1 211
Figure 57. Overflow event when PRBEN=0 212
Figure 58. Overflow event when PRBEN=1 212
Figure 59. Counting timing diagram when the prescaler division is 4 212
Figure 60. General-purpose timer block diagram 216
Figure 61. Control circuit with CK_INT divided by 1 216
Figure 62. Block diagram of external clock mode A 217
Figure 63. Counting in external clock mode A 217
Figure 64. Block diagram of external clock mode B 217
Figure 65. Counting in external clock mode B 218
Figure 66. Counter timing with prescaler value changing from 1 to 4 218
Figure 67. Overflow event when PRBEN=0 219
Figure 68. Overflow event when PRBEN=1 219
Figure 69. Counter timing diagram with internal clock divided by 4 220
Figure 70. Counter timing diagram with internal clock divided by 1 and TMRx_PR=0x32 220
Figure 71. Example of counter behavior in encoder interface mode (encoder mode C) 221
Figure 72. Input/output channel 1 main circuit 221
Figure 73. Channel 1 input stage 221
Figure 74. Capture/compare channel output stage (channel 1 to 4) 222
Figure 75. C1ORAW toggles when counter value matches the C1DT value 223
Figure 76. Upcounting mode and PWM mode A 224
Figure 77. Up/down counting mode and PWM mode A 224
Figure 78. One-pulse mode 225
Figure 79. Clearing CxORAW(PWM mode A) by EXT input 225
Figure 80. Example of reset mode 226

Figure 81. Example of suspend mode 226
Figure 82. Example of trigger mode 227
Figure 83. Master/slave timer connection 227
Figure 84. Using master timer to start slave timer 228
Figure 85. Starting master and slave timers synchronously by an external trigger 229
Figure 86. Block diagram of general-purpose TMR9/12 242
Figure 87. Block diagram of general-purpose TMR10/11/13/14 243
Figure 88. Control circuit with CK_INT divided by 1 243
Figure 89. Block diagram of external clock mode A 244
Figure 90. Counting in external clock mode A 244
Figure 91. Counter timing with prescaler value changing from 1 to 4 245
Figure 92. Overflow event when PRBEN=0 245
Figure 93. Overflow event when PRBEN=1 246
Figure 94. Input/output channel 1 main circuit 246
Figure 95. Channel 1 input stage 246
Figure 96. Capture/compare channel output stage (channel 1) 247
Figure 97. C1ORAW toggles when counter value matches the C1DT value 248
Figure 98. Upcounting mode and PWM mode A 249
Figure 99. One-pulse mode 249
Figure 100. Example of reset mode 250
Figure 101. Example of suspend mode 250
Figure 102. Example of trigger mode 251
Figure 103. Block diagram of advanced-control timer 265
Figure 104. Control circuit with CK_INT divided by 1 266
Figure 105. Block diagram of external clock mode A 266
Figure 106. Counting in external clock mode A 267
Figure 107. Block diagram of external clock mode B 267
Figure 108. Counting in external clock mode B 267
Figure 109. Counter timing with prescaler value changing from 1 to 4 268
Figure 110. Overflow event when PRBEN=0 269
Figure 111. Overflow event when PRBEN=1 269
Figure 112. Counter timing diagram with internal clock divided by 4 269
Figure 113. Counter timing diagram with internal clock divided by 1 and TMRx_PR=0x32 270
Figure 114. OVFIF when RPR=2 270
Figure 115. Example of encoder interface mode C 271
Figure 116. Input/output channel 1 main circuit 271
Figure 117. Channel 1 input stage 271
Figure 118. Channel output stage (channel 1 to 3) 272
Figure 119. Channel 4 output stage 273
Figure 120. C1ORAW toggles when counter value matches the C1DT value 274
Figure 121. Upcounting mode and PWM mode A 274

Figure 122. Up/down counting mode and PWM mode A 275
Figure 123. One-pulse mode 275
Figure 124. Clearing CxORAW(PWM mode A) by EXT input 276
Figure 125. Complementary output with dead-time insertion 276
Figure 126. Example of TMR break function 277
Figure 127. Example of reset mode 278
Figure 128. Example of suspend mode 278
Figure 129. Example of trigger mode 279
Figure 130. Window watchdog block diagram 296
Figure 131. Window watchdog timing diagram 297
Figure 132. WDT block diagram 300
Figure 133. Simplified RTC block diagram 303
Figure 134. RTC second and alarm waveform example with DIV=0004 and TA=00003 305
Figure 135. RTC overflow waveform example with DIV=0004 305
Figure 136. ADC1 block diagram 314
Figure 137. ADC basic operation process 316
Figure 138. ADC power-on and calibration 317
Figure 139. Sequence mode 319
Figure 140. Preempted group auto conversion mode 320
Figure 141. Repetition mode 320
Figure 142. Partition mode 321
Figure 143. Data alignment 321
Figure 144. Block diagram of master/salve mode 323
Figure 145. Regular simultaneous mode 324
Figure 146. Regular simultaneous mode 324
Figure 147. Alternate preempted trigger mode 325
Figure 148. Fast switch mode 325
Figure 149. Fast slow mode 326
Figure 150. DAC1/DAC2 block diagram 338
Figure 151. LFSR register calculation algorithm 340
Figure 152. Trianaular-wave generation 340
Figure 153. Bit timing 348
Figure 154. Transmit interrupt generation 349
Figure 155. Transmit interrupt generation 350
Figure 156. Receive interrupt 0 generation 350
Figure 157. Receive interrupt 1 generation 350
Figure 158. Status error interrupt generation 351
Figure 159. CAN block diagram 352
Figure 160. 32-bit identifier mask mode 354
Figure 161. 32-bit identifier list mode 354
Figure 162. 16-bit identifier mask mode 354

Figure 163. 16-bit identifier list mode 355
Figure 164. Transmit mailbox status 357
Figure 165. Receive FIFO status 359
Figure 166. Transmit and receive mailboxes 374
Figure 167. XMC block diagram 380
Figure 168. XMC memory banks 381
Figure 169. NOR/PSARM mode 1 read access 385
Figure 170. NOR/PSARM mode 1 write access 386
Figure 171. NOR/PSARM mode 2 read access 387
Figure 172. NOR/PSARM mode 2 write access 388
Figure 173. NOR/PSARM mode A read access 390
Figure 174. NOR/PSARM mode A write access 390
Figure 175. NOR/PSARM mode B read access 392
Figure 176. NOR/PSARM mode B write access 393
Figure 177. NOR/PSARM mode C read access 395
Figure 178. NOR/PSARM mode C write access 395
Figure 179. NOR/PSARM mode D read access 397
Figure 180. NOR/PSARM mode D write access 398
Figure 181. NOR/PSARM multiplexed mode read access 399
Figure 182. NOR/PSARM multiplexed mode write access 400
Figure 183. NOR/PSARM synchronous multiplexed mode read access 402
Figure 184. NOR/PSARM synchronous multiplexed mode write access 403
Figure 185. NAND read access 405
Figure 186. NAND wait functionality 406
Figure 187. SDIO “no response” and “no data” operations 418
Figure 188. SDIO multiple block read operation 418
Figure 189. SDIO multiple block write operation 419
Figure 190. SDIO sequential read operation 419
Figure 191. SDIO sequential write operation 420
Figure 192. SDIO block diagram 432
Figure 193. Command channel state machine (CCSM) 435
Figure 194. SDIO command transfer 436
Figure 195. Data channel state machine (DCSM) 436
Figure 196. Structure of 512 bytes endpoint register buffer 453
Figure 197. Buffer description table of regular endpoint vs. double-buffered endpoint 454
Figure 198. ACC interrupt mapping diagram 462
Figure 199. ACC block diagram 464
Figure 200. Cross-return algorithm 464

1 System architecture

HT32F49365/HT32F49395 series microcontrollers consist of 32-bit ARM® Cortex®-M4F processor core, multiple 16-bit and 32-bit timers, DMA controller, RTC, communication interfaces such as SPI, I²C, USART/UART and SDIO, CANs, external memory controller (XMC), USB2.0 full-speed interface, HICK with automatic clock calibration (ACC), 12-bit ADC, 12-bit DAC, programmable voltage monitor (PVM) and other peripherals. Cortex®-M4F processer supports enhanced high-performance DSP instruction set, including extended single cycle 16-bit/32-bit multiply accumulater (MAC), dual 16-bit MAC instruction, optimized 8-bit/16-bit SIMD operation and saturation operation instruction, and single-precision (IEEE-754) float point unit (FPU), as shown in the following diagram:

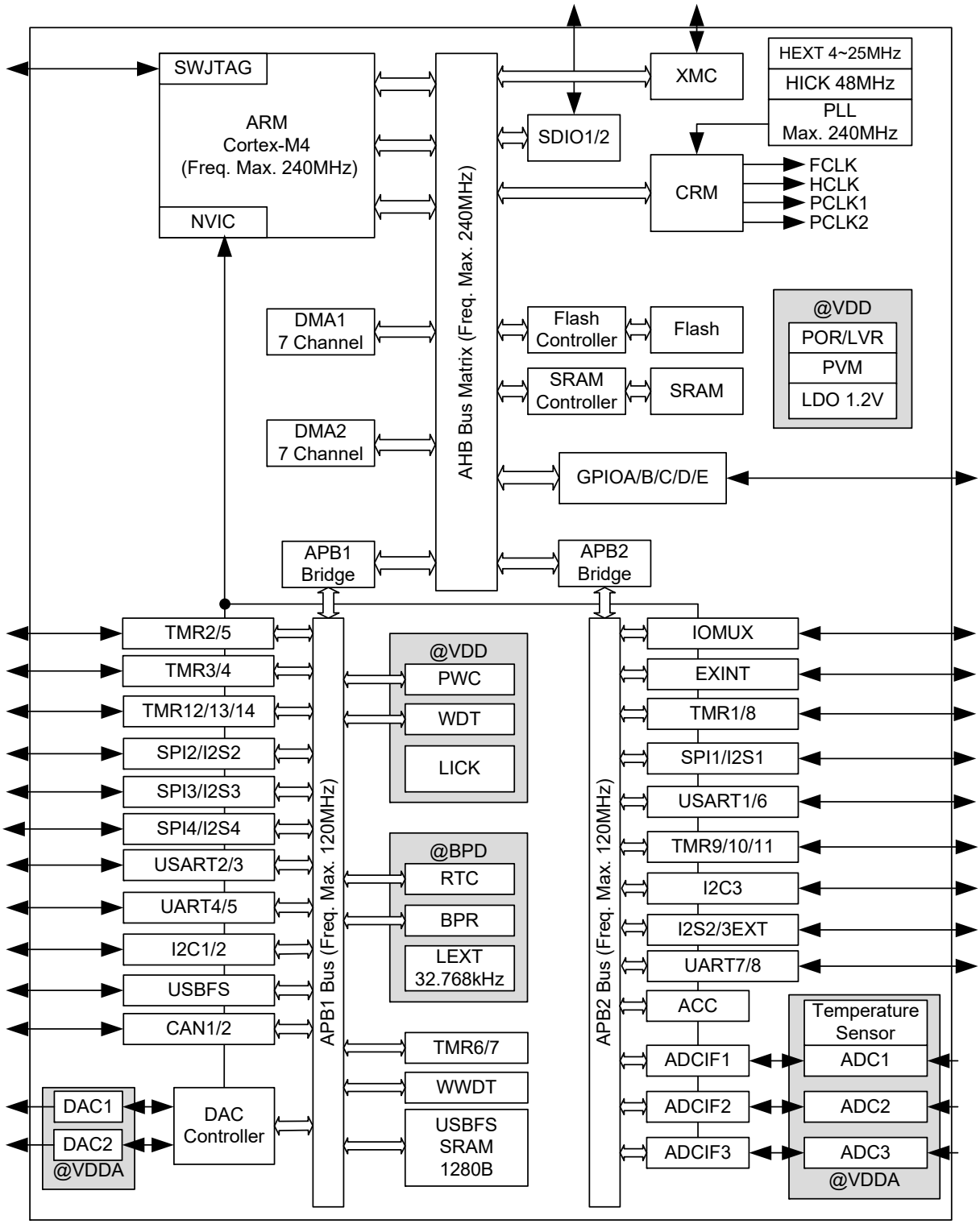


Figure 1. HT32F49365/HT32F49395 Series microcontrollers system architecture

Feature

- Core: ARM®32-bit Cortex®-M4F CPU with FPU
  - 240 MHz maximum frequency, with a Memory Protection Unit (MPU), single-cycle multiplication and hardware division
  - Floating Point Unit (FPU)
  - DSP instructions
- Memories
  - 256 to 1024 KBytes of Flash memory
  - sLib: configure any part of main Flash as a library area that is code excutable but secured and non-readable
  - SPIM interface: extra interfacing up to 16 Mbytes of external SPI Flash (as instruction/data memory)
  - Up to 96 + 128 KBytes of SRAM
  - External memory controller (XMC) with 2 Chip Select, supporting multiplexed SRAM/NOR/PSRAM and NAND memories
  - LCD parallel interface, 8080/6800 modes
- Clock, Reset, and Power management
  - 2.6 V ~ 3.6 V application suppy and I/Os
  - Power on reset (POR)/ low voltage reset (LVR), and power voltage monitor (PVM)
  - 4 to 25 MHz crystal (HEXT)
  - Internal 48 MHz factory-trimmed RC (offering 1% accuracy at T_A=25 °C, 2.5 % accuracy at T_A=-40 to +105 °C), with automatic clock calibration (ACC)
  - Internal 40 kHz RC oscillator (LICK)
  - 32.768 kHz crystal oscillator (LEXT)
- Low power consumption
  - Sleep, Deepsleep, and Standby modes
  - V_{BAT} supply for RTC and 42 × 16-bit battery powered registers (BPR)
- 3 × 12-bit 0.5 μs A/D converters, up to 16 channels
  - Conversion range: 0 V to 3.6 V
  - Triple sample and hold capability
  - Temparature sensor
- 2 × 12-bit D/A converters
- DMA: 14-channel DMA controller
  - Peripherals supported: timers, ADCs, SDIOs, I^2Ss, SPIs, I^2Cs, and USARTs
- Debug Mode
  - Serial wire debug (SWD) and JTAG interface
  - Cortex®-M4F Embedded Trace Macrocell (ETM™)
- Up to 80 Fast I/O Interfaces
  - 37/51/80 multifunctional and bidirectional I/Os, all mappable to 16 external interrupt vectors and almost 5 V-tolerant
  - All fast I/Os, control registers accessable with f_{AHB} speed
- Up to 17 Timers
  - Up to 8 × 16-bit general-purpose timers + 2 × 32-bit general-purpose timers; each with 4 IC/OC/PWM or pulse counter and quadrature (incremental) encoder input.

- Up to 2 × 16-bit motor control PWM advanced timers with dead-time generator and emergency brake
- 2 × Watchdog timers
- SysTick timer: 24-bit downcounter
- 2 × 16-bit basic timers to drive the DAC

■ Up to 20 Communication Interfaces
- Up to 3 × I²C interfaces (SMBus/PMBus)
- Up to 8 × USARTs (ISO7816 interface, LIN, IrDA capability, and modem control)
- Up to 4 × SPIs (50 Mbit/s), all with I²S interface multiplexed,. I²S2/ I²S3 support full-duplex
- Up to 2 × CAN interfaces (2.0B Active)
- USB2.0 full-speed interface supporting Crystal-less
- Up to 2 × SDIO interfaces

■ CRC Calculation Unit

■ 96-bit unique ID (UID)

■ Packages
- LQFP100 (14 × 14 mm)
- LQFP64 (10 × 10 mm)
- LQFP48 (7 × 7 mm)
- QFN48 (6 × 6 mm)

■ List of Models

Internal Flash	Part number
1024 KBytes	HT32F49395
256 KBytes	HT32F49365

System overview

ARM Cortex™-M4F processor

Cortex®-M4F processor is a low power consumption processor featuring low gate count, low interrupt latency, and low-cost debug. It supports DSP instruction set and FPU, and is applicable to deeply-embedded applications that require quicker response to interruption. Cortex®-M4F processor is based on ARMv7-M architecture, supporting both Thumb instruction set and DSP instruction set.

The following diagram shows the internal block diagram of Cortex®-M4F processor. Please refer to ARM Cortex® -M4 Technical Reference Manual for more information.

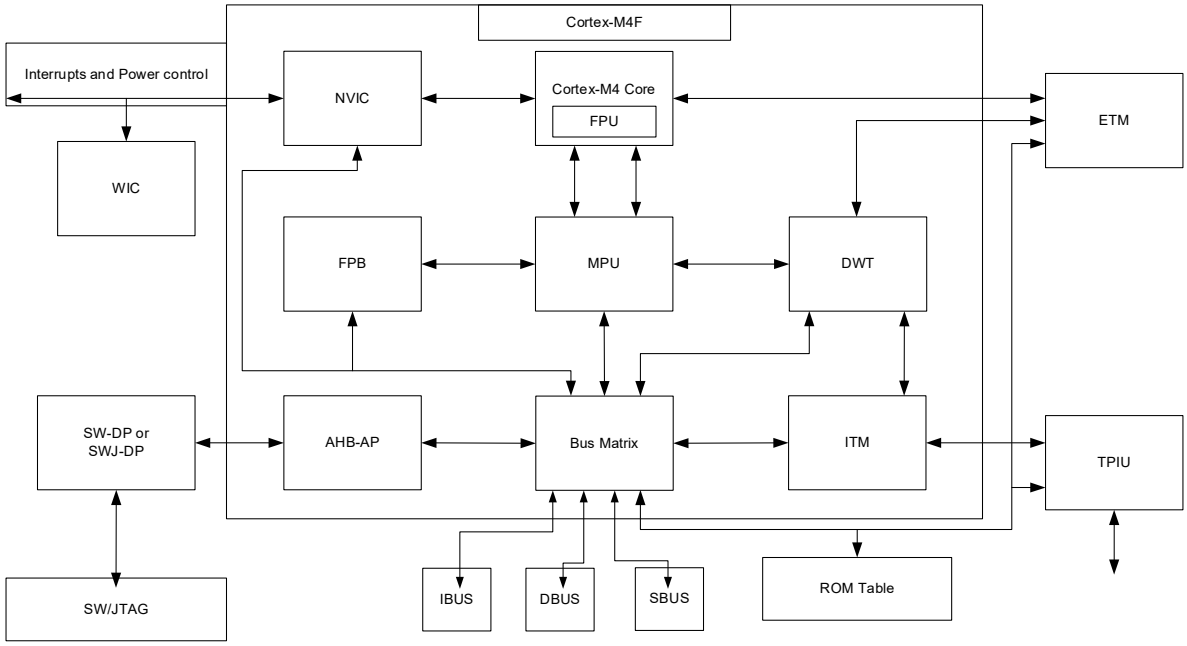


Figure 2. Internal block diagram of Cortex®-M4F

Bit band

Through bit-band operations, read and write access to a single bit can be performed using common load/store operations. The Cortex®-M4F memory includes two bit-band regions: the least significant 1M byte of SRAM and the least significant 1M byte of peripherals. In addition to access to bit-band addresses, their respective bit-band alias area can be used to access to any bit of any address. The bit-band alias area transforms each bit into a 32-bit word. Thus, accessing to one bit in the alias area has the same effect as read-modify-write operation on the bit in the bit-band region.

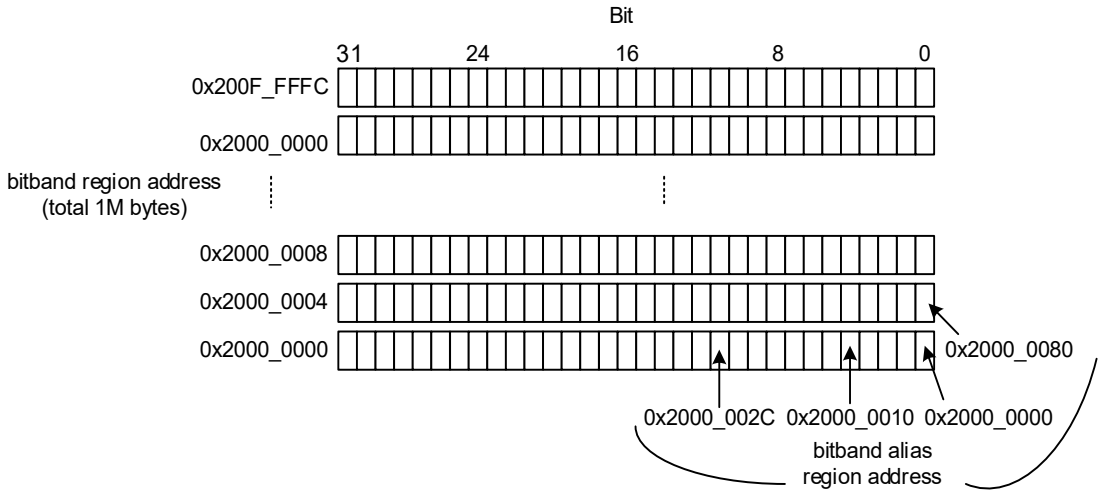


Figure 3. Comparison between bit-band region and its alias region: image A

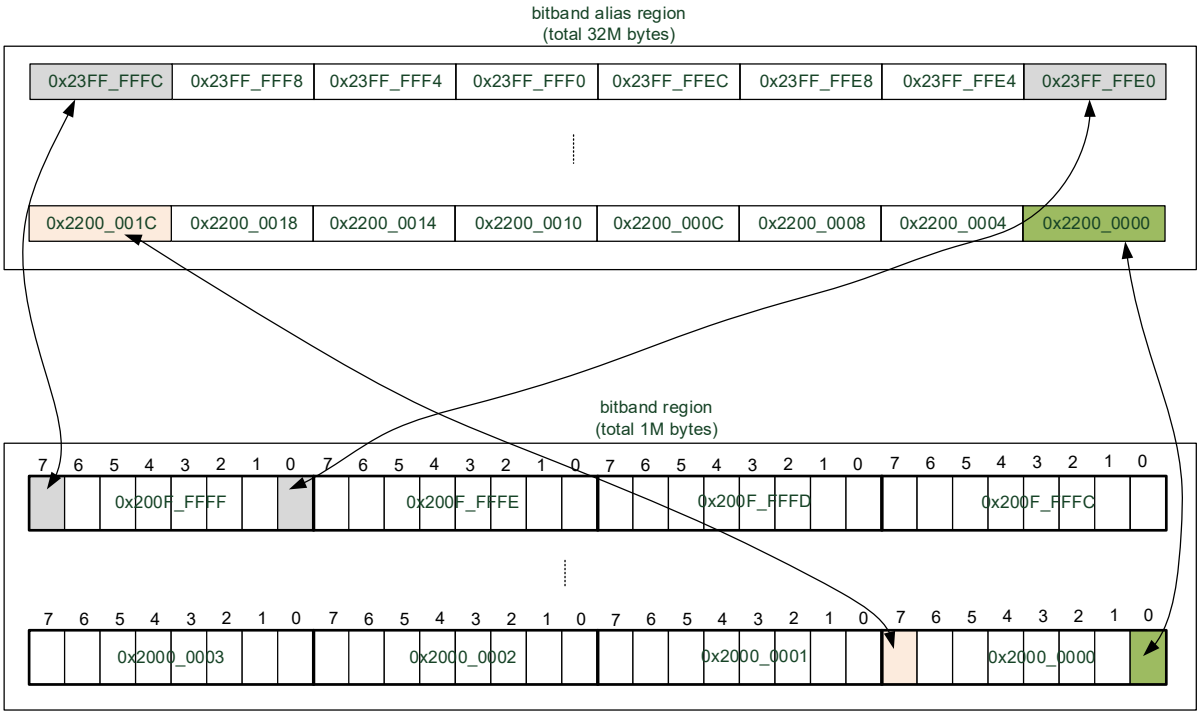


Figure 4. Comparison between bit-band region and its alias region: image B

Bit-band region: address area supporting bit-band operations

Bit-band alias region: access to the alias region has the same effect as read-modify-write operation on the bit-band region

Each bit in the bit-band region is mapped into a word (LSB) of the alias region. When accessing to the address in the bit-band alias region, such address is transformed into the bit-band address. For a read operation, read one word in the bit-band region, then move the required bit to the right to LSB, and then return the LSB. For a write opearation, first move the targeted bit to the left to the corresponding bit number, then perform read-modify-write operation on bit level.

The address ranges of two memories supporting bit-band operations:

The lowest 1 Mbyte of SRAM: 0x2000_0000~0x200F_FFFF

The lowest 1 Mbyte of peripherals: 0x4000_0000~0x400F_FFFF

For a bit in the SRAM bit-band region, if the byte address is A, the bit number is n ($0 \le n \le 7$), then the alias address where the bit is :

AliasAddr = 0x2200_0000 + (A-0x2000_0000) × 32 + n × 4

For a bit in the peripheral bit-band region, if the byte address is A, the bit number is n ($0 \le n \le 7$), then the alias address where the bit is:

AliasAddr = 0x4200_0000 + (A-0x4000_0000) × 32 + n × 4

The following table shows the mapping between bit-band region and alias region in SRAM:

Table 1. Bit-band address mapping in SRAM

Bit-band region	Equivalent alias address
0x2000_0000.0	0x2200_0000.0
0x2000_0000.1	0x2200_0004.0
0x2000_0000.2	0x2200_0008.0
…	…
0x2000_0000.31	0x2200_007C.0
0x2000_0004.0	0x2200_0080.0
0x2000_0004.1	0x2200_0084.0
0x2000_0004.2	0x2200_0088.0
…	…
0x200F_FFFC.31	0x23FF_FFFC.0

The following table shows the mapping between bit-band region and alias region in the peripheral area:

Table 2. Bit-band address mapping in the peripheral area

Bit-band region	Equivalent alias address
0x4000_0000.0	0x4200_0000.0
0x4000_0000.1	0x4200_0004.0
0x4000_0000.2	0x4200_0008.0
…	…
0x4000_0000.31	0x4200_007C.0
0x4000_0004.0	0x4200_0080.0
0x4000_0004.1	0x4200_0084.0
0x4000_0004.2	0x4200_0088.0
…	…
0x400F_FFFC.31	0x43FF_FFFC.0

In terms of bit-band operation, one of the advantages is to control LED ON/OFF independently via GPIO pins. On the other hand, it brings great conveninence for serial interface operations. In short, it is best suited to hardware I/O-intensive low-level applications.

In addition, bit-band operations can also simplify jump process. When jump operation is based on a bit level, the previous steps are:

- Read the whole register
- Mask the undesired bits
- Compare and jump

For now, you just need do:

- Read the bit status from the bit-band alias region
- Compare and jump

Apart from making code more concise, its important function is also reflected in multi-task environment. When it comes to multiples tasks, it turns the read-modify-write operations into a hardware-supported atomic operation to avoid the scenario where the read-modify-write opearion is disrupted, resulting in disorder.

Interrupt and exception vectors

Table 3. HT32F49365/HT32F49395 series vector table

Pos.	Priority	Priority Type	Name	Description	Address
	—	—	—	Reserved	0x0000_0000
	-3	Fixed	Reset	Reset	0x0000_0004
	-2	Fixed	NMI	Non maskable interrupt Clock Fail Detector (CFD) is linked to NMI vector	0x0000_0008
	-1	Fixed	HardFault	All class of fault	0x0000_000C
	0	Configurable	MemoryManage	Memory management	0x0000_0010
	1	Configurable	BusFault	Pre-fetch fault, memory access fault	0x0000_0014
	2	Configurable	UsageFault	Undefined instruction or illegal state	0x0000_0018
	—	—	—	Reserved	0x0000_001C ~0x0000_002B
	3	Configurable	SVCall	System service call via SWI instruction	0x0000_002C
	4	Configurable	DebugMonitor	Debug monitor	0x0000_0030
	—	—	—	Reserved	0x0000_0034
	5	Configurable	PendSV	Pendable request for system service	0x0000_0038
	6	Configurable	SysTick	System tick timer	0x0000_003C
0	7	Configurable	WWDT	Window watchdog timer	0x0000_0040
1	8	Configurable	PVM	PVM from EXINT interrupt	0x0000_0044
2	9	Configurable	TAMPER	Tamper interrupt	0x0000_0048
3	10	Configurable	RTC	RTC global interrupt	0x0000_004C
4	11	Configurable	FLASH	Flash global interrupt	0x0000_0050
5	12	Configurable	CRM	Clock Reset manage interrupt	0x0000_0054
6	13	Configurable	EXINT0	EXINT line0 interrupt	0x0000_0058
7	14	Configurable	EXINT1	EXINT line1 interrupt	0x0000_005C
8	15	Configurable	EXINT2	EXINT line2 interrupt	0x0000_0060
9	16	Configurable	EXINT3	EXINT line3 interrupt	0x0000_0064
10	17	Configurable	EXINT4	EXINT line4 interrupt	0x0000_0068
11	18	Configurable	DMA1 channel1	DMA1 channel1 global interrupt	0x0000_006C
12	19	Configurable	DMA1 channel2	DMA1 channel2 global interrupt	0x0000_0070
13	20	Configurable	DMA1 channel3	DMA1 channel3 global interrupt	0x0000_0074
14	21	Configurable	DMA1 channel4	DMA1 channel4 global interrupt	0x0000_0078
15	22	Configurable	DMA1 channel5	DMA1 channel5 global interrupt	0x0000_007C
16	23	Configurable	DMA1 channel6	DMA1 channel6 global interrupt	0x0000_0080
17	24	Configurable	DMA1 channel7	DMA1 channel7 global interrupt	0x0000_0084
18	25	Configurable	ADC1_2	ADC1 and ADC2 global interrupt	0x0000_0088
19	26	Configurable	USBFS_H_CAN1_TX	USBFS high priority or CAN1 TX interrupt	0x0000_008C
20	27	Configurable	USBFS_L_CAN1_RX0	USBFS low priority or CAN1 RX0 interrupt	0x0000_0090
21	28	Configurable	CAN1_RX1	CAN1 RX1 interrupt	0x0000_0094
22	29	Configurable	CAN_SE	CAN state error interrupt	0x0000_0098
23	30	Configurable	EXINT9_5	EXINT line[9:5] interrupt	0x0000_009C

Pos.	Priority	Priority Type	Name	Description	Address
24	31	Configurable	TMR1_BRK_TMR9	TMR1 break interrupt and TMR9 global interrupt	0x0000_00A0
25	32	Configurable	TMR1_OVF_TMR10	TMR1 overflow and TMR10 global interrupt	0x0000_00A4
26	33	Configurable	TMR1_TRG_HALL_TMR11	TMR1 trigger and HALL interrupt and TMR11 global interrupt	0x0000_00A8
27	34	Configurable	TMR1_CH	TMR1 channel interrupt	0x0000_00AC
28	35	Configurable	TMR2	TMR2 global interrupt	0x0000_00B0
29	36	Configurable	TMR3	TMR3 global interrupt	0x0000_00B4
30	37	Configurable	TMR4	TMR4 global interrupt	0x0000_00B8
31	38	Configurable	I²C1_EVT	I²C1 event interrupt	0x0000_00BC
32	39	Configurable	I²C1_ERR	I²C1 error interrupt	0x0000_00C0
33	40	Configurable	I²C2_EVT	I²C2 event interrupt	0x0000_00C4
34	41	Configurable	I²C2_ERR	I²C2 error interrupt	0x0000_00C8
35	42	Configurable	SPI1	SPI1 global interrupt	0x0000_00CC
36	43	Configurable	SPI2_I²S2EXT	SPI2 and I²S2EXT global interrupt	0x0000_00D0
37	44	Configurable	USART1	USART1 global interrupt	0x0000_00D4
38	45	Configurable	USART2	USART2 global interrupt	0x0000_00D8
39	46	Configurable	USART3	USART3 global interrupt	0x0000_00DC
40	47	Configurable	EXINT15_10	EXINT line[15:10] global interrupt	0x0000_00E0
41	48	Configurable	RTCAlarm	RTC alarm through EXINT interrupt	0x0000_00E4
42	49	Configurable	USBFS_WAKEUP	USBFS wakeup through EXINT interrupt	0x0000_00E8
43	50	Configurable	TMR8_BRK_TMR12	TMR8 break interrupt and TMR12 global interrupt	0x0000_00EC
44	51	Configurable	TMR8_OVF_TMR13	TMR8 overflow interrupt and TMR13 global interrupt	0x0000_00F0
45	52	Configurable	TMR8_TRG_HALL_TMR14	TMR8 trigger and HALL interrupt and TMR14 global interrupt	0x0000_00F4
46	53	Configurable	TMR8_CH	TMR8 channel interrupt	0x0000_00F8
47	54	Configurable	ADC3	ADC3 global interrupt	0x0000_00FC
48	55	Configurable	XMC	XMC global interrupt	0x0000_0100
49	56	Configurable	SDIO	SDIO global interrupt	0x0000_0104
50	57	Configurable	TMR5	TMR5 global interrupt	0x0000_0108
51	58	Configurable	SPI3_I²S3EXT	SPI3 and I²S3EXT global interrupt	0x0000_010C
52	59	Configurable	UART4	UART4 global interrupt	0x0000_0110
53	60	Configurable	UART5	UART5 global interrupt	0x0000_0114
54	61	Configurable	TMR6	TMR6 global interrupt	0x0000_0118
55	62	Configurable	TMR7	TMR7 global interrupt	0x0000_011C
56	63	Configurable	DMA2 channel1	DMA2 channel1 global interrupt	0x0000_0120
57	64	Configurable	DMA2 channel2	DMA2 channel2 global interrupt	0x0000_0124
58	65	Configurable	DMA2 channel3	DMA2 channel3 global interrupt	0x0000_0128
59	66	Configurable	DMA2 channel4_5	DMA2 channel4 and DMA2 channel5 global interrupt	0x0000_012C
60	67	Configurable	SDIO2	SDIO2 global interrupt	0x0000_0130
61	68	Configurable	I²C3_EVT	I²C3 event interrupt	0x0000_0134

Pos.	Priority	Priority Type	Name	Description	Address
62	69	Configurable	I²C3_ERR	I²C3 error interrupt	0x0000_0138
63	70	Configurable	SPI4	SPI4 global interrupt	0x0000_013C
64	71	—	—	Reserved	0x0000_0140
65	72	—	—	Reserved	0x0000_0144
66	73	—	—	Reserved	0x0000_0148
67	74	—	—	Reserved	0x0000_014C
68	75	Configurable	CAN2_TX	CAN2 TX interrupt	0x0000_0150
69	76	Configurable	CAN2_RX0	CAN2 RX0 interrupt	0x0000_0154
70	77	Configurable	CAN2_RX1	CAN2 RX1 interrupt	0x0000_0158
71	78	Configurable	CAN2_SE	CAN2 status error interrupt	0x0000_015C
72	79	Configurable	ACC	ACC interruupt	0x0000_0160
73	80	Configurable	USBFS_MAPH(Note)	USBFS remap high priority interrupt	0x0000_0164
74	81	Configurable	USBFS_MAPL(Note)	USBFS remap low priority interrupt	0x0000_0168
75	82	Configurable	DMA2 channel6_7	DMA2 channel6 and DMA2 channel7 global interrupt	0x0000_016C
76	83	Configurable	USART6	UART6 global interrupt	0x0000_0170
77	84	Configurable	UART7	UART7 global interrupt	0x0000_0174
78	85	Configurable	UART8	UART8 global interrupt	0x0000_0178

Note: USBFS module interrupt supports remap through the USBINTMAP bit in the CRM_INTMAP register. When USBINTMAP=0, use USBFS_H (19th) and USBFS_ L (20th) interrupts; when USBINTMAP=1, use USB_ MAPH (73rd) andUSB_MAPL (74th) interrupts.

System Tick (SysTick)

The System Tick is a 24-bit downcounter. It will be reloaded with the initial value automatically when it is decremented to zero. It can generate periodic interrupts, so it is often used as multi-task scheduling counter for embedded operating system, and also to call the periodic tasks for non-embedded system.

The System Tick calibration value is fixed to 9000, which gives a reference time base of 1 ms when the System Tick clock is set to 9 MHz.

Reset

The processor reads the first two words from the CODE memory after a system reset and before program execution.

- Get the initial value of the main stack pointer (MSP) from address 0x0000_0000
- Get the initial value of the program counter (PC) from address 0x0000_0004. This value is a reset vector and LSB must be 1. Then take the instructions from the address corresponding to this value.

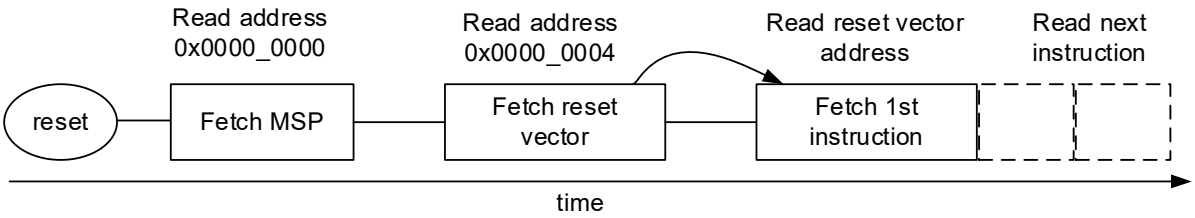


Figure 5. Reset process

Cortex™-M4F uses a full stack that increases downward, so the initial value of the main stack pointer (MSP) must be the end address of the stack memory plus 1. For example, if the stack area is set between 0x2000_7C00 and 0x2000_7FFF, then the initial value of MSP must be defined as 0x2000_8000.

The vector table follows the initial value of MSP. Cortex™ -M4F operates in Thumb state, and thus each data value in the vetor table must set the LSB to 1. In the following diagram, 0x0000_0101 is used to represent 0x0000_0100. After the instruction at 0x0000_0100 is executed, the program starts running formally. Before that, it is a must for initializing MSP, because the first instruction may be interrupted by NMI or other faults before being executed. After the completion of MSP initialization, it is ready to prepare stack room for its service routines.

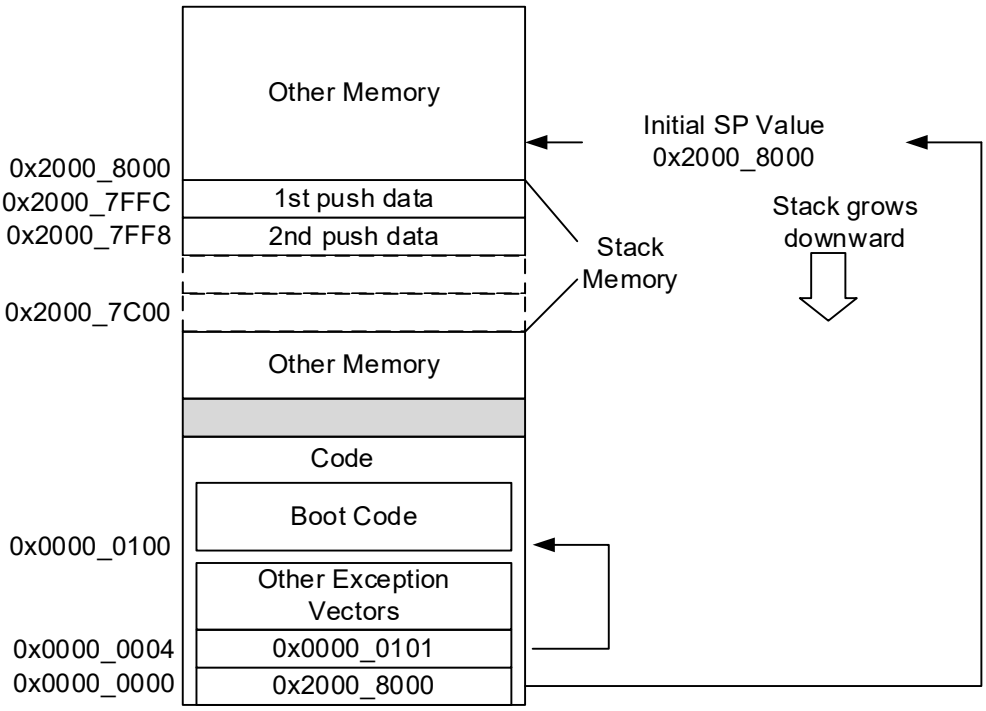


Figure 6. Example of MSP and PC initialization

In the HT32F49365/HT32F49395 series, the main Flash memory, Boot code or SRAM can be remapped to the code area between 0x0000_0000 and 0x07FF_FFFF. BOOT1 and BOOT0 are used to set the specific memory from which CODE starts.

{BOOT1, BOOT0}=00/10, CODE starts from the main Flash memory

{BOOT1, BOOT0}=01, CODE starts from Boot code

{BOOT1, BOOT0}=11, CODE starts from SRAM

After a system reset or when leaving from Standby mode, the pin values of both BOOT1 and BOOT0 will be relatched.

Boot code memory contains an embedded boot loader program that provides not only Flash programming function through USART1, USART2 or USB interface, but also provides extra firmware including communication protocol stacks that can be called for use by software developer through API.

List of abbreviations for registers

Table 4. List of abbreviations for registers

Register type	Description
rw	Software can read and write to this bit.
ro	Software can only read this bit.
wo	Software can only write to the bit. Reading it returns its reset value.
rrc	Software can read this bit. Reading this bit automaticaly clears it.
rw0c	Software can read this bit and clear it by writing 0. Writing 1 has no effect on this bit.
rw1c	Software can read this bit and clear it by writing 1. Writing 0 has no effect on this bit.
rw1s	Software can read this bit and set it by writing 1. Writing 0 has no effect on this bit.
tog	Software can read this bit and torggle it by writing 1. Writing 0 has no effect on this bit.
rwt	Software can read this bit. Writng any value will trigger an event.
resd	Reserved.

Device characteristics information

Table 5. List of abbreviations for registers

Register abbr.	Base address	Reset value
F_SIZE	0x1FFF F7E0	0xXXXX
UID[31:0]	0x1FFF F7E8	0xXXXX XXXX
UID[63:32]	0x1FFF F7EC	0xXXXX XXXX
UID[95:64]	0x1FFF F7F0	0xXXXX XXXX

Flash memory size register

This register contains the information about Flash memory size.

Bit	Register	Reset value	Type	Descrption
Bit 15:0	F_SIZE	0xXXXX	ro	Flash size, in terms of KByte For example:0x0080 = 128 KByte

Device electronic signature

The device electronic signature contains the memory size and the unique device ID (96 bits). It is stored in the information block of the Flash memory. The 96-bit ID is unique for any device, and cannot be altered by users. It can be used for the following:

- Serial number: such as USB string serial number
- Part of security keys

Bit	Register	Reset value	Type	Description
Bit 31:0	UID[31:0]	0xXXXX XXXX	ro	UID for bit 31 to bit 0

Bit	Register	Reset value	Type	Description
Bit 31:0	UID[63:32]	0xXXXX XXXX	ro	UID for bit 63 to bit 32

Bit	Register	Reset value	Type	Description
Bit 31:0	UID[95:64]	0xXXXX XXXX	ro	UID for bit 95 to bit 64

Note: UID[95:88] is series ID, which is 0x99 for HT32F49365/HT32F49395.

2 Memory resources

Internal memory address map

Internal memory contains program memory (Flash), data memory (SRAM), peripheral registers and core registers. Their respective address mapping are shown in the following diagram.

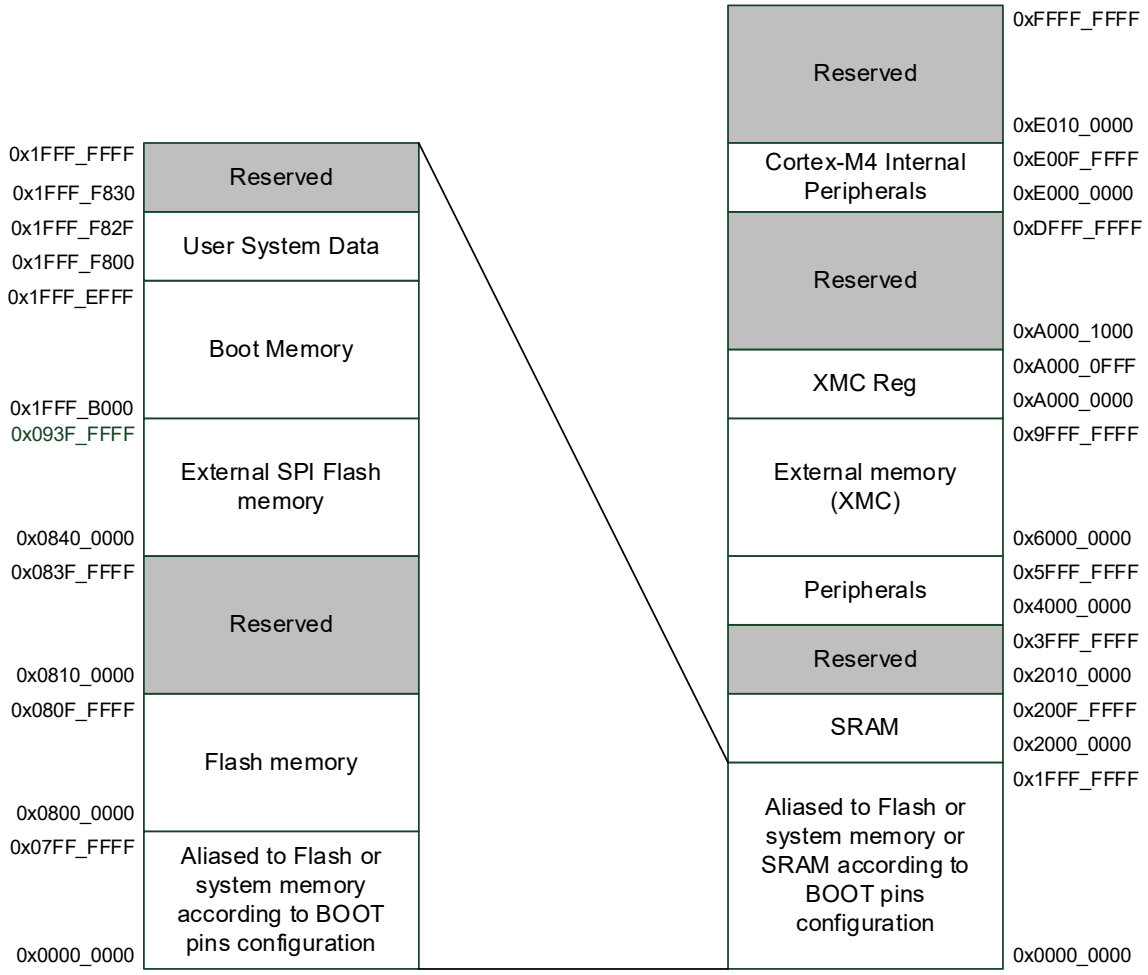


Figure 7. HT32F49365/HT32F49395 address mapping

Flash memory

HT32F49365/HT32F49395 series provide up to 1024 KB of on-chip Flash memory, supporting a zero wait state single cycle 32-bit read operation.

Refer to Chapter 5 for more details about Flash memory controller and register configuration.

Flash memory organization (1024K)

The main memory is divided into bank 1 and bank 2. 512 Kbytes, 256 pages per bank, and 2 Kbytes per page.

External memory size can be up to 16 Mbytes, including 4096 pages, and 4 Kbytes per page.

Table 6. Flash memory organization (1024K main memory)

Block		Name	Address range
Main memory	Bank 1 512 KB	Page 0	0x0800 0000 – 0x0800 07FF
		Page 1	0x0800 0800 – 0x0800 0FFF
		Page 2	0x0800 1000 – 0x0800 17FF
		…	…
		Page 255	0x0807 F800 – 0x0807 FFFF
	Bank2 512 KB	Page 256	0x0808 0000 – 0x0808 07FF
		Page 257	0x0808 0800 – 0x0808 0FFF
		Page 258	0x0808 1000 – 0x0808 17FF
		…	…
		Page 511	0x080F F800 – 0x080F FFFF
External memory	16 MB	Page 0	0x0840 0000 – 0x0840 0FFF
		Page 1	0x0840 1000 – 0x0840 1FFF
		Page 2	0x0840 2000 – 0x0840 2FFF
		…	…
		Page 4095	0x093F F000 – 0x093F FFFF
Information block		Boot loader 16 KB	0x1FFF B000 – 0x1FFF EFFF
		User system data 48 B	0x1FFF F800 – 0x1FFF F82F

Flash memory organization (512K)

The main memory contains only one bank of 512 Kbytes, including 256 pages and 2 Kbytes per page.

External memory size can be up to 16 Mbytes, including 4096 pages, and 4 Kbytes per page.

Table 7. Flash memory organization (512K main memory)

Block		Name	Address range
Main memory	Bank 1 512 KB	Page 0	0x0800 0000 – 0x0800 07FF
		Page 1	0x0800 0800 – 0x0800 0FFF
		Page 2	0x0800 1000 – 0x0800 17FF
		…	…
		Page 255	0x0807 F800 – 0x0807 FFFF
External memory	16 MB	Page 0	0x0840 0000 – 0x0840 0FFF
		Page 1	0x0840 1000 – 0x0840 1FFF
		Page 2	0x0840 2000 – 0x0840 2FFF
		…	…
		Page 4095	0x093F F000 – 0x093F FFFF

Block	Name	Address range
Information block	Boot loader 16KB	0x1FFF B000 – 0x1FFF EFFF
	User system data 48B	0x1FFF F800 – 0x1FFF F82F

Flash memory organization (256K)

The main memory contains only one bank of 256 Kbytes, including 128 pages and 2 Kbytes per page.

External memory size can be up to 16 Mbytes, including 4096 pages, and 4 Kbytes per page.

Table 8. Flash memory organization (256K main memory)

Block		Name	Address range
Main memory	Bank 1 256 KB	Page 0	0x0800 0000 – 0x0800 07FF
		Page 1	0x0800 0800 – 0x0800 0FFF
		Page 2	0x0800 1000 – 0x0800 17FF
		…	…
		Page 127	0x0803 F800 – 0x0803 FFFF
External memory	16 MB	Page 0	0x0840 0000 – 0x0840 0FFF
		Page 1	0x0840 1000 – 0x0840 1FFF
		Page 2	0x0840 2000 – 0x0840 2FFF
		…	…
		Page 4095	0x093F F000 – 0x093F FFFF
Information block		Boot loader 16 KB	0x1FFF B000 – 0x1FFF EFFF
		User system data 48 B	0x1FFF F800 – 0x1FFF F82F

SRAM memory

The HT32F49365/HT32F49395 series contain up to 96 KB of on-chip SRAM which starts at the address 0x2000_0000. It supports byte, half-word (16 bit) and word (32 bit) accesses. In addition, HT32F49365/HT32F49395 also provide a special mode that supports dynamic switch between 96 KB and 224 KB. This is done by setting the EOPB0 bit. In 224 KB mode, Flash memory size (zero wait sate) is limited to 128 KB, while in 96 KB mode, the zero-wait-state Flash size is limited to 256 KB.

Peripheral address map

Table 9. Peripheral boundary address

Bus	Boundary address	Peripherals
AHB	0A000 1000 - 0xFFFF FFFF	Reserved
	0A000 0000 - 0xA000 0FFF	XMC_REG
	0x6000 0000 - 0x9FFF FFFF	XMC_MEM
	0x4002 A000 - 0x5FFF FFFF	Reserved
	0x4002 8000 - 0x4002 9FFF	Reserved
	0x4002 3400 - 0x4002 7FFF	SDIO2
	0x4002 3000 - 0x4002 33FF	CRC
	0x4002 2000 - 0x4002 23FF	Flash memory interface (FLASH)
	0x4002 1400 - 0x4002 1FFF	Reserved
	0x4002 1000 - 0x4002 13FF	Clock reset manage (CRM)
	0x4002 0800 - 0x4002 0FFF	Reserved
	0x4002 0400 - 0x4002 07FF	DMA2
	0x4002 0000 - 0x4002 03FF	DMA1
	0x4001 8400 - 0x4001 7FFF	Reserved
	0x4001 8000 - 0x4001 83FF	SDIO
APB2	0x4001 7400 - 0x4001 7FFF	Reserved
	0x4001 7000 - 0x4001 73FF	I^2S3EXT
	0x4001 6C00 - 0x4001 6FFF	I^2S2EXT
	0x4001 6800 - 0x4001 6BFF	UART8
	0x4001 6400 - 0x4001 67FF	UART7
	0x4001 6000 - 0x4001 63FF	USART6
	0x4001 5C00 - 0x4001 5FFF	I^2C3
	0x4001 5800 - 0x4001 5BFF	ACC
	0x4001 5400 - 0x4001 57FF	TMR11 timer
	0x4001 5000 - 0x4001 53FF	TMR10 timer
	0x4001 4C00 - 0x4001 4FFF	TMR9 timer
	0x4001 4400 - 0x4001 4BFF	Reserved
	0x4001 4000 - 0x4001 43FF	Reserved
	0x4001 3C00 - 0x4001 3FFF	ADC3
	0x4001 3800 - 0x4001 3BFF	USART1
	0x4001 3400 - 0x4001 37FF	TMR8 timer
	0x4001 3000 - 0x4001 33FF	SPI1/I^2S1
	0x4001 2C00 - 0x4001 2FFF	TMR1 timer
	0x4001 2800 - 0x4001 2BFF	ADC2
	0x4001 2400 - 0x4001 27FF	ADC1
	0x4001 2000 - 0x4001 23FF	Reserved
	0x4001 1C00 - 0x4001 1FFF	Reserved
	0x4001 1800 - 0x4001 1BFF	GPIO port E
	0x4001 1400 - 0x4001 17FF	GPIO port D
	0x4001 1000 - 0x4001 13FF	GPIO port C
	0X4001 0C00 - 0x4001 0FFF	GPIO port B
	0x4001 0800 - 0x4001 0BFF	GPIO port A
	0x4001 0400 - 0x4001 07FF	EXINT
	0x4001 0000 - 0x4001 03FF	IOMUX

Bus	Boundary address	Peripherals
APB1	0x4000 8400 - 0x4000 FFFF	Reserved
	0x4000 7800 - 0x4000 83FF	USBFS 1280 bytes buffer(Note)
	0x4000 7400 - 0x4000 77FF	DAC
	0x4000 7000 - 0x4000 73FF	Power control (PWC)
	0x4000 6C00 - 0x4000 6FFF	Backup registers (BPR)
	0x4000 6800 - 0x4000 6BFF	CAN2
	0x4000 6400 - 0x4000 67FF	CAN1
	0x4000 6000 - 0x4000 63FF	USBFS 512 bytes buffer(Note)
	0x4000 5C00 - 0x4000 5FFF	USBFS
	0x4000 5800 - 0x4000 5BFF	I^2C2
	0x4000 5400 - 0x4000 57FF	I^2C1
	0x4000 5000 - 0x4000 53FF	UART5
	0x4000 4C00 - 0x4000 4FFF	UART4
	0x4000 4800 - 0x4000 4BFF	USART3
	0x4000 4400 - 0x4000 47FF	USART2
	0x4000 4000 - 0x4000 43FF	SPI4/I^2S4
	0x4000 3C00 - 0x4000 3FFF	SPI3/I^2S3
	0x4000 3800 - 0x4000 3BFF	SPI2/I^2S2
	0x4000 3400 - 0x4000 37FF	Reserved
	0x4000 3000 - 0x4000 33FF	Watchdog timer (WDT)
	0x4000 2C00 - 0x4000 2FFF	Window watchdog timer (WWDT)
	0x4000 2800 - 0x4000 2BFF	RTC
	0x4000 2400 - 0x4000 27FF	Reserved
	0x4000 2000 - 0x4000 23FF	TMR14 timer
	0x4000 1C00 - 0x4000 1FFF	TMR13 timer
	0x4000 1800 - 0x4000 1BFF	TMR12 timer
	0x4000 1400 - 0x4000 17FF	TMR7 timer
	0x4000 1000 - 0x4000 13FF	TMR6 timer
	0x4000 0C00 - 0x4000 0FFF	TMR5 timer
	0x4000 0800 - 0x4000 0BFF	TMR4 timer
	0x4000 0400 - 0x4000 07FF	TMR3 timer
	0x4000 0000 - 0x4000 03FF	TMR2 timer

Note: When USBBUFS = 0, USBFS buffer size is 512 bytes, its address is 0x4000 6000 ~ 0x400063FF. When USBBUFS = 1, USBFS buffer size is 768 ~ 1280 bytes, its address is 0x4000 7800 ~ 0x4000 83FF. If both CAN1 and CAN2 are not enabled, the maximum USBFS buffer can be set to 1280 bytes; If any one of them is enabled, the maximum USBFS buffer can be up to 1024 bytes; If both are enabled, the maximum USB buffer can be set to 768 bytes.

3 Power control (PWC)

Introduction

For HT32F49365/HT32F49395 series, its operating voltage supply is 2.6 V ~ 3.6 V, with a temperature range of -40 ~ +105 °C. To reduce power consumption, this series provides three types of power saving modes, including Sleep, Deepsleep and Standby modes so as to achieve the best tradeoff among the conflicting demands of CPU operating time, speed and power consumption.The HT32F49365/HT32F49395 series have three power domains – V_{DD}/V_{DDA} domain, 1.2 V domain and battery powered domain. The V_{DD}/V_{DDA} domain is supplied directly by external power, the 1.2 V domain is powered by the embedded LDO in the V_{DD}/V_{DDA} domain, and the battery powered domain is supplied through a V_{BAT} pin.

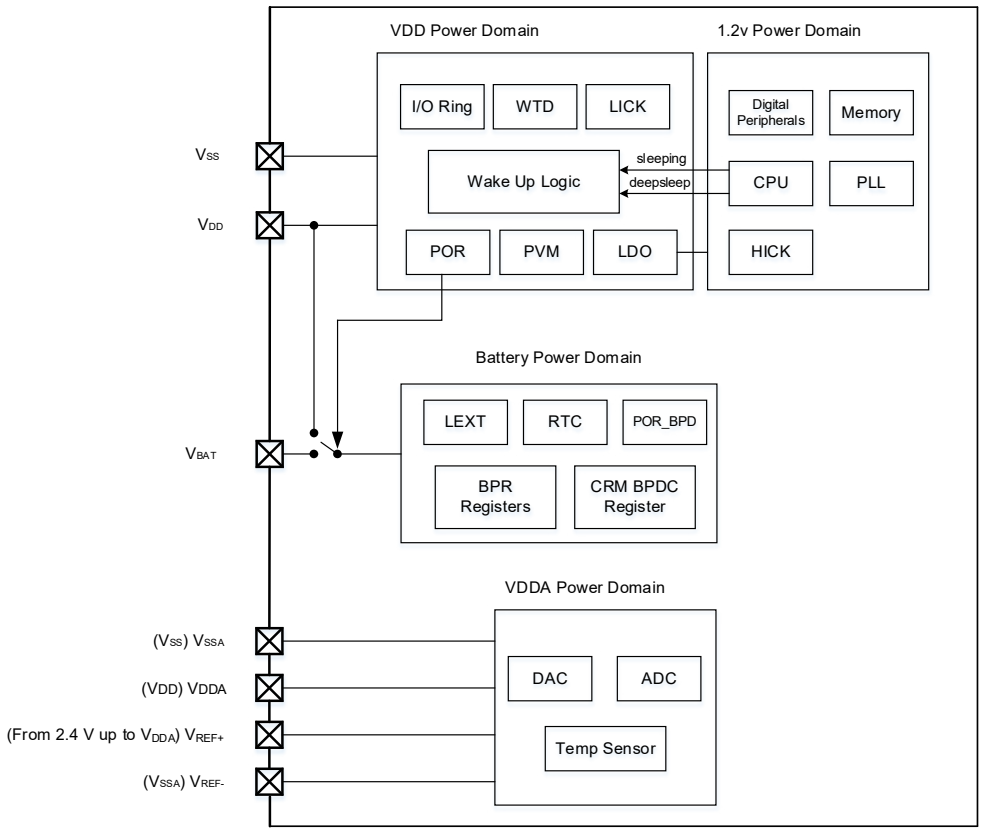


Figure 8. Block diagram of each power supply

Main Features

- Three power domains: V_{DD}/V_{DDA} domain, 1.2 V domain and battery powered domain
- Three types of power saving modes: Sleep mode, Deepsleep mode, and Standby mode
- Internal voltage regulator supplies 1.2 V voltage source for the core domain
- Power voltage detector is provided to issue an interrupt when the supply voltage is lower or higher than a programmed threshold
- The battery powered domain is powered by V_{BAT} when V_{DD} is powered off
- V_{DD}/V_{DDA} applies separated digital and analog module to reduce noise on external power

POR/LVR

A POR analog module embedded in the V_{DD}/V_{DDA} domain is used to generate a power reset. The power reset signal is released at V_{POR} when the V_{DD} is increased from 0 V to the operating voltage, or it is triggered at V_{LVR} when the V_{DD} drops from the operating voltage to 0 V. During the power-on reset period, the reset signal has certain amount of time delay compared to V_{DD} boost process. At the same time, hysteresis occurs in power-on reset (POR) and low voltage reset (LVR).

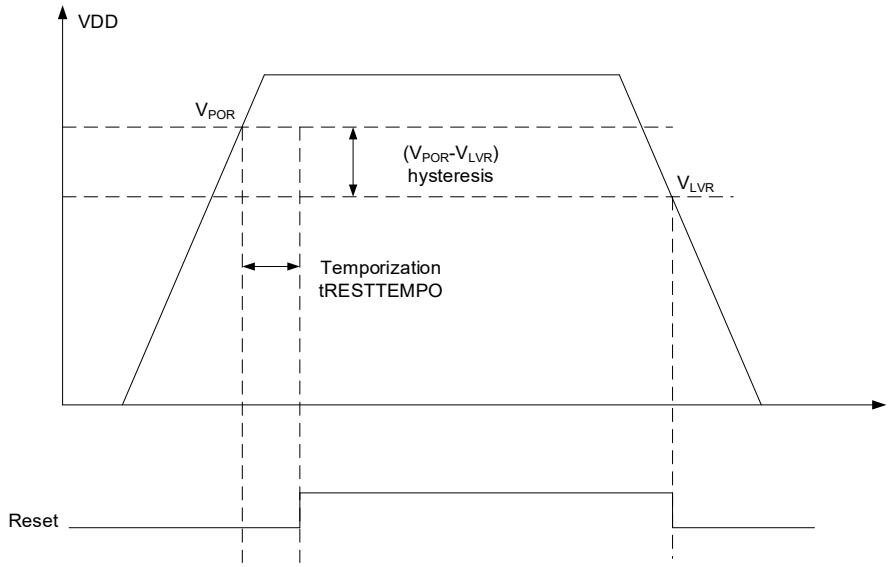


Figure 9. Power-on reset/Low voltage reset waveform

Power voltage monitor (PVM)

The PVM is used to monitor the power supply variations. It is enabled by setting the PVMEN bit in the power control register (PWC_CTRL), and the threshold value for voltage monitor is selected with the PVMSEL[2:0].

After PVM is enabled, the comparison result between V_{DD} and the programmed threshold is indicated by the PVMOF bit in the PWC_CTRLSTS register, with the hysteresis voltage V_{HYS_P} being 100 mv. The PVM interrupt will be generated through the EXTI line 16 when V_{DD} rises above the PVM threshold.

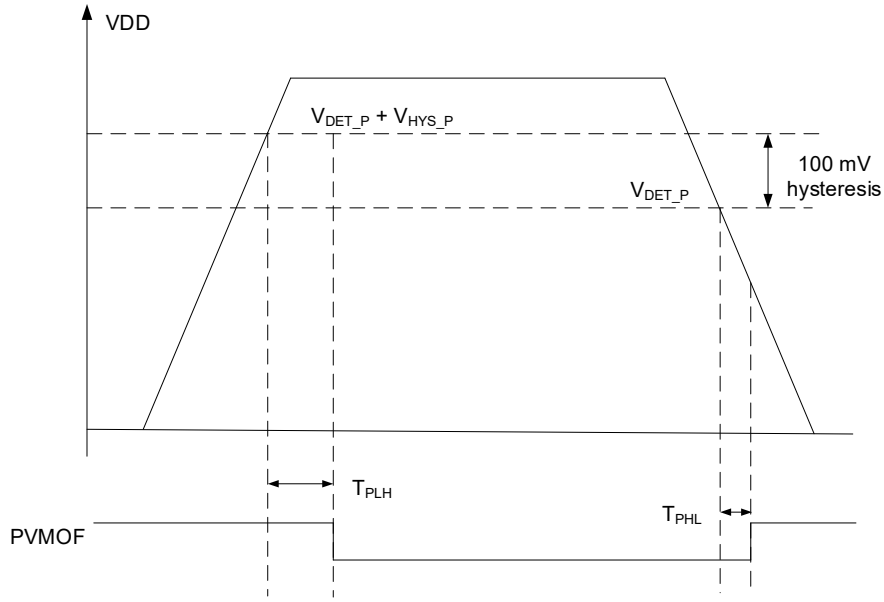


Figure 10. PVM threshold and output

Power domain

1.2 V domain

1.2 V core domain includes a CPU core, SRAM, embedded digital peripherals and Phase Locked Loop (PLL). Such power domain is supplied by LDO (voltage regulator).

V_{DD}/V_{DDA} domain

V_{DD}/V_{DDA} domain includes V_{DD} domain and V_{DDA} domain. The V_{DD} domain contains I/O circuit, power-saving mode wakeup circuit, watchdog timer, power-on reset/low voltage reset (POR/LVR), LDO and all PAD circuits other than PC13, PC14 and PC15. The V_{DDA} domain contains DAC/ADC (DA/AD converters), temperature sensor and so on.

Typically, to ensure a better accuracy of ADC/DAC at a low voltage, the digital circuit is supplied by V_{DD} while the analog circuit is powered by V_{DDA}. On 64-pin packages and packages with less pins, the external reference voltage V_{REF+} and V_{REF-} are connected to the V_{DDA} pin and V_{SSA} pin, respectively.

Battery powered domain

The battery powered domain contains RTC circuit, LEXT oscillator, PC13, PC14 and PC15, which is powered by either V_{DD} or V_{BAT} pin. When the V_{DD} is cut off, the battery powered domain is automatically switched to V_{BAT} pin to ensure that RTC can work normally.

1. When the battery powered domain is powered by V_{DD}, the PC13 can be used as a general-purpose I/O, tamper pin, RTC calibration clock, RTC alarm or second output, while the PC14 and PC15 can be used as a GPIO or LEXT pin. (As an I/O port, PC13, PC14 and PC15 must be limited below 2 MHz, and to the maximum load of 30 pF, and these I/O ports must not be used as current sources)

2. When the battery powered domain is powered by V_{BAT}, the PC13 can be used as a a tamper pin, RTC alarm or second output, while the PC14 and PC15 can only be used as a LEXT pin.

The switch of the battery powered domain will not be disconnected from V_{BAT} because of the V_{DD} being at its rising phrase or due to V_{DD} low voltage reset. If the power switch has not been switched to the V_{DD} when the V_{DD} is powered on quickly, it is recommended to add a low votage drop diode between V_{DD} and V_{BAT} in order to prevent the currents of V_{DD} from being injected to V_{BAT}. If there is no external battery in the application, it is better to connet the V_{BAT} to a 100 nF ceramic filter capacitor that is externally connected to V_{DD}.

Power saving modes

When the CPU does not need to be kept running, there are three types of low-power modes available (Sleep mode, Deepsleep mode and Standby mode) to save power. Users can select the mode that gives the best compromise according to the low-power consumption, short startup time, and available wakeup sources. In addition, the power consumption in Run mode can be reduced by slowing down the system clocks or gating the clocks on the APB and AHB peripherals when they are not used.

Sleep mode

The Sleep mode is entered by executing WFI or WFE command. There are two options to select the Sleep mode entry mechanism through the SLEEPONEXIT bit in the Cortex®-M4F system control register.

SLEEP-NOW mode

When SLEEPDEEP=0 and SLEEPONEXIT=0, the MCU enters Sleep mode as soon as WFI or WFE instruction is executed.

SLEEP-ON-EXIT mode

When SLEEPDEEP=0 and SLEEPONEXIT=1, by execitomg the WFI instruction, the MCU enters Sleep mode as soon as the system exits the lowest-priority interrupt service routine.

In Sleep mode, all clocks and LDO work normally except CPU clocks (stopped), and all I/O pins keep the same state as in Run mode. The LDO provides an 1.2 V power (for CPU core, memory and embedded peripherals) as it is in normal power consumption mode.

1. If the WFI is executed to enter Sleep mode, any peripheral interrupt can wake up the device from Sleep mode.

2. If the WFE is executed to enter Sleep mode, the MCU exits Sleep mode as soon as an event occurs. The wakeup event can be generated by the following:
   - Enabling a peripheral interrupt (it is not enabled in the NVIC) and enabling the SEVONPEND bit. When the MCU resumes, the peripheral interrupt pending bit and NVIC channel pending bit must be cleared.
   - Configuring an internal EXINT line as an event mode to generate a wakeup event.

The wakeup time required by a WFE command is the shortest, since no time is wasted on interrupt entry/exit.

Deepsleep Mode

Deepsleep mode is entered by setting the SLEEPDEEP bit in the Cortex™-M4F system control register and clearing the LPSEL bit in the power control register before executing WFI or WFE instruction.

The LDO status is selected by setting the VRSEL bit in the power control register (PWC_CTRL). When VRSEL=0, the LDO works in normal mode. When VRSEL=1, the LDO is set in low-power consumption mode.

In Deepsleep mode, all clocks in 1.2 V domain are stopped, and both HICK and HEXT oscillators are disabled. The LDO supplies power to the 1.2 V domain in normal mode or low-power mode. All I/O pins keep the same state as in Run mode. SRAM and register contents are preserved.

1. When the Sleep mode is entered by executing a WFI instruction, the interrupt generated on any external interrupt line in Interrupt mode can wake up the system from Deepsleep mode.
2. When the Sleep mode is entered by executing a WFE instruction, the interrupt generated on any external interrupt line in Event mode can wake up the system from Deepsleep mode.

When the MCU exits the Deepsleep mode, the HICK RC oscillator is enabled and selected as a system clock after stabilization.When the LDO operates in low-power mode, an additional wakeup delay is incurred for the reason that the LDO must be stabilized before the system is waken from the Deepsleep mode.

Standby Mode

Standby mode can achieve the lowest power consumption for the device. In this mode, the LDO is disabled. The whole 1.2 V domain, PLL, HICK and HEXT oscillators are also powered off. SRAM and register contents are lost. Only registers in the battery powered domain and standby circuitry remain supplied.

The Standby mode is entered by the following procedures:

- Set the SLEEPDEE bit in the Cortex™-M4F system control register
- Set the LPSEL bit in the power control register (PWC_CTRL)
- Clear the SWEF bit in the power control/status register (PWC_CTRLSTS)
- Execute a WFI/WFE instruction

In Standby mode, all I/O pins remain in a high-impedance state except reset pins, TAMPER pins that are set as anti-tamper or calibration output, and the wakeup pins enabled.

The MCU leaves the Standby mode when an external reset (NRST pin), an WDT reset, a rising edge on the WKUP pin or the rising edge of an RTC alarm even occurs.

Debug mode

By default, the debug connection is lost if the MCU enters Deepsleep mode or Standby mode while debugging. The reason is that the Cortex™-M4F core is no longer clocked. However, the software can be debugged even in the low-power mode by setting some configuration bits in the DEBUG register (DEBUG_CTRL).

PWC registers

The peripheral registers can be accessed by half-words (16 bit) or words (32 bit)

Table 10. PW register map and reset values

Register abbr.	Offset	Reset value
PWC_CTRL	0x00	0x0000 0000
PWC_CTRLSTS	0x04	0x0000 0000

Power control register (PWC_CTRL)

Bit	Register	Reset value	Type	Description
Bit 31:9	Reserved	0x000000	resd	Kept at its default value.
Bit 8	BPWEN	0x0	rw	Battery powered domain write enable 0: Disabled 1: Enabled Note: After reset, the battery powered domain write access is disabled. To write, this bit must be enabled.
Bit 7:5	PVMSEL	0x0	rw	Power voltage monitoring boundary select 000: Unused, not configurable 001: 2.3 V 010: 2.4 V 011: 2.5 V 100: 2.6 V 101: 2.7 V 110: 2.8 V 111: 2.9 V
Bit 4	PVMEN	0x0	rw	Power voltage monitoring enable 0: Disabled 1: Enabled
Bit 3	CLSEF	0x0	wo	Clear SEF flag 0: No effect 1: Clear the SEF flag Note: This bit is cleared by hardware after clearing the SEF flag. Reading this bit at any time will return all zero.
Bit 2	CLSWEF	0x0	wo	Clear SWEF flag 0: No effect 1: Clear the SWEF flag Note: Clear the SWEF flag after two system clock cycles. This bit is cleared by hardware after clearing the SWEF flag. Reading this bit at any time will return all zero.
Bit 1	LPSEL	0x0	rw	Low power mode select in sleepdeep 0: Enter DEEPSLEEP mode 1: Enter Standby mode
Bit 0	VRSEL	0x0	rw	LDO state select in deepsleep mode 0: Enabled 1: Low-power consumption mode

Power control/status register (PWC_CTRLSTS)

Additional APB cycles are needed to read this register versus a standard APB read.

Bit	Register	Reset value	Type	Description
Bit 31:9	Reserved	0x000000	resd	Kept at its default value.
Bit 8	SWPEN	0x0	rw	Standby wake-up pin enable 0: Disabled (this pin is used for general-purpose I/O) 1: Enabled (this pin is forced in input pull-down mode, and no longer used for general-purpose I/O) Note: This bit is cleared by hardware after system reset.
Bit 7:3	Reserved	0x00	resd	Keep at its default value.
Bit 2	PVMOF	0x0	ro	Power voltage monitoring output flag 0: Power voltage is higher than the threshold 1: Power voltage is lower than the threshold Note: The power voltage monitor is stopped in Standby mode.
Bit 1	SEF	0x0	ro	Standby mode entry flag 0: Device is not in Standby mode 1: Device is in Standby mode Note: This bit is set by hardware (enter Standby mode) and cleared by POR/LVR or by setting the CLSEF bit.
Bit 0	SWEF	0x0	ro	Standby wake-up event flag 0: No wakeup event occurred 1: A wakeup event occurred Note: This bit is set by hardware (on an wakeup event), and cleared by POR/LVR or by setting the CLSWEF bit. A wakeup event is generated by one of the following: When the rising edge on the Standby wakeup pin occurs; When the RTC alarm event occurs; If the Standby wakeup pin is enabled when the Standby wakeup pin level is high.

4 Clock and reset manage (CRM)

Clock

HT32F49365/HT32F49395 series provide different clock sources: HEXT oscillator clock, HICK oscillator clock, PLL clock, LEXT oscillator and LICK oscillator.

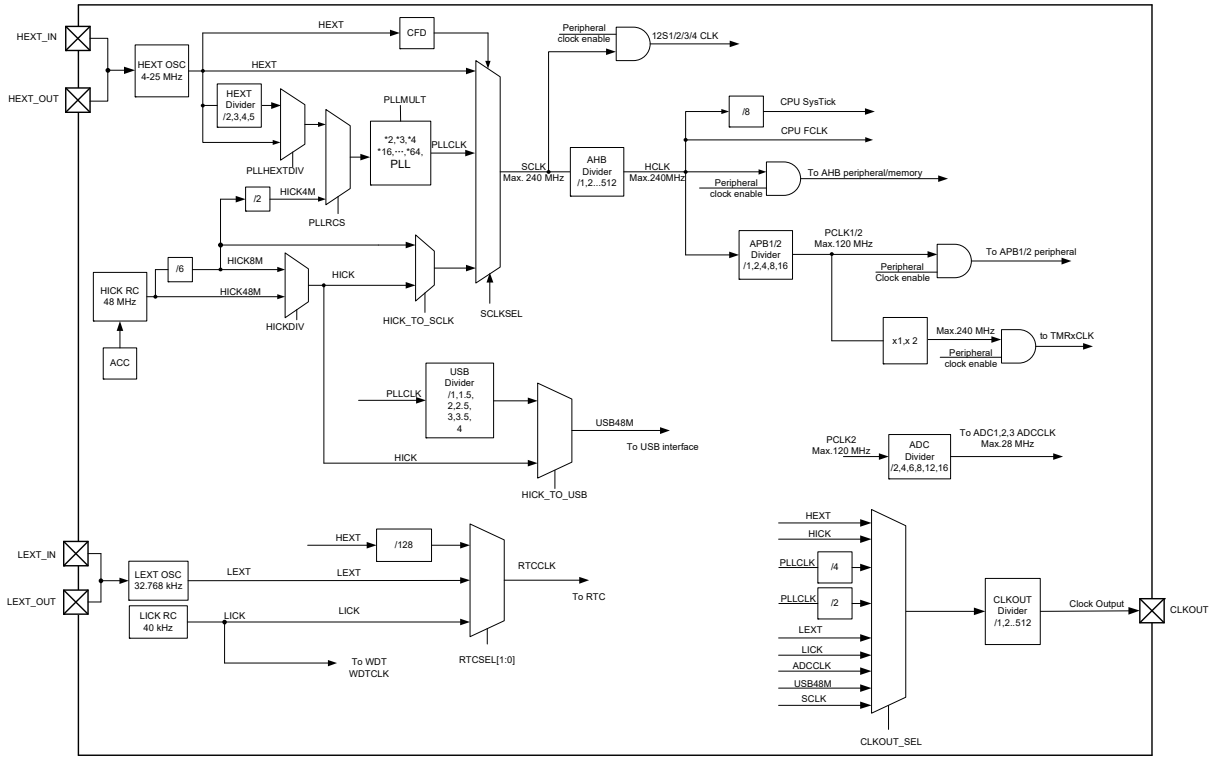


Figure 11. Clock tree

AHB, APB1 and APB2 all support multiple frequency division. The AHB domain has a maximum of 240 MHz, and both APB1 and APB2 are up to 120 MHz.

Clock sources

- High speed external oscillator (HEXT)

  The HEXT includes two clock sources: crystal/ceramic resonator and bypass clock.

  The HEXT crystal/ceramic resonator is connected externally to a 4~25MHz HEXT crystal that produces a highly accurate clock for the system. The HEXT clock signal is not released until it becomes stable.

  An external clock source can be provided by HEXT bypass. Its frequency can be up to 25 MHz. The external clock signal should be connected to the HEXT_IN pin while the HEXT_OUT pin should be left floating.

- High speed internal clock (HICK)

  The HICK oscillator is clocked by a high-speed RC in the microcontroller. The internal frequency of the HICK clock is 48 MHz. Although it is less accurate, its startup time is shorter than the HEXT crystal oscillator. The HICK clock frequency of each device is calibrated for 1% accuracy (25°C) in factory. The factory calibration value is loaded in the HICKCAL[7:0] bit of the clock control register. The RC oscillator speed may be affected by voltage or temperature variations. Thus the HICK frequency can be trimmed using the HICKTRIM[5:0] bit in the clock control register.

  The HICK clock signal is not released until it becomes stable.

- PLL clock

  The HICK or HEXT clock can be used as an input clock source of PLL, and the input clock rangs from 2 M to 16 MHz. The input clock source and multiplication factor must be configured before enabling the PLL. Otherwise, once the PLL is enabled, these parameters cannot be changed. The PLL clock signal is not released until it becomes stable.

- Low speed external oscillator (LEXT)

  The LEXT oscillator provides two clock sources: LEXT crystal/ceramic resonator and LEXT bypass.

  LEXT crystal/ceramic resonator:

  The LEXT crystal/ceramic resonator provides a 32.768 KHz low-speed clock source. The LEXT clock signal is not released until it becomes stable.

- LEXT bypass clock

  In this mode, an external clock source with a frequency of 32.768 kHzcan be provided. The external clock signal should be connected to the LEXT_IN pin while the LEXT_OUT remains floating.

- Low speed internal RC oscillator (LICK)

  The LICK oscillator is clocked by an internal low-speed RC oscillator. The clock frequency is between 30 kHz and 60 kHz. It acts as a low-power clock source that can be kept running in Deepsleep mode and Standby mode for watchdog and auto-wakeup unit.

  The LICK clock signal is not released until it becomes stable.

System clock

After a system reset, the HICK oscillator is selected as system clock. The system clock can make flexible switch among HICK oscillator, HEXT oscillator and PLL clock. However, a switch from one clock source to another occurs only if the target clock source becomes stable. When the HICK oscillator is used directly or indirectly through the PLL as the system clock, it cannot be stopped.

Peripheral clock

Most peripherals use HCLK, PCLK1 or PCLK2 clock. The individual peripherals have their dedicated clocks.

System Tick timer (SysTick) is clocked by HCLK or HCLK/8.

ADCs are clocked by APB2 divided by 2, 4, 6, 8, 12 or 16.

The timers are clocked by APB1/2. In particular, if the APB prescaler is 1, the timer clock frequency is equal to that of APB1/2; otherwise, the timer clock frequency is set to double the APB1/2 frequency.

The USB clock source can be switched between HICK and PLL frequency divider. If the HICK is selected as the clock source, the USB clock should be set as 48 MHz; If the PLL frequency divider is selected as the clock source, the USB frequency divider provides 48 MHz USBCLK, and thus the PLL needs to be set as 48 × N × 0.5 MHz (N=2, 3, 4, 5…)

RTC clock sources: HEXT/128, LEXT oscillator and LICK oscillator. Once the clock source is selected, it cannot be altered without resetting the battery powered domain. If the LEXT is used as RTC clock, the RTC is not affected when the VDD is powered off. If the HEXT or LICK is selected as RTC clock, the RTC state is not guaranteed when both HEXT and LICK are powered off.

Watchdog is clocked by LICK oscillator. If the watchdog is enabled by either hardware option or software access, the LICK oscillator is forced ON. The clock is provided to the watchdog only after the LICK oscillator temporization.

Clock fail detector

The clock fail detector (CFD) is designed to respond to HEXT clock failure when the HEXT is used as the system clock ,directly or indirectly. If a failure is detected on the HEXT clock, a clock failure event is sent to the break input of TMR1 and TMR8 and an interrupt is generated. This interrrpt is directly linked to CPU NMI so that the software can perform rescue operations. The NMI interrupt keeps executing until the CFD interrupt pending bit is cleared. This is why the CFD interrupt has to be cleared in the NMI service rounte. The HEXT clock failure will result in a switch of the system clock to the HICK clock, the CFD to be disabled , HEXT clock to be stopped, and even PLL to be disabled if the HEXT clock is selected as the system clock through PLL.

Auto step-by-step system clock switch

The automatic frequency switch is designed to ensure a smooth and stable switch of system frequency when the system clock source is switched from others to the PLL or when the AHB prescaler is changed from large to small. When the operational target is larger than 108 MHz, it is recommended to enable the automatic frequency switch. Once it is enabled, the AHB bus is halted by hardware till the completion of the switch. During this switch period, the DMA remain working, and the interrupt events are recorded and then handled by NVIC when the AHB bus resumes.

Internal clock output

The microcontroller allows the internal clock signal to be output to an external CLKOUT pin. That is, ADC CLK, USB48M, SCLK, LICK, LEXT, HICK, HEXT, PLL/2 and PLL/4 can be output to the CLKOUT pin.

Interrupts

The microcontroller specifies a stable flag for each clock source. As a result, when a clock source is enabled, it is possible to determine if the clock is stable by checking the flag pertaining to the clock source. An interrupt request is generated when the interrupt corresponding to the clock source is enabled.

If a failure is detected on the HEXT clock, the CFD interrupt is generated. Such interrrpt is directly linked to CPU NMI.

Reset

System reset

HT32F49365/HT32F49395 series provide the following system reset sources:

- NRST reset: on the external NRST pin
- WDT reset: watchdog overflow
- WWDT reset: window watchdog overflow
- CPU software reset: Cortex™-M4F software reset
- Low-power management reset: this type of reset is enabled when entering Standby mode (by clearing the nSTDBY_RST bit in the user system data); this type of reset is enabled when entering Deepsleep mode (by clearing the nDEPSLP_RST in the user system data).
- POR reset: power-on reset
- LVR reset: low voltage reset
- When exiting Standby mode

NRST reset, WDT reset, WWDT reset, software reset and low-power management reset sets all registers to their reset values except the clock control/status register (CRM_CTRLSTS) and the battery powered domain; the power-on reset, low-voltage reset or reset generated when exiting Standby mode sets all registers to their reset values except the battery powered domain registers.

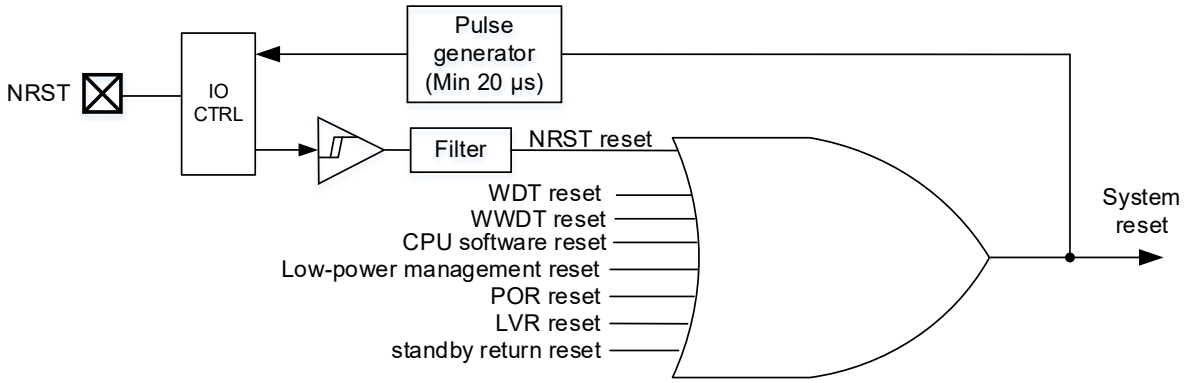


Figure 12. System reset circuit

Battery powered domain reset

Battery powered domain has two specific reset sources:

- Software reset: triggered by setting the BPDRST bit in the battery powered domain control register (CRM_BPDC)
- V_{DD} or V_{BAT} power on, if both supplies have been powered off.

Software reset affects only the battery powered domain.

CRM registers

These peripheral registers have to be accessed by bytes (8 bits), half words (16 bits) or words (32 bits).

Table 11. CRM register map and reset values

Register	Offset	Reset value
CRM_CTRL	0x000	0x0000 XX83
CRM_CFG	0x004	0x0000 0000
CRM_CLKINT	0x008	0x0000 0000
CRM_APB2RST	0x00C	0x0000 0000
CRM_APB1RST	0x010	0x0000 0000
CRM_AHBEN	0x014	0x0000 0014
CRM_APB2EN	0x018	0x0000 0000
CRM_APB1EN	0x01C	0x0000 0000
CRM_BPDC	0x020	0x0000 0000
CRM_CTRLSTS	0x024	0x0C00 0000
CRM_AHBRST	0x028	0x0000 0000
CRM_MISC1	0x030	0x0000 0000
CRM_MISC2	0x050	0x0000 0000
CRM_MISC3	0x054	0x0000 000D
CRM_INTMAP	0x05C	0x0000 0000

Clock control register (CRM_CTRL)

Bit	Register	Reset value	Type	Description
Bit 30:26	Reserved	0x00	resd	Kept at its default value.
Bit 25	PLLSTBL	0x0	ro	PLL clock stable This bit is set by hardware after PLL is ready. 0: PLL clock is not ready 1: PLL clock is ready
Bit 24	PLLEN	0x0	rw	PLL enable This bit is set and cleared by software. It can also be cleared by hardware when entering Standby or Deepsleep mode. When the PLL clock is used as the system clock, this bit cannot be cleared. 0: PLL is OFF 1: PLL is ON
Bit 23:20	Reserved	0x0	resd	Kept at its default value.
Bit 19	CFDEN	0x0	rw	Clock failure detector enable 0: OFF 1: ON

Bit	Register	Reset value	Type	Description
Bit 18	HEXTBYPS	0x0	rw	High speed external crystal bypass This bit can be written only if the HEXT is disabled. 0: OFF 1: ON
Bit 17	HEXTSTBL	0x0	ro	High speed external crystal stable This bit is set by hardware after HEXT becomes stable. 0: HEXT is not ready 1: HEXT is ready
Bit 16	HEXTEN	0x0	rw	High speed external crystal enable This bit is set and cleared by software. It can also be cleared by hardware when entering Standby or Deepsleep mode. When the HEXT clock is used as the system clock, this bit cannot be cleared 0: OFF 1: ON
Bit 15:8	HICKCAL	0xXX	rw	High speed internal clock calibration The default value of this field is the initial factory calibration value. When the HICK output frequency is 48 MHz, it needs adjust 240 kHz (design value) based on this frequency for each HICKCAL value change; when HICK output frequency is 8 MHz (design value), it needs adjust 40 kHz based on this frequency for each HICKCAL value change. Note: This bit can be written only if the HICKCAL_KEY[7:0] is set as 0x5A.
Bit 7:2	HICKTRIM	0x20	rw	High speed internal clock trimming These bits work with the HICKCAL[7:0] to determine the HICK oscillator frequency. The default value is 32, which can trim the HICK to be ±1%.
Bit 1	HICKSTBL	0x1	ro	High speed internal clock stable This bit is set by hardware after the HICK is ready. 0: Not ready 1: Ready
Bit 0	HICKEN	0x1	rw	High speed internal clock enable This bit is set and cleared by software. It can also be set by hardware when exiting Standby or Deepsleep mode. When a HEXT clock failure occurs. This bit can also be set. When the HICK is used as the sytem clock, this bit cannot be cleared. 0: Disabled 1: Enabled

Clock configuration register (CRM_CFG)

Access: 0 to 2 wait states. 1 or 2 wait states are inserted only when the access occurs during a clock source switch.

Bit	Register	Reset value	Type	Description
Bit 31	PLLRANGE	0x0	rw	PLL clock output range 0: PLL output ≤ 72 MHz 1: PLL output > 72 MHz
Bit 26:24	CLKOUT_SEL	0x0	rw	Clock output selection CLKOUT_SEL[3] is in bit 16 of the CRM_MISC1 register. 0000: Not clock output 0001: Reserved 0010: LICK 0011: LEXT 0100: SCLK 0101: HICK 0110: HEXT 0111: PLL/2 1100: PLL/4 1101: USB 1110: ADC
Bit 27 Bit 23:22	USBDIV	0x0	rw	USB division factor The divided PLL clock is used as USB clock. 000: PLL/1.5 001: PLL 010: PLL/2.5 011: PLL/2 100: PLL/3.5 101: PLL/3 110: PLL/4 111: PLL/4
Bit 30:29 Bit 21:18	PLLMULT	0x00	rw	PLL multiplication factor { Bit 30:29, Bit 21:18} 000000: PLLx 2 000001: PLLx 3 000010: PLL outputx 4 000011: PLLx 5 …… 001100: PLLx 14 001101: PLLx 15 001110: PLLx 16 001111: PLLx 16 010000: PLLx 17 010001: PLLx 18 010010: PLLx 19 010011: PLLx 20 …… 111110: PLLx 63 111111: PLLx 64 Note: The configuration of the PLLRANGE bit has to take into account of the PLL multiplication value.
Bit 17	PLLHEXTDIV	0x0	rw	HEXT division selection for PLL entry clock 0: HEXT is not divided 1: HEXT is divided according to the setting of HEXTDIV
Bit 16	PLLRCS	0x0	rw	PLL entry clock select 0: HICK clock divided (4MHz) to be PLL entry clock 1: HEXT clock is used as PLL entry clock

Bit	Register	Reset value	Type	Description
Bit 28 Bit 15:14	ADCDIV	0x0	rw	ADC division The divided PCLK is used as ADC clock. 000: PCLK/2 001: PCLK/4 010: PCLK/6 011: PCLK/8 100: PCLK/2 101: PCLK/12 110: PCLK/8 111: PCLK/16
Bit 13:11	APB2DIV	0x0	rw	APB2 division HCLK frequency division is used as APB2 clock. 0xx: HCLK is not divided 100: HCLK is divided by 2 101: HCLK is divided by 4 110: HCLK is divided by 8 111: HCLK is divided by 16 Note: These bit must be configured by software to ensure that the APB2 clock frequency is less than 120 MHz.
Bit 10:8	APB1DIV	0x0	rw	APB1 division HCLK frequency division is used as APB1 clock. 0xx: HCLK is not divided 100: HCLK is divided by 2 101: HCLK is divided by 4 110: HCLK is divided by 8 111: HCLK is divided by 16 Note: These bit must be configured by software to ensure that the APB1 clock frequency is less than 120 MHz.
Bit 7:4	AHBDIV	0x0	rw	AHB division SCLK frequency division is used as AHB clock. 0xxx: SCLK is not divided 1000: SCLK is divided by 2 1001: SCLK is divided by 4 1010: SCLK is divided by 8 1011: SCLK is divided by 16 1100: SCLK is divided by 64 1101: SCLK is divided by 128 1110: SCLK is divided by 256 1111: SCLK is divided by 512
Bit 3:2	SCLKSTS	0x0	ro	System clock select status 00: HICK 01: HEXT 10: PLL 11: Reserved, default value
Bit 1:0	SCLKSEL	0x0	rw	System clock select 00: HICK 01: HEXT 10: PLL 11: Reserved, default value

Clock interrupt register (CRM_CLKINT)

Bit	Register	Reset value	Type	Description
Bit 31:24	Reserved	0x00	resd	Kept at its default value.
Bit 23	CFDFC	0x0	wo	Clock failure detection flag clear Writing 1 by software to clear CFDF. 0: No effect 1: Clear
Bit 22:21	Reserved	0x0	resd	Kept at its default value.
Bit 20	PLLSTBLFC	0x0	wo	PLL stable flag clear Writing 1 by software to clear PLLSTBLF. 0: No effect 1: Clear
Bit 19	HEXTSTBLFC	0x0	wo	HEXT stable flag clear Writing 1 by software to clear HEXTSTBLF. 0: No effect 1: Clear
Bit 18	HICKSTBLFC	0x0	wo	HICK stable flag clear Writing 1 by software to clear HICKSTBLF. 0: No effect 1: Clear
Bit 17	LEXTSTBLFC	0x0	wo	LEXT stable flag clear Writing 1 by software to clear LEXTSTBLF. 0: No effect 1: Clear
Bit 16	LICKSTBLFC	0x0	wo	LICK stable flag clear Writing 1 by software to clear LICKSTBLF. 0: No effect 1: Clear
Bit 15:13	Reserved	0x0	resd	Kept at its default value.
Bit 12	PLLSTBLIEN	0x0	rw	PLL stable interrupt enable 0: Disabled 1: Enabled
Bit 11	HEXTSTBLIEN	0x0	rw	HEXT stable interrupt enable 0: Disabled 1: Enabled
Bit 10	HICKSTBLIEN	0x0	rw	HICK stable interrupt enable 0: Disabled 1: Enabled
Bit 9	LEXTSTBLIEN	0x0	rw	LEXT stable interrupt enable 0: Disabled 1: Enabled
Bit 8	LICKSTBLIEN	0x0	rw	LICK stable interrupt enable 0: Disabled 1: Enabled
Bit 7	CFDF	0x0	ro	Clock Failure Detection flag This bit is set by hardware when the HEXT clock failure occurs. 0: No clock failure 1: Clock failure
Bit 6:5	Reserved	0x0	resd	Kept at its default value.

Bit	Register	Reset value	Type	Description
Bit 4	PLLSTBLF	0x0	ro	PLL stable flag Set by hardware. 0: PLL is not ready. 1: PLL is ready.
Bit 3	HEXTSTBLF	0x0	ro	HEXT stable flag Set by hardware. 0: HEXT is not ready 1: HEXT is ready
Bit 2	HICKSTBLF	0x0	ro	HICK stable flag Set by hardware. 0: HICK is not ready 1: HICK is ready
Bit 1	LEXTSTBLF	0x0	ro	LEXT stable flag Set by hardware. 0: LEXT is not ready 1: LEXT is ready
Bit 0	LICKSTBLF	0x0	ro	LICK stable interrupt flag Set by hardware. 0: LICK is not ready 1: LICK is ready

APB2 peripheral reset register (CRM_APB2RST)

Bit	Register	Reset value	Type	Description
Bit 31:27	Reserved	0x00	resd	Kept at its default value.
Bit 26	UART8RST	0x0	rw	UART8 reset 0: No effect 1: Reset
Bit 25	UART7RST	0x0	rw	UART7 reset 0: No effect 1: Reset
Bit 24	UART6RST	0x0	rw	UART6 reset 0: No effect 1: Reset
Bit 23	I2C3RST	0x0	rw	I^2C3 reset 0: No effect 1: Reset
Bit 22	ACCRST	0x0	rw	ACC reset 0: No effect 1: Reset
Bit 21	TMR11RST	0x0	rw	TMR11 reset 0: No effect 1: Reset
Bit 20	TMR10RST	0x0	rw	TMR10 reset 0: No effect 1: Reset
Bit 19	TMR9RST	0x0	rw	TMR9 reset 0: No effect 1: Reset
Bit 18:16	Reserved	0x0	resd	Kept at its default value.

Bit	Register	Reset value	Type	Description
Bit 15	ADC3RST	0x0	rw	ADC3 reset 0: No effect 1: Reset
Bit 14	USART1RST	0x0	rw	USART1 reset 0: No effect 1: Reset
Bit 13	TMR8RST	0x0	rw	TMR8 reset 0: No effect 1: Reset
Bit 12	SPI1RST	0x0	rw	SPI1 reset 0: No effect 1: Reset
Bit 11	TMR1RST	0x0	rw	TMR1 reset 0: No effect 1: Reset
Bit 10	ADC2RST	0x0	rw	ADC2 reset 0: No effect 1: Reset
Bit 9	ADC1RST	0x0	rw	ADC1 reset 0: No effect 1: Reset
Bit 8:7	Reserved	0x0	resd	Kept at its default value.
Bit 6	GPIOERST	0x0	rw	GPIOE reset 0: No effect 1: Reset
Bit 5	GPIODRST	0x0	rw	GPIOD reset 0: No effect 1: Reset
Bit 4	GPIOCRST	0x0	rw	GPIOC reset 0: No effect 1: Reset
Bit 3	GPIOBRST	0x0	rw	GPIOB reset 0: No effect 1: Reset
Bit 2	GPIOARST	0x0	rw	GPIOA reset 0: No effect 1: Reset
Bit 1	EXINTRST	0x0	rw	EXINT reset 0: No effect 1: Reset Note: Always read as 0.
Bit 0	IOMUXRST	0x0	rw	IOMUX reset 0: No effect 1: Reset

APB1 peripheral reset register (CRM_APB1RST)

Bit	Register	Reset value	Type	Description
Bit 31:30	Reserved	0x0	resd	Kept at its default value.
Bit 29	DACRST	0x0	rw	DAC reset 0: No effect 1: Reset
Bit 28	PWCRST	0x0	rw	PWC reset 0: No effect 1: Reset
Bit 27	BPRRST	0x0	rw	Battery powered register reset 0: No effect 1: Reset
Bit 26	CAN2RST	0x0	rw	CAN2 reset 0: No effect 1: Reset
Bit 25	CAN1RST	0x0	rw	CAN1 reset 0: No effect 1: Reset
Bit 24	Reserved	0x0	resd	Kept at its default value.
Bit 23	USBRST	0x0	rw	USB reset 0: No effect 1: Reset
Bit 22	I2C2RST	0x0	rw	I^2C2 reset 0: No effect 1: Reset
Bit 21	I2C1RST	0x0	rw	I^2C1 reset 0: No effect 1: Reset
Bit 20	UART5RST	0x0	rw	UART5 reset 0: No effect 1: Reset
Bit 19	UART4RST	0x0	rw	UART4 reset 0: No effect 1: Reset
Bit 18	USART3RST	0x0	rw	USART3 reset 0: No effect 1: Reset
Bit 17	USART2RST	0x0	rw	USART2 reset 0: No effect 1: Reset
Bit 16	SPI4RST	0x0	rw	SPI4 reset 0: No effect 1: Reset
Bit 15	SPI3RST	0x0	rw	SPI3 reset 0: No effect 1: Reset
Bit 14	SPI2RST	0x0	rw	SPI2 reset 0: No effect 1: Reset
Bit 13:12	Reserved	0x0	resd	Kept at its default value.

Bit	Register	Reset value	Type	Description
Bit 11	WWDTRST	0x0	rw	WWDT reset 0: No effect 1: Reset
Bit 10:9	Reserved	0x0	resd	Kept at its default value.
Bit 8	TMR14RST	0x0	rw	TMR14 reset 0: No effect 1: Reset
Bit 7	TMR13RST	0x0	rw	TMR13 reset 0: No effect 1: Reset
Bit 6	TMR12RST	0x0	rw	TMR12 reset 0: No effect 1: Reset
Bit 5	TMR7RST	0x0	rw	TMR7 reset 0: No effect 1: Reset
Bit 4	TMR6RST	0x0	rw	TMR6 reset 0: No effect 1: Reset
Bit 3	TMR5RST	0x0	rw	TMR5 reset 0: No effect 1: Reset
Bit 2	TMR4RST	0x0	rw	TMR4 reset 0: No effect 1: Reset
Bit 1	TMR3RST	0x0	rw	TMR3 reset 0: No effect 1: Reset
Bit 0	TMR2RST	0x0	rw	TMR2 reset 0: No effect 1: Reset

APB peripheral clock enable register (CRM_AHBEN)

Bit	Register	Reset value	Type	Description
Bit 31:29	Reserved	0x0	resd	Kept at its default value.
Bit 28	Reserved	0x0	resd	Kept at its default value.
Bit 27:17	Reserved	0x0	resd	Kept at its default value.
Bit 16	Reserved	0x0	resd	Kept at its default value.
Bit 15	Reserved	0x0	resd	Kept at its default value.
Bit 14	Reserved	0x0	resd	Kept at its default value.
Bit 13:12	Reserved	0x0	resd	Kept at its default value.
Bit 11	SDIO2EN	0x0	rw	SDIO2 clock enable 0: Disabled 1: Enabled
Bit 10	SDIO1EN	0x0	rw	SDIO1 clock enable 0: Disabled 1: Enabled
Bit 9	Reserved	0x0	rw	Kept at its default value.
Bit 8	XMCEN	0x0	rw	XMC clock enable 0: Disabled 1: Enabled
Bit 7	Reserved	0x0	resd	Kept at its default value.
Bit 6	CRCEN	0x0	rw	CRC clock enable 0: Disabled 1: Enabled
Bit 5	Reserved	0x0	resd	Kept at its default value.
Bit 4	FLASHEN	0x1	rw	Flash clock enable This bit is used to enable Flash clock in Sleep or Deepsleep mode. 0: Disabled 1: Enabled
Bit 3	Reserved	0x0	resd	Kept at its default value.
Bit 2	SRAMEN	0x1	rw	SRAM clock enable This bit is used to enable SRAM clock in Sleep or Deepsleep mode. 0: Disabled 1: Enabled
Bit 1	DMA2EN	0x0	rw	DMA2 clock enable 0: Disabled 1: Enabled
Bit 0	DMA1EN	0x0	rw	DMA1 clock enable 0: Disabled 1: Enabled

APB2 peripheral clock enable register (CRM_AHB2EN)

Bit	Register	Reset value	Type	Description
Bit 31:27	Reserved	0x00	resd	Kept at its default value.
Bit 26	UART8EN	0x0	rw	UART8 clock enable 0: Disabled 1: Enabled
Bit 25	UART7EN	0x0	rw	UART7 clock enable 0: Disabled 1: Enabled
Bit 24	USART6EN	0x0	rw	USART6 clock enable 0: Disabled 1: Enabled
Bit 23	I2C3EN	0x0	rw	I^2C3 clock enable 0: Disabled 1: Enabled
Bit 22	ACCEN	0x0	rw	ACC clock enable 0: Disabled 1: Enabled
Bit 21	TMR11EN	0x0	rw	TMR11 clock enable 0: Disabled 1: Enabled
Bit 20	TMR10EN	0x0	rw	TMR10 clock enable 0: Disabled 1: Enabled
Bit 19	TMR9EN	0x0	rw	TMR9 clock enable 0: Disabled 1: Enabled
Bit 18:16	Reserved	0x0	resd	Kept at its default value.
Bit 15	ADC3EN	0x0	rw	ADC3 clock enable 0: Disabled 1: Enabled
Bit 14	USART1EN	0x0	rw	USART1 clock enable 0: Disabled 1: Enabled
Bit 13	TMR8EN	0x0	rw	TMR8 clock enable 0: Disabled 1: Enabled
Bit 12	SPI1EN	0x0	rw	SPI1 clock enable 0: Disabled 1: Enabled
Bit 11	TMR1EN	0x0	rw	TMR1 clock enable 0: Disabled 1: Enabled
Bit 10	ADC2EN	0x0	rw	ADC2 clock enable 0: Disabled 1: Enabled
Bit 9	ADC1EN	0x0	rw	ADC 1 clock enable 0: Disabled 1: Enabled
Bit 8:7	Reserved	0x0	rw	Kept at its default value.

Bit	Register	Reset value	Type	Description
Bit 6	GPIOEEN	0x0	rw	GPIOE clock enable 0: Disabled 1: Enabled
Bit 5	GPIODEN	0x0	rw	GPIOD clock enable 0: Disabled 1: Enabled
Bit 4	GPIOCEN	0x0	rw	GPIOC clock enable 0: Disabled 1: Enabled
Bit 3	GPIOBEN	0x0	rw	GPIOB clock enable 0: Disabled 1: Enabled
Bit 2	GPIOAEN	0x0	rw	GPIOA clock enable 0: Disabled 1: Enabled
Bit 1	Reserved	0x0	rw	Keep at its default value.
Bit 0	IOMUXEN	0x0	rw	IOMUX clock enable 0: Disabled 1: Enabled

APB1 peripheral clock enable register (CRM_AHB1EN)

Bit	Register	Reset value	Type	Description
Bit 31:30	Reserved	0x0	resd	Kept at its default value.
Bit 29	DACEN	0x0	rw	DAC clock enable 0: Disabled 1: Enabled
Bit 28	PWCEN	0x0	rw	Power control clock enable 0: Disabled 1: Enabled
Bit 27	BPREN	0x0	rw	BPR clock enable 0: Disabled 1: Enabled
Bit 26	CAN2EN	0x0	rw	CAN2 clock enable 0: Disabled 1: Enabled
Bit 25	CAN1EN	0x0	rw	CAN1 clock enable 0: Disabled 1: Enabled
Bit 24	Reserved	0x0	resd	Kept its default value.
Bit 23	USBEN	0x0	rw	USB clock enable 0: Disabled 1: Enabled
Bit 22	I2C2EN	0x0	rw	I^2C2 clock enable 0: Disabled 1: Enabled
Bit 21	I2C1EN	0x0	rw	I^2C1 clock enable 0: Disabled 1: Enabled

Bit	Register	Reset value	Type	Description
Bit 20	UART5EN	0x0	rw	UART5 clock enable 0: Disabled 1: Enabled
Bit 19	UART4EN	0x0	rw	UART4 clock enable 0: Disabled 1: Enabled
Bit 18	USART3EN	0x0	rw	USART3 clock enable 0: Disabled 1: Enabled
Bit 17	USART2EN	0x0	rw	USART2 clock enable 0: Disabled 1: Enabled
Bit 16	SPI4EN	0x0	rw	SPI4 clock enable 0: Disabled 1: Enabled
Bit 15	SPI3EN	0x0	rw	SPI3 clock enable 0: Disabled 1: Enabled
Bit 14	SPI2EN	0x0	rw	SPI2 clock enable 0: Disabled 1: Enabled
Bit 13:12	Reserved	0x0	resd	Kept at its default value.
Bit 11	WWDTEN	0x0	rw	WWDT clock enable 0: Disabled 1: Enabled
Bit 10:9	Reserved	0x0	resd	Kept its default value.
Bit 8	TMR14EN	0x0	rw	TMR14 clock enable 0: Disabled 1: Enabled
Bit 7	TMR13EN	0x0	rw	TMR13 clock enable 0: Disabled 1: Enabled
Bit 6	TMR12EN	0x0	rw	TMR12 clock enable 0: Disabled 1: Enabled
Bit 5	TMR7EN	0x0	rw	TMR7 clock enable 0: Disabled 1: Enabled
Bit 4	TMR6EN	0x0	rw	TMR6 clock enable 0: Disabled 1: Enabled
Bit 3	TMR5EN	0x0	rw	TMR5 clock enable 0: Disabled 1: Enabled
Bit 2	TMR4EN	0x0	rw	TMR4 clock enable 0: Disabled 1: Enabled
Bit 1	TMR3EN	0x0	rw	TMR3 clock enable 0: Disabled 1: Enabled

Bit	Register	Reset value	Type	Description
Bit 0	TMR2EN	0x0	rw	TMR2 clock enable 0: Disabled 1: Enabled

Battery powered domain control register (CRM_BPDC)

Access: 0 to 3 wait states. Wait states are inserted in the case of consecutive accesses to this register.

Note: LEXTEN, LEXTBYPS, RTCSEL, and RTCEN bits of the battery powered domain control register (CRM_BDC) are in the battery powered domain. As a result, these bits are write protected after reset, and can only be modified by setting the BPWEN bit in the power control register (PWR_CTRL). These bits could be reset only by battery powered domain reset. Any internal or external Reset does not affect these bits.

Bit	Register	Reset value	Type	Description
Bit 31:17	Reserved	0x0000	resd	Kept at its default value.
Bit 16	BPDRST	0x0	rw	Battery powered domain software reset 0: No effect 1: Reset
Bit 15	RTCEN	0x0	rw	RTC clock enable Set and cleared by software. 0: Disabled 1: Enabled
Bit 14:10	Reserved	0x00	resd	Kept at its default value.
Bit 9:8	RTCSEL	0x0	rw	RTC clock selection Once the RTC clock source is selected, it cannot be changed until the BPDRST bit is reset. 00: No clock 01: LEXT 10: LICK 11: HEXT/128
Bit 7:3	Reserved	0x00	resd	Kept at its default value.
Bit 2	LEXTBYPS	0x0	rw	Low speed external crystal bypass 0: Disabled 1: Enabled
Bit 1	LEXTSTBL	0x0	ro	Low speed external oscillator stable Set by hardware after the LEXT is ready. 0: LEXT is not ready 1: LEXT is ready
Bit 0	LEXTEN	0x0	rw	External low-speed oscillator enable 0: Disabled 1: Enabled

Control/status register (CRM_CTRLSTS)

Reset flag can only be cleared by power reset, while others are cleared by system reset.

Access: 0 to 3 wait states. Wait states are inserted in the case of consecutive accesses to this register.

Bit	Register	Reset value	Type	Description
Bit 31	LPRSTF	0x0	ro	Low-power reset flag Sety by hardware. Cleared by writing to the RSTFC bit. 0: No low-power reset occurs 1: Low-power reset occurs
Bit 30	WWDTRSTF	0x0	ro	Window watchdogtimer reset flag Sety by hardware. Cleared by writing to the RSTFC bit. 0: No window watchdogtimer reset occurs 1: Window watchdogtimer reset occurs
Bit 29	WDTRSTF	0x0	ro	Watchdog timer reset flag Sety by hardware. Cleared by writing to the RSTFC bit. 0: No watchdog timer reset occurs 1: Watchdog timer reset occurs
Bit 28	SWRSTF	0x0	ro	Software reset flag Sety by hardware. Cleared by writing to the RSTFC bit. 0: No software reset occurs 1: Software reset occurs
Bit 27	PORRSTF	0x1	ro	POR/LVR reset flag Sety by hardware. Cleared by writing to the RSTFC bit. 0: No POR/LVR reset occurs 1: POR/LVR reset occurs
Bit 26	NRSTF	0x1	rw	NRST pin reset flag Sety by hardware. Cleared by writing to the RSTFC bit. 0: No NRST pin reset occurs 1: NRST pin reset occurs
Bit 25	Reserved	0x0	resd	Kept at its default value.
Bit 24	RSTFC	0x0	rw	Reset flag clear Cleared by writing 1 through software. 0: No effect 1: Clear the reset flag
Bit 23:2	Reserved	0x000000	resd	Kept at its default value.
Bit 1	LICKSTBL	0x0	ro	LICK stable 0: LICK is not ready 1: LICK is ready
Bit 0	LICKEN	0x0	rw	LICK enable 0: Disabled 1: Enabled

AHB peripheral reset register (CRM_AHBRST)

Bit	Register	Reset value	Type	Description
Bit 31:5	Reserved	0x00000	resd	Kept at its default value.
Bit 14	Reserved	0x0	resd	Kept at its default value.
Bit 13:0	Reserved	0x0000	resd	Kept at its default value.

Additional register1 (CRM_MISC1)

Bit	Register	Reset value	Type	Description
Bit 31:28	CLKOUTDIV	0x0	rw	Clock output division Set the frequency division of CLKOUT. 0xxx: Clock output 1000: Clock output divided by 2 1001: Clock output divided by 4 1010: Clock output divided by 8 1011: Clock output divided by 16 1100: Clock output divided by 64 1101: Clock output divided by 128 1110: Clock output divided by 256 1111: Clock output divided by 512
Bit 27:26	Reserved	0x0	resd	Kept its default value.
Bit 25	HICKDIV	0x0	rw	HICK 6 divider selection This bit is used to select HICK or HICK /6. If the HICK/6 is selected, the clock frequency is 8 MHz. Otherwise, the clock frequency is 48 MHz. 0: HICK/6 1: HICK Note: In any case, HICK always input 4 MHz to PLL.
Bit 24	USBBUFS	0x0	rw	USB buffer size selection 0: USB buffer size is 512 byte 1: USB buffer size is 768~1280 byte.
Bit 23:17	Reserved	0x00	resd	Kept at its default value.
Bit 16	CLKOUT_SEL[3]	0x0	rw	Clock output selection Used with CRM_CFG register bit 26:24.
Bit 15:8	Reserved	0x00	resd	Keep at its default value.
Bit 7:0	HICKCAL_KEY	0x00	rw	HICK calibration key The HICKCAL[7:0] can be written only when this field is set 0x5A.

Additional register2 (CRM_MISC2)

Bit	Register	Reset value	Type	Description
Bit 31:17	Reserved	0x0000	resd	Kept at its default value.
Bit 16	CLK_TO_TMR	0x0	rw	CLKOUT internally connected to timer 10 channel 1 0: Not connected 1: Connected
Bit 15:0	Reserved	0x0000	resd	Kept its default value.

Additional register3 (CRM_MISC3)

Bit	Register	Reset value	Type	Description
Bit 31:16	Reserved	0x0000	resd	Kept at its default value.
Bit 15	Reserved	0x0	resd	Kept at its default value.
Bit 14	Reserved	0x0	resd	Kept at its default value.
Bit 13:12	HEXTDIV	0x0	rw	HEXT division 00: HEXT clock is divided by 2 01: HEXT clock is divided by 3 10: HEXT clock is divided by 4 11: HEXT clock is divided by 5 Note: The division control bit should be configured only after the HEXT clock is ready. Otherwise, it will be ignored.
Bit 11:10	Reserved	0x0	resd	Kept at its default value.
Bit 9	HICK_TO_SCLK	0x0	rw	HICK as system clock frequency select When the HICK is selected as the clock source SCLKSEL, the frequency of SCLK is: 0: Fixed 8 MHz, that is, HICK/6 1: 48 MHz or 8 MHz, depending on theHICKDIV
Bit 8	HICK_TO_USB	0x0	rw	USB 48 MHz clock source select 0: PLL or PLL division 1: HICK or HICK/6 Note: Since USB must work at 48 MHz, HICKDIV=1 must be guaranteed to ensure that the HICK 48 MHz is selected as the clock source of USB 48 MHz.
Bit 7:6	Reserved	0x0	resd	Kept its default value.
Bit 5:4	AUTO_STEP_EN	0x0	rw	Auto step-by-step system clock switch enable When the system clock source is switched from others to the PLL or when the AHB prescaler is changed from large to small (system frequency is from small to large), it is recommended to enable the auto step-by-step system clock switch if the operational target is larger than 108 MHz. Once it is enabled, the AHB bus is halted by hardware till the completion of the switch. During this switch period, the DMA remain working, and the interrupt events are recorded and then handled by NVIC when the AHB bus resumes. 00: Disabled 01: Reserved 10: Reserved 11: Enabled. When AHBDIV or SCLKSEL is modified, the auto step-by-step system clock switch is activated automatically.
Bit 3:0	Reserved	0xd	resd	It is fixed to 0xd. Do not change.

Interrupt map register (CRM_INTMAP)

Bit	Register	Reset value	Type	Description
Bit 31:1	Reserved	0x0000 0000	resd	Kept at its default value.
Bit 0	USBINTMAP	0x0	rw	USBFS interrupt remap 0: USBFS uses the 19th USBFS_H and the 20th USBFS_L interrupt 1: USBDEV uses the 73rd USBFS_MAPH and the 74th USBFS_MAPL interrupt.

5 Flash memory controller (FLASH)

Flash memory introduction

Flash memory is divided into four parts: main Flash memory, external memory, information block and Flash memory registers.

- Main Flash memory is up to 1024 KB, including bank 1 and bank 2.
- Externa memory is up to 16 MB
- Information block consists of 16 KB boot loader and the user system data area. The boot loader uses USART1, USART2 or USB (DFU) serial interface for ISP programming.

Main Flash memory (1024 KB) is divided into bank 1 and bank 2, 512 KB/256 pages per bank, and 2 K per page.

External memory size is up to16 MB, including 4096 pages, 4 K per page.

Table 12. Flash memory architecture(1024 K)

Block		Name	Address range
Main Flash memory	Bank 1 512 KB	Page 0	0x0800 0000 – 0x0800 07FF
		Page 1	0x0800 0800 – 0x0800 0FFF
		Page 2	0x0800 1000 – 0x0800 17FF
		...	…
		Page 255	0x0807 F800 – 0x0807 FFFF
	Bank 2 512 KB	Page 256	0x0808 0000 – 0x0808 07FF
		Page 257	0x0808 0800 – 0x0808 0FFF
		Page 258	0x0808 1000 – 0x0808 17FF
		…	…
		Page 511	0x080F F800 – 0x080F FFFF
External memory	16 MB	Page 0	0x0840 0000 – 0x0840 0FFF
		Page 1	0x0840 1000 – 0x0840 1FFF
		Page 2	0x0840 2000 – 0x0840 2FFF
		…	…
		Page 4095	0x093F F000 – 0x093F FFFF
Information block		Boot loader 16 KB	0x1FFF B000 – 0x1FFF EFFF
		User option bytes 48 B	0x1FFF F800 – 0x1FFF F82F

Main Flash memory (512 KB) has only bank 1, including 256 pages, 2 K per page.

External memory size is up to 16 MB, including 4096 pages, 4 K per page.

Table 13. Flash memory architecture(512 K)

Block		Name	Address range
Main Flash memory	Bank 1 512 KB	Page 0	0x0800 0000 – 0x0800 07FF
		Page 1	0x0800 0800 – 0x0800 0FFF
		Page 2	0x0800 1000 – 0x0800 17FF
		…	…
		Page 255	0x0807 F800 – 0x0807 FFFF
External memory	16 MB	Page 0	0x0840 0000 – 0x0840 0FFF
		Page 1	0x0840 1000 – 0x0840 1FFF
		Page 2	0x0840 2000 – 0x0840 2FFF
		…	…
		Page 4095	0x093F F000 – 0x093F FFFF
Information block		Boot loader 16 KB	0x1FFF B000 – 0x1FFF EFFF
		User option bytes 48 B	0x1FFF F800 – 0x1FFF F82F

Main Flash memory (256 KB) has only bank 1, including 128 pages, 2 K per page.

External memory size is up to 16 MB, including 4096 pages, 4 K per page.

Table 14. Flash memory architecture(256 K)

Block		Name	Address range
Main Flash memory	Bank 1 256 KB	Page 0	0x0800 0000 – 0x0800 07FF
		Page 1	0x0800 0800 – 0x0800 0FFF
		Page 2	0x0800 1000 – 0x0800 17FF
		…	…
		Page 127	0x0803 F800 – 0x0803 FFFF
External memory	16 MB	Page 0	0x0840 0000 – 0x0840 0FFF
		Page 1	0x0840 1000 – 0x0840 1FFF
		Page 2	0x0840 2000 – 0x0840 2FFF
		…	…
		Page 4095	0x093F F000 – 0x093F FFFF
Information block		Boot loader 16 KB	0x1FFF B000 – 0x1FFF EFFF
		User option bytes 48 B	0x1FFF F800 – 0x1FFF F82F

External memory

External Flash memory controls the external SPI Flash through SPIM transmission interface. It supports ciphertext protection. User can decide whether or not to encrypte the data by setting the EXT_FLASH_KEYx bit of the user system data area, and can control the range to be encrypted with the FLASH_DA register.

AHB clock (HCLK), used as a reference clock of SPIM, provides HCLK/2 clock for external SPIM via SPIM interface.

SPIM=External SPI Flash memory expansion (program execution/data store/program and data encrypted)

Note: SPIM has to be accessed by word or half-word.

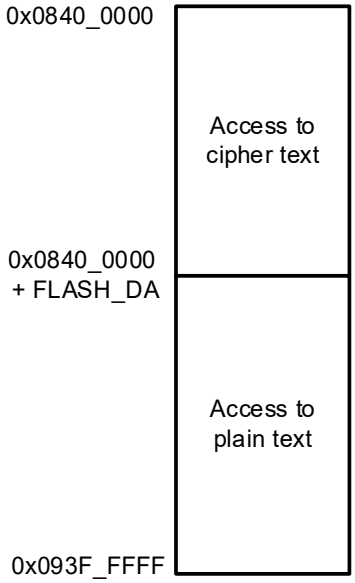


Figure 13. External memory ciphertext protection

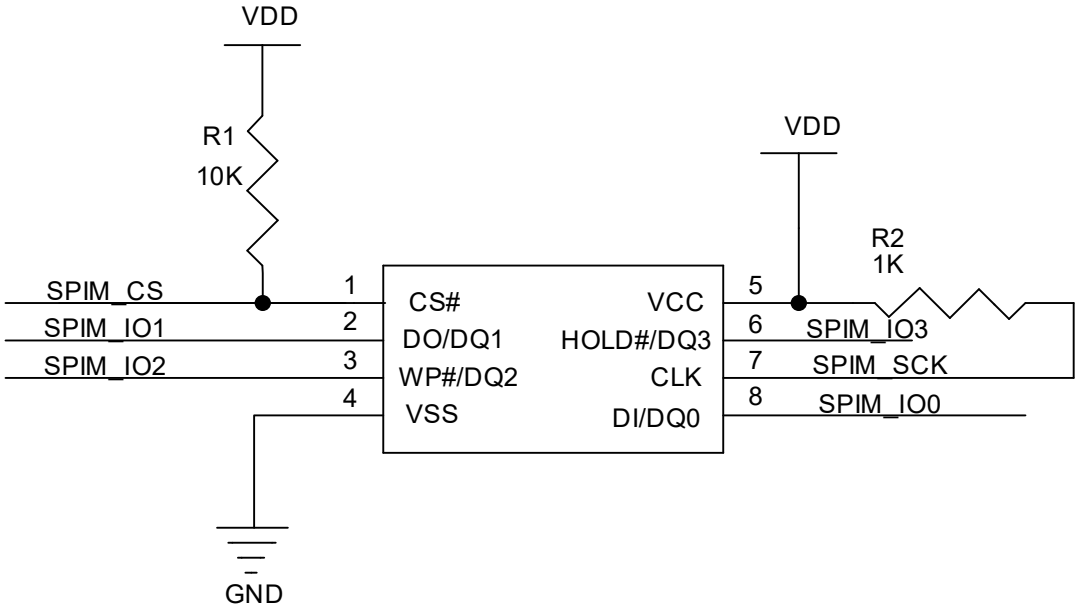


Figure 14. Reference circuit for external memory

Table 15. Instruction set supported by external memory

Instruction	Code	FLASH_SELECT register config.	Description
Write Enable	0x06	0x1/0x2	Both 0x1 and 0x2 Flash must support 0x06 instruction
Quad Page Program	0x32	0x1/0x2	Both 0x1 and 0x2 Flash must support 0x32 instruction
Sector Erase	0x20	0x1/0x2	Both 0x1 and 0x2 Flash must support 0x20 instruction
Chip Erase	0xC7	0x1/0x2	Both 0x1 and 0x2 Flash must support 0XC7 instruction
Read Status Register	0x05	0x1/0x2	Both 0x1 and 0x2 Flash must support 0x05 instruction
Quad I/O Read	0xEB	0x1/0x2	Both 0x1 and 0x2 Flash must support 0xEB instruction 24-bit Addr + 6x Dummy cycle
Volatile status Register write enable	0x50	0x1	When 0x1 Flash is selected, the hardware sends a command automatically to configure the Quad Enable bit of the Flash Status Register 0x1 Flash memory must support: 0x50 and 0x01 or 0x50 and 0x31
Write Status Register-1	0x01		
Write Status Register-2	0x31		

Note: 1. If it is mandatory to set the QE to 1 in the Status Register before executing 0x32 and 0xEB, then 0x1 Flash is selected by seting the FLASH_SELECT register to 0x1.
2. If it is not required to set the QE bit before executing 0x32 and 0xEB, then 0x2 Flash is selected by seting the FLASH_SELECT register to 0x2.

For example,

- If FLASH_SELECT register is set to 0x1:
  D25Q127C, GD25Q64C,GD25Q32C,GD25Q16C,GD25Q80C and W25Q128V Flash memory are supported.
- If FLASH_SELECT register is set as 0x2:
  EN25F20A and EN25QH128A Flash memory are supported.

User system data area

The system data will be read from the information block of Flash memory whenever a system reset occurs, and is saved in the user system data register (FLASH_USD)and erase programming protection status register (FLASH_EPPS).

Each system data occupies two bytes, where the low bytes corresponds to the contents in the system data area, and the high bytes represent the inverse code that is used to verify the correctness of the selected bit. When the high byte is not equal to the inverse code of the low byte (except when both high and low byte are all 0xFF), the system data loader will issue a system data error flag (USDERR) and the corresponding system data and their inverse codes are forced 0xFF.

Note: The update of the contents in the user system data area becomes effective only after a system reset.

Table 16. User system data area

Address	Bit	Description	
0x1FFF_F800	[7:0]	FAP[7:0]: Flash memory access protection (Access protection enable/disable result is stored in the FLASH_USD[1] register) 0xA5: Disabled Others: Enabled	
	[15:8]	nFAP[7:0]: Inverse code of FAP[7:0]	
	[23:16]	SSB[7:0]: System configuration byte (it is stored in the FLASH_USD[9:2] register)	
		Bit 7:4	Reserved
		Bit 3 (BTOPT)	0: When booting from main Flash memory, if there is no boot loader in the bank 2, it will starts from bank 1, otherwise, bank 2. 1: When booting from main Flash memory, it starts from bank 1.
		Bit 2 (nSTDBY_RST)	0: Reset occurs when entering Standby mode 1: No reset occurs when entering Standby mode
		Bit 1 (nDEPSLP_RST)	0: Reset occurs when entering Deepsleep mode 1: No reset occurs when entering Deepsleep mode
		Bit 0 (nWDT_ATO_EN)	0: Watchdog is enabled 1: Watchdog is disabled
	[31:24]	nSSB[7:0]: Inverse code of SSB[7:0]	
0x1FFF_F804	[7:0]	Data0[7:0]: User data 0 (It is stored in the FLASH_USD[17:10] register)	
	[15:8]	nData0[7:0]: Inverse code of Data0[7:0]	
	[23:16]	Data1[7:0]: User data 1 (It is stored in the FLASH_USD[25:18] register)	
	[31:24]	nData1[7:0]: Inverse code of Data1[7:0]	
0x1FFF_F808	[7:0]	EPP0[7:0]: Flash erase/write protection byte 0 (in the FLASH_EPPS[7:0]) This field is used to protect page 0 ~page 15 of main Flash memory. Each bit takes care of two pages (2 KB/page) 0: Erase/write protection is enabled 1: Erase/write protection is disabled	
	[15:8]	nEPP0[7:0]: Inverse code of EPP0[7:0]	
	[23:16]	EPP1[7:0]: Flash erase/write protection byte 1 (stored in the FLASH_EPPS[15:8]) This field is used to protect page 16~page 31 of main Flash memory. Each bit takes care of two pages (2 KB/page) 0: Erase/write protection is enabled 1: Erase/write protection is disabled	
	[31:24]	nEPP1[7:0]: Inverse code of EPP1[7:0]	

Address	Bit	Description
0x1FFF_F80C	[7:0]	EPP2[7:0]: Flash erase/write protection byte 2 (stored in the FLASH_EPPS[23:16]) This field is used to protect page 32~page 47 of main Flash memory. Each bit takes care of two pages (2 KB/page) 0: Erase/write protection is enabled 1: Erase/write protection is disabled
	[15:8]	Inverse code of nEPP2[7:0]: EPP2[7:0]
	[23:16]	EPP3[7:0]: Flash erase/write protection byte 3 (stored in the FLASH_EPPS[31:24]) Bit 6:0 is used to protect page 48~page 61 of main Flash memory. Each bit takes care of two pages (2 KB/page) Bit 7 is used to protect page 62 and beyond, as well as external memory. 0: Erase/write protection is enabled 1: Erase/write protection is disabled
	[31:24]	nEPP3[7:0]: Inverse code of EPP3[7:0]
0x1FFF_F810	[7:0]	EOPB0[7:0]: Extended system options Bit 7:1: Reserved Bit 0: 0: 224 KB of on-chip SRAM 1: 96 KB of on-chip SRAM Note: Switching from 1 to 0 is valid only when the security library is disabled.
	[15:8]	nEOPB0[7:0]: Inverse code of EOPB0[7:0]
	[31:16]	Reserved
0x1FFF_F814	[7:0]	Data2[7:0]: User data 2
	[15:8]	nData2[7:0]: Inverse code of Data2[7:0]
	[23:16]	Data3[7:0]: User data 3
	[31:24]	nData3[7:0]: Inverse code of Data3[7:0]
0x1FFF_F818	[7:0]	Data4[7:0]: User data 4
	[15:8]	nData4[7:0]: Inverse code of Data4[7:0]
	[23:16]	Data5[7:0]: User data 5
	[31:24]	nData5[7:0]: Inverse code of Data5[7:0]
0x1FFF_F81C	[7:0]	Data6[7:0]: User data 6
	[15:8]	nData6[7:0]: Inverse code of Data6[7:0]
	[23:16]	Data7[7:0]: User data 7
	[31:24]	nData7[7:0]: Inverse code of Data7[7:0]
0x1FFF_F820	[7:0]	EXT_FLASH_KEY0[7:0]: External memory ciphertext access area encryption key byte 0 The situations for non-encryption includes: Both EXT_FLASH_KEYx and nEXT_FLASH_KEYx are 0xFF (namely default erase status) Write 0x00 to EXT_FLASH_KEYx Write 0xFF to EXT_FLASH_KEYx That is, {nEXT_FLASH_KEYx, EXT_FLASH_KEYx } are all 0xFFFF, 0xFF00 and 0x00FF.
	[15:8]	nEXT_FLASH_KEY0[7:0]: Inverse code of EXT_FLASH_KEY0[7:0]
	[23:16]	EXT_FLASH_KEY1[7:0]: External memory ciphertext encryption key byte 1
	[31:24]	nEXT_FLASH_KEY1[7:0]: Inverse code of EXT_FLASH_KEY1[7:0]

Address	Bit	Description
0x1FFF_F824	[7:0]	EXT_FLASH_KEY2[7:0]: External memory ciphertext encryption key byte 2
	[15:8]	nEXT_FLASH_KEY2[7:0]: Inverse code of EXT_FLASH_KEY2[7:0]
	[23:16]	EXT_FLASH_KEY3[7:0]: External memory ciphertext encryption key byte3
	[31:24]	nEXT_FLASH_KEY3[7:0]: Inverse code of EXT_FLASH_KEY3[7:0]
0x1FFF_F828	[7:0]	EXT_FLASH_KEY4[7:0]: External memory ciphertext encryption key byte 4
	[15:8]	nEXT_FLASH_KEY4[7:0]: Inverse code of EXT_FLASH_KEY4[7:0]
	[23:16]	EXT_FLASH_KEY5[7:0]: External memory ciphertext encryption key byte 5
	[31:24]	nEXT_FLASH_KEY5[7:0]: Inverse code of EXT_FLASH_KEY5[7:0]
0x1FFF_F82C	[7:0]	EXT_FLASH_KEY6[7:0]: External memory ciphertext encryption key byte 6
	[15:8]	nEXT_FLASH_KEY6[7:0]: Inverse code of EXT_FLASH_KEY6[7:0]
	[23:16]	EXT_FLASH_KEY7[7:0]: External memory ciphertext encryption key byte 7
	[31:24]	nEXT_FLASH_KEY7[7:0]: Inverse code of EXT_FLASH_KEY7[7:0]

Flash memory operation

Unlock/lock

After reset, Flash memory is protected, by default. FLASH_CTRLx cannot be written. Write and erase operation can be performed only when the Flash memory is unlocked.

Unlock procedure:

Flash memory block can be unlocked by writing KEY1 (0x45670123) and KEY2 (0xCDEF89AB) to the FLASH_UNLOCKx register.

Note: Writing incorrect key sequence leads to a bus error and the Flash memory is also locked until the next reset.

Lock procedure:

Flash memory block can be locked by setting the OPLK bit in the FLASH_CTRLx register.

Erase operation

Erase operation must be done before programming. Flash memory erase includes mass erase and page erase.

Page erase

Any page in the Flash memory can be erased with page erase function. Below should be followed during page erase:

- Check the OBF bit in the FLASH_STSx register to confirm that there is no other programming operation in progress;
- Write the page to be erased in the FLASH_ADDRx register
- Set the SECERS and ERSTR bit in the FLASH_CTRLx register to enable page erase
- Wait until the OBF bit becomes “0” in the FLASH_STSx register. Read the EPPERR bit and ODF bit in the FLASH_STSx register to verify the erased pages.

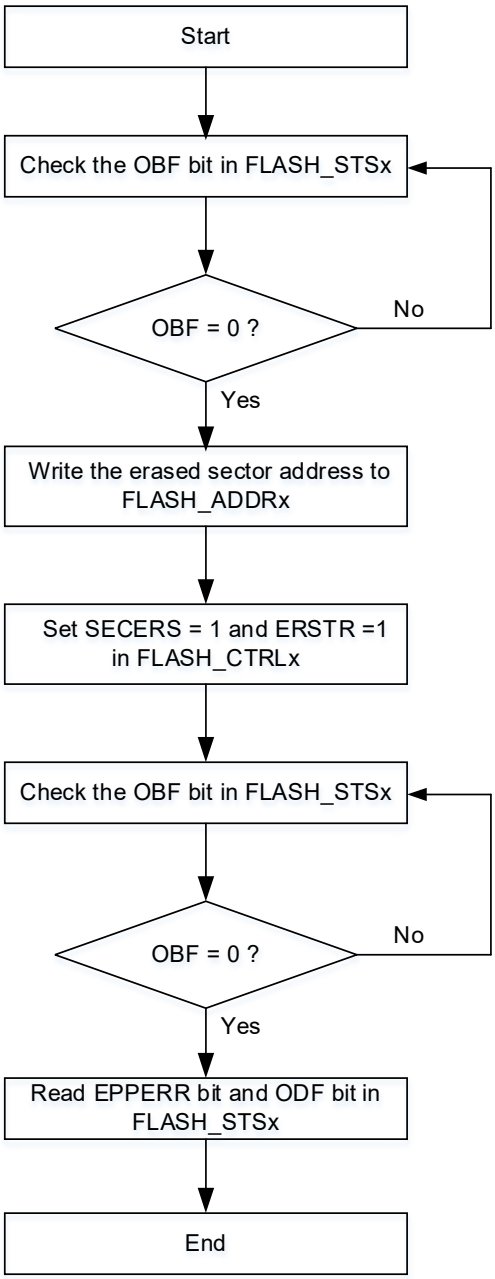


Figure 15. Flash memory page erase process

Mass erase

Mass erase function can erase all the Flash memory.

The following process is recommended:

- Check the OBF bit in the FLASH_STSx register to confirm that there is no other programming operation in progress;
- Set the BANKERS and ERSTR bit in the FLASH_CTRLx register to enable mass erase;
- Wait until the OBF bit becomes "0" in the FLASH_STSx register. Read the EPPERR bit and ODF bit in the FLASH_STSx register to verify the erased pages.

Note: Read operation to the Flash memory during erase will halt CPU until the completion of erase.

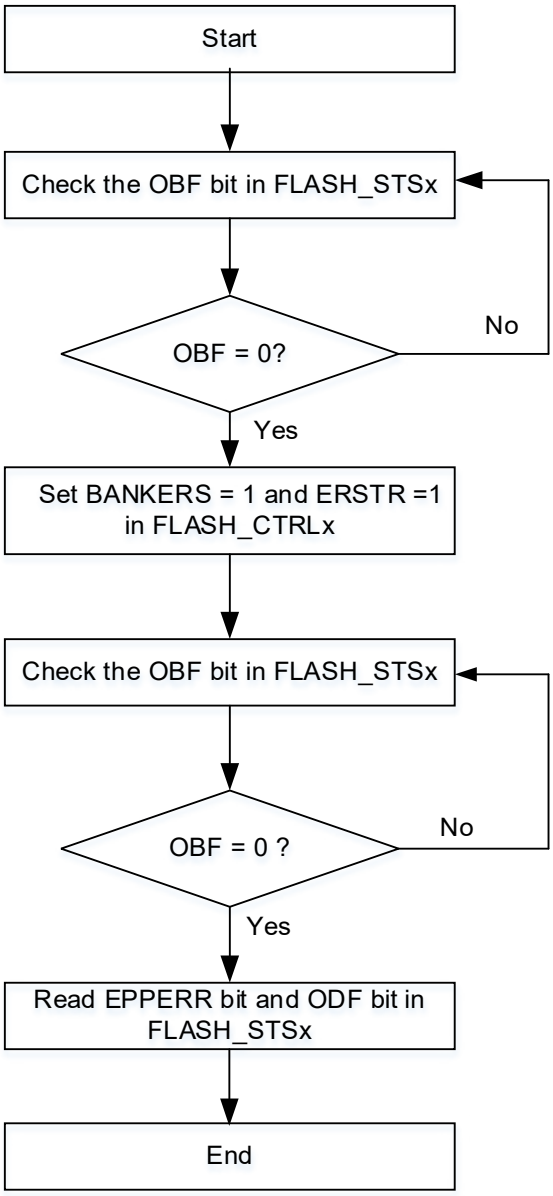


Figure 16. Flash memory mass erase process

Programming operation

The Flash memory can be programmed with 32 bits, 16 bits or 8 bits at a time.

The following process is recommended:

- Check the OBF bit in the FLASH_STSx register to confirm that there is no other programming operation in progress;
- Set the FPRGM bit in the FLASH_CTRLx register, so that the Flash memory programming instructions can be received;
- Write the data (word/half-word/byte) to be programmed to the designated address;
- Wait until the OBF bit in the FLASH_STSx register becomes "0", read the EPPERR, PRGMERR and ODF bit to verify the programming result.

Note: 1. When the address to be written is not erased in advance, the programming operation is not executed unless the data to be written is all 0. In this case, a programming error is reported by the PRGMERR bit in the FLASH_STSx register.
2. Read operation to the Flash memory during tprogramming will halt CPU until the completion of programming.

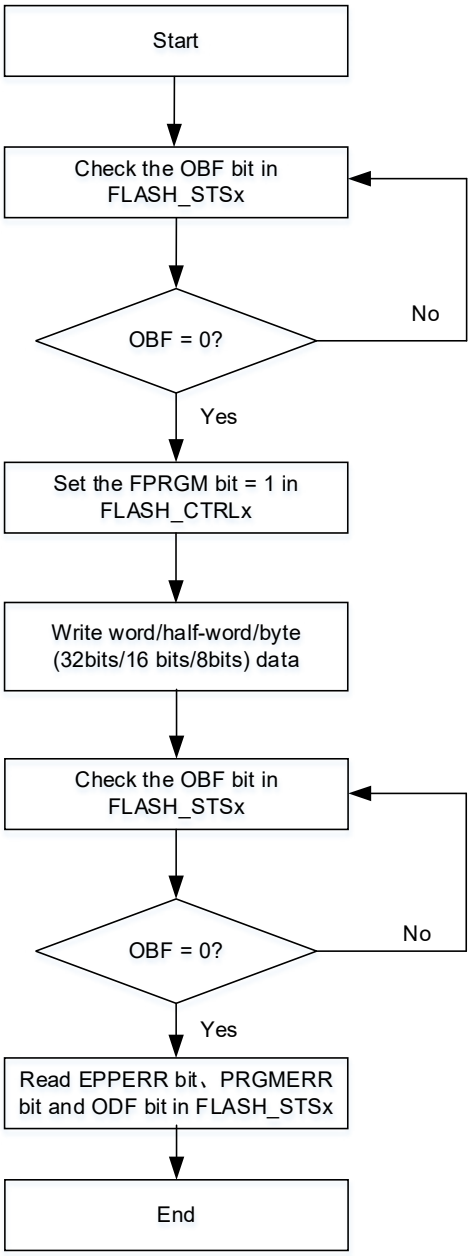


Figure 17. Flash memory programming process

Read operation

Flash memory can be accessed through AHB bus of the CPU.

External memory operation

External memory has the same operation method as that of Flash memory, including read, unlock, erase and programming except that the external memory only supports 32-bit and 16-bit operations, rather than 8 bits.

User system data area operation

Unlock/lock

After reset, user system data area is protected, by default. Write and erase operations can be performed only after the Flash memory is unlocked before the unlock operation for the user system data area.

Unlock procedure:

Flash memory block can be unlocked by writing KEY1 (0x45670123) and KEY2 (0xCDEF89AB) to the FLASH_UNLOCK register;

When KEY1 (0x45670123) and KEY2 (0xCDEF89AB) is written to the FLASH_USD_UNLOCK register, the USDULKS bit in the FLASH_CTRL register will be automatically set by hardware, indicating that it supports write/erase operation to the user system data area.

Note: Writing an incorrect key sequence leads to bus error and the Flash memory is also locked until the next reset.

Lock procedure:

User system data area is locked by clearing the USDULKS bit in the FLASH_CTRL register by software.

Erase operation

Erase operation must be done before programming. User system data area can perform erase operation independently.

Below should be followed during page erase:

- Check the OBF bit in the FLASH_STS register to confirm that there is no other programming operation in progress;
- Set the USDERS and ERSTR bit in the FLASH_CTRL register to enable erase operation;
- Wait until the OBF bit becomes “0” in the FLASH_STS register. Read the ODF bit in the FLASH_STSx register to verify the erase result.

Note: Read operation to the Flash memory during programming will halt CPU until the completion of erase.

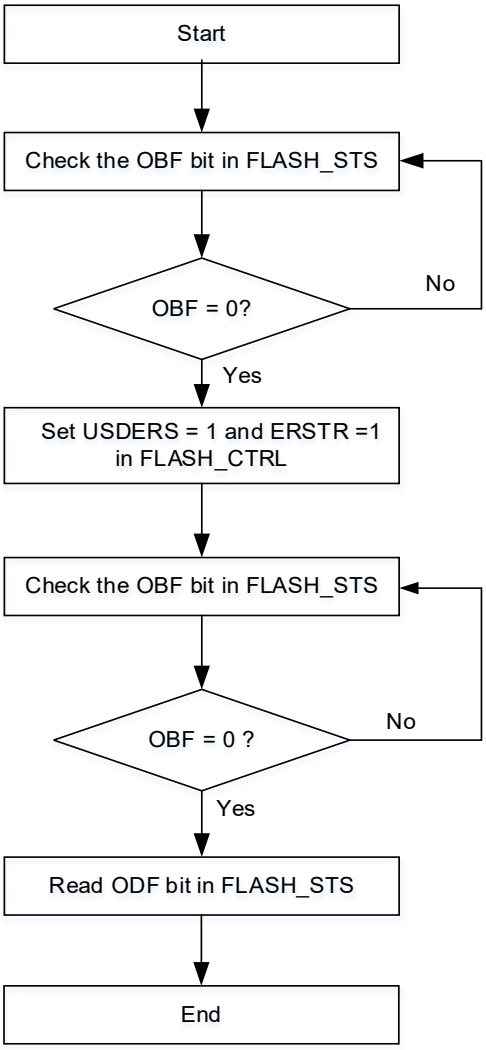


Figure 18. System data area erase process

Programming operation

The User system data area can be programmed with 16 bits at a time.

The following process is recommended:

- Check the OBF bit in the FLASH_STS register to confirm that there is no other programming operation in progress;
- Set the USDPRGM bit in the FLASH_CTRL register, so that the programming instructions for the user system data area can be received;
- Write the data (half-word) to be programmed to the designated address;
- Wait until the OBF bit in the FLASH_STS register becomes “0”, read the PRGMERR and ODF bit to verify the programming result.

Note: Read operation to the Flash memory during programming will halt CPU until the completion of programming.

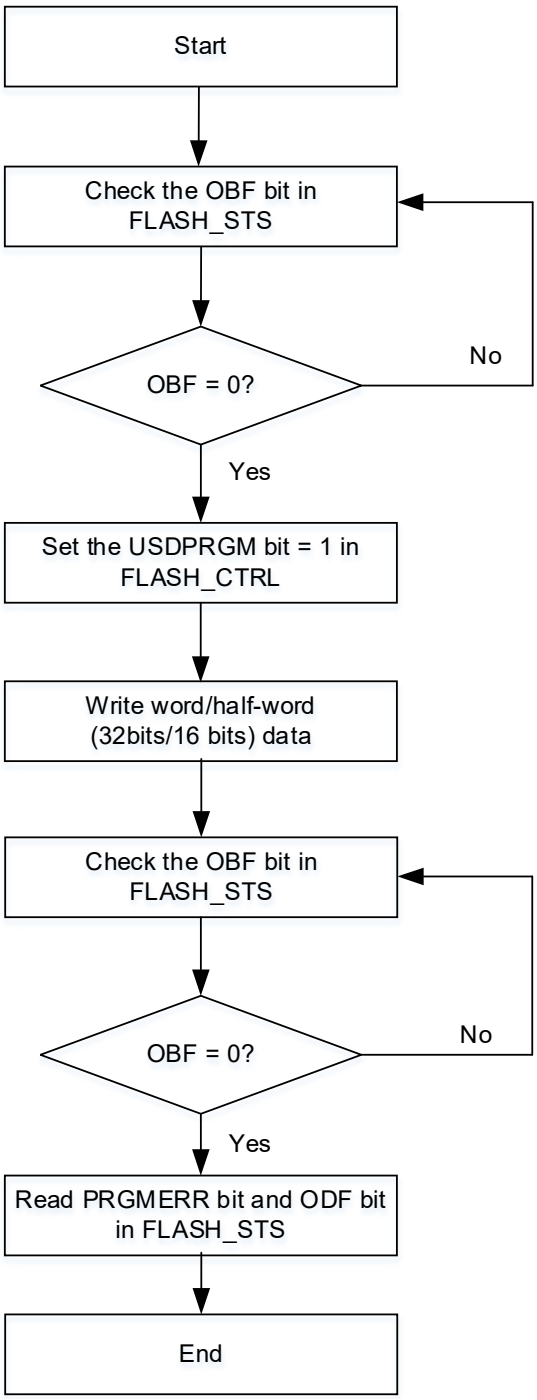


Figure 19. System data area programming process

Read operation

User system data area can be accessed through AHB bus of the CPU.

Flash memory protection

Flash memory includes access and erase/program protection.

Access protection

When the contents in the nFAP and FAP byte are different from 0x5A and 0xA5, the Flash memory will activate access protection after a system reset. In this case, only the Flash program is allowed to read Flash memory data. This read operation is not permitted in debug mode or by booting from non-Flash memory.

When the Flash access is protected, the user can re-erase the system data area, and unlock Flash access protection (switching from protected to unprotected state will trigger mass erase on the Flash memory automatically) by wrting 0xA5 to FAP byte, and then perform a system reset. Subsequently, the system data loader will be reloaded with system data and updated with Flash memory access protection disable state (FAP byte)

Note: If the access protection bit is set in debug mode, then the debug mode has to be cleared by POR instead of system reset in order to resume access to Flash memory data.

The following table shows Flash memory access limits when Flash access protection is enabled.

Table 17. Flash memory access limit

Block	Access limits					
	In debug mode or boot from SRAM and boot loader			Boot from main Flash memory		
	Read	Write	Erase	Read	Write	Erase
Main Flash memory	Not allowed		Not allowed(1)(2)	Accessible		
External memory	Not allowed		Not allowed(2)	Accessible		
User system data area	Not allowed	Accessible		Accessible		

Note: 1. Main Flash memory is cleared automatically by hardware only when the access protection is disabled;
2. Only page erase is forbidden. Bank 1 and bank 2 and external memory mass erase are not affected.

Erase/program protection

If Flash memory is 256K and above, the erase/program protection is based on two-page level.

This is used to protect the contents in the Flash memory against inadvertent operation when the program crash occurs.

Erase/program operation is not permitted under one of the following events, and the EPPERR bit is set accordingly:

- The pages with erase/program protection enabled (main Flash memory and external memory)
- Bank1, bank2 and external memory with erase/program protection enabled
- When the Flash access protection is enabled, page 0 and page 1 in the main Flash memory will be protected against erase/program automatically,
- Once the Flash access protection is enabled, the main Flash memory is protected against erase/program when it is in debug mode or when it is started from non-main Flash memory.

Special functions

Security library settings

Security library is a defined area protected by a code in the main memory. This area is only executable but cannot be read (Except for I-Code and D-code buses), written, or deleted, unless a correct code is keyed in. Security library includes instruction security library and data security library; users can select part of or the whole security library for instruction storage, but using the whole security library for storing data is not supported.

Advantages of security library:

Security library is protected by codes so that solution providers can program core algorithm into this area;

Security library cannot be read or deleted (including ISP/IAP/SWD) but only executed unless code defined by the solution provider is keyed in;

The rest of the area can be used for secondary development by solution providers;

Solution providers can sell core algorithm with security library function and do not have to develop full solutions for every customer.

Security library helps prevent from deliberate damage or changing terminal application codes.

Note: Security library can only be located in the main Flash memory;

Security library code must be programmed by page, with its start address aligned with the main memory address;

Interrupt vector table will be placed on the first page of Flash memory (page 0), which should not be configured as security library;

Program codes to be protected by the security library should not be placed on the first page of Flash memory;

Only I-Code bus is allowed to read instruction security library;

Only D-Code bus is allowed to read data security library;

When writing or deleting security library code, a warning message will be issued by WRPRTFLR =1 in the FLASH_STS register;

Executing mass erase in the main memory will not erase the security library.

By default, security library setting register is unreadable and write protected. To enable write access to this register, security library should be unlocked first, write 0xA35F6D24 to the SLIB_UNLOCK register, and check the SLIB_ULKF bit in the SLIB_MISC_STS register to verify if it is unlocked successfully. Then write a value into the security library setting register.

Optional CRC check for security library code is based on a page level.

The steps to enable security library are as follows:

- Check the OBF bit in the FLASH_STS register to ensure that there is no other ongoing programming operation;
- Write 0xA35F6D24 to the SLIB_KEYR register to unlock security library.
- Check the SLIB_ULKF bit of SLIB_MISC_STS register to verify that it is unlocked successfully.

- Set the area to be protected in the SLIB_SET_RANGE register, including the addresses of instruction area and data area.
- Wait until the OBF bit becomes "0";
- Set security library password in the SLIB_SET_PSW register;
- Wait until the OBF bit becomes "0";
- Program the code to be saved in security library;
- Perform system reset, and then reload security library setting word;
- Read the SLIB_STS0/STS1 register to verify the security library setting.

Note: Security library should be enabled when the Flash access protection is not activated.

Steps to unlock security library:

- Write the previously set security library password to the SLIB_PWD_CLR register.
- Wait until the OBF bit becomes "0";
- Perform system reset, and then reload security library setting word;
- Read the SLIB_STS0 register to check the security library setting.

Note: Disabling the security library will automatically perform mass erase for the main memory and page erase for security library setting block.

Flash memory registers

The following table lists Flash register map and their reset values. These peripheral registers must be accessed by words (32 bits).

Table 18. Flash memory interface – Register map and reset value

Register	Offset	Reset value
FLASH_PSR	0x00	0x0000 0030
FLASH_UNLOCK	0x04	0xXXXX XXXX
FLASH_USD_UNLOCK	0x08	0xXXXX XXXX
FLASH_STS	0x0C	0x0000 0000
FLASH_CTRL	0x10	0x0000 0080
FLASH_ADDR	0x14	0x0000 0000
FLASH_USD	0x1C	0x03FF FFFC
FLASH_EPPS	0x20	0xFFFF FFFF
FLASH_UNLOCK2	0x44	0xXXXX XXXX
FLASH_STS2	0x4C	0x0000 0000
FLASH_CTRL2	0x50	0x0000 0080
FLASH_ADDR2	0x54	0x0000 0000
FLASH_UNLOCK3	0x84	0xXXXX XXXX
FLASH_SELECT	0x88	0x0000 0000
FLASH_STS3	0x8C	0x0000 0000
FLASH_CTRL3	0x90	0x0000 0080
FLASH_ADDR3	0x94	0x0000 0000
FLASH_DA	0x98	0x0000 0000
SLIB_STS0	0xCC	0x0000 0000
SLIB_STS1	0xD0	0x0000 0000

Register	Offset	Reset value
SLIB_PWD_CLR	0xD4	0x0000 0000
SLIB_MISC_STS	0xD8	0x0100 0000
SLIB_SET_PWD	0xDC	0x0000 0000
SLIB_SET_RANGE	0xE0	0x0000 0000
SLIB_UNLOCK	0xF0	0x0000 0000
FLASH_CRC_CTRL	0xF4	0x0000 0000
FLASH_CRC_CHKR	0xF8	0x0000 0000

Flash performance select register (FLASH_PSR)

Bit	Register	Reset value	Type	Description
Bit 31:0	Reserved	0x0000 0030	resd	Kept at its default value.

Flash unlock register (FLASH_UNLOCK)

Only used in Flash memory bank 1.

Bit	Register	Reset value	Type	Description
Bit 31:0	UKVAL	0xXXXX XXXX	wo	Unlock key value This is used to unlock Flash memory bank 1.

Note: All these bits are write-only, and return 0 when being read.

Flash user system data unlock register (FLASH_USD_UNLOCK)

Bit	Register	Reset value	Type	Description
Bit 31:0	USD_UKVAL	0xXXXX XXXX	wo	User system data Unlock key value

Note: All these bits are write-only, and return 0 when being read.

Flash status register (FLASH_STS)

Only used in Flash memory bank 1.

Bit	Register	Reset value	Type	Description
Bit 31:6	Reserved	0x0000000	resd	Kept at its default value
Bit 5	ODF	0x0	rw	Operation done flag This bit is set by hardware when Flash memory operations (program/erase) is completed. It is cleared by writing “1”.
Bit 4	EPPERR	0x0	rw	Erase/program protection error This bit is set by hardware when programming the erase/program- protected Flash memory address. It is cleared by writing “1”.
Bit 3	Reserved	0x0	resd	Kept at its default value.
Bit 2	PRGMERR	0x0	rw	Programming error When the programming addess is not “0xFFFF”, this bit is set by hardware. It is cleared by writing “1”.
Bit 1	Reserved	0x0	resd	Kept at its default value.
Bit 0	OBF	0x0	ro	Operation busy flag When this bit is set, it indicates that Flash memory operation is in process. It is cleared when operation is completed.

Flash control register (FLASH_CTRL)

Only used in Flash memory bank 1.

Bit	Register	Reset value	Type	Description
Bit 31:13	Reserved	0x00000	resd	Kept at its default value
Bit 12	ODFIE	0x0	rw	Operation done flag interrupt enable 0: Interrupt is disabled; 1: Interrupt is enabled.
Bit 11,8,3	Reserved	0x0	resd	Kept its default value
Bit 10	ERRIE	0x0	rw	Error interrupt enable This bit enables EPPERR or PROGERR interrupt. 0: Interrupt is disabled; 1: Interrupt is enabled.
Bit 9	USDULKS	0x0	rw	User system data unlock success This bit is set by hardware when the user system data is unlocked properly, indicating that erase/program operation to the user system data is allowed. This bit is cleared by writing “0”, which will re-lock the user system data area.
Bit 7	OPLK	0x1	rw	Operation lock This bit is set by default, indicating that Flash memory is protected against operations. This bit is cleared by hardware after unlock, indicating that erase/program operation to Flash memory is allowed. Writing “1” can re-lock Flash memory operations.
Bit 6	ERSTR	0x0	rw	Erase start An erase operation is triggered when this bit is set. This bit is cleared by hardware after the completion of the erase operation.
Bit 5	USDERS	0x0	rw	User system data erase It indicates the user system data erase.
Bit 4	USDPRGM	0x0	rw	User system data program It indicates the user system data program.
Bit 2	BANKERS	0x0	rw	Bank erase It indicates bank erase operation.
Bit 1	SECERS	0x0	rw	Page erase It indicates page erase operation.
Bit 0	FPRGM	0x0	rw	Flash program It indicates Flash program operation.

Flash address register (FLASH_ADDR)

Only used in Flash memory bank 1.

Bit	Register	Reset value	Type	Description
Bit 31:0	FA	0x0000 0000	wo	Flash address Select the address of the pages to be erased in page erase operation.

User system data register (FLASH_USD)

Bit	Register	Reset value	Type	Description
Bit 31:26	Reserved	0x00	resd	Kept at its default value
Bit 25:18	USER_D1	0xFF	ro	User data 1
Bit 17:10	USER_D0	0xFF	ro	User data 0
Bit 9:2	SSB	0xFF	ro	System setting byte Includes the system setting bytes in the loaded user system data area Bit [9:6]: Unused Bit 5: BTOPT Bit 4: nSTDBY_RST Bit 3: nDEPSLP_RST Bit 2: nWDT_ATO_EN
Bit 1	FAP	0x0	ro	Flash access protection Access to Flash memory is not allowed when this bit is set.
Bit 0	USDERR	0x0	ro	User system data error When this bit is set, it indicates that certain byte does not match its inverse code in the user system data area. At this point, this byte and its inverse code will be forced to0xFF when being read.

Erase/program protection status register (FLASH_EPPS)

Bit	Register	Reset value	Type	Description
Bit 31:0	EPPS	0xFFFF FFFF	ro	Erase/Program protection status This register reflects the erase/program protection byte status in the loaded user system data.

Flash unlock register2 (FLASH_UNLOCK2)

Only used in Flash memory bank 2.

Bit	Register	Reset value	Type	Description
Bit 31:0	UKVAL	0xXXXX XXXX	wo	Unlock key value This register is used to unlock Flash memory bank 2.

Note: All these bits are write-only, and return 0 when being read.

Flash status register2 (FLASH_STS2)

Only used in Flash memory bank 2.

Bit	Register	Reset value	Type	Description
Bit 31:6	Reserved	0x0000000	resd	Kept at its default value
Bit 5	ODF	0x0	rw	Operation done flag This bit is set by hardware when Flash memory operations (program/erase) is completed. It is cleared by writing “1”.
Bit 4	EPPERR	0x0	rw	Erase/Program protection error This bit is set by hardware when programming the erase/program- protected Flash memory address. It is cleared by writing “1”.
Bit 3	Reserved	0x0	resd	Kept at its default value

Bit	Register	Reset value	Type	Description
Bit 2	PRGMERR	0x0	rw	Program error When the programming addess is not “0xFFFF”, this bit is set by hardware. It is cleared by writing “1”.
Bit 1	Reserved	0x0	resd	Kept at its default value
Bit 0	OBF	0x0	ro	Operation busy flag When this bit is set, it indicates that Flash memory operation is in process. It is cleared when operation is completed.

Flash control register2 (FLASH_CTRL2)

Only used in Flash memory bank 2.

Bit	Register	Reset value	Type	Description
Bit 31:13	Reserved	0x00000	resd	Kept at its default value
Bit 12	ODFIE	0x0	rw	Operation done flag interrupt enable 0: Interrupt is disabled 1: Interrupt is enabled
Bit 11	Reserved	0x0	resd	Keep its default value
Bit 10	ERRIE	0x0	rw	Error interrupt enable This bit enables EPPERR or PROGERR interrupt. 0: Interrupt is disabled 1: Interrupt is enabled
Bit 9,8	Reserved	0x0	resd	Keep at its default value
Bit 7	OPLK	0x1	rw	Operation lock This bit is set by default, indicating that Flash memory is protected against operations. This bit is cleared by hardware after unlock, indicating that erase/program operation to Flash memory is allowed. Writing “1” can re-lock Flash memory operations.
Bit 6	ERSTR	0x0	rw	Erase start An erase operation is triggered when this bit is set. This bit is cleared by hardware after the completion of the erase operation.
Bit 5,4,3	Reserved	0x0	resd	Kept at its default value
Bit 2	BANKERS	0x0	rw	Bank erase It indicates bank erase operation.
Bit 1	SECERS	0x0	rw	Page erase It indicates page erase operation.
Bit 0	FPRGM	0x0	rw	Flash program It indicates Flash program operation.

Flash address register2 (FLASH_ADDR2)

Only used in Flash memory bank 2.

Bit	Register	Reset value	Type	Description
Bit 31:0	FA	0x0000 0000	wo	Flash address Select the address of the pages to be erased in page erase operation.

Flash unlock register3 (FLASH_UNLOCK3)

Only used in external memory.

Bit	Register	Reset value	Type	Description
Bit 31:0	UKVAL	0xXXXX XXXX	wo	Unlock key value This register is used to unlock SPIM.

Flash select register (FLASH_SELECT)

Only used in external memory.

Bit	Register	Reset value	Type	Description
Bit 31:0	SELECT	0x0000 0000	wo	SPIM supports extended SPI Flash chip selection 0x0001: Refer to Table 15 0x0002: Refer to Table 15 Others: Reserved

Flash status register3 (FLASH_STS3)

Only used in external memory.

Bit	Register	Reset value	Type	Description
Bit 31:6	Reserved	0x0000000	resd	Kept at its default value
Bit 5	ODF	0x0	rw	Operation done flag This bit is set by hardware when Flash memory operations (program/erase) is completed. It is cleared by writing “1”.
Bit 4	EPPERR	0x0	rw	Erase/Program protection error This bit is set by hardware when programming the erase/program- protected Flash memory address. It is cleared by writing “1”.
Bit 3	Reserved	0x0	resd	Kept at its default value
Bit 2	PRGMERR	0x0	rw	Programming error When the programming addess is not “0xFFFF”, this bit is set by hardware. It is cleared by writing “1.
Bit 1	Reserved	0x0	resd	Kept at its default value
Bit 0	OBF	0x0	ro	Operation busy flag When this bit is set, it indicates that Flash memory operation is in process. It is cleared when operation is completed.

Flash control register3 (FLASH_CTRL3)

Only used in external memory.

Bit	Register	Reset value	Type	Description
Bit 31:13	Reserved	0x00000	resd	Kept at its default value
Bit 12	ODFIE	0x0	rw	Operation done flag interrupt enable 0: Interrupt is disabled 1: Interrupt is enabled
Bit 11	Reserved	0x0	resd	Kept at its default value
Bit 10	ERRIE	0x0	rw	Error interrupt enable 0: Interrupt is disabled 1: Interrupt is enabled
Bit 9,8	Reserved	0x0	resd	Kept at its default value
Bit 7	OPLK	0x1	rw	Operation lock This bit is set by default, indicating that Flash memory is protected against operations. This bit is cleared by hardware after unlock, indicating that erase/program operation to Flash memory is allowed. Writing “1” can re-lock Flash memory operations.
Bit 6	ERSTR	0x0	rw	Erase start An erase operation is triggered when this bit is set. This bit is cleared by hardware after the completion of the erase operation.
Bit 5,4,3	Reserved	0x0	resd	Kept at its default value
Bit 2	CHPERS	0x0	rw	Mass erase Perform mass erase on the external memory.
Bit 1	SECERS	0x0	rw	Page erase It indicates page erase operation.
Bit 0	FPRGM	0x0	rw	Flash program It indicates Flash program operation.

Flash address register3 (FLASH_ADDR3)

Only used in external memory.

Bit	Register	Reset value	Type	Description
Bit 31:0	FA	0x0000 0000	wo	Flash address Select the address of the pages to be erased in external memory.

Flash decrption address register (FLASH_DA)

Only used in external memory.

Bit	Register	Reset value	Type	Description
Bit 31:0	FDA	0x0000 0000	wo	Flash decryption address Set the encryption range of external memory through FLASH_DA register in the user program. 0x0840_0000 ~ (0x0840_0000+FDA-0x1): the cipher text of external memory (0x0840_0000 +FDA) ~ 0x093F FFFF: the plain text of external memory Note: The setting value of FDA must be a multiple of 4, aligned by word.

Flash security library status register (SLIB_STS0)

Only used in Flash security library.

Bit	Register	Reset value	Type	Description
Bit 31:4	Reserved	0x0000000	resd	Kept at its default value
Bit 3	SLIB_ENF	0x0	ro	SLIB_ENF: sLib enable flag When this bit is set, it indicates that the main Flash memory is partially or completely (depending on the setting of SLIB_STS1) used as security library code.
Bit 2:0	Reserved	0x0	resd	Kept at its default value

Flash security library status register1 (SLIB_STS1)

Only used in Flash security library.

Bit	Register	Reset value	Type	Description
Bit 31:22	SLIB_ES	0x000	ro	Security library end page 0: Page 0 1: Page 1 2: Page 2 … 511: Page 511
Bit 21:11	SLIB_DAT_SS	0x000	ro	Security library data start page 0: Invalid page 1: Page 1 2: Page 2 … 511: Page 511 0x7FF: No security library data area
Bit 10:0	SLIB_SS	0x000	ro	Security library start page 0: Page 0 1: Page 1 2: Page 2 … 511: Page 511

Flash security library password clear register (SLIB_PWD_CLR)

Only used in Flash security library.

Bit	Register	Reset value	Type	Description
Bit 31:0	SLIB_PCLR_VAL	0x0000 0000	wo	Security library password clear value Entering correct security library password will unlock security library functions. The write status of this register is reflected in the bit 0 and bit 1 in the SLIB_MISC_STS register.

Security library additional status register (SLIB_MISC_STS)

Only used in Flash security library.

Bit	Register	Reset value	Type	Description
Bit 31:25	Reserved	0x00	resd	Kept at its default value
Bit 24:16	SLIB_RCNT	0x100	ro	Security library remaining count It is decremented from 256 to 0.
Bit 15:3	Reserved	0x0000	resd	Kept at its default value
Bit 2	SLIB_ULKF	0x0	ro	Security library unlock flag When this bit is set, it indicates that sLib-related setting registers can be configured.
Bit 1	SLIB_PWD_OK	0x0	ro	Security library password ok This bit is set by hardware when the password is correct.
Bit 0	SLIB_PWD_ERR	0x0	ro	Security library password error This bit is set by hardware when the password is incorrect and the setting value of the password clear register is different from 0xFFFF FFFF. Note: When this bit is set, the hardware will no longer agree to re-program the password clear register until the next reset.

Security library password setting register (SLIB_SET_PWD)

Only used for Flash security library password setting.

Bit	Register	Reset value	Type	Description
Bit 31:0	SLIB_PSET_VAL	0x0000 0000	wo	Security library password setting value Note: This register can be written only after unlocking security library lock. It is used to set up the startup password of security library. Values of 0xFFFF_FFFF and 0x0000_0000 are invalid.

Security library address setting register (SLIB_SET_RANGE)

Only used for Flash security library address setting.

Bit	Register	Reset value	Type	Description
Bit 31:22	SLIB_ES_SET	0x000	wo	Security library end page setting Theses bits are used to set the security library end page. 0: Page 0 1: Page 1 2: Page 2 … 511: Page 511
Bit 21:11	SLIB_DSS_SET	0x000	wo	Security library data start page setting These bits are used to set the security library start page. 0: Invalid page. Setting it will cause security library to fail to start. 1: Page 1 2: Page 2 … 511: Page 511 0x7FF: No security library data area.
Bit 10:0	SLIB_SS_SET	0x000	wo	Security library start page setting These bits are used to set the security library start page. 0: Page 0 1: Page 1 2: Page 2 … 511: Page 511

Note: All these bits are write-only, and return 0 when being read. This register can be written only after unlocking security library lock.

Security library unlock register (SLIB_UNLOCK)

Only used for Flash security library unlock setting.

Bit	Register	Reset value	Type	Description
Bit 31:0	SLIB_UKVAL	0x0000 0000	wo	Security library unlock key value Fixed key value is 0xA35F_6D24, used for security library setting register unlock

Note: All these bits are write-only, and return 0 when being read.

Flash CRC check control register (FLASH_CRC_CTRL)

Only used in main Flash memory.

Bit	Register	Reset value	Type	Description
Bit 31	CRC_STRT	0x0	wo	CRC start Set this bit to enable user code or security library code CRC calibration. This bit is cleared automatically after the hardware enables CRC.
Bit 30:24	Reserved	0x00	wo	Kept at its default value
Bit 23:12	CRC_SN	0x000	wo	CRC calibration page numbler Set the number of the CRC calibarion, in terms of pages.

Bit	Register	Reset value	Type	Description
Bit 11:0	CRC_SS	0x000	wo	CRC calibration start page Set the start page for this CRC calibration. 0x0: Page 0 0x1: Page 1 …

Note: All these bits are write-only. Reading them has no effect.

Flash CRC check result register (FLASH_CRC_CHKR)

Only used in Flash or security library.

Bit	Register	Reset value	Type	Description
Bit 31:0	CRC_CHKR	0x0000 0000	ro	CRC check result

Note: All these bits are write-only. Reading them has no effect.

6 General-purpose I/Os (GPIOs)

Introduction

HT32F49365/HT32F49395 support up to 80 bidirectional I/O pins, which are grouped as five categories, namely PA, PB, PC, PD and PE. Each of the GPIO group provides up to 16 I/O pins that feature communication, control and data collection. In addition, their main features also include:

- Supports general-purpose I/O (GPIO) or multiplexed function I/O (IOMUX), which will be detailed in this chapter and the subsequent sections
- Each pin can be configured by software as floating input, pull-up/pull-down input, analog input/output, push-pull/open-drain output, multiplexed push-pull/open-drain output
- Each pin's output drive capability and output signal slope is configureable by software
- Each pin can be configured as external interrupt input
- Each pin can be locked

Functional descriptions

GPIO structure

Each of the GPIO pins can be configured by software as four input modes (floating, pull-up/pull-down and analog input) and four output modes (open-drain, push-pull, alternate function push-pull/open-drain output)

Each I/O port bit can be programmed flexibly. However, I/O port registers must be accessed by 32-bit words (half-word/byte access is not supported).

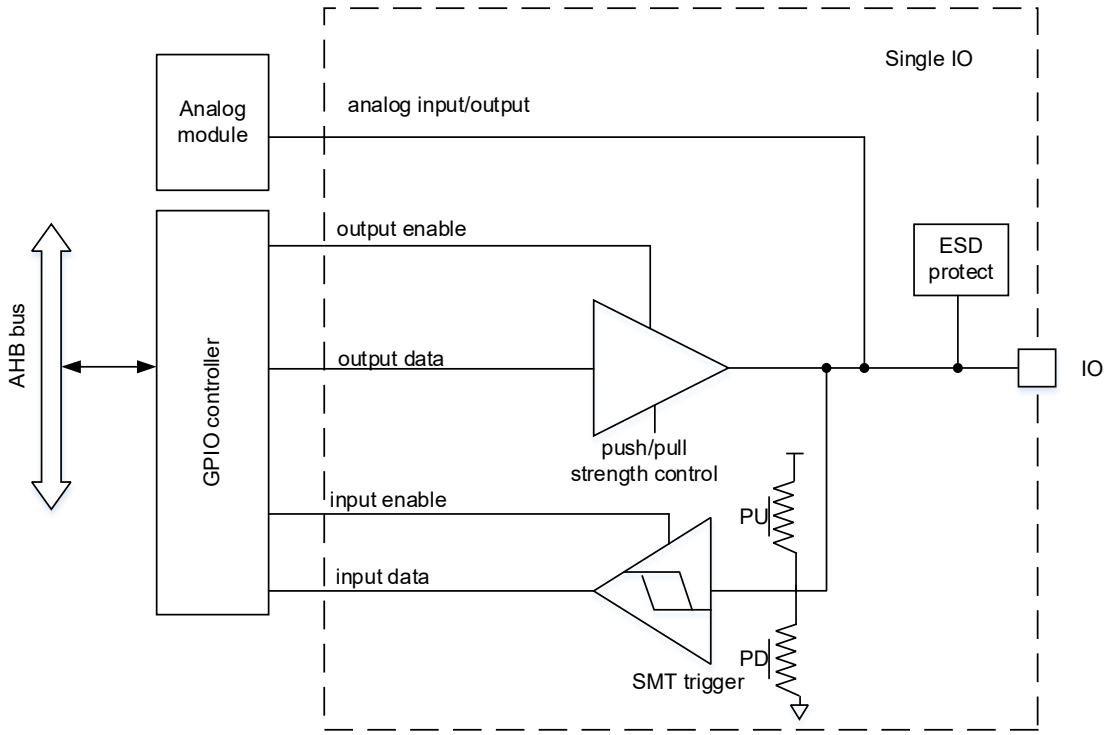


Figure 20. GPIO basic structure

GPIO reset status

After power-on or system reset, all pins are configured as floating input mode except JATG-related pins. JTAG pin configuration are as follows:

- PA15/JTDI, PA13/JTMS and PB4/JNTRST in pull-up input mode;
- PA14/JTCK in pull-down input mode;
- PB3/TDO in floating input mode.

General-purpose input configuration

Table 19. GPIO input mode configuration

Mode	IOFC	HDRV	IOMC[1]	IOMC[0]	ODT register
Floating input	01	000			Unused
Pull-down input	10				0
Pull-up input					1

When I/O port is configured as input:

- Get I/O states by reading input data register.
- Floating input, pull-up/pull-down input is configurable
- Schmitt-trigger input is activated.
- Output is disabled.

Note: In floating input mode, it is recommended to set the unused pins as analog input mode in order to avoid leakage caused by interference from unused pins in a complex environment.

Analog input/output configuration

Table 20. GPIO analog input/output mode configuration

Mode	IOFC	HDRV	IOMC[1]	IOMC[0]	ODT register
Analog input/output	00	000			Unused

When I/O port is configured as analog input:

- Schmitt-trigger input is disabled.
- Digital input/output is disabled.
- Without any pull-up/pull-down resistor.

General-purpose output configuration

Table 21. GPIO output mode configuration

Mode	IOFC	HDRV	IOMC[1]	IOMC[0]	ODT register
Push-Pull	00	000/100: Input mode 001: Output mode, large sourcing/sinking strength 010: Output mode, normal sourcing/sinking strength 011: Output mode, normal sourcing/sinking strength 1xx: Output mode, Maximum sourcing/sinking strength			0 or 1
Open-Drain	01				0 or 1

When I/O port is configured as output:

- Schmitt-trigger input is enabled.
- Output through output register.
- Pull-up/pull-down resistors are disabled.
- In open-drain mode, forced output 0, use pull-up resistor to output 1.
- In push-pull mode, output register is used to output 0/1.
- When CONF=10 or 11, it is used as a multiplexed output, refer to IOMUX for more details.

I/O port protection

Locking mechanism can freeze the I/O configuration for the purpose of protection. When LOCK is applied to a port bit, its configuration cannot be modified until the next reset or power on.

GPIO registers

The following table lists GPIO register map and their reset values. These peripheral registers must be accessed by words (32 bits).

Table 22. GPIO register map and reset values

Register	Offset	Reset value
GPIOx_CFGLR	0x00	0x4444 4444
GPIOx_CFGHR	0x04	0x4444 4444
GPIOx_IDT	0x08	0x0000 XXXX
GPIOx_ODT	0x0C	0x0000 0000
GPIOx_SCR	0x10	0x0000 0000
GPIOx_CLR	0x14	0x0000 0000
GPIOx_WPR	0x18	0x0000 0000
GPIOx_SRCTR	0x20	0x0000 0000
GPIOx_HDRV	0x3C	0x0000 0000

GPIO configuration register low (GPIOx_CFGLR) (x = A..E)

Bit	Register	Reset value	Type	Description
Bit 31:30 Bit 27:26 Bit 23:22 Bit 19:18 Bit 15:14 Bit 11:10 Bit 7:6 Bit 3:2	IOFCy	0x1	rw	GPIOx function configuration (y=0~7) In input mode (IOMCy[1:0]=00): 00: Analog mode 01: Floating input (reset state) 10: Pull-up/pull-down input 11: Reserved In output mode (IOMCy[1:0]!=00): 00: General-purpose push-pull output 01: General-purpose open-drain output 10: Alternate function push-pull output 11: Alternate function open-drain output
Bit 29:28 Bit 25:24 Bit 21:20 Bit 17:16 Bit 13:12 Bit 9:8 Bit 5:4 Bit 1:0	IOMCy	0x0	rw	GPIOx mode configuration (y=0~7) 00: Input mode (reset state) 01: Output mode, large sourcing/sinking strength 10: Output mode, normal sourcing/sinking strength 11: Output mode, normal sourcing/sinking strength

Note: Some port registers have different reset values. For example, some PA pins are JTAG/SWD with pull-up input by default.

GPIO configuration register high (GPIOx_CFGHR) (x = A..E)

Bit	Register	Reset value	Type	Description
Bit 31:30 Bit 27:26 Bit 23:22 Bit 19:18 Bit 15:14 Bit 11:10 Bit 7:6 Bit 3:2	IOFCy	0x1	rw	GPIOx function configuration (y=8~15) In input mode (IOMCy[1:0]=00): 00: Analog mode 01: Floating input (reset state) 10: Pull-up/pull-down input 11: Reserved In output mode (IOMCy[1:0]!=00): 00: General-purpose push-pull output 01: General-purpose open-drain output 10: Alternate function push-pull output 11: Alternate function open-drain output
Bit 29:28 Bit 25:24 Bit 21:20 Bit 17:16 Bit 13:12 Bit 9:8 Bit 5:4 Bit 1:0	IOMCy	0x0	rw	GPIOx mode configuration (y=8~15) 00: Input mode (reset state) 01: Output mode, large sourcing/sinking strength 10: Output mode, normal sourcing/sinking strength 11: Output mode, normal sourcing/sinking strength

Note: Some port registers have different reset values. For example, some PB pins are JTAG/SWD with pull-up input by default.

GPIO input data register (GPIOx_IDT) (x = A..E)

Bit	Register	Reset value	Type	Description
Bit 31:16	Reserved	0x0000	resd	Kept at its default value.
Bit 15:0	IDT	0xXXXX	ro	GPIOx input data Indicates the input status of I/O port. Each bit corresponds to an I/O.

GPIO output data register (GPIOx_ODT) (x = A..E)

Bit	Register	Reset value	Type	Description
Bit 31:16	Reserved	0x0000	resd	Kept at its default value.
Bit 15:0	ODT	0x0000	rw	GPIOx output data Each bit represents an I/O port. As output: it indicates the output status of I/O port. 0: Low 1: High As input: it indicates the pull-up/pull-down status of I/O port. 0: Pull-down 1: Pull-up

GPIO set/clear register (GPIOx_SCR) (x = A..E)

Bit	Register	Reset value	Type	Description
Bit 31:16	IOCB	0x0000	wo	GPIOx clear bit The corresponding ODT register bits are cleared by writing “1” to these bits. Writing 0 has no effect on the ODT register bits, which is equivalent to ODT register bit operations. 0: No action to the corresponding ODT bits 1: Clear the correspoinding ODT bits
Bit 15:0	IOSB	0x0000	wo	GPIOx set bit The corresponding ODT register bits are set by writing “1” to these bits.Writing 0 has no effect on the ODT register bits, which is equivalent to ODT register bit operations. If both IOCB and IOSB are set, the IOSB has priority. 0: No action to the correspoinding ODT bits 1: Set the correspoinding ODT bits

GPIO clear register (GPIOx_CLR) (x = A..E)

Bit	Register	Reset value	Type	Description
Bit 31:16	Reserved	0x0000	resd	Kept at its default value.
Bit 15:0	IOCB	0x0000	wo	GPIOx clear bit The corresponding ODT register bits are cleared by writing “1” to these bits. Writing o has no effect on the ODT register bit remains unchanged, which is equivalent to ODT register bit operations. 0: No action to the correspoinding ODT bits 1: Clear the correspoinding ODT bits

GPIO write protection register (GPIOx_WPR) (x = A..E)

Bit	Register	Reset value	Type	Description
Bit 31:17	Reserved	0x0000	resd	Kept at its default value.
Bit 16	WPSEQ	0x0	rw	Write protect sequence Write protect enable sequence bit and WPEN bit must be enabled at the same time to achieve write protection for some I/O bits. Write protect enable bit is executed four times in the order below: write “1” → write “0” → write “1” → read. Note that the value of WPSEL bit cannot be modified during this period.
Bit 15:0	WPEN	0x0000	rw	Write protect enable Each bit corresponds to an I/O port. 0: No effect. 1: Write protect

GPIO huge current control register (GPIOx_HDRV) (x = A..E)

Bit	Register	Reset value	Type	Description
Bit 31:16	Reserved	0x0000	resd	Kept at its default value.
Bit 15:0	HDRV	0x0000	rw	(Portx Huge sourcing/sinking strength control) (y = 0…15) Each bit corresponds to an I/O port. 0: GPIO is configured as large or normal sourcing/sinking strength, depending on IOMCy[1:0]. 1: GPIO is configured as maximum sourcing/sinking strength, ignoring IOMCy[1:0].

7 Multiplex function I/Os (IOMUX)

Introduction

HT32F49365/HT32F49395 support up to 80 bi-directional I/O pins, which are grouped as five categories, namely PA, PB, PC, PD and PE. Each of the GPIO group provides up to 16 I/O pins that feature communication, control and data collection. In addition, their main features also include:

- Supports general-purpose I/O (GPIO) or multiplex I/O (IOMUX), which will be detailed in this chapter.
- Can be configured as multiplex function input/output ports by setting GPIOx_CFGLR or GPIOx_CFGHR
- Most pins support output map for several peripherals. Select different peripheral input/output through IOMUX register
- Supports external interrupt

Functional descriptions

IOMUX structure

While being used as multiplexed function input, the I/O port should be configured as input modes (floating, pull-up and pull-down input)

To enable multiplexed function output, the port is configured as multiplexed function output mode (push-pull or open-drain) by setting GPIOx_CFGLR or GPIOx_CFGHR register. In this case, the pins are disconnected from GPIO controller, and controlled by IOMUX controller, instead.

To achieve bidirectional multiplexed function, the port needs to be configured as multiplexed function output modes (push-pull or open-drain), controlled by IOMUX controller.

In MUX output mode, it is possible that an I/O pin is used as output for several peripherals. Select the required multiplexed function output by setting IOMUX registers. However, when a pin is programmed as MUX IO without activating the corresponding peripheral , its output will not specified.

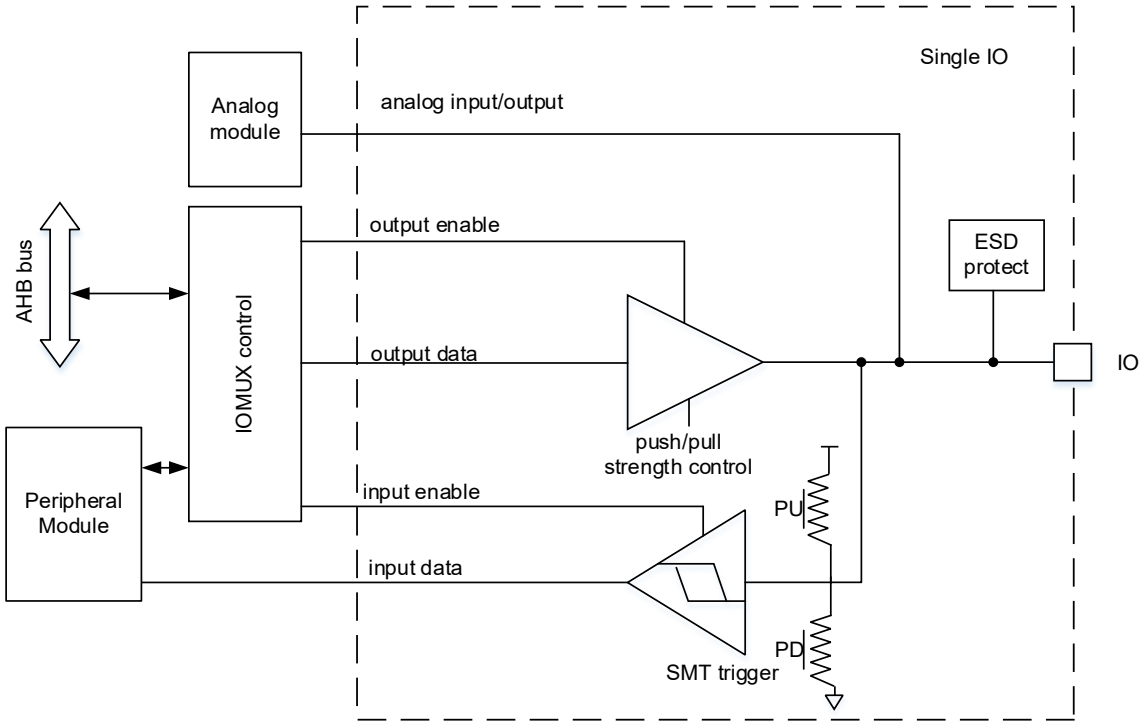


Figure 21. Basic structure of IOMUX basic structure

MUX Input configuration

When I/O ports are configured as multiplexed function input:

- Get I/O pin state by reading input data registers
- The pin be configured as floating input, pull-up or pull-down input
- Schmitt-trigger input is activated.
- Pin output is disabled.

Table 23. IOMUX input configuration

Mode	IOFC	HDRV	IOMC[1]	IOMC[0]	ODT register
Floating input	01	000			Unused
Pull-down input	10				0
Pull-up input					1

MUX output or bidirectional MUX configuration

When an I/O port is configured as MUX output or a bidirectional MUX:

- I/O pin output depends on the peripherals.
- Schmitt-trigger input is activated.
- Pull-up/pull-down resistor is disabled.
- If the I/O pin is set as several MUX outputs by mistake, the pin output depends on map priority, refer to next section for details.

- In open-drain mode, get an I/O port state by reading input data register
- In push-pull mode, get an I/O port state by reading input data register

The MUX functions of some peripherals can be remapped to different pins. Therefore, it is necessary to select the number of the desired peripheral IOMUX functions in different packages. Pin mapping is achieved by setting the IOMUX_REMAP and IOMUX_REMAPx registers.

Table 24. IOMUX output configuration

Mode	IOFC	HDRV	IOMC[1]	IOMC[0]
Push-Pull	10	001: Output mode, large sourcing/sinking strength 010: Output mode, normal sourcing/sinking strength 011: Output mode, normal sourcing/sinking strength 1xx: Output mode, maximum sourcing/sinking strength		
Open-Drain	11			

Note: For MUX function output or bidirectional MUX function, IOMC[1:0] > 00 must be met.

Peripheral MUX function configuration

When IOMUX function is used:

- To use a peripheral pin in MUX output, it is configured as multiplexed push-pull/open-drain output.
- To use a peripheral pin in MUX input, it is configured as floating input/pull-up/pull-down input.
- For ADC peripherals, the pins of analog channels should be configured as analog input/output mode.
- For I²C peripherals that intend to use pins as bidirectional functions, open-drain mode is required.

IOMUX map priority

When several peripheral MUX functions are mapped to the same pin, the priorirty below should be respected:

- Hardware preemption
- JTAG debug port
- Non-timer peripherals has priority over timer peripherals
- No priority applied among several non-timer peripherals, MUX function is overlapped to the same pin

Hardware preemption

Certain pins are occupied by specific hardware functions regardless of the GPIO configuration.

Table 25. Hardware preemption

Pin	Enable bit	Description
PA0	PWC_CTRLSTS[8] = 1	Once enabled, PA0 pin acts as WKUP function of PWC.
PA4	DAC_CTRL[2] = 1	Once enabled, PA4 pin acts as DAC1 analog channel.
PA5	DAC_CTRL[18] = 1	Once enabled, PA5 pin acts as DAC2 analog channel.
PA11	CRM_APB1EN[23] = 1	Once enabled, PA11 pin acts as USB_DM channel.
PA12	CRM_APB1EN[23] = 1	Once enabled, PA12 pin acts as USB_DP channel.
PC13	CRM_APB1EN[27] = 1& (BPR_CTRL[0] = 1 | BPR_RTCCAL[8] = 1 | BPR_RTCCAL[7] = 1)	Once enabled, PC13 pin acts as RTC channel.
PC14	CMR_BPDC[0] = 1	Once enabled, PC14 pin acts as LEXT channel.
PC15	CMR_BPDC[0] = 1	Once enabled, PC15 pin acts as LEXT channel.

Debug port priority

The programmed debug pins will remain its state during device debugging, regardless of their GPIO register configuration. By doing this, can the debug port be free from disturbance imposed by other peripherals.

To utilize more pins during this period, the above-mentioned remap configuration can be changed by setting the SWJTAG_MUX[2:0] bit in the IOMUX_REMAP register and SWJTAG_GMUX[2:0] bit in the IOMUX_REMAP7 register.

Table 26. Debug port map

SWJTAG_MUX[2:0] or SWJTAG_GMUX[2:0]	SWJI/O pin allocation				
	PA13/JTMS/ SWDIO	PA14/JTCK/ SWCLK	PA15/ JTDI	PB3/JTDO/ TRACESWO	PB4/ NJTRST
000	√	√	√	√	√
001	√	√	√	√	x
010	√	√	x	x	x
100	x	x	x	x	x
Others	—	—	—	—	—

Note: √ indicates that this pin is forcibly allocated to debug port,while x indicates that this pin can be released to other peripherals.

Other peripheral output priority

For other peripherals, their output priority are as follows:

- Non-timer peripherals have priority over timers. In other words, when other peripherals and timers are mapped to the same pin at the same time, the timer can not be output.
- When multiple non-timer peripherals are mapped to the same pin, their output are overlapped to this pin.

External interrupt/wake-up lines

Each pin can be used as an external interrupt input. The corresponding pin should be configured as input mode.

IOMUX registers

The following table shows IOMUX register map and their reset values, These peripheral registers must be accessed by words (32 bits).

Table 27. IOMUX register map and reset value

Register	Offset	Reset value
IOMUX_EVTOUT	0x00	0x0000 0000
IOMUX_REMAP	0x04	0x0000 0000
IOMUX_EXINTC1	0x08	0x0000
IOMUX_EXINTC2	0x0C	0x0000
IOMUX_EXINTC3	0x10	0x0000
IOMUX_EXINTC4	0x14	0x0000
IOMUX_REMAP2	0x1C	0x0000 0000
IOMUX_REMAP3	0x20	0x0000 0000
IOMUX_REMAP4	0x24	0x0000 0000

Register	Offset	Reset value
IOMUX_REMAP5	0x28	0x0000 0000
IOMUX_REMAP6	0x2C	0x0000 0000
IOMUX_REMAP7	0x30	0x0000 0000
IOMUX_REMAP8	0x34	0x0000 0000

Note: IOMUX clock must be enabled before read/write access to IOMUX_EVCTRL, IOMUX_REMAPx and IOMUX_EXINTx.

Event output control register (IOMUX_EVTOUT)

Bit	Register	Reset value	Type	Description
Bit 31:8	Reserved	0x000000	resd	Kept at its default value.
Bit 7	EVOEN	0x0	rw	Event output enable Once enabled, the EVENTOUT signal of Cortex-M is directed to the allocated I/O port.
Bit 6:4	SELPORT	0x0	rw	Selection IO port Select the GPIO port for EVENTOUT signal output: 000: GPIOA 001: GPIOB 010: GPIOC 011: GPIOD 100: GPIOE
Bit 3:0	SELPIN	0x0	rw	Selection IO pin (x=A…E) Select the I/O pin of GPIOx for EVENTOUT output: 0000: Pin 0 0001: Pin 1 0010: Pin 2 0011: Pin 3 0100: Pin 4 0101: Pin 5 0110: Pin 6 0111: Pin 7 1000: Pin 8 1001: Pin 9 1010: Pin 10 1011: Pin 11 1100: Pin 12 1101: Pin 13 1110: Pin 14 1111: Pin 15

IOMUX remap register (IOMUX_REMAP)

Bit	Register	Reset value	Type	Description
Bit 31	SPI1_MUX	0x0	rw	SPI1 IO multiplexing Refer to bit 0 for more details.
Bit 30	Reserved	0x0	resd	Kept at its default value.
Bit 29	TMR2ITR1_MUX	0x0	rw	TMR2 internal trigger 1 multiplexing This bit is used to select TMR2 ITR1. 0: TMR8_TRGO is used as TMR2ITR1 input. 1: Ethernet PTP output is used as TMR2 ITR1 input. Note: This pin is reserved for future use.
Bit 28	SPI3_MUX	0x0	rw	SPI3 IO multiplexing This bit is used to select CS, SCK, MISO and MOS for SPI3 IO. 0: CS/PA15, SCK/PB3, MISO/PB4 and MOSI/PB5. 1: CS/PA4, SCK/PC10, MISO/PC11 and MOSI/PC12. Note: This pin is reserved for future use.
Bit 27	Reserved	0x0	resd	Kept at its default value.
Bit 26:24	SWJTAG_MUX	0x0	rw	SWD JTAG mutiplexing These bits are used to configure SWJTGA-related I/Os as GPIOs. 000: Supports SWD and JTAG. All SWJTAG pins cannot be used as GPIOs. 001: Supports SWD and JTAG. NJTRST is disabled. PB4 can be used as GPIO. 010: Supports SWD but JTAG is disabled. PA15/PB3/PB4 can be used as GPIOs. 100: SWD and JTAG are disabled. All SWJTAG pins can be used as GPIOs. Others: No effect.
Bit 23	MII_RMII_SEL	0x0	rw	MII or RMII selection Select Ethernet MII or RMII interface. 0: MII 1: RMII Note: This pin is reserved for future use.
Bit 22	CAN2_MUX	0x0	rw	CAN2 IO multiplexing This bit is used to select IO multiplexing for CAN2_TX and CAN2_RX. 0: RX/PB12, TX/PB13 1: RX/PB5, TX/PB6 Note: This pin is reserved for future use.
Bit 21	Reserved	0x0	resd	Kept at its default value.
Bit 20	ADC2_ETO_MUX	0x0	rw	ADC2 external trigger ordinary conversion multiplexing Select external trigger input for ADC2 ordinary conversion. 0: ADC2 external trigger ordinary conversion is connected to EXINT11 1: ADC2 external trigger ordinary conversion is connected to TMR8_TRGO
Bit 19	ADC2_ETP_MUX	0x0	rw	ADC2 external trigger preempted conversion mutiplexing Select external trigger input for ADC2 preempted conversion. 0: ADC2 external trigger preempted conversion is connected to EXINT15. 1: ADC2 external trigger preempted conversion to TMR8 channel 4.

Bit	Register	Reset value	Type	Description
Bit 18	ADC1_ETO_MUX	0x0	rw	ADC1 external trigger regular conversion mutiplexing Select external trigger input for ADC1 ordinary conversion. 0: ADC1 external trigger ordinary conversion is connected to EXINT11. 1: ADC1 external trigger ordinary conversion TMR8_TRGO.
Bit 17	ADC1_ETP_MUX	0x0	rw	ADC1 external trigger preempted conversion multiplexing. Select external trigger input for ADC1 preempted conversion. 0: ADC1 external trigger preempted conversion is connected to EXINT15. 1: ADC1 external trigger preempted conversion is connected to TMR8 channel 4.
Bit 16	TMR5CH4_MUX	0x0	rw	TMR5 channel4 multiplexing Select internal map for TMR5 channel 4. 0: TMR5_CH4 is connected to PA3. 1: TMR5_CH4 is connected to LICK. LICK can be calibrated.
Bit 15	PD01_MUX	0x0	rw	PD0/PD1 mapped on HEXT_IN / HEXT_OUT Select GPIO function map for PD0 and PD1. This is only applicable to 48-pin/64-pin packages. 0: Not PD0 and PD1 mapping 1: PD0 is mapped to HEXT_IN, while PD1 to HEXT_OUT
Bit 14:13	CAN_MUX	0x0	rw	CAN IO multiplexing Select IO multiplexing for CAN_TX and CAN_RX. 00: RX/PA11, TX/PA12 01: Unused 10: RX/ PB8, TX/ PB9 11: RX/ PD0, TX/ PD1
Bit 12	TMR4_MUX	0x0	rw	TMR4 IO multiplexing Select IO multiplexing for TMR4. 0: CH1/PB6, CH2/PB7, CH3/PB8 and CH4/PB9 1: CH1/PD12, CH2/PD13, CH3/PD14 and CH4/PD15
Bit 11:10	TMR3_MUX	0x0	rw	TMR3 IO multiplexing Select IO multiplexing for TMR3. 00: CH1/PA6, CH2/PA7, CH3/PB0 and CH4/PB1 01: Unused 10: CH1/PB4, CH2/PB5, CH3/PB0 and CH4/PB1 11: CH1/PC6, CH2/PC7, CH3/PC8 and CH4/PC9 Note: IO multiplexing has no impact on TMR3_EXT on PD2.
Bit 9:8	TMR2_MUX	0x0	rw	TMR2 IO multiplexing Select IO multiplexing for TMR2. 00: CH1/EXT/PA0, CH2/PA1, CH3/PA2 and CH4/PA3 01: CH1/EXT/PA15, CH2/PB3, CH3/PA2 and CH4/PA3 10: CH1/EXT/PA0, CH2/PA1, CH3/PB10 and CH4/PB11 11: CH1/EXT/PA15, CH2/PB3, CH3/PB10 and CH4/PB11

Bit	Register	Reset value	Type	Description
Bit 7:6	TMR1_MUX	0x0	rw	TMR1 IO multiplexing Select IO multiplexing for TMR1. 00: EXT/PA12, CH1/PA8, CH2/PA9 and CH3/PA10, CH4/PA11, BRK/PB12, CH1C/PB13, CH2C/PB14, CH3C/PB15 01: EXT/PA12, CH1/PA8, CH2/PA9, CH3/PA10, CH4/PA11, BRK/PA6, CH1C/PA7, CH2C/PB0, CH3C/PB1 10: Unused 11: EXT/PE7, CH1/PE9, CH2/PE11, CH3/PE13, CH4/PE14, BRK/PE15, CH1C/PE8, CH2C/PE10, CH3C/PE12
Bit 5:4	USART3_MUX	0x0	rw	USART3 IO multiplexing Select IO multiplexing for USART3. 00: TX/PB10, RX/PB11, CK/PB12, CTS/PB13, RTS/PB14 01: TX/PC10, RX/PC11, CK/PC12, CTS/PB13, RTS/PB14 10: Unused 11: TX/PD8, RX/PD9, CK/PD10, CTS/PD11, RTS/PD12
Bit 3	USART2_MUX	0x0	rw	USART2 IO mutiplexing Select IO multiplexing for USART2. 0: CTS/PA0, RTS/PA1, TX/PA2, RX/PA3 and CK/PA4 1: CTS/PD3, RTS/PD4, TX/PD5, RX/PD6 and CK/PD7
Bit 2	USART1_MUX	0x0	rw	USART1 IO multiplexing Select USART1 IO multiplexing 0: TX/PA9, RX/PA10 1: TX/PB6, RX/PB7
Bit 1	I2C1_MUX	0x0	rw	I^2C1 IO multiplexing Select I^2C1 IO multiplexing. 0: SCL/PB6, SDA/PB7 SMBA/PB5 1: SCL/PB8, SDA/PB9 SMBA/PB5
Bit 0	SPI1_MUX	0x0	rw	SPI1 IO multiplexing Select SPI1 IO multiplexing. SPI1_MUX[1] is in bit 31. 00: CS/PA4, SCK/PA5, MISO/PA6, MOSI/PA7, MCK/PB0 01: CS/PA15, SCK/PB3, MISO/PB4, MOSI/PB5, MCK/PB0 10: CS/PA4, SCK/PA5, MISO/PA6, MOSI/PA7, MCK/PB6 11: CS/PA15, SCK/PB3, MISO/PB4, MOSI/PB5, MCK/PB6

IOMUX external interrupt configuration register1 (IOMUX_EXINTC1)

Bit	Register	Reset value	Type	Description
Bit 31:16	Reserved	0x0000	resd	Kept at its default value.
Bit 15:12	EXINT3	0x0000	rw	EXINT3 input source configuration Select the input source for EXINT3 external interrupt. 0000: GPIOA pin3 0001: GPIOB pin3 0010: GPIOC pin3 0011: GPIOD pin3 0100: GPIOE pin3 Others: Reserved
Bit 11:8	EXINT2	0x0000	rw	EXINT2 input source configuration Select the input source for EXINT2 external interrupt. 0000: GPIOA pin2 0001: GPIOB pin2 0010: GPIOC pin2 0011: GPIOD pin2 0100: GPIOE pin2 Others: Reserved
Bit 7:4	EXINT1	0x0000	rw	EXINT1 input source configuration Select the input source for EXINT1 external interrupt. 0000: GPIOA pin1 0001: GPIOB pin1 0010: GPIOC pin1 0011: GPIOD pin1 0100: GPIOE pin1 Others: Reserved
Bit 3:0	EXINT0	0x0000	rw	EXINT0 input source configuration Select the input source for EXINT0 external interrupt. 0000: GPIOA pin0 0001: GPIOB pin0 0010: GPIOC pin0 0011: GPIOD pin0 0100: GPIOE pin0 Others: Reserved

IOMUX external interrupt configuration register2 (IOMUX_EXINTC2)

Bit	Register	Reset value	Type	Description
Bit 31:16	Reserved	0x0000	resd	Kept at its default value.
Bit 15:12	EXINT7	0x0000	rw	EXINT7 input source configuration Select the input source for EXINT7 external interrupt. 0000: GPIOA pin7 0001: GPIOB pin7 0010: GPIOC pin7 0011: GPIOD pin7 0100: GPIOE pin7 Others: Reserved
Bit 11:8	EXINT6	0x0000	rw	EXINT6 input source configuration Select the input source for EXINT6 external interrupt. 0000: GPIOA pin6 0001: GPIOB pin6 0010: GPIOC pin6 0011: GPIOD pin6 0100: GPIOE pin6 Others: Reserved.
Bit 7:4	EXINT5	0x0000	rw	EXINT5 input source configuration Select the input source for EXINT5 external interrupt. 0000: GPIOA pin5 0001: GPIOB pin5 0010: GPIOC pin5 0011: GPIOD pin5 0100: GPIOE pin5 Others: Reserved
Bit 3:0	EXINT4	0x0000	rw	EXINT4 input source configuration Select the input source for EXINT4 external interrupt. 0000: GPIOA pin4 0001: GPIOB pin4 0010: GPIOC pin4 0011: GPIOD pin4 0100: GPIOE pin4 Others: Reserved

IOMUX external interrupt configuration register3 (IOMUX_EXINTC3)

Bit	Register	Reset value	Type	Description
Bit 31:16	Reserved	0x0000	resd	Kept at its default value.
Bit 15:12	EXINT11	0x0000	rw	EXINT11 input source configuration Select the input source for EXINT11 external interrupt. 0000: GPIOA pin11 0001: GPIOB pin11 0010: GPIOC pin11 0011: GPIOD pin11 0100: GPIOE pin11 Others: Reserved
Bit 11:8	EXINT10	0x0000	rw	EXINT10 input source configuration Select the input source for EXINT10 external interrupt. 0000: GPIOA pin10 0001: GPIOB pin10 0010: GPIOC pin10 0011: GPIOD pin10 0100: GPIOE pin10 Others: Reserved
Bit 7:4	EXINT9	0x0000	rw	EXINT9 input source configuration Select the input source for EXINT9 external interrupt. 0000: GPIOA pin9 0001: GPIOB pin9 0010: GPIOC pin9 0011: GPIOD pin9 0100: GPIOE pin9 Others: Reserved
Bit 3:0	EXINT8	0x0000	rw	EXINT8 input source configuration Select the input source for EXINT8 external interrupt. 0000: GPIOA pin8 0001: GPIOB pin8 0010: GPIOC pin8 0011: GPIOD pin8 0100: GPIOE pin8 Others: Reserved

IOMUX external interrupt configuration register4 (IOMUX_EXINTC4)

Bit	Register	Reset value	Type	Description
Bit 31:16	Reserved	0x0000	resd	Kept at its default value.
Bit 15:12	EXINT15	0x0000	rw	EXINT15 input source configuration Select the input source for EXINT15 external interrupt. 0000: GPIOA pin15 0001: GPIOB pin15 0010: GPIOC pin15 0011: GPIOD pin15 0100: GPIOE pin15 Others: Reserved
Bit 11:8	EXINT14	0x0000	rw	EXINT14 input source configuration Select the input source for EXINT14 external interrupt. 0000: GPIOA pin14 0001: GPIOB pin14 0010: GPIOC pin144 0011: GPIOD pin14 0100: GPIOE pin14 Others: Reserved
Bit 7:4	EXINT13	0x0000	rw	EXINT13 input source configuration Select the input source for EXINT13 external interrupt. 0000: GPIOA pin13 0001: GPIOB pin13 0010: GPIOC pin13 0011: GPIOD pin13 0100: GPIOE pin13 Others: Reserved
Bit 3:0	EXINT12	0x0000	rw	EXINT12 input source configuration Select the input source for EXINT12 external interrupt. 0000: GPIOA pin12 0001: GPIOB pin12 0010: GPIOC pin12 0011: GPIOD pin12 0100: GPIOE pin12 Others: Reserved

IOMUX remap register2 (IOMUX_REMAP2)

Bit	Register	Reset value	Type	Description
Bit 31:22	Reserved	0x000	resd	Keep at its default value.
Bit 21	SPIM_EN	0x0	rw	SPIM enable Select whether to use SPI Flash.
Bit 20:19	SDIO2_MUX	0x0	rw	SDIO2_MUX[1:0]: SDIO2 IO multiplexing 00: D0/PC0 D1/PC1 D2/PC3 D4/PA4 D5/PA5 D6/PA6 D7/PA7 CK/PC4 CMD/PC5 01: D0/PA4 D1/PA5 D2/PA6 D3/PA7 CK/PA2 CMD/PA3 10, 11: Unused
Bit 18	I2C3_MUX	0x0	rw	I^2C3_MUX: I^2C3 IO mutiplexing Select IO multiplexing for I^2C3. 0: SCL/PA8 SDA/PC9 SMBA/PA9 1: SCL/PA8 SDA/PB4 SMBA/PA9
Bit 17	SPI4_MUX	0x0	rw	SPI4_MUX: SPI4 IO multiplexing Select IO multiplexing for SPI4. 0: CS/PE4 SCK/PE2 MISO/PE5 MOSI/PE6 MCK/PC8 1: CS/PE12 SCK/PE11 MISO/PE13 MOSI/PE14 MCK/PC8
Bit 16:11	Reserved	0x00	resd	Keep its default value.
Bit 10	XMC_NADV_MUX	0x0	rw	XMC_NADV_MUX: XMC NADV connect Select whether to use XMC_NADV signal. 0: XMC_NADV is connected to the corresponding pin (by default) 1: XMC_NADV is not used. The corresponding pin canbe used for other peripherals.
Bit 9:6	Reserved	0x0	resd	Kept at its default value.
Bit 5	TMR9_MUX	0x0	rw	TMR9_MUX: TMR9 IO multiplexing Select IO multiplexing for TMR9. 0: CH1/PA2, CH2/PA3 1: CH1/PE5, CH2/PE6
Bit 4:0	Reserved	0x00	resd	Kept at its default value.

IOMUX remap register3 (IOMUX_REMAP3)

Bit	Register	Reset value	Type	Description
Bit 31:4	Reserved	0x0000000	resd	Kept at its default value.
Bit 3:0	TMR9_GMUX	0x0	rw	TMR9 IO general multiplexing Select IO multiplexing for TMR9. 0000: CH1/PA2, CH2/PA3 0001: CH1/PE5, CH2/PE6

IOMUX remap register4 (IOMUX_REMAP4)

Bit	Register	Reset value	Type	Description
Bit 31:20	Reserved	0x000	resd	Kept at its default value.
Bit 19	TMR5CH4_GMUX	0x0	rw	TMR5 channel4 general multiplexing Select TMR5 channel4 general multiplexing 0: TMR5_CH4 is connected to PA3. 1: LICK is connected to TMR5_CH4 to get calibration
Bit 18:16	Reserved	0x0	resd	Kept at its default value.
Bit 15:12	TMR4_GMUX	0x0	rw	TMR4 IO general multiplexing Select IO multiplexing for TMR4. 0000: CH1/PB6 CH2/PB7 CH3/PB8 CH4/PB9 0001: CH1/PD12 CH2/PD13 CH3/PD14 CH4/PD15
Bit 11:8	TMR3_GMUX	0x0	rw	TMR3 IO general multiplexing Select IO multiplexing for TMR3. 0000: CH1/PA6 CH2/PA7 CH3/PB0 CH4/PB1 0001: CH1/PB4 CH2/PB5 CH3/PB0 CH4/PB1
Bit 7:6	TMR2ITR1_GMUX	0x0	rw	TMR2 internal trigger 1 general multiplexing Select TMR2_ITR1 general multiplexing 00: TMR8_TRGO is used as input source of TMR2 ITR1 01: Reserved. Do not use. 10: Reserved. Do not use. 11: SB SOF is used as input source of TMR2_ITR1 (This selection will cause TMR2_GMUX/TMR2_MUX failure. Pay more attention to this restriction)
Bit 5:4	TMR2_GMUX	0x0	rw	TMR2 IO general multiplexing Select IO multiplexing for TMR2. 00: CH1_EXT/PA0 CH2/PA1 CH3/PA2 CH4/PA3 01: CH1_EXT/PA15 CH2/PB3 CH3/PA2 CH4/PA3 10: CH1_EXT/PA0 CH2/PA1 CH3/PB10 CH4/PB11 11: CH1_EXT/PA15 CH2/PB3 CH3/PB10 CH4/PB11
Bit 3:0	TMR1_GMUX	0x0	rw	TMR1 IO general multiplexing Select IO multiplexing for TMR1. 0000: EXT/PA12, CH1/PA8, CH2/PA9, CH3/PA10, CH4/PA11, BRK/PB12, CH1C/PB13, CH2C/PB14, CH3C/PB15 0001: EXT/PA12, CH1/PA8, CH2/PA9, CH3/PA10, CH4/PA11, BRK/PA6, CH1C/PA7, CH2C/PB0, CH3C/PB1 0011: EXT/PE7, CH1/PE9, CH2/PE11, CH3/PE13, CH4/PE14, BRK/PE15, CH1C/PE8, CH2C/PE10, CH3C/PE12 Others: Unused

IOMUX remap register5 (IOMUX_REMAP5)

Bit	Register	Reset value	Type	Description
Bit 31:28	SPI4_GMUX	0x0	rw	SPI4 IO general multiplexing Select IO multiplexing for SPI4. 0000: CS/PE4 SCK/PE2 MISO/PE5 MOSI/PE6 MCK/PC8 0001: CS/PE12 SCK/PE11 MISO/PE13 MOSI/PE14 MCK/PC8 0010: CS/PB6 SCK/PB7 MISO/PB8 MOSI/PB9 MCK/PC8 0011: CS/PB6 SCK/PB7 MISO/PB8 MOSI/PB9 MCK/PA10 Others: Unused
Bit 27:24	SPI3_GMUX	0x0	rw	SPI3 IO general multiplexing Select IO multiplexing for SPI3. 0000: CS/PA15, SCK/PB3, MISO/PB4, MOSI/PB5 MCK/PC7 0001: CS/PA4, SCK/PC10, MISO/PC11, MOSI/PC12 MCK/PC7 0010: CS/PA15, SCK/PB3, MISO/PB4, MOSI/PB5 MCK/PB10 0011: CS/PA4, SCK/PC10, MISO/PC11, MOSI/PC12 MCK/PB10 Others: Unused
Bit 23:20	SPI2_GMUX	0x0	rw	SPI2 IO general multiplexing Select IO multiplexing for SPI2. 0000: MCK/PC6 0001: MCK/PA3 0010: MCK/PA6 Others: Unused
Bit 19:16	SPI1_GMUX	0x0	rw	SPI1 IO general multiplexing Select IO multiplexing for SPI1. 0000: CS/PA4, SCK/PA5, MISO/PA6, MOSI/PA7 MCK/PB0 0001: CS/PA15, SCK/PB3, MISO/PB4, MOSI/PB5 MCK/PB0 0010: CS/PA4, SCK/PA5, MISO/PA6, MOSI/PA7 MCK/PB6 0011: CS/PA15, SCK/PB3, MISO/PB4, MOSI/PB5 MCK/PB6 Others: Unused
Bit 15:12	I2C3_GMUX	0x0	rw	I^2C3 IO general multiplexing Select IO multiplexing for I^2C3. 0000: SCL/PA8 SDA/PC9 SMBA/PA9 0001: SCL/PA8 SDA/PB4 SMBA/PA9 Others: Unused
Bit 11:8	Reserved	0x0	resd	Kept at its default value.
Bit 7:4	I2C1_GMUX	0x0	rw	I^2C1 IO general multiplexing Select IO multiplexing for I^2C1. 0000: SCL/PB6, SDA/PB7, SMBA/PB5 0001: SCL/PB8, SDA/PB9, SMBA/PB5 Others: Unused

Bit	Register	Reset value	Type	Description
Bit 3:0	USART5_GMUX	0x0	rw	USART5 IO general multiplexing Select IO multiplexing for USART5. 0000: TX/PC12 RX/PD12 0001: TX/PB9 RX/PB8 Others: Unused

IOMUX remap register6 (IOMUX_REMAP6)

Bit	Register	Reset value	Type	Description
Bit 31:28	UART4_GMUX	0x0	rw	IO general multiplexing Select IO multiplexing for UART4. 0000: TX/PC10 RX/PC11 0010: TX/PA0 RX/PA1 Others: Unused
Bit 27:24	USART3_GMUX	0x0	rw	USART3 IO general multiplexing Select IO multiplexing for USART3. 0000: TX/PB10, RX/PB11, CK/PB12, CTS/PB13, RTS/PB14 0001: TX/PC10, RX/PC11, CK/PC12, CTS/PB13, RTS/PB14 0011: TX/PD8, RX/PD9, CK/PD10, CTS/PD11, RTS/PD12 Others: Unused
Bit 23:20	USART2_GMUX	0x0	rw	USART2 IO general multiplexing Select IO multiplexing for USART2. 0000: CTS/PA0, RTS/PA1, TX/PA2, RX/PA3, CK/PA4 0001: CTS/PD3, RTS/PD4, TX/PD5, RX/PD6, CK/PD7 Others: Unused
Bit 19:16	USART1_GMUX	0x0	rw	USART1 IO general multiplexing Select IO multiplexing for USART1. 0000: TX/PA9, RX/PA10 0001: TX/PB6, RX/PB7 Others: Unused
Bit 15:12	SDIO2_GMUX	0x0	rw	SDIO2 IO general multiplexing Select IO multiplexing for SDIO2. 0000: D0/PC0 D1/PC1 D2/PC3 D4/PA4 D5/PA5 D6/PA6 D7/PA7 CK/PC4 CMD/PC5 0001: D0/PA4 D1/PA5 D2/PA6 D3/PA7 CK/PA2 CMD/PA3 Others: Unused
Bit 11:8	Reserved	0x0	resd	Kept at its default value.
Bit 7:4	CAN2_GMUX	0x0	rw	CAN2 IO general multiplexing Select IO multiplexing for CAN2. 0000: RX/PB12, TX/PB13 0001: RX/PB5, TX/PB6 Others: Unused
Bit 3:0	CAN1_GMUX	0x0	rw	CAN1 IO general multiplexing Select IO multiplexing for CAN1. 00: RX/PA11, TX/PA12 10: RX/ PB8, TX/ PB9 11: RX/ PD0, TX/ PD1 Others: Unused

IOMUX remap register7 (IOMUX_REMAP7)

Bit	Register	Reset value	Type	Description
Bit 31:28	Reserved	0x0	resd	Kept at its default value.
Bit 27	XMC_NADV_GMUX	0x0	rw	XMC NADV IO general multiplexing Select whether to use XMC_NADV signal. 0: XMC_NADV is connected to the corresponding pin (by default) 1: XMC_NADV is not used. The corresponding pin can be used by other pheripherals.
Bit 26:24	XMC_GMUX	0x0	rw	XMC IO general multiplexing Select IO multiplexing for XMC. 0000: NEW/PD5 D0/PD14 D1/PD15 D2/PD0 D3/PD1 D4/PE7 D5/PE8 D6/PE9 D7/PE10 D13/PD8 NOE/PD4 0001: NEW/PD2 D0/PB14 D1/PC6 D2/PC11 D3/PC12 D4/PA2 D5/PA3 D6/PA4 D7/PA5 D13/PB12 NOE/PC5 0010: NEW/PC2 D0/PB14 D1/PC6 D2/PC11 D3/PC12 D4/PA2 D5/PA3 D6/PA4 D7/PA5 D13/PB12 NOE/PC5 Others: Unused
Bit 23:21	Reserved	0x0	resd	Kept at its default value.
Bit 20	PD01_GMUX	0x0	rw	PD0/PD1 mapped onto HEXT_IN / HEXT_OUT Select GPIO mapping for PD0 and PD1. This is applied to only 48-pin and 64-pin packages. 0: No PD0 and PD1 mapping 1: PD0 is mapped to HEXT_IN, while PD1 is mapped to HEXT_OUT.
Bit 19	Reserved	0x0	resd	Kept at its default value.
Bit 18:16	SWJTAG_GMUX	0x0	rw	SWD JTAG IO general mutiplexing These bits are used to configure SWJTAG-related IOs as GPIO. 000: Supports SWD and JTAG. All SWJTAG pins cannot be used as GPIO. 001: Supports SWD and JTAG. NJTRST is disabled. PB4 can be used as GPIO. 010: Supports SWD. But JTAG is disabled. PA15/PB3/PB4 can be used as GPIO. 100: SWD and JTAG are disabled. All SWJTAG pins can be used as GPIO Others: No effect.
Bit 15:10	Reserved	0x00	resd	Kept at its default value.
Bit 9	ADC2_ETO_GMUX	0x0	rw	ADC2 external trigger regular conversion general multiplexing Select the input source for ADC2 external trigger regular conversion 0: ADC2 external trigger regular conversion is connected to EXINT11 1: ADC2 external trigger regular conversion is connected to TMR8_TRGO.

Bit	Register	Reset value	Type	Description
Bit 8	ADC2_ETP_GMUX	0x0	rw	ADC2 external trigger preempted conversion general multiplexing Select the input source for ADC2 external trigger preempted conversion 0: ADC2 external trigger preempted conversion is connected to EXINT15 1: ADC2 external trigger preempted conversion is connected to TMR8 channel 4
Bit 7:6	Reserved	0x0	resd	Keep at its default value.
Bit 5	ADC1_ETO_GMUX	0x0	rw	ADC1 external trigger regular conversion general multiplexing Select the input source for ADC1 external trigger regular conversion. 0: ADC1 external trigger regular conversion is connected to EXINT11 1: ADC1 external trigger regular conversion is connected to TMR8_TRGO
Bit 4	ADC1_ETP_GMUX	0x0	rw	ADC1 External trigger preempted conversion general multiplexing Select the input source for ADC1 External trigger preempted conversion. 0: ADC1 External trigger preempted conversion is connected to EXINT15 1: ADC1 External trigger preempted conversion is connected to TMR8 channel 4
Bit 3	EXT_SPIM_GEN	0x0	rw	SPIM enable Select whether to use SPI Flash.
Bit 2:0	EXT_SPIM_GMUX	0x0	rw	Select IO multiplexing for SPIM 000: SCK/PB1 CS/PA8 IO0/PA11 IO1/PA12 IO2/PB7 IO3/PB6 001: SCK/PB1 CS/PA8 IO0/PB10 IO1/PB11 IO2/PB7 IO3/PB6 Others: Unused

IOMUX remap register8 (IOMUX_REMAP8)

Bit	Register	Reset value	Type	Description
Bit 31:28	UART8_GMUX	0x0	rw	UART8 IO general multiplexing Select IO multiplexing for UART8. 0000: TX/PE1 RX/PE0 0001: TX/PC2 RX/PC3 Others: Unused
Bit 27:24	UART7_GMUX	0x0	rw	UART7 IO general multiplexing Select IO multiplexing for UART7. 0000: TX/PE8 RX/PE7 0001: TX/PB4 RX/PB3 Others: Unused
Bit 23:20	USART6_GMUX	0x0	rw	USART6 IO general multiplexing Select IO multiplexing for USART6. 0000: TX/PC6 RX/PC7 0001: TX/PA4 RX/PA5 Others: Unused
Bit 19	Reserved	0x0	resd	Kept at its default value.
Bit 18	Reserved	0x0	resd	Kept at its default value.
Bit 17:16	Reserved	0x0	resd	Kept at its default value.
Bit 15:0	Reserved	0x0000	resd	Kept at its default value.

8 External interrupt/Event controller (EXINT)

EXINT introduction

EXINT consists of 19 interrupt lines EXINT_LINE[18:0], each of which can generate an interrupt or event by edge detection trigger or software trigger. EXINT can enable or disable an interrupt or event independenttly through software configuration, and utilizes different edge detection modes (rising edge, falling edge or both edges) as well as trigger modes (edge detection, software trigger or both tirggers) to respond to the trigger source in order to generate an interrupt or event.

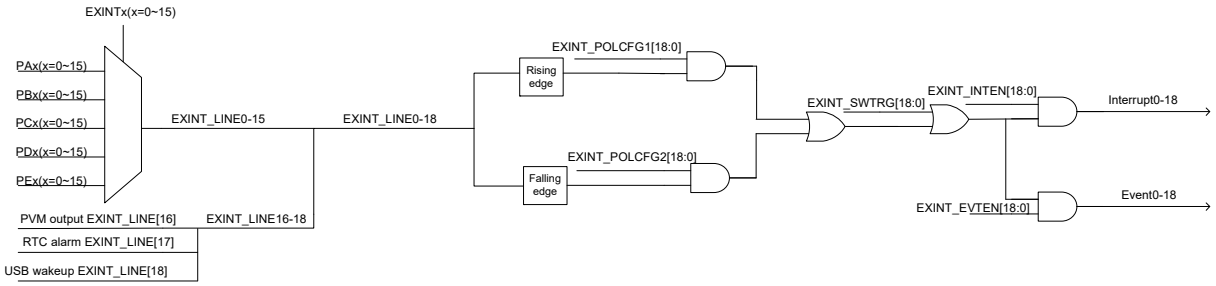


Figure 22. External interrupt/Event controller block diagram

Main features

- EXINT 0~15 mapping IO can be configured independently
- Independent trigger selection on each interrupt line
- Independent enable bit on each interrupt
- Independent enable bit on each event
- Up to 19 software trigger that can be generated and cleared independently.
- Independent status bit on each interrupt
- Each interrupt can be cleared independently.

Function descriptions and configuration procedure

With up to 19 interrupt lines EXINT_LINE[18:0], EXINT can detect not only GPIO external interrupt sources but also three internal sources such as PVM output, RTC alarm events and USB wakeup events through edge detection mechanism, where, GPIO interrupt sources can be selected with IOMUX_EXINTCx register. It should be noted that these input sources are mutually exclusive. For example, EXINT_LINE0 is allowed to select only one of PA0/PB0/PC0/PD0 pins, instead of taking both PA0 and PB0 as the input sources at the same time.

EXINT supports several edge detection modes, including rising edge, falling edge or both edges, selected by EXINT_POLCFG1 and EXINT_POLCFG2 register. Active edge trigger detected on the interrupt line can be used to generate an events or interrupt.

In addition, EXINT supports independent software trigger for the generation of an event or interrupt. This is achieved by setting the corresponding bits in the EXINT_SWTRG register.

EXINT can enable or disable an interrupt or event independently through software configuration such as EXINT_INTEN and EXINT_EVTEN register, indicating that the corresponding interrupt or event must be enabled prior to either edge detection or software trigger.

EXINT also features an independent interrupt status bit. Reading access to EXINT_INTSTS register can obtain the corresponding interrupt status. The status flag is cleared by writing "1" to this register.

Interrupt initialization procedure

- Select an interrupt source by setting IOMUX_EXINTCx register (This is required if GPIO is used as an interrupt source)
- Select an trigger mode by setting EXINT_POLCFG1 and EXINT_POLCFG2 register
- Enable interrupt or event by setting EXINT_INTEN and EXINT_EVTEN register
- Generate software trigger by setting EXINT_SWTRG register (This is applied to only software trigger interrupt)

Interrupt clear procedure

- Writing "1" to the EXINT_INTSTS register to clear the interrupts generated, and the corresponding bits in the EXINT_SWTRG register.

EXINT registers

These peripheral registers must be accessed by words (32 bits). The following table shows EXINT register map and their reset value.

Table 28. External interrupt/Event controller register map and reset value

Register	Offset	Reset value
EXINT_INTEN	0x00	0x0000 0000
EXINT_EVTEN	0x04	0x0000 0000
EXINT_ POLCFG1	0x08	0x0000 0000
EXINT_ POLCFG2	0x0C	0x0000 0000
EXINT_ SWTRG	0x10	0x0000 0000
EXINT_ INTSTS	0x14	0x0000 0000

Interrupt enable register (EXINT_INTEN)

Bit	Register	Reset value	Type	Description
Bit 31:20	Reserved	0x000	resd	Forced to 0 by hardware.
Bit 19:0	INTENx	0x00000	rw	Interrupt enable or disable on line x 0: Interrupt request is disabled 1: Interrupt request is enabled Note: Bit 19 is reserved for future use.

Event enable register (EXINT_EVTEN)

Bit	Register	Reset value	Type	Description
Bit 31:20	Reserved	0x000	resd	Forced to 0 by hardware.
Bit 19:0	EVTENx	0x00000	rw	Event enable or disable on line x 0: Event request is disabled 1: Event request is enabled Note: Bit 19 is reserved for future use.

Polarity configuration register1 (EXINT_ POLCFG1)

Bit	Register	Reset value	Type	Description
Bit 31:20	Reserved	0x000	resd	Forced to 0 by hardware.
Bit 19:0	RPx	0x00000	rw	Rising edge event configuration bit on line x These bits are used to select rising edge to trigger an interrupt and event on line x. 0: Rising trigger on line x is disabled 1: Rising trigger on line x is enable Note: Bit 19 is reserved for future use.

Polarity configuration register2 (EXINT_ POLCFG2)

Bit	Register	Reset value	Type	Description
Bit 31:20	Reserved	0x000	resd	Forced to 0 by hardware.
Bit 19:0	FPx	0x00000	rw	Falling edge event configuration bit on line x These bits are used to select falling edge to trigger an interrupt and event on line x. 0: Falling trigger on line x is disabled 1: Falling trigger on line x is enabled Note: Bit 19 is reserved for future use.

Software trigger register (EXINT_ SWTRG)

Bit	Register	Reset value	Type	Description
Bit 31:20	Reserved	0x000	resd	Forced to 0 by hardware.
Bit 19:0	SWTx	0x00000	rw	Software triggle on line x If the corresponding bit in EXINT_INTEN register is 1, the software writes to this bit. The hardware sets the corresponding bit in the EXINT_INTSTS automatically to generate an interrupt. If the corresponding bit in the EXINT_EVTEN register is 1, the software writes to this bit. The hardward generates an event on the corresponding interrupt line automatically. 0: Default value 1: Sotware trigger generated Note: This bit is cleared by writing 1 to the corresponding bit in the EXINT_INTSTS register. Bit 19 is reserved for future use.

Interrupt status register (EXINT_ INTSTS)

Bit	Register	Reset value	Type	Description
Bit 31:20	Reserved	0x000	resd	Forced to 0 by hardware.
Bit 19:0	LINEx	0x00000	rw	Line x status bit 0: No interrupt occurred 1: Interrupt occurred Note: This bit is cleared by writing 1. Bit 19 is reserved for future use.

9 DMA controller (DMA)

Introduction

Direct memory access (DMA) controller is designed for 32-bit MCU applications with the aim of enhancing system performance and reducing the generation of interrupts.

There are two DMA controllers in the microcontroller. Each controller contains 7 DMA channels. Each channel manages memory access requests from one or more peripherals. An arbiter is available for coordinating the priority of each DMA request.

Main features

- AMBA compliant (Rev. 2.0)
- Only support AHB OKAY and ERROR responses
- HBUSREQ and HGRANT of AHB master interface are not supported
- Support 7 channels
- Peripheral-to-memory, memory-to-peripheral, and memory-to-memory transfers
- Support hardware handshake
- Support 8-bit, 16-bit and 32- bit data transfers
- Programmable amount of data to be transferered: up to 65535
- Support flexible mapping

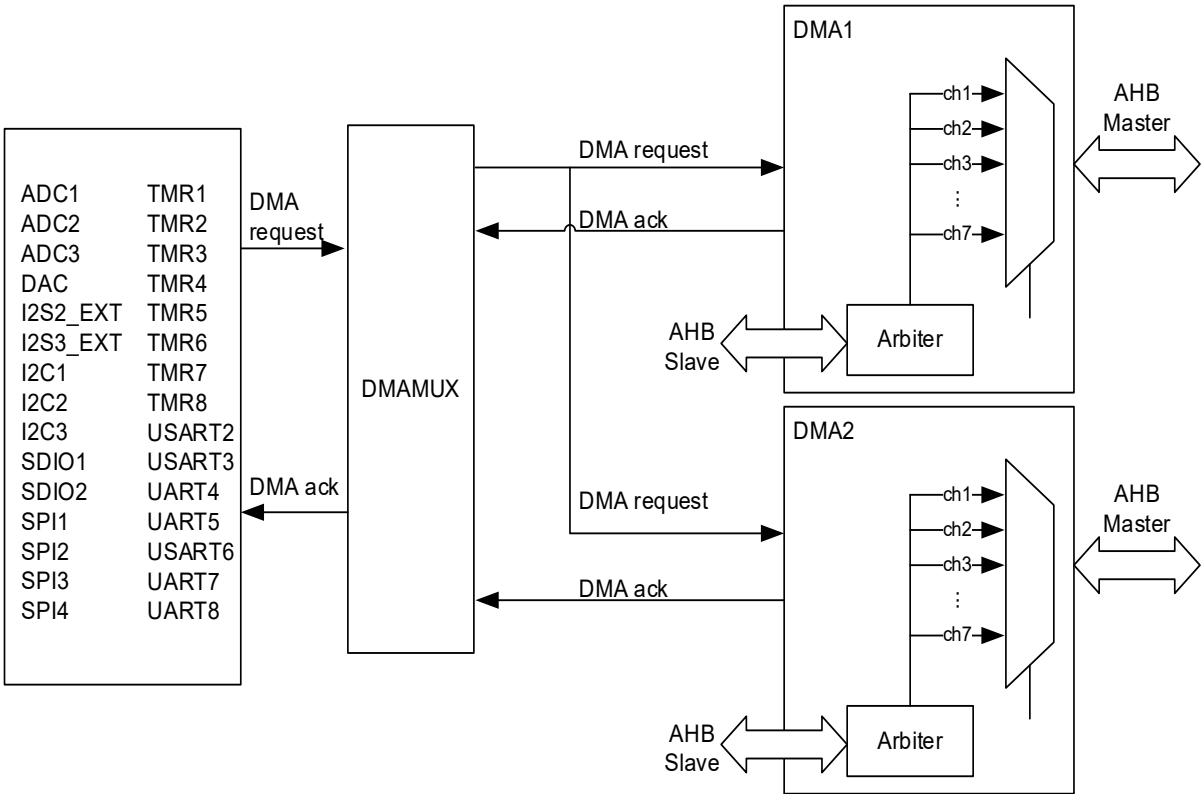


Figure 23. DMA block diagram

Note: The number of DMA peripherals in the above diagram may decrease depending on different models.

Functional descriptions

DMA configuration

1. Set the peripheral address in the DMA_CxPADD register

   The initial peripheral address for data transfer remains unchanged during transmission.

2. Set the memory address in the DMA_CxMADDR register

   The initial memory address for data transfer remains unchanged during transmission.

3. Configure the amount of data to be transferred in the DMA_CxDTCNT register

   Programmable data transfer size is up to 65535. This value is decremented after each data transfer.

4. Configure the channel setting in the DMA_CxCTRL register

   Including channel priority, data transfer direction/width, address incremented mode, circular mode and interrupt mode.

   - Channel priority (CHPL)
     There are four levels, including very high priority, high priority, medium priority and low priority.

If the two channels have the same priority level, then the channel with lower number will get priority over the one with higher number. For example, channel 1 has priority over channel 2.

- Data transfer direction (DTD)
  Memory-to-peripheral (M2P), peripheral-to-memory (P2M)
- Address incremented mode (PINCM/MINCM)
  In incremented mode, the subsequent transfer address is the previous address plus transfer width (PWIDTH/MWIDTH).
- Circular mode (LM)
  In circular mode, the contents in the DMA_CxDTCNT register is automatically reloaded with the initially programmed value after the completion of the last transfer.
- Memory-to-memory mode (M2M)
  This mode indicates that DMA channels perform data transfer without requests from peripherals. Circular mode and memory-to-memory mode cannot be used at the same time.

5. Enable DMA transfer by setting the CHEN bit in the DMA_CxCTRL register

Handshake mechanism

In P2M and M2P mode, the peripherals need to send a request signal to the DMA controller. The DMA channel will send the peripheral transfer request (single) until the signal is acknowledged. After the completion of peripheral transmission, the DMA controller sends the acknowledge signal to the peripheral. The peripheral then releases its request as soon as it receives the acknowledge signal. At the same time, the DMA controller releases the acknowledge signal as well.

Arbiter

When several channels are enabled simultaneously, the arbiter will restart arbitration after full data transfer by the master controller. The channel with very high priority waits until the channel of the master controller has completed data transfers before taking control of it. The master controller will re-arbitrate to serve other channels as long as the channel completes a single transfer based on the master controller priority.

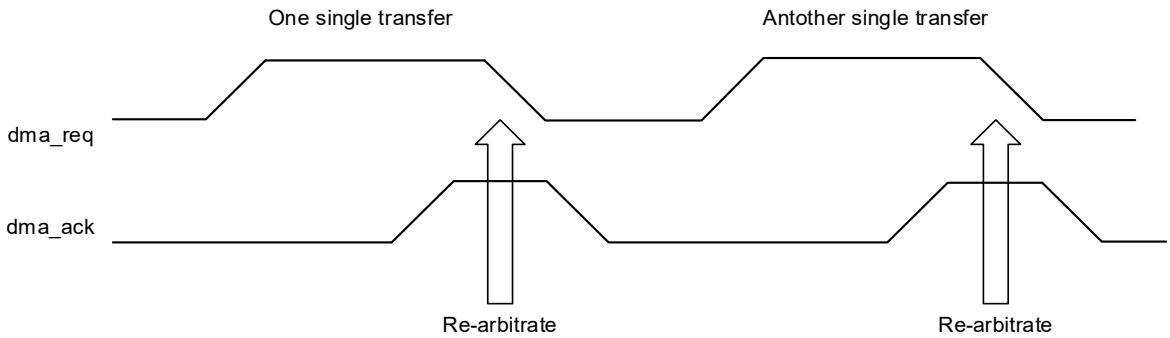


Figure 24. Re-arbitrae after request/acknowledge

Programmable data transfer width

Transfer width of the source data and destination data is programmable through the PWIDTH and MWIDTH bits in the DMA_CxCTRL register. When PWIDTH is not equal to MWIDTH, it can be aligned according to the settings of PWIDTH/ MWIDTH.

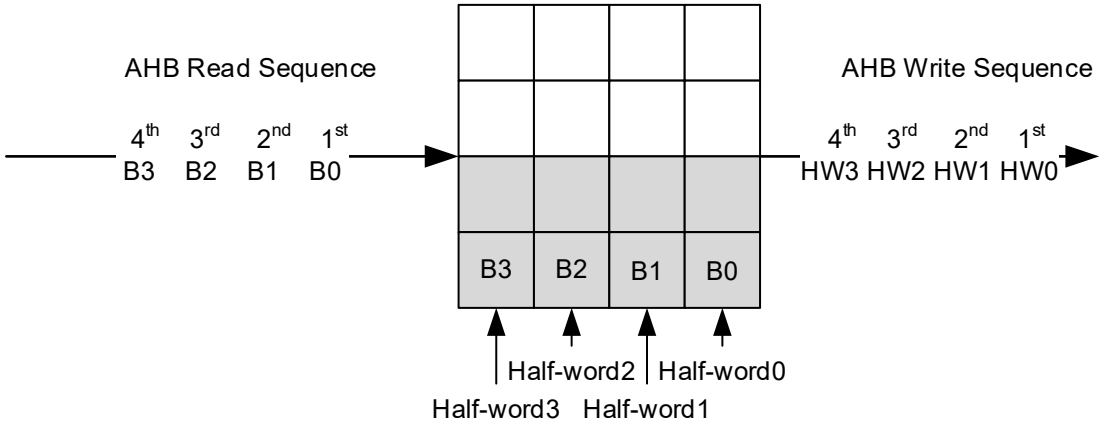


Figure 25. PWIDTH: byte, MWIDTH: half-word

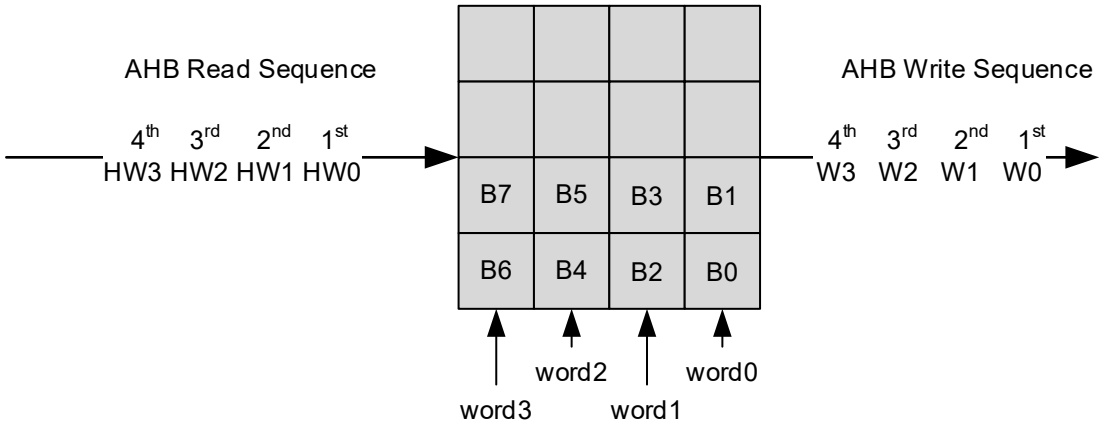


Figure 26. PWIDTH: half-word, MWIDTH: word

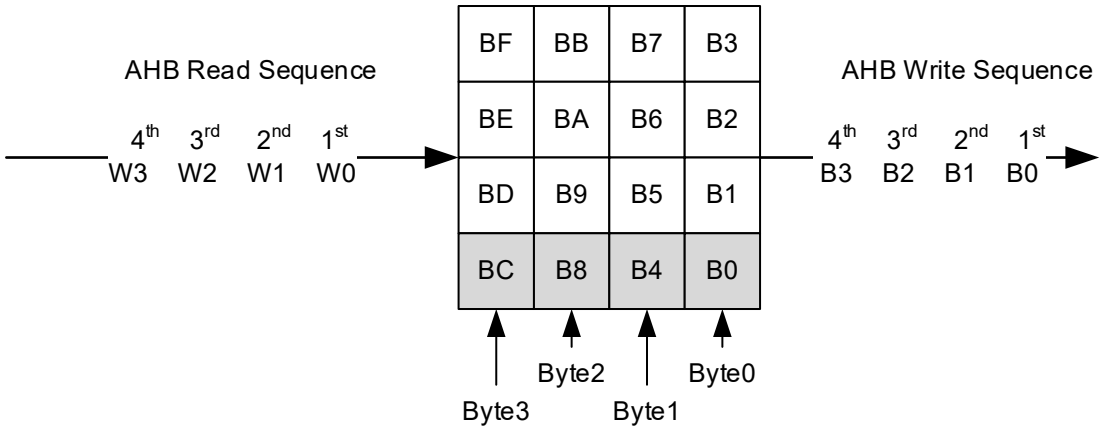


Figure 27. PWIDTH: word, MWIDTH: byte

Errors

Table 29. DMA error event

Error event	
Transfer error	AHB response error occurred during DMA read/write access

Interrupts

An interrupt can be generated on a DMA half-transfer, transfer complete and transfer error. Each channel has its specific interrupt flag, clear and enable bits, as shown in the table below.

Table 30. DMA interrupt requests

Interrupt event	Event flag bit	Clear control bit	Enable control bit
Half transfer	HDTF	HDTFC	HDTIEN
Transfer completed	FDTF	FDTFC	FDTIEN
Transfer error	DTERRF	DTERRFC	DTERRIEN

Note: DMA 2 channel 4/channel 5, channel 6/channel 7 interrupts are mapped onto the same interrupt vector.

Fixed DMA request mapping

Several peripheral requests are mapped to the same DMA channel through logic ORed. This means that only one request can be enabled at a time.

The peripheral DMA requests can be independently activated/de-activated by setting the control bits in the corresponding peripheral registers.

Table 31. DMA1 requests for each channel

Peripherals	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5	Channel 6	Channel 7
ADC1	ADC1						
SPI/I²S		SPI1/I²S1_RX	SPI1/I²S1_TX	SPI2/I²S2_RX	SPI2/I²S2_TX		
USART		USART3_TX	USART3_RX	USART1_TX	USART1_RX	USART2_RX	USART2_TX
I²C		I²C3_TX	I²C3_RX	I²C2_TX	I²C2_RX	I²C1_TX	I²C1_RX
TMR1		TMR1_CH1	TMR1_CH2	TMR1_CH4 TMR1_TRIG TMR1_HALL	TMR1_UP	TMR1_CH3	
TMR2	TMR2_CH3	TMR2_UP			TMR2_CH1		TMR2_CH2 TMR2_CH4
TMR3		TMR3_CH3	TMR3_CH4 TMR3_UP			TMR3_CH1 TMR3_TRIG	
TMR4	TMR4_CH1			TMR4_CH2	TMR4_CH3		TMR4_UP

Table 32. DMA2 requests for each channel

Peripherals	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5	Channel 6	Channel 7
ADC3					ADC3		
SPI/I²S	SPI3/I²S3_RX	SPI3/I²S3_TX	SPI4/I²S4_RX	SPI4/I²S4_TX			
UART4			UART4_RX		UART4_TX		
SDIO				SDIO			
SDIO2					SDIO2		
TMR5	TMR5_CH4 TMR5_TRIG	TMR5_CH3 TMR5_UP		TMR5_CH2	TMR5_CH1		
TMR6/ DAC CH1			TMR6_UP/ DAC CH1				

Peripherals	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5	Channel 6	Channel 7
TMR7/ DAC CH2				TMR7_UP/ DAC CH2			
TMR8	TMR8_CH3 TMR8_UP	TMR8_CH4 TMR8_TRIG TMR8_HALL	TMR8_CH1		TMR8_CH2		

Flexible DMA request mapping

In flexible request mapping mode (DMA_FLEX_EN = 1), the request source of each channel is programmed by CHx_SRC[x = 1 ~ 7]. For example, to specify DMA channel 1 as USART3_TX, and channel 3 as USART3_RX, and others are unused, then it is necessary that DMA_FLEX_EN = 1, CH1_SRC = 30, CH3_SRC = 29 and CH[2/4/5/6/7]_SRC = 0.

The table below shows CHx_SRC values and their corresponding request sources.

Table 33. Flexible DMA requests for each channel

CHx_ SRC	Request source	CHx_ SRC	DMA source	CHx_ SRC	Request source	CHx_ SRC	Request source
0	No select	1	ADC1	2	reserved	3	ADC3
4	reserved	5	DAC1	6	DAC2	7	reserved
8	reserved	9	SPI1_RX	10	SPI1_TX	11	SPI2_RX
12	SPI2_TX	13	SPI3_RX	14	SPI3_TX	15	SPI4_RX
16	SPI4_TX	17	I²S2EXT_RX	18	I²S2EXT_TX	19	I²S3EXT_RX
20	I²S3EXT_TX	21	reserved	22	reserved	23	reserved
24	reserved	25	USART1_RX	26	USART1_TX	27	USART2_RX
28	USART2_TX	29	USART3_RX	30	USART3_TX	31	UART4_RX
32	UART4_TX	33	UART5_RX	34	UART5_TX	35	USART6_RX
36	USART6_TX	37	UART7_RX	38	UART7_TX	39	UART8_RX
40	UART8_TX	41	I²C1_RX	42	I²C1_TX	43	I²C2_RX
44	I²C2_TX	45	I²C3_RX	46	I²C3_TX	47	reserved
48	reserved	49	SDIO1	50	SDIO2	51	reserved
52	reserved	53	TMR1_TRIG	54	TMR1_HALL	55	TMR1_UP
56	TMR1_CH1	57	TMR1_CH2	58	TMR1_CH3	59	TMR1_CH4
60	reserved	61	TMR2_TRIG	62	reserved	63	TMR2_UP
64	TMR2_CH1	65	TMR2_CH2	66	TMR2_CH3	67	TMR2_CH4
68	reserved	69	TMR3_TRIG	70	reserved	71	TMR3_UP
72	TMR3_CH1	73	TMR3_CH2	74	TMR3_CH3	75	TMR3_CH4
76	reserved	77	TMR4_TRIG	78	reserved	79	TMR4_UP
80	TMR4_CH1	81	TMR4_CH2	82	TMR4_CH3	83	TMR4_CH4
84	reserved	85	TMR5_TRIG	86	reserved	87	TMR5_UP
88	TMR5_CH1	89	TMR5_CH2	90	TMR5_CH3	91	TMR5_CH4
92	reserved	93	reserved	94	reserved	95	TMR6_UP
96	reserved	97	reserved	98	reserved	99	reserved
100	reserved	101	reserved	102	reserved	103	TMR7_UP
104	reserved	105	reserved	106	reserved	107	reserved
108	reserved	109	TMR8_TRIG	110	TMR8_HALL	111	TMR8_UP
112	TMR8_CH1	113	TMR8_CH2	114	TMR8_CH3	115	TMR8_CH4
116	reserved	117	reserved	118	reserved	119	reserved

CHx_ SRC	Request source	CHx_ SRC	DMA source	CHx_ SRC	Request source	CHx_ SRC	Request source
120	reserved	121	reserved	122	reserved	123	reserved
124	reserved	125	reserved	126	reserved	127	reserved
128	reserved	129	reserved	130	reserved	131	reserved
132	reserved	133	reserved	134	reserved	135	reserved
136	reserved	137	reserved	138	reserved	139	reserved
140	reserved	141	reserved	142	reserved	143	reserved
144	reserved	145	reserved	146	reserved	147	reserved
148	reserved	149	reserved	150	reserved	151	reserved
152	reserved	153	reserved	154	reserved	155	reserved
156	reserved	157	reserved	158	reserved	159	reserved
160	reserved	161	reserved	162	reserved	163	reserved
164	reserved	165	reserved	166	reserved	167	reserved
168	reserved	169	reserved	170	reserved	171	reserved
172	reserved	173	reserved	174	reserved	175	reserved

Note: The request mapping mode must be identical for DMA1 and DMA2 (DMA_FLEX_EN must be set either 1 or 0 for both DMA 1 and DMA 2)

DMA registers

The following table shows DMA register map and their reset values.

Table 34. DMA register map and reset value

Register	Offset	Reset value
DMA_STS	0x00	0x0000 0000
DMA_CLR	0x04	0x0000 0000
DMA_C1CCTRL	0x08	0x0000 0000
DMA_C1DTCNT	0x0C	0x0000 0000
DMA_C1PADDR	0x10	0x0000 0000
DMA_C1MADDR	0x14	0x0000 0000
DMA_C2CTRL	0x1C	0x0000 0000
DMA_C2DTCNT	0x20	0x0000 0000
DMA_C2PADDR	0x24	0x0000 0000
DMA_C2MADDR	0x28	0x0000 0000
DMA_C3CTRL	0x30	0x0000 0000
DMA_C3DTCNT	0x34	0x0000 0000
DMA_C3PADDR	0x38	0x0000 0000
DMA_C3MADDR	0x3C	0x0000 0000
DMA_C4CTRL	0x44	0x0000 0000
DMA_C4DTCNT	0x48	0x0000 0000
DMA_C4PADDR	0x4C	0x0000 0000
DMA_C4MADDR	0x50	0x0000 0000
DMA_C5CTRL	0x58	0x0000 0000
DMA_C5DTCNT	0x5C	0x0000 0000
DMA_C5PADDR	0x60	0x0000 0000
DMA_C5MADDR	0x64	0x0000 0000

Register	Offset	Reset value
DMA_C6CTRL	0x6C	0x0000 0000
DMA_C6DTCNT	0x70	0x0000 0000
DMA_C6PADDR	0x74	0x0000 0000
DMA_C6MADDR	0x78	0x0000 0000
DMA_C7CTRL	0x80	0x0000 0000
DMA_C7DTCNT	0x84	0x0000 0000
DMA_C7PADDR	0x88	0x0000 0000
DMA_C7MADDR	0x8C	0x0000 0000
DMA_SRC_SEL0	0xA0	0x0000 0000
DMA_SRC_SEL1	0xA4	0x0000 0000

Note: In the following registers, all bits related to channel 6 and channel 7 are not relevant for DMA 2 fixed request mapping since it has only 5 channels. They are applied to DMA 2 flexible request mapping, instead, supporting up to 7 channels.

DMA interrupt status register (DMA_STS)

Bit	Register	Reset value	Type	Description
Bit 31:28	Reserved	0x0	resd	Kept at its default value.
Bit 27	DTERRF7	0x0	ro	Channel 7 data transfer error event flag 0: No transfer error occurred 1: Transfer error occurred
Bit 26	HDTF7	0x0	ro	Channel half transfer event flag 0: No half-transfer event occurred 1: Half-transfer event occurred
Bit 25	FDTF7	0x0	ro	Channel 7 transfer complete event flag 0: No transfer complete event occurred 1: Transfer complete event occurred
Bit 24	GF7	0x0	ro	Channel global event flag 0: No transfer error, half transfer or transfer complete event occurred. 1: Transfer error, half transfer or transfer complete event occurred.
Bit 23	DTERRF6	0x0	ro	Channel 6 data transfer error event flag 0: No transfer error occurred 1: Transfer error occurred
Bit 22	HDTF6	0x0	ro	Channel 6 half transfer event flag 0: No half-transfer event occurred 1: Half-transfer event occurred
Bit 21	FDTF6	0x0	ro	Channel 6 transfer complete event flag 0: No transfer complete event occurred 1: Transfer complete event occurred
Bit 20	GF6	0x0	ro	Channel 6 global event flag 0: No transfer error, half transfer or transfer complete event occurred. 1: Transfer error, half transfer or transfer complete event
Bit 19	DTERRF5	0x0	ro	Channel 5 data transfer error event flag 0: No transfer error occurred 1: Transfer error occurred

Bit	Register	Reset value	Type	Description
Bit 18	HDTF5	0x0	ro	Channel 5 half transfer event flag 0: No half-transfer event occurred 1: Half-transfer event occurred
Bit 17	FDTF5	0x0	ro	Channel 5 transfer complete event flag 0: No transfer complete event occurred 1: Transfer complete event occurred
Bit 16	GF5	0x0	ro	Channel 5 global event flag 0: No transfer error, half transfer or transfer complete event occurred. 1: Transfer error, half transfer or transfer complete event
Bit 15	DTERRF4	0x0	ro	Channel 4 data transfer error event flag 0: No transfer error occurred 1: Transfer error occurred
Bit 14	HDTF4	0x0	ro	Channel 4 half transfer event flag 0: No half-transfer event occurred 1: Half-transfer event occurred
Bit 13	FDTF4	0x0	ro	Channel 4 transfer complete event flag 0: No transfer complete event occurred 1: Transfer complete event occurred
Bit 12	GF4	0x0	ro	Channel 4 global event flag 0: No transfer error, half transfer or transfer complete event occurred. 1: Transfer error, half transfer or transfer complete event
Bit 11	DTERRF3	0x0	ro	Channel 3 data transfer error event flag 0: No transfer error occurred 1: Transfer error occurred
Bit 10	HDTF3	0x0	ro	Channel 3 half transfer event flag 0: No half-transfer event occurred 1: Half-transfer event occurred
Bit 9	FDTF3	0x0	ro	Channel 3 transfer complete event flag 0: No transfer complete event occurred 1: Transfer complete event occurred
Bit 8	GF3	0x0	ro	Channel 3 global event flag 0: No transfer error, half transfer or transfer complete event occurred. 1: Transfer error, half transfer or transfer complete event
Bit 7	DTERRF2	0x0	ro	Channel 2 data transfer error event flag 0: No transfer error occurred 1: Transfer error occurred
Bit 6	HDTF2	0x0	ro	Channel 2 half transfer event flag 0: No half-transfer event occurred 1: Half-transfer event occurred
Bit 5	FDTF2	0x0	ro	Channel 2 transfer complete event flag 0: No transfer complete event occurred 1: Transfer complete event occurred
Bit 4	GF2	0x0	ro	Channel 2 global event flag 0: No transfer error, half transfer or transfer complete event occurred. 1: Transfer error, half transfer or transfer complete event

Bit	Register	Reset value	Type	Description
Bit 3	DTERRF1	0x0	ro	Channel 1 data transfer error event flag 0: No transfer error occurred 1: Transfer error occurred
Bit 2	HDTF1	0x0	ro	Channel 1 half transfer event flag 0: No half-transfer event occurred 1: Half-transfer event occurred
Bit 1	FDTF1	0x0	ro	Channel 1 transfer complete event flag 0: No transfer complete event occurred 1: Transfer complete event occurred
Bit 0	GF1	0x0	ro	Channel 1 global event flag 0: No transfer error, half transfer or transfer complete event occurred. 1: Transfer error, half transfer or transfer complete event

DMA interrupt flag clear register (DMA_CLR)

Bit	Register	Reset value	Type	Description
Bit 31:28	Reserved	0x0	resd	Kept at its default value.
Bit 27	DTERRFC7	0x0	rw1c	Channel 7 data transfer error flag clear 0: No effect 1: Clear the DTERRF flag in the DMA_STS register
Bit 26	HDTFC7	0x0	rw1c	Channel 7 half transfer flag clear 0: No effect 1: Clear the HDTF7 flag in the DMA_STS register
Bit 25	FDTFC7	0x0	rw1c	Channel 7 transfer complete flag clear 0: No effect 1: Clear the FDTF7 flag in the DMA_STS register
Bit 24	GFC7	0x0	rw1c	Channel 7 global interrupt flag clear 0: No effect 1: Clear the DTERRF7, HDTF7, FDTF7 and GF7 flag in the DMA_STS register
Bit 23	DTERRFC6	0x0	rw1c	Channel 6 data transfer error flag clear 0: No effect 1: Clear the DTERRF6 flag in the DMA_STS register
Bit 22	HDTFC6	0x0	rw1c	Channel 6 half transfer flag clear 0: No effect 1: Clear the HDTF6 flag in the DMA_STS register
Bit 21	FDTFC6	0x0	rw1c	Channel 6 transfer complete flag clear 0: No effect 1: Clear the FDTF6 flag in the DMA_STS register
Bit 20	GFC6	0x0	rw1c	Channel 6 global interrupt flag clear 0: No effect 1: Clear the DTERRF6, HDTF6, FDTF6 and GF6 flag in the DMA_STS register
Bit 19	DTERRFC5	0x0	rw1c	Channel 5 data transfer error flag clear 0: No effect 1: Clear the DTERRF5 flag in the DMA_STS register
Bit 18	HDTFC5	0x0	rw1c	Channel 5 half transfer flag clear 0: No effect 1: Clear the HDTF5 flag in the DMA_STS register

Bit	Register	Reset value	Type	Description
Bit 17	FDTFC5	0x0	rw1c	Channel 5 transfer complete flag clear 0: No effect 1: Clear the FDTF5 flag in the DMA_STS register
Bit 16	GFC5	0x0	rw1c	Channel 5 global interrupt flag clear 0: No effect 1: Clear the DTERRF5, HDTF5, FDTF5 and GF5 in the DMA_STS register
Bit 15	DTERRFC4	0x0	rw1c	Channel 4 data transfer error flag clear 0: No effect 1: Clear the DTERRF4 flag in the DMA_STS register
Bit 14	HDTFC4	0x0	rw1c	Channel 4 half transfer flag lear 0: No effect 1: Clear the HDTF4 flag in the DMA_STS register
Bit 13	FDTFC4	0x0	rw1c	Channel 4 transfer complete flag clear 0: No effect 1: Clear the FDTF4 flag in the DMA_STS register
Bit 12	GFC4	0x0	rw1c	Channel 4 global interrupt flag clear 0: No effect 1: Clear the DTERRF4, HDTF4, FDTF4 and GF4 flag in the DMA_STS register
Bit 11	DTERRFC3	0x0	rw1c	Channel 7 data transfer error flag clear 0: No effect 1: Clear the DTERRF7 flag in the DMA_STS register
Bit 10	HDTFC3	0x0	rw1c	Channel 7 half transfer flag clear 0: No effect 1: Clear the HDTF7 flag in the DMA_STS register
Bit 9	FDTFC3	0x0	rw1c	Channel 3 transfer complete flag clear 0: No effect 1: Clear the FDTF3 flag in the DMA_STS register
Bit 8	GFC3	0x0	rw1c	Channel 3 global interrupt flag clear 0: No effect 1: Clear the DTERRF3, HDTF3, FDTF3 and GF3 flag in the DMA_STS register
Bit 7	DTERRFC2	0x0	rw1c	Channel 2 data transfer error flag clear 0: No effect 1: Clear the DTERRF2 flag in the DMA_STS register
Bit 6	HDTFC2	0x0	rw1c	Channel 2 half transfer flag clear 0: No effect 1: Clear the HDTF2 flag in the DMA_STS register
Bit 5	FDTFC2	0x0	rw1c	Channel 2 transfer complete flag clear 0: No effect 1: Clear the FDTF2 flag in the DMA_STS register
Bit 4	GFC2	0x0	rw1c	Channel 2 global interrupt flag clear 0: No effect 1: Clear the DTERRF2, HDTF2, FDTF2 and GF2 in the DMA_STS register
Bit 3	DTERRFC1	0x0	rw1c	Channel 1 data transfer error flag clear 0: No effect 1: Clear the DTERRF1 flag in the DMA_STS register

Bit	Register	Reset value	Type	Description
Bit 2	HDTFC1	0x0	rw1c	Channel 1 half transfer flag clear 0: No effect 1: Clear the HDTF1 flag in the DMA_STS register
Bit 1	FDTFC1	0x0	rw1c	Channel 1 transfer complete flag clear 0: No effect 1: Clear the FDTF1 flag in the DMA_STS register
Bit 0	GFC1	0x0	rw1c	Channel 1 global interrupt flag clear 0: No effect 1: Clear the DTERRF1, HDTF1, FDTF1 and GF1 in the DMA_STS register

DMA channelx configuration register (DMA_CxCTRL) (x = 1…7)

Bit	Register	Reset value	Type	Description
Bit 31:15	Reserved	0x00000	resd	Kept at its default value.
Bit 14	M2M	0x0	rw	Memory to memory mode 0: Disabled 1: Enabled
Bit 13:12	CHPL	0x0	rw	Channel priority level 00: Low 01: Medium 10: High 11: Very high
Bit 11:10	MWIDTH	0x0	rw	Memory data bit width 00: 8 bits 01: 16 bits 10: 32 bits 11: Reserved
Bit 9:8	PWIDTH	0x0	rw	Peripheral data bit width 00: 8 bits 01: 16 bits 10: 32 bits 11: Reserved
Bit 7	MINCM	0x0	rw	Memory address increment mode 0: Disabled 1: Enabled
Bit 6	PINCM	0x0	rw	Peripheral address increment mode 0: Disabled 1: Enabled.
Bit 5	LM	0x0	rw	Circular mode 0: Disabled 1: Enabled
Bit 4	DTD	0x0	rw	Data transfer direction 0: Read from peripherals 1: Read from memory
Bit 3	DTERRIEN	0x0	rw	Data transfer error interrupt enable 0: Disabled 1: Enabled
Bit 2	HDTIEN	0x0	rw	Half-transfer interrupt enable 0: Disabled 1: Enabled

Bit	Register	Reset value	Type	Description
Bit 1	FDTIEN	0x0	rw	Transfer complete interrupt enable 0: Disabled 1: Enabled
Bit 0	CHEN	0x0	rw	Channel enable 0: Disabled 1: Enabled

DMA channelx number of data register (DMA_CxDTCNT) (x = 1…7)

Bit	Register	Reset value	Type	Description
Bit 31:16	Reserved	0x0000	resd	Kept at its default value.
Bit 15:0	CNT	0x0000	rw	Number of data to transfer The number of data to transfer is from 0x0 to 0xFFFF. This register can only written when the CHEN bit in the correspoinding channel is set 0. The value is decremented after each DMA transfer. Note: This register holds the number of data to transfer, instead of transfer size. The transfer size is calculated by data width.

DMA channelx peripheral address register (DMA_CxPADDR) (x = 1…7)

Bit	Register	Reset value	Type	Description
Bit 31:0	PADDR	0x0000 0000	rw	Peripheral base address Base address of peripheral data register is the source or destination of data transfer. Note: The register can only be written when the CHEN bit in the corresponding channel is set 0.

DMA channelx memory address register (DMA_CxMADDR) (x = 1…7)

Bit	Register	Reset value	Type	Description
Bit 31:0	MADDR	0x0000 0000	rw	Memory base address Memory address is the source or destination of data transfer. Note: The register can only be written when the CHEN bit in the corresponding channel is set 0.

Channel source register (DMA_SRC_SEL0)

Bit	Register	Reset value	Type	Description
Bit 31:24	CH4_SRC	0x00	rw	CH4 source select When DMA_FLEX_EN=1, CH4_SRC selects channel 4 source, please refer to “Flexible DMA request mapping”.
Bit 23:16	CH3_SRC	0x00	rw	CH3 source select When DMA_FLEX_EN=1, CH3_SRC selects channel 3 source, please refer to “Flexible DMA request mapping”.
Bit 15:8	CH2_SRC	0x00	rw	CH2 source select When DMA_FLEX_EN=1, CH2_SRC selects channel 2 source, please refer to “Flexible DMA request mapping”.
Bit 7:0	CH1_SRC	0x00	rw	CH1 source select When DMA_FLEX_EN=1, CH1_SRC selects channe 1 source, please refer to “Flexible DMA request mapping”.

Channel source register1 (DMA_SRC_SEL1)

Bit	Register	Reset value	Type	Description
Bit 31:25	Reserved	0x00	resd	Kept at its default value.
Bit 24	DMA_FLEX_EN	0x0	rw	DMA flexible request mapping enable 0: DMA fixed request mapping mode 1: DMA flexible request mapping mode
Bit 23:16	CH7_SRC	0x00	rw	CH7 source select When DMA_FLEX_EN=1, CH7_SRC selects channel 7 source, please refer to “Flexible DMA request mapping”.
Bit 15:8	CH6_SRC	0x00	rw	CH6 source select When DMA_FLEX_EN=1, CH6_SRC selects channel 6 source, please refer to “Flexible DMA request mapping”.
Bit 7:0	CH5_SRC	0x00	rw	CH5 source select When DMA_FLEX_EN=1, CH5_SRC selects channel source, please refer to “Flexible DMA request mapping”.

10 CRC calculation unit (CRC)

CRC introduction

The Cyclic Redundancy Check (CRC) is an independent peripheral with CRC check feature. It follows CRC32 standard.

The CRC_CTRL register is used to select input data reverse (word, REVOD=1) or output data reverse (byte, REVID=01; half-word, REVID=0; word: REVID=11). CRC calculation unit is also equipped with initialization function. After each CRC reset, the value in the CRC_IDT register is loaded with data register (CRC_DT) by CRC.

Users can write the data to go through CRC check and read the calculated result through CRC_DT register. Note that the calculation result is the combination of the previous result and the current value to be calculated.

Main features

- Use CRC-32 code
- 4 HCLK cycles for each CRC calculation
- Support input/output data format toggle
- Perform write/read operation through CRC_DT register
- Set an initialization value with the CRC_IDT register. The value is loaded with CRC_DT register after each CRC reset.

CRC registers

Table 35. CRC register map and reset value

Register	Offset	Reset value
CRC_DT	0x00	0xFFFF FFFF
CRC_CDT	0x04	0x0000 0000
CRC_CTRL	0x08	0x0000 0000
CRC_IDT	0x10	0xFFFF FFFF
CRC_POLY	0x14	0x04C1 1DB7

Data register (CRC_DT)

Bit	Register	Reset value	Type	Description
Bit 31:0	DT	0xFFFF FFFF	rw	Data register bits Used as input register when writing new data. Return CRC calculation results when it is read.

Common data register (CRC_CDT)

Bit	Register	Reset value	Type	Description
Bit 31:8	Reserved	0x000000	resd	Kept at its default value.
Bit 7	REVOD	0x0	resd	Reverse output data Set and cleared by software. This bit is used to control whether to reverse output data. 0: No effect 1: Word reverse
Bit 6:5	REVID	0x0	rw	Reverse input data Set and cleared by software. This bit is used to control how to reverse input data. 00: No effect 01: Byte reverse 10: Half-word reverse 11: Word reverse
Bit 4:1	Reserved	0x0	resd	Kept at its default value.
Bit 0	RST	0x0	rw	Reset CRC calculation unit Set by software. Cleared by hardware. To reset CRC calculation unit, the data register is set as 0xFFFF FFFF. 0: No effect 1: Reset

Control register (CRC_CTRL)

Bit	Register	Reset value	Type	Description
Bit 31:8	Reserved	0x000000	resd	Kept at its default value.
Bit 7	REVOD	0x0	resd	Reverse output data Set and cleared by software. This bit is used to control whether to reverse output data. 0: No effect 1: Word reverse
Bit 6:5	REVID	0x0	rw	Reverse input data Set and cleared by software. This bit is used to control how to reverse input data. 00: No effect 01: Byte reverse 10: Half-word reverse 11: Word reverse
Bit 4:1	Reserved	0x0	resd	Kept at its default value.
Bit 0	RST	0x0	rw	Reset CRC calculation unit Set by software. Cleared by hardware. To reset CRC calculation unit, the data register is set as 0xFFFF FFFF. 0: No effect 1: Reset

Initialization register (CRC_IDT)

Bit	Register	Reset value	Type	Description
Bit 31:0	IDT	0xFFFF FFFF	rw	Initialization data register When CRC reset is triggered by the RST bit in the CRC_CTRL register, the value in the initialization register is written into the CRC_DT register as an initial value.

Polynomial register (CRC_POLY)

Bit	Register	Reset value	Type	Description
Bit 31:0	POLY	0x04C1 1DB7	rw	Polynomial coefficient The generated polynomial is a divisior in CRC calculation. Using CRC32 mode, this polynomial coefficient is 0x4C11DB7. Users can also set the polynomial coefficient according to their needs.

11 I²C interface

I²C introduction

I²C (inter-integrated circuit) bus interface manages the communication between the microcontroller and serial I²C bus. It supports master and slave modes, with up to 400 kbit/s of communication speed.

Main features

- I²C bus
  - Master and slave modes
  - Multimaster capability
  - Stand speed (100 kHz) and fast speed (400 kHz)
  - 7-bit and 10-bit address modes
  - Broadcast call mode
  - Status flag
  - Error flag
  - Clock stretching capability
  - Communication event interrupts
  - Error interrupts
- Support DMA transfer
- Support partial SMBus2.protocol
  - PEC generation and verification
  - SMBus reminder function
  - ARP(address resolution protocol)
  - Timeout mechanism
- PMBus

Functional descriptions

I²C bus consists of a data line (SDA) and clock line (SCL). It can achieve a maximum of 100 kHz speed in standard mode, while up to 400kHz in fast mode. A frame of data transmission begins with a Start condition and ends with a Stop condition. The bus is kept in busy state after receiving the Start condition, and becomes idle as long as it receives the Stop condition.

Start condition: SDA switches from high to low when SCL is set high.

Stop condition: SDA switches from low to high when SCL is set high.

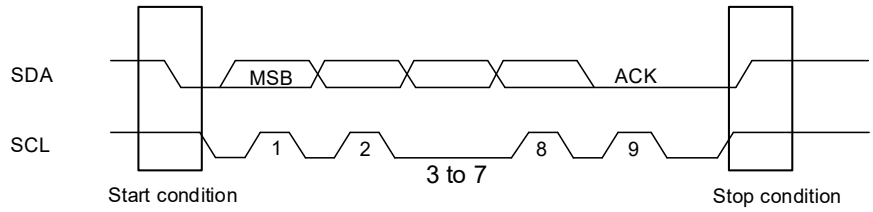


Figure 28. I²C bus protocol

I²C interface

The following diagram shows the block diagram of I²C function

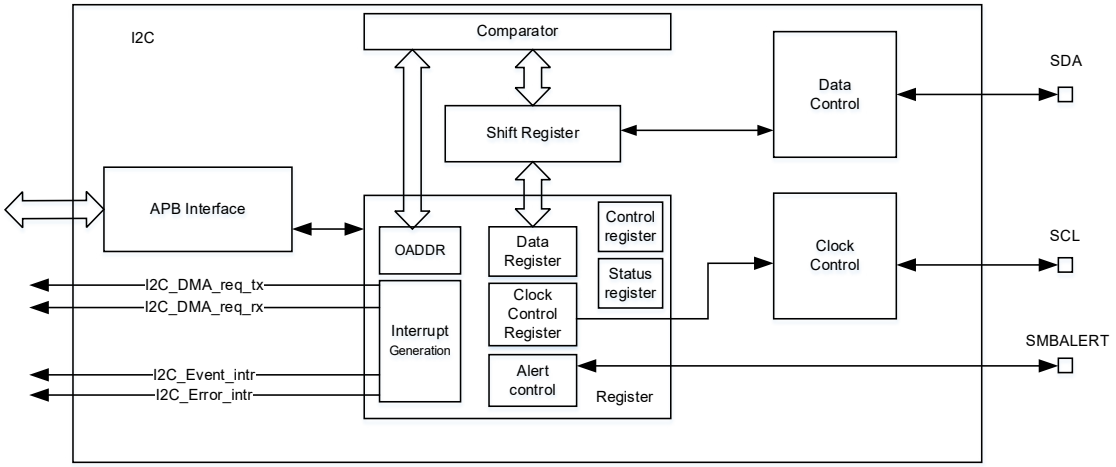


Figure 29. I²C function block diagram

I²C clock

I²C is clocked by either APB1 or APB2. The I²C clock division is achieved by setting the CLKFREQ[7:0] in the I2C_CTRL2 register. The minimum clock frequency varies from one mode to another, that is, at least 2 MHz in standard mode, but 4 MHz in fast mode.

Operation mode

I²C bus interface can operate both in master mode and slave mode. Switching from master mode to slave mode, vice versa, is supported as well. By default, the interface operates in slave mode. When GENSTART=1 is set (Start condition is activated), the I²C bus interface switches from slave mode to master mode, and returns to slave mode automatcially at the end of data transfer (Stop condition is triggered).

- Master transmitter
- Master receiver
- Slave transmitter
- Slave receiver

Communication process

- Master mode communication:
  - Start condition generation
  - Address transmission
  - Data Tx or Rx
  - Stop condition generation
  - End of communication
- Slave mode communication:
  - Wait until the address is matched.
  - Data Tx or Rx

- Wait for the generation of Stop condition
- End of communication

Address control

Both master and slave support 7-bit and 10-bit addressing modes.

Slave address mode:

- In 7-bit mode
  - ADDR2EN=0 stands for single address mode: only match OADDR1
  - DUALEN=1 stands for dual address mode: match OADDR1 and OADDR2
- In 10-bit mode
  - Only match OADDR1

Support special slave address:

- Broadcast call address (0b0000000x): This address is enabled when GCAEN=1.
- SMBus device default address (0b1100001x): This address is enabled for SMBus address resolution protocol in SMBus device mode.
- SMBus master default address (0b0001000x): This address is enabled for SMBus master notification protocol in SMBus master mode.
- SMBus alert address (0b0001100x): This address is enabled for SMBus alert response address protocol in SMBus master mode when SMBALERT = 1
  Refer to SMBus2.0 protocol for more information.

Slave address matching procedure:

- Receive Start condition
- Address matching
- The slave sends ACK if address is matched.
- ADDR7F is set 1, with DIRF indicating the transmission direction
  - When DIRF=0, slave enters receiver mode, starting receiving data.
  - When DIRF=1, slave enters transmitter mode, starting transmitting data

Clock stretching capability

Clock stretching is enabled by setting the STRETCH bit in the I2C_CTRL1 register. Once enabled, when the slave cannot process data in a timely manner on certain conditions, it will pull down SCL line to low level to stop communication in order to prevent data lost.

- Transmitter mode:
  - Clock stretching enable: If no data is written to the I2C_DT register before the next byte transimission (the first SCL rising edge of the next data), the I^2C interface will pull down SCL bus and wait until the data is written to the I2C_DT
  - Clock stretching disable: if no data is written to the I2C_DT register before the next byte transmission (the first SCL rising edge of next data), an underrun error will happen.
- Receiver mode
  - Clock stretching enable: When the shift register has received another byte before the data in the I2C_DT register is read, the I^2C will hold the SCL bus low to wait for the software to read I2C_DT register
  - Clock stretching disable: The data in the I2C_DT register is not yet read when the shift register receives another byte. In this case, if another data is received, an overrun error occurs.

I²C slave communication flow

Initialization

Enable I²C peripheral clock, and configure the clock-related bits in the I2C_CTRL2 register for a correct timing, and then wait for I²C master to send a Start condition.

Transmitter

The following diagram shows the transfer sequence of slave transmitter.

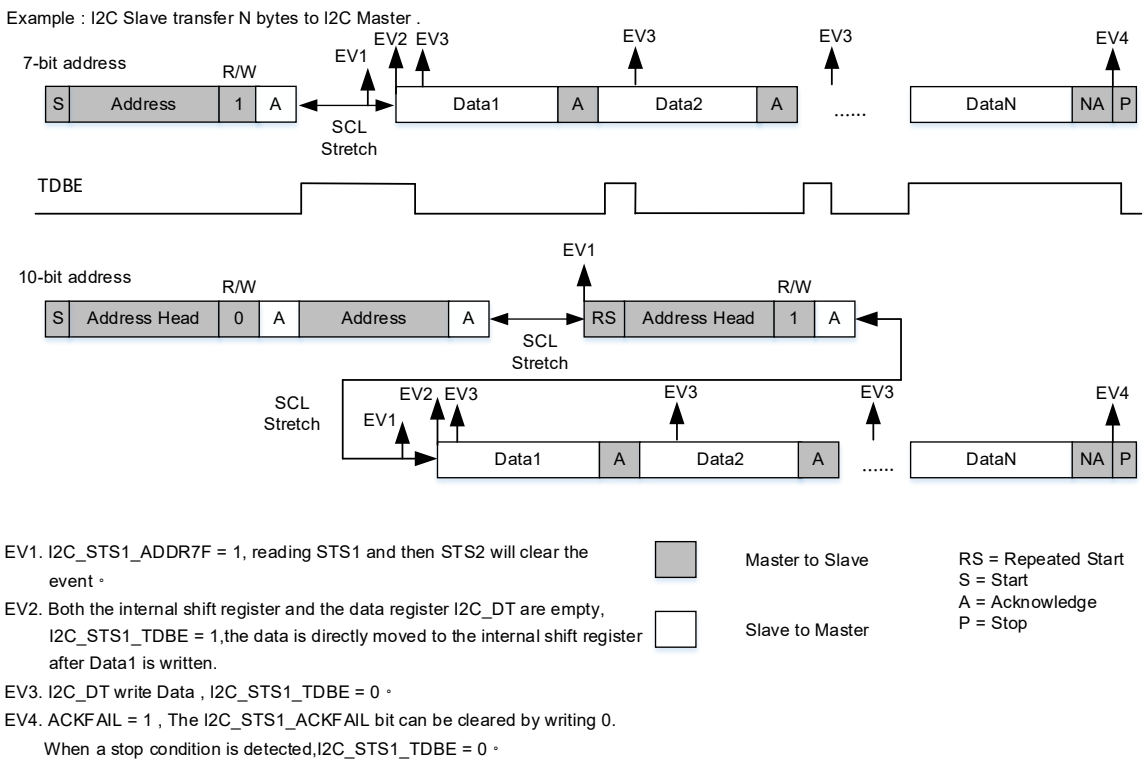


Figure 30. Transfer sequence of slave transmitter

7-bit address mode:

1. Wait for the master to send addresses
2. EV1: Address is matched (ADDR7F=1), and then the slave pulls the SCL bus low. Read STS1 and then STS2 by software will clear the ADDR7F bit. At this point, it enters transmission stage, in which both DT register and internal shift register are empty. The TDBE bit is set 1 by hardware.
3. EV2: When the data is written to the DT register, it is directly moved to the shift register and the SCL bus is released. The TDBE bit is still set 1 at this time.
4. EV3: At this point, the DT register is empty, but the shift register is not. Writing to the DT register will clear the TDBE bit.
5. EV4: After receiving the ACKFAIL event from the master, the ACKFIAL=1. Writing 0 to the ACKFIAL will clear the event.
6. End of communication.

10-bit address mode:

1. Wait for the master to send address
2. EV1: Address is matched (ADDR7F=1), and then the slave pulls the SCL bus low. Read STS1 and then STS2 by software will clear the ADDR7F bit. Wait for the master to re-send Start condition.
3. EV1: Address is matched (ADDR7F=1). Read STS1 and then STS2 will re-clear the ADDR7F bit. At this point, it enters transmission stage. Both DT register and shift register are empty. The TDBE is set 1 by hardware.
4. EV2: When the data is written to DT register, it is directly moved to the shift register, and SCL bus is released. The TDBE is still set 1 at this time.
5. EV3: At this point, the DT register is empty but the shift register is not. Writing to the DT register will clear the TDBE bit.
6. EV4: After receiving the ACKFAIL event from the master, ACKFIAL=1. Writing 0 to the ACKFIAL bit will clear the event.
7. End of communication.

Slave receiver

The following diagram shows the transfer sequence of slave receiver.

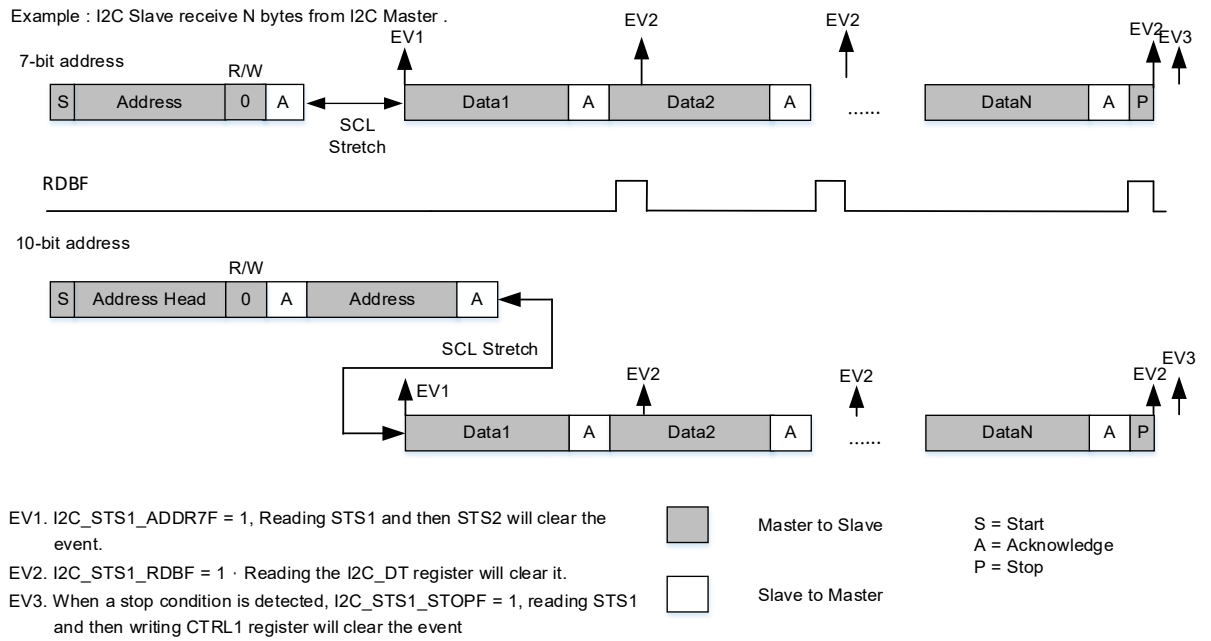


Figure 31. Transfer sequence of slave receiver

7-bit address mode:

1. Wait for the master to send address.
2. EV1: Address is matched (ADDR7F=1), and the slave pulls the SCL bus low. Read STS1 and then STS2 by software will clear the ADDR7F bit. At this point, the SCL bus is released, and enters receive stage.
3. The internal shift register receives the bus data and stores it to DT register.
4. EV2: After receiving the byte, the RDBF bit is set 1. Read the I2C_DT register will clear the RDBF bit.
5. EV3: After receiving the Stop condition from the master, STOPF=1. Read STS1 and then write to CTRL1 register will clear the event.
6. End of communication.

10-bit address mode:

1. Wait for the master to send address.
2. EV1: Address is matched (ADDR7F=1). The slave pulls the SCL bus low. Read STS1 and then STS2 by software will clear the ADDR7F bit. At this point, the SCL bus is released, and enters receive stage.
3. The internal shift register receives the bus data and stores it to DT register.
4. EV2: After receiving the byte, the RDBF bit is set 1. Read the I2C_DT register will clear the RDBF bit.
5. EV3: After receiving the Stop condition from the master, STOPF=1. Read STS1 and then write to CTRL1 register will clear the event.
6. End of communication.

I²C master communication flow

Initialization

1. Porgram input clock to generate correct timing through the CLKFREQ bit in the I2C_CTRL2 register;
2. Program I²C communication speed through the I2C_CLKCTRL bit in the clock control register;
3. Program the maximum rising time of bus through the I2C_TMRISE register;
4. Program the control register1 I2C_CTRL1;
5. Enable peripherals, if the GENSTART bit is set, a Start condition is generated on the bus, and the device enters master mode.

Slave address transmission

Slave address is divided into 7-bit and 10-bit modes. Whether it is transmitter mode or receiver mode depends on the lowest address bit.

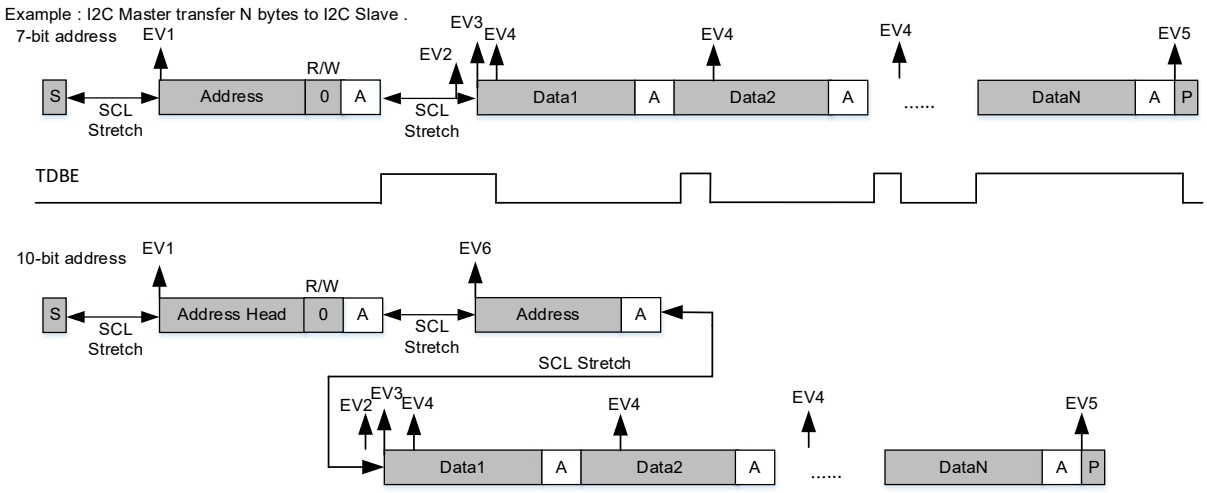
7-bit address mode:

- Transmitter: When the lowest bit of the address sent is 0, the master enters transmitter mode.
- Receiver: When the lowest bit of the address sent is 1, the master enters receiver mode.

10-bit address mode:

- Transmitter: First send address head 0b11110xx0 (where xx refers to address [9:8]) and then slave address [7:0], the master enters transmitter mode.
- Receiver: First send slave address head 0b11110xx0 (where xx refers to address [9:8]) and then address [7:0], followed by the address head 0b11110xx1 (where xx refers to address [9:8]), the master enters receiver mode.

Master transmitter



EV1. I2C_STS1_STARTF=1, reading STS1 and write the address to I2C_DT will clear the event.
EV2. I2C_STS1_ADDR7F= 1, reading STS1 and then STS2 will clear the event.
EV3. Both the internal shift register and the data register I2C_DT are empty, I2C_STS1_TDBE = 1, the data is directly moved to the internal shift register after Data1 is written.
EV4. I2C_DT writes Data, I2C_STS1_TDBE = 0。
EV5. I2C_STS1 register TDBE and TDC bit = 1, software sets the stop condition to clear the event
EV6. I2C_STS1_ADDRHF= 1,reading STS1 and write I2C_DT register will clear the event.

Master to Slave

Slave to Master

S = Start
A = Acknowledge
P = Stop

Figure 32. Transfer sequence of master transmitter

7-bit address mode:

1. Generate a Start condition (GENSTART=1)
2. EV1: Start condition is ready (STARTF=1). Read STS1 and write the address to DT register.
3. EV2: Address is matched successfully (ADDR7F=1). Read STS1 and then STS2 will clear the ADDR7F bit. In this case, the master enters transmit stage, and both DT register and internal shift regiser are empty. The TDBE bit is set 1 by hardware.
4. EV3: When the data is written to the DT register, it is directly moved to the shift register and the SCL bus is released. The TDBE bit is still set 1 at this time.
5. EV4: At this point, the DT register is empty but the shift register is full. Writing to the DT register will clear the TDBE bit.
6. The TDBE bit is set only after the second-to-last byte is sent.
7. EV5: TDC=1 indicates that the byte transmission is complete. The master sends Stop condition (STOPF=1). The TDBE bit and TDC bit is cleared by hardware.
8. End of communication.

10-bit address mode:

1. Generate Start condition (GENSTART=1)
2. EV1: Start condition is ready (STARTF=1). Read STS1 and write the address to DT register.
3. EV6: 10-bit address head sequence is sent. Read STS1 and write to DT register can clear the ADDRHF bit.
4. EV2: Address is matched successfully (ADDR7F=1). Read STS1 and then STS2 will clear the ADDR7F bit. In this case, the master enters transmit stage, and both DT register and internal shift regiser are empty. The TDBE bit is set 1 by hardware.
5. EV3: When the data is written to the DT register, it is directly moved to the shift register and the SCL bus is released. The TDBE bit is still set 1 at this time.
6. EV4: At this point, the DT register is empty but the shift register is full. Writing to the DT register will clear the TDBE bit.
7. The TDBE bit is set only after the second-to-last byte is sent.
8. EV5: TDC=1 indicates that the byte transmission is complete. The master sends Stop condition (STOPF=1). The TDBE bit and TDC bit is cleared by hardware.
9. End of communication.

Master receiver

Data reception depends on I²C interrupt priority:

1. Very high priority

- When the second-to-last byte is being read, clear the ACKEN bit and set the GENSTOP bit in the I2C_CTRL1 register to generate Stop condition.
- If only one byte is received, clear the ADDR7F flag and set the ACKEN and GENSTOP bit in the I2C_CTRL1 register.

- After the byte is received, the I2C_STS1_RDBF bit is set 1 by hardware, and it is cleared after the software read the I2C_DT register.

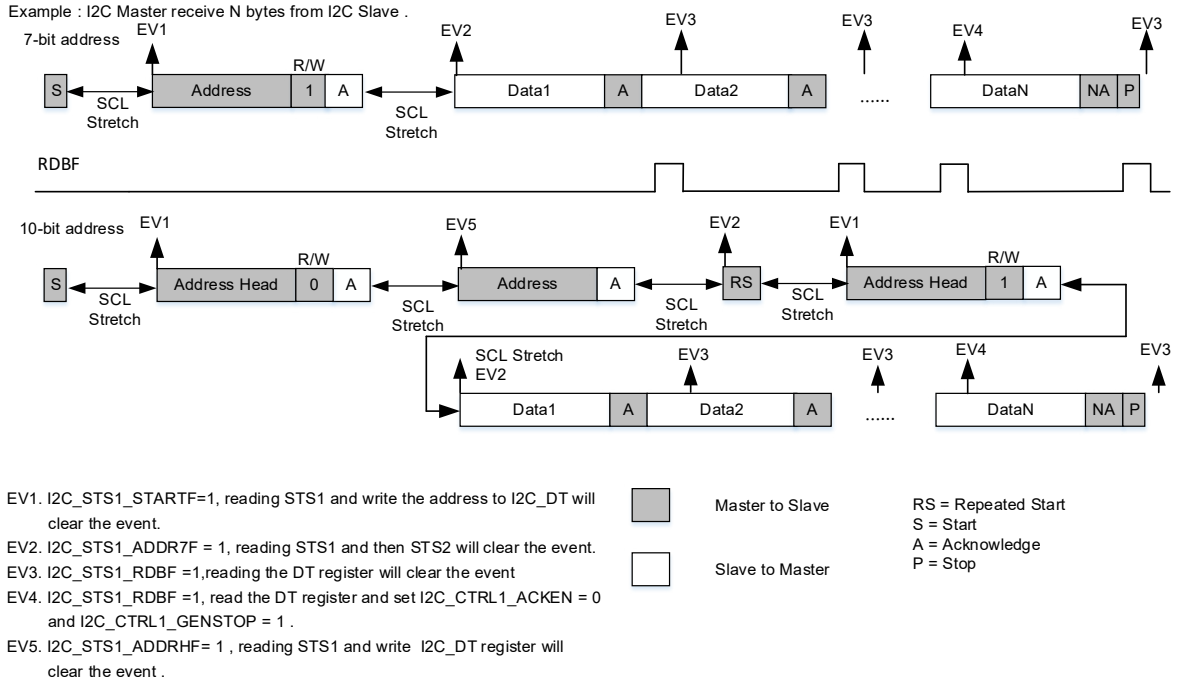


Figure 33. Transfer sequence of master receiver

7-bit address mode:

1. Generate a Start condition (GENSTART=1)
2. EV1: Start condition is ready (STARTF=1). Read STS1 and write the address to DT register.
3. EV2: Address is matched successfully (ADDR7F=1). Read STS1 and then STS2 will clear the ADDR7F bit. In this case, the master enters receive stage.
4. EV3: The RDBF bit is set 1 after the byte is received. It is cleared when the I2C_DT register is read.
5. EV4: The ACKEN bit is cleared and the GENSTOP is set as soon as the second-to-last byte is received.
6. EV3: The RDBF bit is set 1 after receiving the byte, and it is cleared when the I2C_DT register is read.
7. End of communication.

10-bit address mode:

1. Generate Start condition (GENSTART=1)
2. EV1: Start condition is ready (STARTF=1). Read STS1 and write the address to DT register.
3. EV5: 10-bit address head sequence is sent. Read STS1 and write to DT register can clear the ADDRHF bit.

4. EV2: Address is matched successfully (ADDR7F=1). Read STS1 and then STS2 will clear the ADDR7F bit, and the master re-send Start condition (GENSTART=1).
5. EV1: Start condition is ready (STARTF=1). Read STS1 and write the address to DT register.
6. EV2: Address is matched successfully (ADDR7F=1). Read STS1 and then STS2 will clear the ADDR7F bit. The master enters receive stage at this time.
7. EV3: The RDBF bit is set 1 after receiving the byte, and it is cleared when the I2C_DT register is read.
8. EV4: The ACKEN bit is cleared and the GENSTOP is set as soon as the second-to-last byte is received.
9. EV3: The RDBF bit is set 1 after receiving the byte, and it is cleared when the I2C_DT register is read.
10. End of communication.

2. When I²C interrupt priority is not very high but the number of bytes to receive is grearter than 2

- The third-to-last byte (N-2) is not read when being received. It is read only after the ACKEN bit in the I2C_CTRL1 register is cleared when the second-to-last byte (N-1) is received. Then the second-to-last byte (N-1) is read after the GENSTOP bit in the I2C_CTRL1 register is set. Afterwards, the bus starts to receive the last one byte.

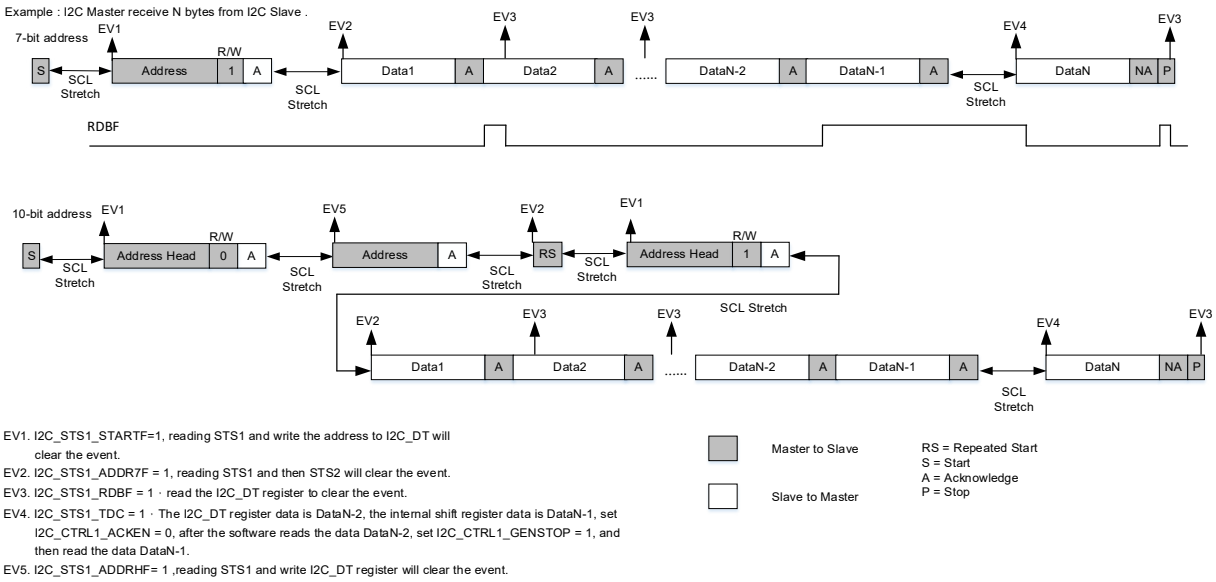


Figure 34. Transfer sequence of master receiver when N>2

7-bit address mode:

1. Generate a Start condition (GENSTART=1)
2. EV1: Start condition is ready (STARTF=1). Read STS1 and write the address to DT register.
3. EV2: Address is matched successfully (ADDR7F=1). Read STS1 and then STS2 will clear the ADDR7F bit, and the master enters receive stage.
4. EV3: The RDBF bit is set 1 after receiving the byte, and it is cleared when the I2C_DT register is read.
5. EV4: TDC=1, the contens in the I2C_DT is N-2, and that of the shift register is N-1. The ACKEN is set 0 by software and the data N-2 is read, afterwards, the GENSTOP=1, and data N-1 is read.
6. EV3: The RDBF bit is set 1 after receiving the byte, and it is cleared when the I2C_DT register is read.
7. End of communication.

10-bit address mode:

1. Generate Start condition (GENSTART=1)
2. EV1: Start condition is ready (STARTF=1). Read STS1 and write the address to DT register.
3. EV5: 10-bit address head sequence is sent. Read STS1 and write to DT register can clear the ADDRHF bit.
4. EV2: Address is matched successfully (ADDR7F=1). Read STS1 and then STS2 will clear the ADDR7F bit, and the master re-send Start condition.
5. EV1: Start condition is ready (STARTF=1). Read STS1 and write the address to the DT register.
6. EV2: Address is matched successfully (ADDR7F=1). Read STS1 and then STS2 will clear the ADDR7F bit, and the master enters receive stage.
7. EV3: The RDBF bit is set 1 after receiving the byte, and it is cleared when the I2C_DT register is read.
8. EV4: TDC=1, the contens in the I2C_DT is N-2, and that of the shift register is N-1. The ACKEN is set 0 by software and the data N-2 is read, afterwards, the GENSTOP=1, and data N-1 is read.
9. EV3: The RDBF bit is set 1 after receiving the byte, and it is cleared when the I2C_DT register is read.
10. End of communication.

3. When I²C interrupt priority is not very high but the number of bytes to receive is equal to 2

- Set the MACKCTRL bit in the I2C_CTRL1 register before data reception. When the address is matched, clear ACKEN bit and then the ADDR7F bit. When the TDC bit is set 1, set the GENSTOP bit in the I2C_CTRL1 register, and then read the DT register.

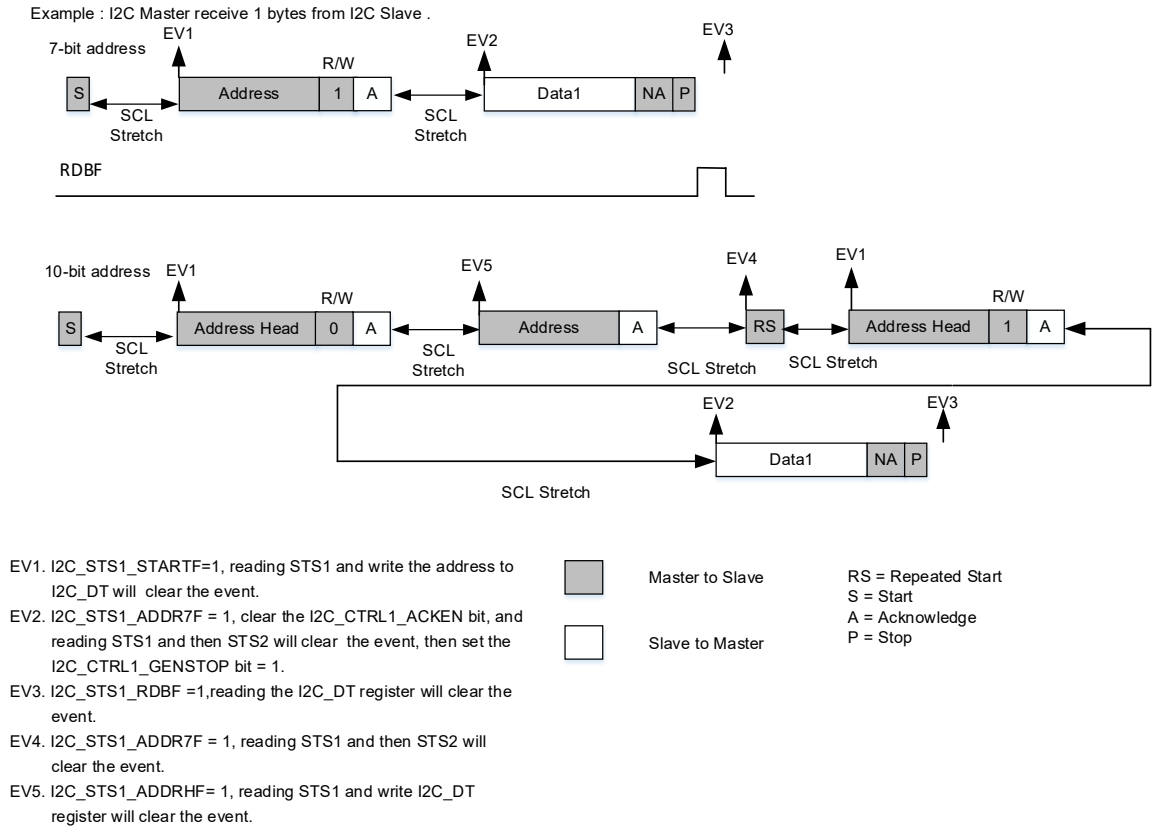


Figure 35. Transfer sequence of master receiver when N=2

7-bit address mode:

1. Set MACKCTRL=1 in the I2C_CTRL1 register
2. Generate Start condition (GENSTART=1)
3. EV1: Start condition is ready (STARTF=1). Read STS1 and write the address to DT register.
4. EV2: Address is matched successfully (ADDR7F=1). Clear the ACKEN bit and read STS1 before reading STS2 and clearing ADDR7F bit, the master enters receive state at this time.
5. EV2: TDC=1, set GENSTOP=1, and then read the I2C_DT register twice.
6. End of communication.

10-bit address mode:

1. Set MACKCTRL=1 in the I2C_CTRL1 register
2. Generate Start condition (GENSTART=1)

3. EV1: Start condition is ready (STARTF=1). Read STS1 and write the address to DT register.
4. EV4: 10-bit address head is sent. Read STS1 and write to DT register can clear the ADDRHF bit.
5. EV2: Address is matched successfully (ADDR7F=1). Read STS1 and then STS2 will clear the ADDR7F bit, and the master re-send Start condition.
6. EV1: Start condition is ready (STARTF=1). Read STS1 and write the address to the DT register.
7. EV2: Address is matched successfully (ADDR7F=1). Clear the ACKEN bit and read STS1 before reading STS2 and clearing ADDR7F bit, the master enters receive state at this time.
8. EV3: TDC=1, set GENSTOP=1, and then read the I2C_DT register twice.
9. End of communication.

4. When I²C interrupt priority is not very high but the number of bytes to receive is equal to 1

- After the address is matched, clear the ACKEN bit and then ADDR7F bit, then set the GENSTOP bit in the I2C_CTRL1 register. Wait until the RDBF bit is set 1 before reading the DT register.

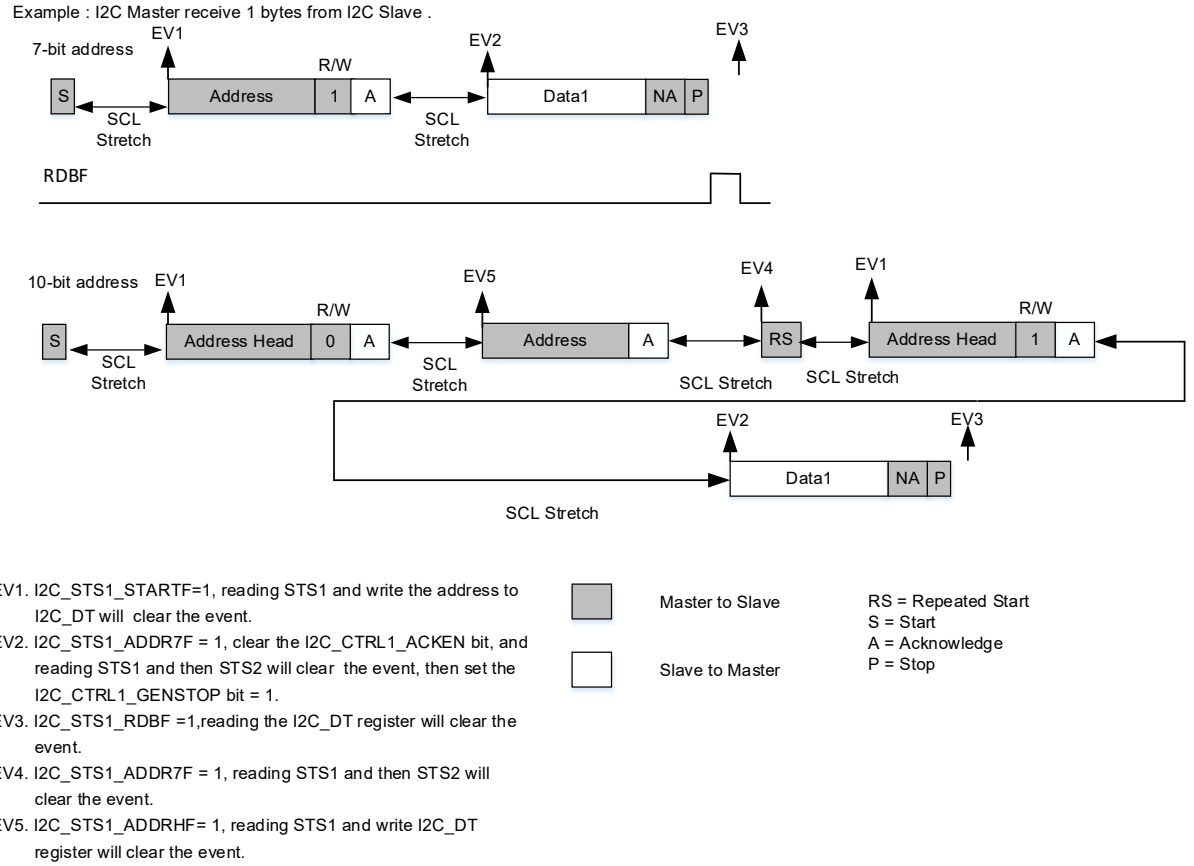


Figure 36. Transfer sequence of master receiver when N=1

7-bit address mode:

1. Generate a Start condition (GENSTART=1)
2. EV1: Start condition is ready (STARTF=1). Read STS1 and write the address to DT register.
3. EV2: Address is matched successfully (ADDR7F=1). Clear the ACKEN bit, read STS1 and then STS2 will clear the ADDR7F bit. Afterwards, set GENSTOP=1, the master enters receive stage at this time.
4. EV3: RDBF=1. It is cleared when the I2C_DT register is read.
5. End of communication.

10-bit address mode:

1. Generate a Start condition (GENSTART=1)
2. EV1: Start condition is ready (STARTF=1). Read STS1 and write the address to DT register.
3. EV5: 10-bit address head is sent. Read STS1 and write to DT register can clear the ADDRHF bit.
4. EV4: Address is matched successfully (ADDR7F=1). Read STS1 and STS2 will clear the ADDR7F bit, the master re-sends Start condition (GENSTART=1).
5. EV1: Start condition is ready (STARTF=1). Read STS1 and write the address to the DT register.
6. EV2: Address is matched successfully (ADDR7F=1). Clear the ACKEN bit, read STS1 and then STS2 will clear the ADDR7F bit. Afterwards, set GENSTOP=1, the master enters receive stage at this time.
7. EV3: RDBF=1. It is cleared when the I2C_DT register is read.
8. End of communication.

Utilize DMA for data transfer

I²C data transfer can be done using DMA controller. An interrupt is generated by enabling the transfer complete interrupt bit. The DATAIEN bit in the I2C_CTRL2 register must be set 0 when using DMA for data transfer. The following sequence is for data transfer with DMA.

Transmission using DMA

1. Set the peripheral address (DMA_CxPADDR= I2C_DT address)
2. Set the memory address (DMA_CxMADDR=data memory address)
3. The transmission direction is set from memory to peripheral (DTD=1 in the DMA_CHCTRL register)
4. Configure the total number of bytes to be transferred in the DMA_CxDTCNT register
5. Configure other parameters such as priority, memory data width, peripheral data width, interrupts, etc in the DMA_CHCTRL register
6. Enable the DMA channel by setting CHEN=1 in the DMA_CxCTRL register
7. Enable I²C DMA request by setting DMAEN=1 in the I2C_CTRL2 register. Once the TDBE bit in the I2C_STS1 register is set, the data is loaded from the programmed memory to the I2C_DT register through DMA

8. When the number of data transfers, programmed in the DMA controller, is reached (DMA_CxDTCNT=0), the data transfer is complete (An interrupt is generated if enabled).
9. Master transmitter: Once the TDC flag is set, the STOP condition is generated, indicating that transfer is complete.

   Slave transmitter: Once the ACKFAIL flag is set, clear the ACKFAIL flag, transfer is complete.

Reception using DMA

1. Set the peripheral address (DMA_CxPADDR = I2C_DT address)
2. Set the memory address (DMA_CxMADDR = memory address)
3. The transmission directionis set from peripheral to memory (DTD=0 in the DMA_CHCTRL register)
4. Configure the total number of bytes to be transferred in the DMA_CxDTCNT register
5. Configure other parameters such as priority, memory data width, peripheral data width, interrupts, etc in the DMA_CHCTRL register
6. Enable the DMA channel by setting CHEN = 1 in the DMA_CxCTRL register
7. Enable I²C DMA request by setting DMAEN = 1 in the I2C_CTRL2 register. Once the RDBE bit in the I2C_STS1 register is set, the data is loaded from the I2C_DT register to the programmed memory through DMA
8. When the number of data transfers, programmed in the DMA controller, is reached (DMA_CxDTCNT = 0), the data transfer is complete (An interrupt is generated if enabled).
9. Master receiver: Clear the ACKFAIL flag, the STOP condition is generated, indicating that the transfer is complete (when the number of bytes to be transferred is greater ≥ 2 and DMAEND = 1, the NACK signal is generated automatically after transfer complete (DMA_CxDTCNT = 0)).

   Slave receiver: Once the STOPF flag is set, clear the STOPF flag, and the transfer is complete.

SMBus

The System Management Bus (SMBus) is a two-wire interface through which various devices can communicate with each other. It is based on I²C. With SMBus, the device can provide manufacturer information, tell the system its model/part number, report different types of errors and accept control parameters and so on. For more information, refer to SMBus 2.0 protocol.

Differences between SMBus and I²C

1. SMBus requires a minimum speed of 10 kHz for the purpose of management and monitor. It is quite easy to know whether the bus is in Idle state or not as long as a parameter is input while running on a certain transmission speed, without the need of detecting the STOP signals one after another, or even keeping STOP and other parameter monitor. There is no limit for I²C.
2. SMBus transmission speed ranges from 10 kHz to 100 kHz. In contrast,I²C has no minimum requirement, and its maximum speed varies from one mode to another, namely, 100 kHz in standard mode and 400 kHz in fast mode.
3. After reset, SMBus needs 35 ms of timeout, but there is no limit for I²C in this regard.

SMBus applications

1. The I²C interface is set in SMBus mode by setting PERMODE=1 in the I2C_CTRL1 register.
2. Select SMBus mode:
   SMBMODE=1: SMBus host
   SMBMODE=0: SMBus device
3. Other configurations are the same as those of I²C.

SMBus protocols are implemented by the user software, while I²C interface only provide the address identification of these protocols

SMBus address resolution protocol (ARP)

SMBus address conflicts can be resolved by dynamically assigning a new uique address to each device. Refer to SMBus 2.0 protocol for more information about ARP.

Setting the ARPEN bit can enable the I²C interface to recognize the default device address (0b1100001x). However, unique device identifier (UDID) and the detailed protocol implementation should be handled by software.

SMBus host notify protocol

The slave device can send data to the master device through SMBus host notify protocol. For example, the slave can notify the host to implement ARP with this protocol. Refer to SMBus 2.0 protocol for details on SMBus host notify protocol.

When the ARP mode is enabled (ARPEN=1) in host mode (SMBMODE=1), the I²C interface is enabled to recognize the 0b0001000x (default host address)

SMBus Alert

SMBus Alert is an optional signal that connects the ALERT pin between the host and the salve. With this signal, the slave notifies the host to access the slave. SMBALERT is a wired-AND signal. For more information about SMBus Alert, refer to SMBus2.0 protocol.

The detailed sequences are as follows:

SMBus host:

1. Enable SMBus Alert mode by setting SMBALERT=1
2. Enable ALERT interrupt if necessary
3. When an alert event occurs on the ALERT pin (ALERT pin changes from high to low)
4. The host will generate ALERT interrupt if enabled
5. The host then processes the interrupt and accesses to all devices through ARA (Alert Response Address 0001100x) so as to get the slave addresses. Only the devices with pulled-down SMBALERT can acknowledge ARA.
6. The host then continues to operate based on the slave addresses available.

SMBus slave:

1. When an alert event occurs and the ALERT pin changes from high to low (SMBALERT=1), the slave responses to ARA (Alert Response Address) address (0001100x)

2. Enable ALERT interrupt if necessary (an interrupt is generated when receiving ARA address)

3. Wait until the host gets the slave addresses through ARA

4. Report its own address, but it continues to wait if the arbitration is lost.

5. Address is reported properly, and the ALERT pin is released (SMBALERT=0).

Packet error checking (PEC)

Packet erro checking (PEC) is used to guarantee the correctness and integrity of data transfer. This is done by using CRC-8 polynominal:

$$C(x) = x^8 + x^2 + x + 1$$

PEC calculation is enabled when PECEN=1 to check address and data. It becomes invalid when the arbitration is lost.

PEC transmission:

- Common mode: Set PECTRA=1 after the last TDBE event so that PEC is transferred after the last transmitted byte.
- DMA mode: The PEC is transferred automatically after the last transmitted byte. For example, if the number of data to be transferred is 8, then DMA_TCNTx=8.

PEC reception:

- Common mode: Set the PECTRA bit after the last RDBF event. The PECTRA must be set before the ACK pulse of the current byte is received.
- DMA mode: When receiving, it will automatically consider the last byte as PECVAL and check it. For example, if the number of data to be transferred is 8, then DMA_TCNTx=9.

In reception mode, the NACK will be generated when PEC fails.

I²C interrupt requests

The following table lists all the I²C interrupt requests.

Table 36. I²C interrupt requests

Interrupt event	Event flag	Enable control bit
Start condition sent (Host)	STARTF	EVTIEN
Address sent (host) or address matched (slave)	ADDR7F	
10-bit address head sent (host)	ADDRHF	
Data transfer complete	TDC	
Stop condition received (slave)	STOPF	
Transmit data buffer empty	TDBE	EVTIEN and DATAIEN
Receive data buffer full	RDBF	
SMBus alert	ALERTF	ERRIEN
Timeout error	TMOUT	
PEC error	PECERR	
Overload/underload	OUF	
Acknowledge failure	ACKFAIL	
Arbitration lost	ARLOST	
Bus error	BUSERR	

I²C debug mode

When the microcontroller enters debug mode (Cortex™-M4F halted), the SMBUS timeout either continues to work or stops, depending on the I2Cx_SMBUS_TIMEOUT configuration bit in the DEBUG module.

I²C registers

These peripheral registers must be accessed by half-word (16 bits) or word (32 bits).

Table 37. I²C register map and reset value

Register	Offset	Reset value
I2C_CTRL1	0x00	0x0000
I2C_CTRL2	0x04	0x0000
I2C_OADDR1	0x08	0x0000
I2C_OADDR2	0x0C	0x0000
I2C_DT	0x10	0x0000
I2C_STS1	0x14	0x0000
I2C_STS2	0x18	0x0000
I2C_CLKCTRL	0x1C	0x0000
I2C_TMRISE	0x20	0x0002

Control register1 (I2C_CTRL1)

Bit	Register	Reset value	Type	Description
Bit 15	RESET	0x0	rw	I²C peripheral reset 0: I²C peripheral is not at reset state. 1: I²C peripheral is at reset state. Note: This bit can be used only when the BUSYF bit is “1”, and no Stop condition is detected on the bus.
Bit 14	Reserved	0x0	resd	Kept at its default value.
Bit 13	SMBALERT	0x0	rw	SMBus alert pin set This bit is set or cleared by software. It is cleared by hardware when I2CEN=0. 0: SMBus alert pin high. 1: SMBus alert pin low.
Bit 12	PECTEN	0x0	rw	Request PEC transfer enable This bit is set or cleared by software. It is cleared by hardware after PECTEN is sent, or under Start/Stop condition. 0: No PEC transfer 1: PEC transfer
Bit 11	MACKCTRL	0x0	rw	Master receive mode acknowledge control 0: ACKEN bit controls ACK of the current byte being transferred 1: ACKEN bit controls ACK of the next byte to be transferred This bit is used only when the number of bytes to receive is equal to 2 so as to ensure that the host responds to ACK in time.

Bit	Register	Reset value	Type	Description
Bit 10	ACKEN	0x0	rw	Acknowledge enable This bit is set or cleared by software. 0: Disabled (no acknowldge sent) 1: Enabled (acknowledge sent)
Bit 9	GENSTOP	0x0	rw	Generate stop condition This bit is set or cleared by software. It is cleared when a Stop condition is detected. It is set by hardware when a timeout error is detected. 0: No Stop condition is generated 1: Stop condition is generate The salve releases the SCL and SDA lines when this bit is set in slave mode.
Bit 8	GENSTART	0x0	rw	Generate start condition This bit is set or cleared by software. It is cleared when a Start condition is sent. 0: No Start condition is generated 1: Start condition is generated
Bit 7	STRETCH	0x0	rw	Clock stretching mode 0: Enabled 1: Disabled Note: This applies to slave mode only.
Bit 6	GCAEN	0x0	rw	General call address enable 0: Enabled 1: Disabled
Bit 5	PECEN	0x0	rw	PEC calculation enable 0: Disabled 1: Enabled
Bit 4	ARPEN	0x0	rw	SMBus address resolution protocol enable 0: Disabled 1: Enabled SMBus host: response to host address 0001000x SMBus slave: response to default device address 0001100x
Bit 3	SMBMODE	0x0	rw	SMBus device mode 0: SMBus slave 1: SMBus host
Bit 2	Reserved	0x0	resd	Forced to be 0 by hardware.
Bit 1	PERMODE	0x0	rw	I^2C peripheral mode 0: I^2C mode 1: SMBus mode
Bit 0	I2CEN	0x0	rw	I^2C peripheral enable 0: Disabled 1: Enabled All bits are cleared as I2CEN = 0 at the end of the communication. In master mode, this bit must not be cleared before the end of the communication.

Note: When the GENSTART, GENSTP or PECTEN bit is set, the I2C_CTRL1 cannot be written by software until the corresponding bit has been cleared by hardware,otherwise, a second GENSTART, GENSTP or PECTEN request may be set.

Control register2 (I2C_CTRL2)

Bit	Register	Reset value	Type	Description
Bit 15:13	Reserved	0x0	resd	Forced to be 0 by hardware.
Bit 12	DMAEND	0x0	rw	End of DMA transfer 0: The next DMA transfer is no the last one. 1: The next DMA transfer is the last one.
Bit 11	DMAEN	0x0	rw	DMA transfer enable 0: Disabled 1: Enabled
Bit 10	DATAIEN	0x0	rw	Data transfer interrupt enable An interrupt is generated when TDBE =1 or RDBF=1. 0: Disabled 1: Enabled
Bit 9	EVTIEN	0x0	rw	Event interrupt enable 0: Disabled 1: Enabled An interrupt is generated in the following conditions: – STARTF = 1 (Master mode) – ADDR7F = 1 (Master/slave mode) – ADDRHF= 1 (Master mode) – STOPF = 1 (Slave mode) – TDC = 1, but no TDBE or RDBF event – If DATAIEN = 1, the TDBE event is 1. – If DATAIEN = 1, the RDBF event is 1.
Bit 8	ERRIEN	0x0	rw	Error interrupt enable 0: Disabled 1: Enabled An interrupt is generated in the following conditions: – BUSERR = 1 – ARLOST = 1 – ACKFAIL = 1 – OVER = 1 – PECERR = 1 – TMOUT = 1 – ALERTF = 1
Bit 7:0	CLKFREQ	0x00	rw	I²C input clock frequency Correct input clock frequency must be set to generate correct timings. The range allowed is between 2 MHz and 120 MHz. 2: 2MHz 3: 3MHz …… 120: 120MHz

Own address register1 (I2C_OADDR1)

Bit	Register	Reset value	Type	Description
Bit 15	ADDR1MODE	0x0	rw	Address mode 0: 7-bit address 1: 10-bit address
Bit 14:10	Reserved	0x00	resd	Kept at its default value.
Bit 9:0	ADDR1	0x000	rw	Own address1 In 7-bit address mode, bit 0 and bit [9:8] don't care.

Own address register2 (I2C_OADDR2)

Bit	Register	Reset value	Type	Description
Bit 15:8	Reserved	0x00	resd	Kept at its default value.
Bit 7:1	ADDR2	0x00	rw	Own address 2 7-bit address
Bit 0	ADDR2EN	0x0	rw	Own address 2 enable 0: In 7-bit address mode, only OADDR1 is recognized. 1: In 7-bit address mode, both OADDR1 and OADDR2 are recognized.

Data register (I2C_DT)

Bit	Register	Reset value	Type	Description
Bit 15:8	Reserved	0x00	resd	Kept at its default value
Bit 7:0	DT[7:0]	0x00	rw	This field is used to store data received or to be transferred. Transmitter mode: Data transfer starts automatically when a byte is written to the DT register. Once the transfer starts (TDE=1), I²C will keep a continuous data transfer flow if the next data to be transferred is written to the DT register in a timely manner. Receiver mode: Btyes received are copied into the DT register (RDNE=1). A continuous data transfer flow can be maintained if the DT register is read before the next word is received (RDNE=1). Note: If an ARLOST event occurs on ACK pulse, the received byte is not copied into the data register, so it cannot be read.

Status register1 (I2C_STS1)

Bit	Register	Reset value	Type	Description
Bit 15	ALERTF	0x0	rw0c	SMBus alert flag In SMBus host mode: 0: No SMBus alert 1: SMBus alert event is received In SMBus slave mode: It indicates the receiving status of the default device address (0001100x) 0: Default device address is not received 1: Default device address is received This bit is cleared by software, or by hardware when I2CEN=0.
Bit 14	TMOUT	0x0	rw0c	SMBus timeout flag 0: No timeout error 1: Timeout This bit is cleared by software, or by hardware when I2CEN=0. Note: This function is valid only in SMBUS mode.
Bit 13	Reserved	0x0	resd	Kept at its default value.
Bit 12	PECERR	0x0	rw0c	PEC receive error flag 0: No PEC error 1: PEC error occurs This bit is cleared by software.
Bit 11	OUF	0x0	rw0c	Overload / underload flag In transmission mode: 0: Normal 1: Underload In reception mode: 0: Normal 1: Overload This bit is cleared by software, or by hardware when I2CEN=0.
Bit 10	ACKFAIL	0x0	rw0c	Acknowledge failure flag 0: No acknowledge failure 1: Acknowledge failure occurs Set by hardware when no acknowledge is returned. This bit is cleared by software, or by hardware when I2CEN=0.
Bit 9	ARLOST	0x0	rw0c	Arbitration lost flag 0: No arbitration lost is detected 1: Arbitration lost is detected This bit is cleared by software, or by hardware when I2CEN=0. On ARLOST even, the I^2C interface switches to slave mode automatically.

Bit	Register	Reset value	Type	Description
Bit 8	BUSERR	0x0	rw0c	Bus error flag 0: No Bus error occurs 1: Bus error occurs Set by hardware when the interface detects a misplaced Start or Stop condition. This bit is cleared by software, or by hardware when I2CEN=0.
Bit 7	TDBE	0x0	ro	Transmit data buffer empty flag 0: The data is being transferred from the DT register to the shift register (the data is still loaded with the data at this point.) 1: The data has been moved from the DT register to the shift register. The data register is empty now. This flag is set when the DT register is empty, and cleared when writing to the DT register. Note: The TDBE bit is not cleared by writing the first data to be transmitted, or by writing data when the TDC is set, since the data register is still empty at this time.
Bit 6	RDBF	0x0	ro	Receive data buffer full flag 0: Data register is empty 1: Data register is full (data received) This flag is cleared when the DT register is read. The RDBF bit is not set at ARLOST event.
Bit 5	Reserved	0x0	resd	Kept at its default value.
Bit 4	STOPF	0x0	ro	Stop condition generation complete flag 0: No Stop condition is detected 1: Stop condition is detected This bit is set by hardware when a Stop condition is detected on the bus by the slave if ACKEN=1. It is cleared by reading STS1 register followed by writing to the CTRL1 register.
Bit 3	ADDRHF	0x0	ro	Master 9~8 bit address head match flag 0: Master 9~8 bit address head mismatch 1: Master 9~8 bit address head match Set by hardware when the first byte is sent by master in 10-bit address mode. Cleared by a write to the CTRL1 register after the STS1 register is read by software, or by hardware when PEN=0. Note: The ADDR10 bit is not set after a NACK reception.
Bit 2	TDC	0x0	ro	Data transfer complete flag 0: Data transfer is not completed yet (the shift register still holds data) 1: Data transfer is completed (shift register is empty) This bit is cleared automatically by read or write access to the DT reginser, or when a Start or Stop condition is received. When STRETCH=0 In reception mode, when a new byte (including ACK pulse) is received and the data register is not read yet (RDBF=1) In transmission mode, when a new byte is sent and the data register is not written yet (TDBE=1) The TDC is set under both conditions.

Bit	Register	Reset value	Type	Description
Bit 1	ADDR7F	0x0	ro	0~7 bit address match flag 0: Address is not sent in host ode or received in slave mode 1: Address is sent in host mode or address is received in slave mode Cleared by read access to STS2 register after the software reads STS1 register. Note: the ADDR7F bit is not set after a NACK reception.
Bit 0	STARTF	0x0	ro	Start condition generation complete flag 0: No Start condition is generated. 1: Start condition is generated. Cleared by write access to the DT register after the software reads the STS1 register.

Status register2 (I2C_STS2)

Bit	Register	Reset value	Type	Description
Bit 15:8	PECVAL	0x00	ro	PEC value Cleared when PECEN is reset.
Bit 7	ADDR2F	0x0	ro	Received address 2 flag 0: Received address matches the contents of OADDR1 1: Received address matches the contents of OADDR2 Cleared when a Stop/Start condition is received, or by hardware when I2CEN=0.
Bit 6	HOSTADDRF	0x0	ro	SMBus host address reception flag 0: SMBus host address is not received. 1: SMBus host address is received. Cleared when a Stop/Start condition is received, or by hardware when I2CEN=0.
Bit 5	DEVADDRF	0x0	ro	SMBus device address reception flag 0: SMBus device address is not received. 1: SMBus device address is received. Cleared when a Stop/Start condition is received, or by hardware when I2CEN=0.
Bit 4	GCADDRF	0x0	ro	General call address reception flag 0: General call address is not received. 1: General call address is received. Cleared when a Stop/Start condition is received, or by hardware when I2CEN=0.
Bit 3	Reserved	0x0	resd	Keep at its default value.
Bit 2	DIRF	0x0	ro	Transmission direction flag 0: Data reception 1: Data transmission Cleared by hardware when a Stop condition is received.
Bit 1	BUSYF	0x0	ro	Bus busy flag transmission mode 0: Bus idle 1: Bus busy Set by hardware on detection of SDA/SCL low, and cleared by hardware on detection of a Stop condition.

Bit	Register	Reset value	Type	Description
Bit 0	TRMODE	0x0	ro	Transmission mode 0: Slave mode 1: Master mode Set by hardware when the GENSTART is set and a Start condition is sent. Cleared by hardware when a Stop condition is detected.

Clock control register (I2C_ CLKCTRL)

Bit	Register	Reset value	Type	Description
Bit 15	SPEEDMODE	0x0	rw	Speed mode selection 0: Standard mode (up to 100 kHz) 1: Fast mode (up to 400 kHz) In fast mode, an accurate 400kHz clock is generated when the I²C clock frequency is an integer multiple of 10 MHz.
Bit 14	DUTYMODE	0x0	rw	Fast mode duty cycle 0: The ratio of High to low is 1:2 1: The ratio of low to high is 9:16
Bit 13:12	Reserved	0x0	resd	Kept at its default value.
Bit 11:0	SPEED	0x000	rw	I²C bus speed config In standard mode: High level = SPEED × T_{I2C_CLK} Low level = SPEED × T_{I2C_CLK} In fast mode: DUTYMODE = 0: High level = SPEED × T_{I2C_CLK} × 1 Low level = SPEED × T_{I2C_CLK} × 2 DUTYMODE = 1: High level = SPEED × T_{I2C_CLK} × 9 Low level = SPEED × T_{I2C_CLK} × 16 The minimum value allowed in standard mode is 4. In fast mode, the minimum value allowed is 1. The CLKCTRL register can be configured only when the I²C is disabled (I2CEN = 0).

Note: The CLKCTRL register can be configured only when the I²C is disabled (I2CEN = 0).

I²C timer rise time register (I2C_TMRISE)

Bit	Register	Reset value	Type	Description
Bit 15:6	Reserved	0x000	resd	Forced to 0 by hardware.
Bit 5:0	RISETIME	0x02	rw	I²C bus rise time Time = RISETIME × T_{I2C_CLK} In standard mode, I²C protocol stand is 1000ns, and the formula as follows: RISETIME = F_{I2C_CLK} + 1 For example, when I²C clock is 48MHz, RISETIME = 48 + 1 In fast mode, I²C protocol stand is 300ns, and the formula as follows: RISETIME = F_{I2C_CLK} × 0.3 + 1 For example, when I²C clock is 48MHz, RISETIME = 48 × 0.3 + 1 Note: RISETIME[5:0] can be configured only when I2CEN=0.

12 Universal synchronous/asynchronous receiver/transmitter (USART)

USART introduction

The universal synchronous/asynchronous receiver/transmitter (USART) serves an interface for communication by means of various configuration and peripherals with different data formats. It supports asynchronous full-duplex and half-duplex as well as synchronous transfer. With a programmable baud rate generator, USART offers up to 7.5MBits/s of baud rate by setting the system frequency and frequency divider, which is also convenient for users to configure the required communication frequency.

In addition to standard NRZ asynchronous and synchronous receiver/transmitter communication protocols, USART also supports widely-used serial communication protocols such as LIN (Local Interconnection Network), IrDA (Infrared Data Association) SIRENDEC specification, Asynchronous SmartCard protcoal defined in ISO7816-3 standard, and CTS/RTS (Clear To Send/ Request To Send) hardware flow operation.

It also allows mutli-processor communication, and supports silent mode waken up by idle frames or ID matching to build up a USART network. Meanwhile, high-speed communication is possible by using DMA.

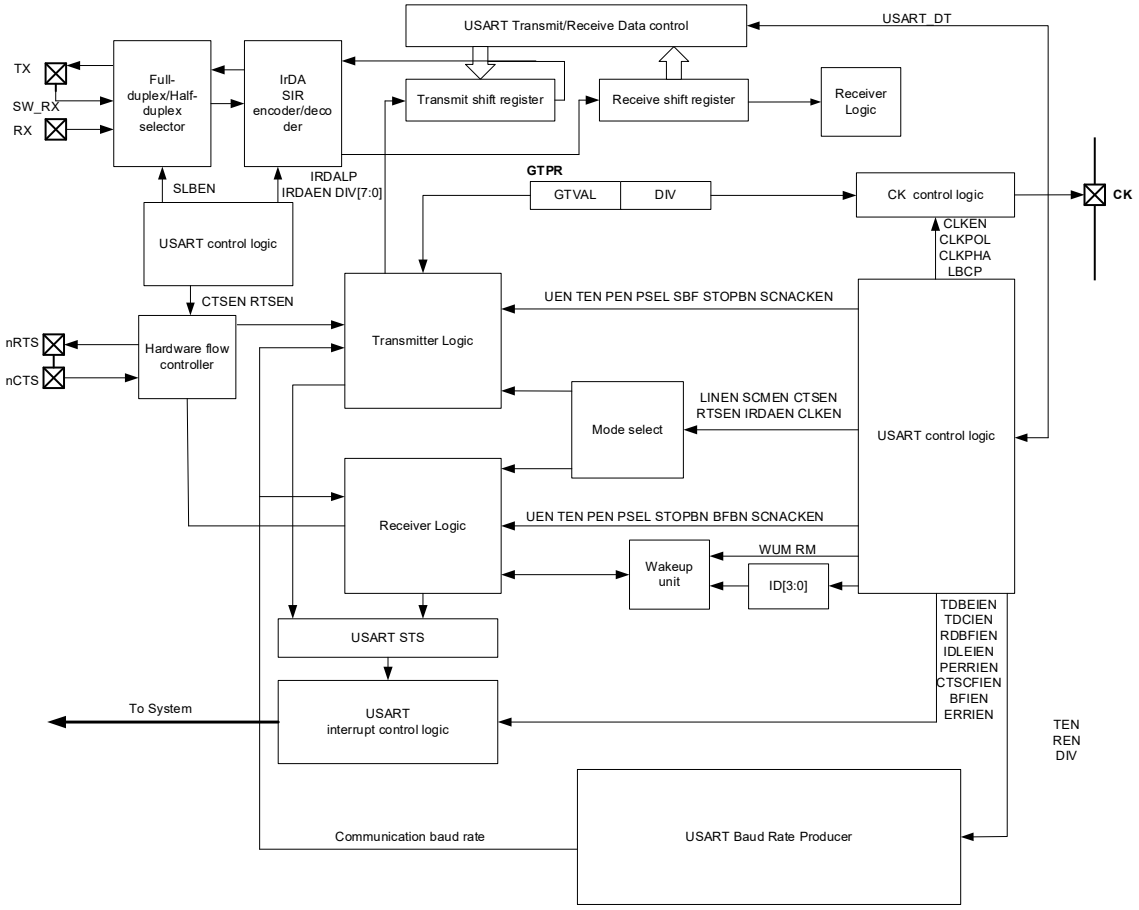


Figure 37. USART block diagram

Main features

- Programmable full-duplex or half-duplex communication
  - Full-duplex, asynchronous communication
  - Half-duplex, single communication
- Programmable communication modes
  - NRZ standard format (Mark/Space)
  - LIN (Local Interconnection Network):
  - IrDA SIR:
  - Asynchronous SmartCard protocol defined in ISO7816-3 standard: Support 0.5 or 1.5 stop bits in Smartcard mode
  - RS-232 CTS/RTS (Clear To Send/Request To Send) hardware flow operation
  - Multi-processor communication with silent mode (waken up by configuraing ID match and bus idle frame)
  - Synchronous mode
- Programmable baud rate generator
  - Shared by transmission and reception, up to 7.5 MBits/s

- Programmable frame format
  - Programmable data word length (8 bits or 9 bits)
  - Programmable stop bits-support 1 or 2 stop bits
  - Programmable parity control: transmitter with parity bit transmission capability, and receiver with received data parity check capability
- Programmable DMA multi-processor communication
- Programmable separate enable bits for transmitter and receiver
- Programmable output CLK phase, polarity and frequency
- Detection flags
  - Receive buffer full
  - Transmit buffer empty
  - Transfer complete flag
- Four error detection flags
  - Overrun error
  - Noise error
  - Framing error
  - Parity error
- Programmable 10 interrupt sources with flags
  - CTSF changes
  - LIN break detection
  - Transmit data register empty
  - Transmission complete
  - Receive data register full
  - Idle bus detected
  - Overrun error
  - Framing error
  - Noise error
  - Parity error

Full-duplex/half-duplex selector

The full-duplex and half-duplex selector enables USART to perform data exchanges with peripherals in full-duplex or half-duplex mode, which is achieved by setting the corresponding registers.

In two-wire unindirectional full-duplex mode (by default), TX pin is used for data output, while the RX pin is used for data input. Since the transmitter and receiver are independent of each other, USART is allowed to send/receive data at the same time so as to achieve full-duplex communication.

When the HALFSEL is set 1, the single-wire bidirectional half-duplex mode is selected for communication. In this case, the LINEN, CLKEN, SCMEN and IRDAEN bits must be set 0. RX pin is inactive, while TX and SW_RX are interconnected inside the USART. For the USART part, TX pins is used for data output, and SW_RX for data input. For the peripheral part, bidirectional data transfer is executed through IO mapped by TX pin.

Mode selector

Introduction

USART mode selector allows USART to work in different operation modes through software configuration so as to enable data exchanges between USART and peripherals with different communication protocols.

USART supports NRZ standard format (Mark/Space), by default. It also supports LIN (Local Interconnection Network), IrDA SIR (Serial Infrared), Asynchronous Smartcard protocol in ISO7816-3 standard, RS-232 CTS/RTS (Clear To Send/Request To Send) hardware flow operation, silent mode and synchronous mode, depending on USART mode selection configuration.

Configuration procedure

Selection of operation mode is done by following the configuration process listed below. In addition, such configuration method, along with that of receiver and transmitter described in the subsequent sections, are used to make USART initialization configuration.

LIN mode

Parameters configuration: LINEN=1, CLKEN=0, STOPBN[1:0]=0, SCMEN=0, SLHDEN=0, IRDAEN=0 and DBN=0.

LIN master has break fram transmission capability, and thus it is able to send 13-bit low-level LIN synchronous break frame by setting SBF=1.

Similarily, LIN slave has break frame detection capability, and thus it is able to detect 11-bit or 12-bit break fame, depending on whether BFBN=1 or BFBN=0.

Smartcard mode

Parameters configuration: SCMEN=1, LINEN=0, SLHDEN=0, IRDAEN=0, CLKEN=1, DBN=1, PEN=1, and STOPBN[1:0]=11.

The polarity, phase and pulse number of the clock can be configured by setting the CLKPOL, CLKPHA and LBCP bits (Refer to Synchronous mode for details).

The assertion of the TDC flag can be delayed by setting the SCGT[7:0] bit (guard time bit). The TDF bit can be asserted high after the guard time counter reaches the value programmed in the SCGT[7:0] bit.

The Smartcard is a single-wire half duplex communication protocol. The SCNACKEN bit is used to select whether to send NACK when a parity error occurs. This is to indicate to the Smarcard that the data has not been correctly received

Infrared mode

Parameters configuration: IRDAEN=1, CLKEN=0, STOPBN[1:0]=0, SCMEN=0 and SLHDEN=0.

The infrared low-power mode can be enabled by setting IRDALP=1. In normal mode the transmitted pulse width is specified as 3/16 bit. In infrared low-power mode, the pulse width can be configurable. And the ISDIV[7:0] bit can be used to achieve the desired low-power frequency.

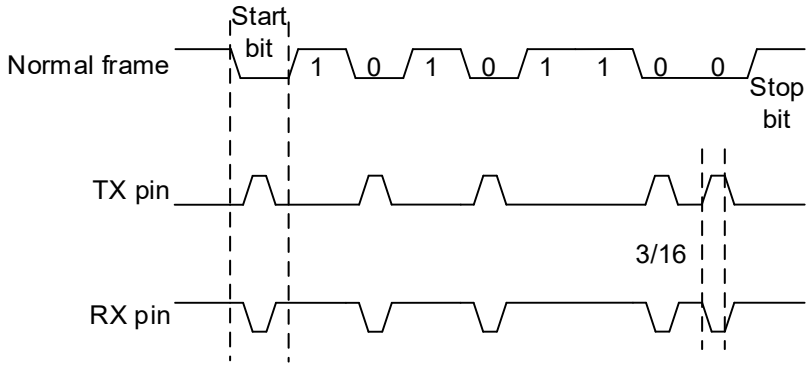


Figure 38. IrDA DATA(3/16) – normal mode

Hardware flow control mode

RTS and CTS flow control can be enabled by setting RTSEN=1 and CTSEN=1, respectively. This is to control serial data flow between two devices.

Silent mode

Silent mode can be entered by setting RM=1. It is possible to wake up from silent mode by setting WUM=1 (ID match) and WUM=0 (idle bus), respectively. The ID[3:0] is configurable. When ID match is selected, if the MSB of data bit is set to 1, it indicates that the current data is ID, and the four LSB represent ID value.

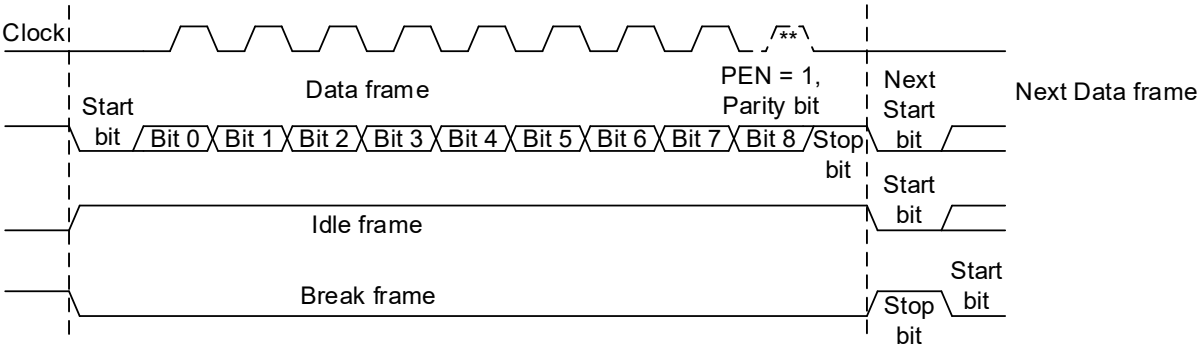
Synchronous mode

By setting the CLKEN bit to 1, synchronous mode and clock pin output are enabled. Select CK pin high or low in idle state by setting the CLKPOL bit (1 or 0). Whether to sample data on the second or the first edge of the clock depends on the CLKPHA bit (1 or 0). The LBCP bit (1 or 0) is used to select whether to output clock on the last data bit. And the ISDIV[4:0] is used to select the required clock output frequency.

USART frame format and configuration

USART data frame consists of start bit, data bit and stop bit, with the last data bit being as a parity bit. USART idle frame size is equal to that of the data frame under current configuration, but all bits are 1. USART break frame size is the current data frame size plus its stop bit. All bits before the stop bit are 0. The DBN bit is used to program 8-bit (DBN=0) or 9-bit (DBN=1) data bits.

9-bit word length (DBN = 1) :



8-bit word length (DBN = 0) :

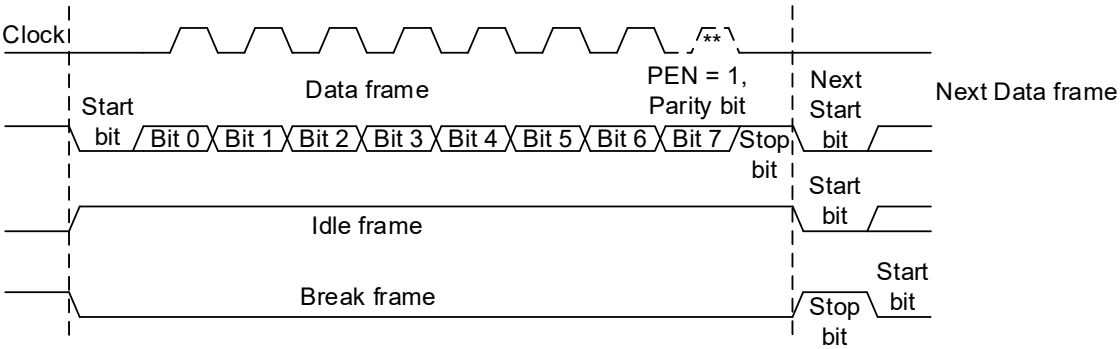


Figure 39. Word length

DMA transfer introduction

Enable transmit data buffer and receive data buffer using DMA to achieve continuous high-speed transmission for USART, which is detailed in subsequent sections. For more information on specific DMA configuration, refer to DMA chapter.

Transmission using DMA

1. Select a DMA channel: Select a DMA channel from DMA channel map table described in DMA chapter.
2. Configure the destination of DMA transfer: Configure the USART_DT register address as the destination address bit of DMA transfer in the DMA control register. Datat will be sent to this address after transmit request is received by DMA.
3. Configure the source of DMA transfer: Configure the memory address as the source of DMA transfer in the DMA control register. Data will be loaded into the USART_DT register from the memory address after transmit request is received by DMA.
4. Configure the total number of bytes to be transferred in the DMA control register.
5. Configure the channel priority of DMA transfer in the DMA control register.

6. Configure DMA interrupt generation after half or full transfer in the DMA control register.
7. Enable DMA transfer channel in the DMA control register.

Reception using DMA

1. Select a DMA transfer channel: Select a DMA channel from DMA channel map table described in DMA chapter.
2. Configure the destination of DMA transfer: Configure the memory address as the destination of DMA transfer in the DMA control register. Data will be loaded from the USART_DT register to the programmed destination after reception request is received by DMA.
3. Configure the source of DMA transfer: Configure the USART_DT register address as the source of DMA transfer in the DMA control register. Data will be loaded from the USART_DT register to the programmed destination after reception request is received by DMA.
4. Configure the total number of bytes to be transferred in the DMA control register.
5. Configure the channel priority of DMA transfer in the DMA control register.
6. Configure DMA interrupt generation after half or full transfer in the DMA control register.
7. Enable a DMA transfer channel in the DMA control register.

Baud rate generation

Introduction

USART baud rate generator uses an internal counter based on PCLK. The DIV (USART_BAUDR[15:0] register) represents the overflow value of the counter. Each time the counter is full, it denotes one-bit data. Thus each data bit width refers to PCLK cycles × DIV.

The receiver and transmitter of USART share the same baud rate generator, and the receiver splits each data bit into 16 equal parts to achieve oversampling, so the data bit width should not be less than 16 PCLK periods, that is, the DIV value must be equal to or greater than 16.

Configuration

User can program the desired baud rate by setting different system clocks and writing different values into the USART_BAUDR register. The calculation format is as follows:

$$\text{TX/RX baud rate} = f_{CK}/\text{DIV}$$

Where, f_{CK} refers to the system clock of USART (i.e. PCLK1/PCLK2)

Note: 1. Write access to the USART_BAUDR register before UEN. The baud rate register value should not be altered when UEN = 1.
2. When USART receiver or transmitter is disabled, the internal counter will be reset, and baud rate interrupt will occur.

Table 38. Error calculation for programmed baud rate

Baud		f_{PCLK}=36MHz			f_{PCLK}=72MHz		
No.	Desired (Kbps)	Actual	Value programmed in the baud register	Error%	Actual	Value programmed in the baud register	Error%
1	2.4	2.4	15000	0%	2.4	30000	0%
2	9.6	9.6	3750	0%	9.6	7500	0%
3	19.2	19.2	1875	0%	19.2	3750	0%
4	57.6	57.6	625	0%	57.6	1250	0%
5	115.2	115.384	312	0.15%	115.2	625	0%
6	230.4	230.769	156	0.16%	230.769	312	0.16%
7	460.8	461.538	78	0.16%	461.538	156	0.16%
8	921.6	923.076	39	0.16%	923.076	78	0.16%
9	2250	2250	16	0%	2250	32	0%
10	4500	NA	NA	NA	4500	16	0%

Taking a baud rate of 115.2Kbps as an example, if f_{PCLK}=36MHz, the value in the baud register should be set to 312(0x38). Based on formula, the calculated baud rate (acutal) is 36000000 / 312 = 115384 = 115.384Kbps. The % error between the desired and actual value is calculated based on the formula: (Calculated actual result-Desired)/desired baud rate × 100%, that is, (115.384 - 115.2) / 115.2 × 100% = 0.15%.

Transmitter

Transmitter introduction

USART transmitter has its individual TEN control bit. The transmitter and receiver share the same baud rate that is programmable. There is a transmit data buffer (TDR) and a transmit shift register in the USART. The TDBE bit is set whenever the TDR is empty, and an interrupt is generated if the TDBEIEN is set.

The data written by software is stored in the TDR register. When the shift register is empty, the data will be moved from the TDR register to the shift register so that the data in the transmit shift register is output on the TX pin in LSB mode. The output format depends on the programmed frame format.

If synchronous transfer or clock output is selected, the clock pulse is output on the CK pin. If the hardware flow control is selected, the control signal is input on the CTS pin.

Note: 1. The TEN bit cannot be reset during data transfer, or the data on the TX pin will be corrupted.
2. After the TEN bit is enabled, the USART will automatically send an idle frame.

Transmitter configuration

1. USART enable: Set the UEN bit.

2. Full-duplex/half-duplex configuration: Refer to 12.2 Full-duplex/half-duplex selector

3. Mode configuration: Refer to 12.3 Mode selector.

4. Frame format configuration: Refer to 12.4 USART frame format and configuration.

5. Interrupt configuration: Refer to 12.9 Interrupt requests.

6. DMA transmission configuration: If the DMA mode is selected, the DMATEN bit (bit 3 in the

USART_CTRL3register) is set, and configure DMA register accordingly.

7. Baud rate configuration: Refer to 12.6 Baud rate generation.
8. Transmitter enable: When the TEN bit is set, the USART transmitter will send an idle frame.
9. Write operation: Wait unitl the TDBE bit is set, the data to be transferred will be loaded into the USART_DT register (This operatin will clear the TDBE bit). Repeat this step in non-DMA mode.
10. After the last data expected to be transferred is written, wait until the TDC is set, indicating the end of transfer. The USART cannot be disabled before the flag is set, or transfer error will occur.
11. When TDC=1, read access to the USART_STS register and write access to the USART_DT register will clear the TDC bit; This bit can also be cleared by writing "0", but this is valid only in DMA mode.

Receiver

Receiver introduction

USART receiver has its individual REN control bit (bit 2 in the USART_CTRL1 register). The transmitter and receiver share the same baud rate that is programmable. There is a receive data buffer (RDR) and a receive shift register in the USART.

The data is input on the RX pin of the USART. When a valid start bit is detected, the receiver ports the data received into the receive shift register in LSB mode. After a full data frame is received, based on the programmed frame format, it will be moved from the receive shift register to the receive data buffer, and the RDBF is set accordingly. An interrupt is generated if the RDBFIEN is set.

If hardware flow control is selected, the control signal is output on the RTS pin.

During data reception, the USART receiver will detect whether there are errors to occur, including framing error, overrun error, parity check error or noise error, depending on software configuration, and whether there are interrupts to generate using the interrupt enable bits.

Receiver configuration

Configuration procedure

1. USART enalbe: UEN bit is set.
2. Full-duplex/half-duplex configuration: Refer to 12.2 Full-duplex/half-duplex selector.
3. Mode configuration: Refer to 12.3 Mode selector.
4. Frame format configuration: Refer to 12.4 USART frame format and configuration.
5. Interrupt configuration: Refer to 12.9 Interrupt requests.
6. Reception using DMA: If the DMA mode is selected, the DMAREN bit is set, and configure DMA register accordingly.
7. Baud rate configuration: Refer to 12.6 Baud rate generation.
8. Receiver enable: REN bit is set.

Character repeption

- The RDBF bit is set. It indicates that the content of the shift register is transferred to the RDR
- (Receiver Data Register). In other words, data is received and can be read (including its associated error flags)
- An interrupt is generated when the RDBFIEN is set.
- The erro flag is set when a framing error, noise error or overrun error is detected during reception.
- In DMA mode, the RDBF bit is set after every byte is received, and it is cleared when the data register is read by DMA.
- In non-DMA mode, the RDBF bit is cleared when read access to the USART_DT register by software. The RDBF flag can also be cleared by writing 0 to it. The RDBF bit must be cleared before the end of next frame reception to avoid overrun error.

Break frame reception

- Non-LIN mode: It is handled as a framing error, and the FERR is set. An interrupt is generated if the corresponding interrupt bit is enabled. Refer to framing error decribed below for details.
- LIN mode: It is handled as a break frame, and the BFF bit is set. An interrupt is generated if the BFIEN is set.

Idle frame reception

- It is handled as a data frame, and the IDLEF bit is set. An interrupt is generated if the IDLEIEN is set.

When a framing error occurs

- The FERR bit is set.
- The USART receiver moves the invalid data from the receive shift register to the receive data buffer.
- In non-DMA mode, both FERR and RDBF are set at the same time. The latter will generate an interrupt. In DMA mode, an interrupt is generated if the ERRIEN. When an overrun error occurs:
- The ROERR bit is set.
- The data in the receive data buffer is not lost. The previous data is still available when the USART_DT register is read.
- The content in the receive shift register is overwritten.Afterwards, any data received will be lost.
- An interrupt is generated if the RDBFIEN is set or both ERRIEN and DMAREN are set.
- The ROERR bit is cleared by reading the USART_STS register and then USART_DT register in order.

Note: 1. If ROERR is set, it indicates that at least one piece of data is lost, with two possibilities:
If RDBF=1, it indicates that the last valid data is still stored in the receive data buffer, and can be read.
If RDBF=0, it indicates that the last valid data in the receive data buffer has already been read.
2. The REN bit cannot be reset during data reception, or the byte that is currently being received will be lost.

Start bit and noise detection

A start bit detection occurs when the REN bit is set. With the oversampling techniques, the USART receiver samples data on the 3rd, 5th, 7th, 8th, 9th and 10th bits to detect the valid start bit and noise.

The following table shows the data sampling over start bit and noise detection.

Table 39. Data sampling over start bit and noise detection

Sampled value (3·5·7)	Sampled value (8·9·10)	NERR bit	Start bit validity
000	000	0	Valid
001/010/100	001/010/100	1	Valid
001/010/100	000	1	Valid
000	001/010/100	1	Valid
111/110/101/011	Any value	1	Invalid
Any value	111/110/101/011	1	Invalid

Note: If the sampling values on the 3rd, 5th, 7th, 8th, 9th, and 10th bits do not match the above mentioned requirements, the USART receiver does not think that a correct start bit is received, and thus it will abort the start bit detection and return to idle state waiting for a falling edge.

The USART receiver has the ability to detect noise. In the non-synchronous mode, the USART receiver samples data on the 7th, 8th and 9th bits, with its oversampling techniques, to distinguish valid data input from noise based on different sampling values, and recover data as well as set NERR (Noise Error Flag) bit.

Table 40. Data sampling over valid data and noise detection

Sampled value	NERR bit	Received bit value	Data validity
000	0	0	Valid
001	1	0	Invalid
010	1	0	Invalid
011	1	1	Invalid
100	1	0	Invalid
101	1	1	Invalid
110	1	1	Invalid
111	0	1	Valid

When noise is detected in a data frame:

- The NERR bit is set at the same time as the RDBF bit
- The invalid data is transferred from the receive shift register to the receive data buffer.
- No interrupt is generated in non-DMA mode. However, since the NERR bit is set at the same time as the RDBF bit, the RDBF bit will generae an interrupt. In DMA mode, an interrupt will be issued if the ERRIEN is set.

The NERR bit is cleared by read access to USART_STS register followed by the USART_DT read operation.

Interrupt requests

USART interrupt generator is the control center of USART interrupts. It is used to monitor the interrupt source inside the USART in real time and the generation of interrupts according to the programmed interrupt control bits. The following table shows the USART interrupt source and interrupt enable control bit. An interrupt will be generated over an event when the corresponding interrupt enable bit is set.

Table 41. USART interrupt request

Interrupt event	Event flag	Enable bit
Transmit data register empty	TDBE	TDBEIEN
CTS flag	CTSCF	CTSCFIEN
Transmit data complete	TDC	TDCIEN
Receive data buffer full	RDBF	RDBFIEN
Receiver overflow error	ROERR	
Idle flag	IDLEF	IDLEIEN
Parity error	PERR	PERRIEN
Break frame flag	BFF	BFIEN
Noise error, overflow error or framing error	NERR or ROERR or FERR	ERRIEN(1)

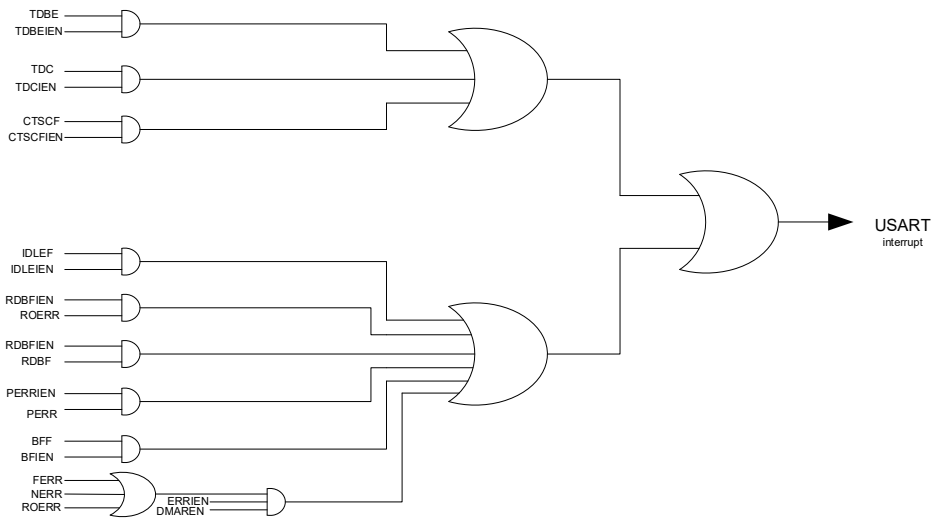


Figure 40. USART interrupt map diagram

I/O pin control

The following five interfaces are used for USART communication.

RX: Serial data input.

TX: Serial data output. In single-wire half-duplex and Smartcard mode, the TX pin is used as an I/O for data transmission and reception.

CK: Transmitter clock output. The output CLK phase, polarity and frequency can be programmable.

CTS: Transmitter input. Send enable signal in hardware flow control mode.

RTS: Receiver output. Send request signal in hardware flow control mode.

USART registers

Table 42. USART register map and reset value

Register	Offset	Reset value
USART_STS	0x00	0x0000 00C0
USART_DT	0x04	0x0000 0000
USART_BAUDR	0x08	0x0000 0000
USART_CTRL1	0x0C	0x0000 0000
USART_CTRL2	0x10	0x0000 0000
USART_CTRL3	0x14	0x0000 0000
USART_GTP	0x18	0x0000 0000

Status register (USART_STS)

Bit	Register	Reset value	Type	Description
Bit 31:10	Reserved	0x000000	resd	Forced to be 0 by hardware.
Bit 9	CTSCF	0x0	rw0c	CTS change flag This bit is set by hardware when the CTS status line changes. It is cleared by software. 0: No change on the CTS status line 1: A change occurs on the CTS status line.
Bit 8	BFF	0x0	rw0c	Break frame flag This bit is set by hardware when a break frame is detected. It is cleared by software. 0: Break frame is not detected. 1: Break frame is detected.
Bit 7	TDBE	0x1	ro	Transmit data buffer empty This bit is set by hardware when the transmit data buffer is empty. It is cleared by a USART_DT register write operation. 0: Data is not transferred to the shift register. 1: Data is transferred to the shift register.
Bit 6	TDC	0x1	rw0c	Transmit data complete This bit is set by hardware at the end of transmission. It is cleared by software. (Option 1: read access to USART_STS register followed by a USART_DT write operation; Option 2: Write “0” to this bit) 0: Transmission is not completed. 1: Transmission is completed.
Bit 5	RDBF	0x0	rw0c	Receive data buffer full This bit is set by hardware when the data is transferred from the shift register to the USART_DT register. It is cleared by software. (Option 1: read USART_DT register; Option 2: write “0” to this bit) 0: Data is not received. 1: Data is received.
Bit 4	IDLEF	0x0	ro	Idle flag This bit is set by hardware when an idle line is detected. It is cleared by software. (Read USART_DT register followed by a USART_DT read operation) 0: No idle line is detected. 1: Idle line is detected.

Bit	Register	Reset value	Type	Description
Bit 3	ROERR	0x0	ro	Receiver overflow error This bit is set by hardware when the data is received while the RDBF is still set. It is cleared by software. (Read USART_STS register followed by a USART_DT read operation) 0: No overflow error 1: Overflow error is detected Note: When this bit iset, the DT regiter content will not be lost, but the subsequent data will be overwritten.
Bit 2	NERR	0x0	ro	Noise error This bit is set by hardware when noise is detect on a received frame. It is cleared by software. (Read USART_STS register followed by a USART_DT read operation) 0: No noise is detected 1: Noise is detected
Bit 1	FERR	0x0	ro	Framing error This bit is set by hardware when a stop bit error (low), excessive noise or break frame is detected. It is cleared by software. USART_STS register followed by a USART_DT read operation). 0: No framing error is detected 1: Framing error is detected
Bit 0	PERR	0x0	ro	Parity error This bit is set by hardware when parity error occurs. It is cleared by software. USART_STS register followed by a USART_DT read operation) 0: No parity error occurs 1: Parity error occurs

Data register (USART_DT)

Bit	Register	Reset value	Type	Description
Bit 31:9	Reserved	0x000000	resd	Forced to be 0 by hardware.
Bit 8:0	DT	0x00	rw	Data value his register provides read and write function. When transmitting with the parity bit enabled, the value written in the MSB bit will be replaced by the parity bit. When receiving with the parity bit enabled, the value in the MSB bit is the received parity bit.

Baud rate register (USART_BAUDR)

Bit	Register	Reset value	Type	Description
Bit 31:16	Reserved	0x0000	resd	Forced to be 0 by hardware.
Bit 15:0	DIV	0x0000	rw	Divider This field define the USART divider.

Note: If the TEN and REN are both disabled, the baud counter stops counting.

Control register1 (USART_CTRL1)

Bit	Register	Reset value	Type	Description
Bit 31:14	Reserved	0x00000	resd	Forced to be 0 by hardware.
Bit 13	UEN	0x0	rw	USART enable 0: USART is disabled. 1: USART is enable.
Bit 12	DBN	0x0	rw	Data bit num This bit is used to program the number of data bits. 0: 8 data bits 1: 9 data bits
Bit 11	WUM	0x0	rw	Wakeup mode This bit determines the way to wake up silent mode. 0: Waken up by idle line 1: Waken up by ID match
Bit 10	PEN	0x0	rw	Parity enable This bit is used to enable hardware parity control (generation of parity bit for transmission; detection of parity bit for reception). When this bit is enabled, the MSB bit of the transmitted data is replaced with the parity bit; Check whether the parity bit of the received data is correct. 0: Parity control is disabled. 1: Parity control is enabled.
Bit 9	PSEL	0x0	rw	Parity selection This bit selects the odd or even parity after the parity control is enabled. 0: Even parity 1: Odd parity
Bit 8	PERRIEN	0x0	rw	PERR interrupt enable 0: Interrupt is disabled. 1: Interrupt is enabled.
Bit 7	TDBEIEN	0x0	rw	TDBE interrupt enable 0: Interrupt is disabled. 1: Interrupt is enabled.
Bit 6	TDCIEN	0x0	rw	TDC interrupt enable 0: Interrupt is disabled. 1: Interrupt is enabled.
Bit 5	RDBFIEN	0x0	rw	RDBF interrupt enable 0: Interrupt is disabled. 1: Interrupt is enabled.
Bit 4	IDLEIEN	0x0	rw	IDLE interrupt enable 0: Interrupt is disabled. 1: Interrupt is enabled.
Bit 3	TEN	0x0	rw	Transmitter enable This bit enables the transmitter. 0: Transmitter is disabled. 1: Transmitter is enabled.
Bit 2	REN	0x0	rw	Receiver enable This bit enables the receiver. 0: Receiver is disabled. 1: Receiver is enabled.

Bit	Register	Reset value	Type	Description
Bit 1	RM	0x0	rw	Receiver mute This bit determines if the receiver is in mute mode or not. It is set or cleared by software. When the idle line is used to wake up from mute mode, this bit is cleared by hardware after wake up. When the address match is used to wake up from mute mode, it is cleared by hardware after wake up. When address mismatches, this bit is set by hardware to enter mute mode again. 0: Receiver is in active mode 1: Receiver is in mute mode
Bit 0	SBF	0x0	rw	Send break frame This bit is used to send a break frame. It can be set or cleared by software. Generally speaking, it is set by software and cleared by hardware at the end of break frame transmission. 0: No break frame is transmitted 1: Break frame is transmitted

Control register2 (USART_CTRL2)

Bit	Register	Reset value	Type	Description
Bit 31:15	Reserved	0x00000	resd	Forced to be 0 by hardware.
Bit 14	LINEN	0x0	rw	LIN mode enable 0: LIN mode is disabled 1: LIN mode is enabled
Bit 13:12	STOPBN	0x0	rw	STOP bit num These bits are used to program the numter of stop bits. 00: 1 stop bit 01: 0.5 stop bit 10: 2 stop bits 11: 1.5 stop bits
Bit 11	CLKEN	0x0	rw	Clock enable This bit is used to enable the clock pin for synchronous mode or Smartcard mode. 0: Clock is disabled 1: Clock is enabled
Bit 10	CLKPOL	0x0	rw	Clock polarity In synchronous mode or Smartcard mode, this bit is used to select the polarity of the clock output on the clock pin in idle state. 0: Clock output low 1: Clock output high
Bit 9	CLKPHA	0x0	rw	Clock phase This bit is used to select the phase of the clock output on the clock pin in synchronous mode or Smartcard mode. 0: Data capture is done on the first clock edge 1: Data capture is done on the second clock edge

Bit	Register	Reset value	Type	Description
Bit 8	LBCP	0x0	rw	Last bit clock pulse This bit is used to select whether the clock pulse of the last data bit transmitted is output on the clock pin in synchronous mode. 0: The clock pulse of the last data bit is no output on the clock pin. 1: The clock pulse of the last data bis is output on the clock pin.
Bit 7	Reserved	0x0	resd	Keep at its default value.
Bit 6	BFIEN	0x0	rw	Break frame interrupt enable 0: Disabled 1: Enabled
Bit 5	BFBN	0x0	rw	Break frame bit num This bit is used to select 11-bit or 10-bit break frame. 0: 10-bit break frame 1: 11-bit break frame
Bit 4	Reserved	0x0	resd	Keep at its default value.
Bit 3:0	ID	0x0	rw	USART identification Configurable USART ID.

Note: These three bits (CLKPOL, CLKPHA and LBCP) should not be changed while the transmission is enabled.

Control register3 (USART_CTRL3)

Bit	Register	Reset value	Type	Description
Bit 31:11	Reserved	0x000000	resd	Forced to be 0 by hardware.
Bit 10	CTSCFIEN	0x0	rw	CTSCF interrupt enable 0: Interrupt is disabled 1: Interrupt is enabled
Bit 9	CTSEN	0x0	rw	CTS enable 0: CTS is disabled 1: CTS is enabled
Bit 8	RTSEN	0x0	rw	RTS enable 0: RTS is disabled 1: RTS is enabled
Bit 7	DMATEN	0x0	rw	DMA transmitter enable 0: DMA transmitter is disabled 1: DMA transmitter is enabled
Bit 6	DMAREN	0x0	rw	DMA receiver enable 0: DMA receiver is disabled 1: DMA receiver is enabled
Bit 5	SCMEN	0x0	rw	Smartcard mode enable 0: Smartcard mode is disabled 1: Smartcard mode is enabled
Bit 4	SCNACKEN	0x0	rw	Smartcard NACK enable This bit is used to send NACK when parity error occurs. 0: NACK is disabled when parity error occurs. 1: NACK is enabled when parity error occurs.
Bit 3	SLBEN	0x0	rw	Single-wire bidirectional half-duplex enable 0: Single-wire bidirectional half-duplex is disabled 1: Single-wire bidirectional half-duplex is enabled

Bit	Register	Reset value	Type	Description
Bit 2	IRDALP	0x0	rw	IrDA low-power mode This bit is used to configure IrDA low-power mode. 0: IrDA low-power mode is disabled 1: IrDA low-power mode is enabled
Bit 1	IRDAEN	0x0	rw	IrDA enable 0: IrDA is disabled 1: IrDA is enabled
Bit 0	ERRIEN	0x0	rw	Error interrupt enable An interrupt is generated when a framing error, overflow error or noise error occurs. 0: Error interrupt is disabled 1: Error interrupt is enabled

Guard time and divider register (GDIV)

Bit	Register	Reset value	Type	Description
Bit 31:16	Reserved	0x0000	resd	Forced to be 0 by hardware.
Bit 15:8	SCGT	0x00	rw	Smartcard guard time value This field specifies the guard time value. The transmission complete flag is set after this guard time in smartcard mode.
Bit 7:0	ISDIV	0x00	rw	IrDA/smartcard division In IrDA mode: 8 bit [7:0] is valid. It is valid in common mode and must be set to 00000001. In low-power mode, it divides the peripheral clock to serve as the period base of the pulse width; 00000000: Reserved–Do not write. 00000001: Divided by 1 00000010: Divided by 2 …… Smartcard mode: the lower 5 bit [4:0] is valid. This division is used to divide the peripheral clock to provide clock for the Smartcard. Configured as follows: 00000: Reserved–Do not write. 00001: Divided by 2 00010: Divided by 4 00011: Divided by 6 ……

13 Serial peripheral interface (SPI)

SPI introduction

The SPI interace supports either the SPI protocol or the I²S protocoal, depending on software configuration. This chapter gives an introduction of the main features and congiruation procedure of SPI used as SPI or I²S.

Functional descriptions

SPI description

The SPI can be configured as host or slave based on software configuration, supporting full-duplex, reception-only full-duplex and transmission-only/reception-only half-duplex modes, DMA transfer, and automatic CRC function of SPI internal hardware.

SPI block diagram:

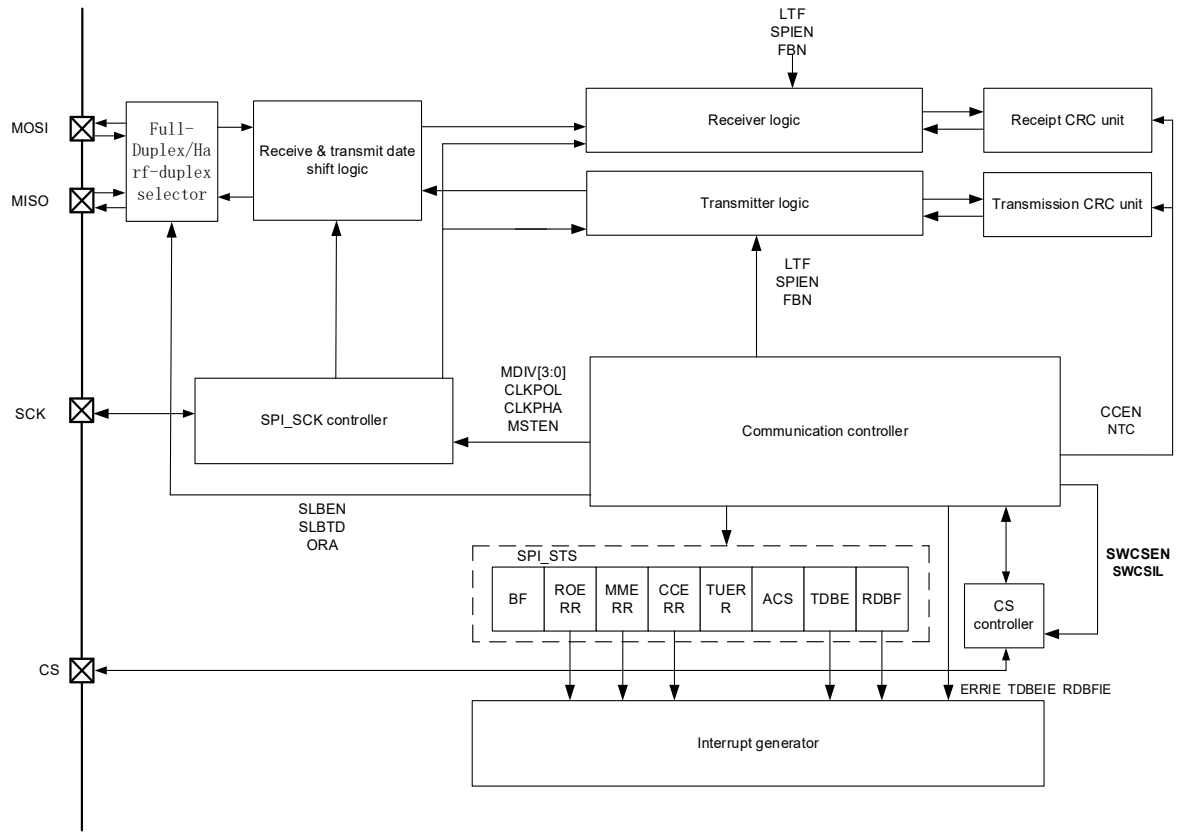


Figure 41. SPI block diagram

Main features as SPI:

- Full-duplex or half-duplex communication
  - Full-duplex synchronous communication (supporting reception-only mode to release IO for transmission)
  - Half-duplex synchronous communication (transfer direction is configurable: receive or transmit)
- Master or slave mode
- CS signal processing mode
  - CS signal processing by hardware
  - CS signal processing by software
- 8-bit or 16-bit frame format
- Communication frequency and prescalers (Frequency up to 48M, and prescalers up to $f_{PCLK}/2$)
- Programmable clock plarity and phase
- Programmable data transfer order (MSB-first or LSB-first)
- Programmable error interrupt flags (receiver overflow error, master mode error and CRC error)
- Programmable transmit data buffer empty interrupt and receive data buffer full interrupt
- Support transmission and reception using DMA
- Support hardware CRC transmission and error checking
- Busy status flag

Full-duplex/half-duplex selector

When used as an SPI interface, it supports four synchronous modes: two-wire unidirectional full-duplex, single-wire unidirectional receive only, single-wire bidirectional half-duplex transmit and single-wire bidirectional half-duplex receive.

The following diagram shows the two-wire unidirectional full-duplex mode and SPI IO connection:

The SPI operats in two-wire unidirectional full-duplex mode when the SLBEN bit and the ORA bit is both 0. In this case, the SPI supports data transmission and reception at the same time. IO connection is as follows:

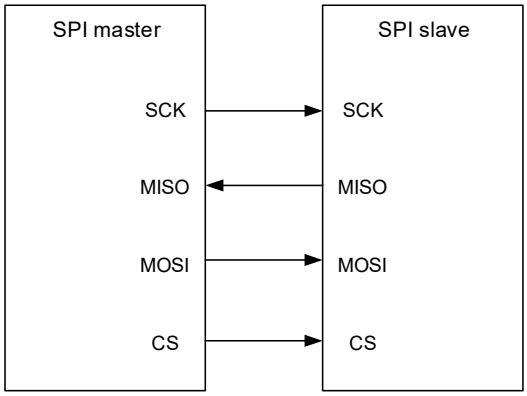


Figure 42. SPI two-wire unidirectional full-duplex connection

In either master or slave mode, it is required to wait until the RDBF bit and TDBE bit is set, and BF=0 before disabling the SPI or entering power-saving mode (or disabling SPI system clock).

The following diagrams show the single-wire unidirectional receive-only mode and SPI IO connection

The SPI operates in single-wire unidirectional receive-only mode when the SLBEN is 0 and the ORA is set. In this case, the SPI can be used only for data reception (transmission is not supported). The MISO pin transmits data in slave mode and receives data in master mode. The MOSI pin transmits data in master mode and receives data in salve mode.

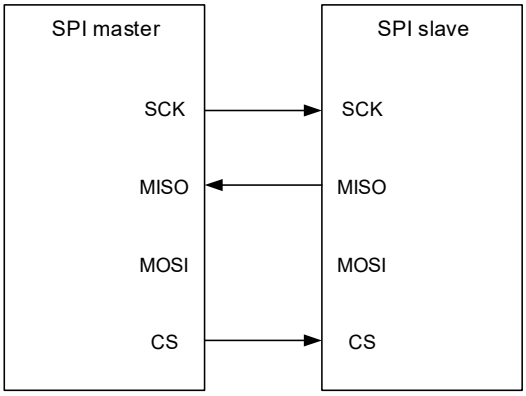


Figure 43. Single-wire unidirectional receive only in SPI master mode

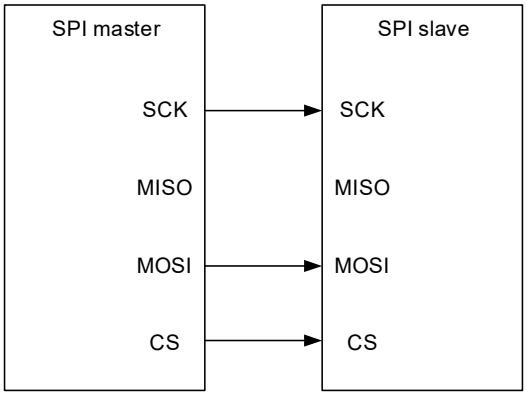


Figure 44. Single-wire unidirectional receive only in SPI slave mode

In master mode, it is necessary to wait until the second-to-last RDBF bit is set and then another SPI_CPK period before disabling the SPI. The last RDBF must be set before entering power-saving mode (or disabling SPI system clock).

In slave mode, there is no need to check any flag before disabling the SPI. However, it is required to wait until the BF becomes 0 before entering power-saving mode.

The following diagram shows single-wire bidirectional half-duplex mode and SPI IO connection

When the SLBEN is set, the SPI operates in single-wire bidirectional half-duplex mode. In this case, the SPI supports data reception and transmission alternately. In master mode, the MOSI pin transmits or receives data in master mode, while the MISO pin is released. In slave mode, the MISO pin transmits or receives data, but the MOSI pin is released.

The SLBTD bit is used by software to configure transfer direction. When the SLBTD bit is set, the SPI can be used only for data transmission; when the SLBTD bit is 0, the SPI can be used only for data reception.

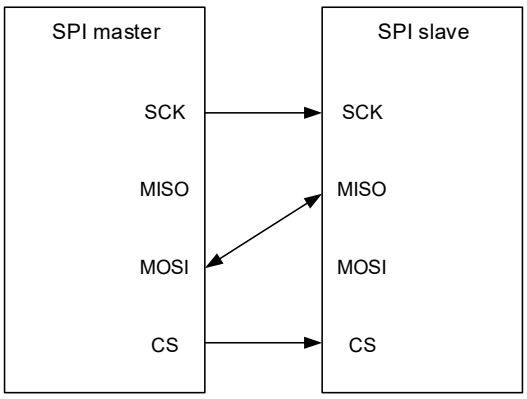


Figure 45. Single-wire bidirectional half-duplex mode

When the SPI is selected for data transmission in single-wire bidirectional half-duplex mode (master or slave), the TDBE bit must be set, and the BF must be 0 before disabling the SPI. The power-saving mode (or disabling SPI system clock) cannot be entered unless the SPI is disabled.

In master mode, when the SPI is selected for data reception in single-wire bidirectional half-duplex mode, it is required to wait until the second-to-last RDBF is set and then another SPI_SCK period before disabling the SPI. And the last RDBF must be set before entering power-saving mode (or disabling SPI system clock).

In slave mode, when the SPI is selected for data reception in single-wire bidirectional half-duplex mode, there is no need to check any flags before disabling the SPI. However, the BT must be 0 before entering power-saving mode (or disabling SPI system clock).

Chip select controller

The Chip select controller (CS) is used to enable hardware or software control for chip select signals through software configuration. This controller is used to select master/slave device in multi-processor mode, and to avoid conflicts on the data lines by enabling the SCK signal output followed by CS signal. The hardware and software configuration procedure is detailed as follows, along with their respective input/output in master and slave mode.

CS hardware configuration procedure:

In master mode with CS being as an output, HWCSOE=1, SWCSEN=0, the CS hardware control is enabled. If the SPI is enabled, low level is output on the CS pin. The CS signal is then released after the SPI is disabled and the transmission is complete.

In master mode with CS being as an input, HWCSOE=0, SWCSEN=0, the CS hardware control is enabled. At this point, the SPI is automatically disabled by hardware and enters slave mode as soon as the CS pin low is detected by master SPI. The mode error flag (MMERR bit) is set at the same time. An interrupt is generated if ERRIE=1. When the MMERR is set, the SPIEN and MSTEN cannot be set by software. The MMERR is cleared by read or write access to the SPI_STS register followed by write operation to the SPI_CTRL1 register.

In slave mode with CS being as an input, HWCSOE=0, SWCSEN=0, the CS hardware control is enabled. The slave selects whether to transmit / receive data based on the level on the CS pin. The slave is selected for data reception and transmission only when the CS pin is low.

CS software configuration procedure:

In master mode with CS being as an input, SWCSEN=1, the CS software control is enabled. When SWCSIL=0, the SPI is automatically disabled by hardware and enters slave mode. The mode error flag (MMERR bit) is set at this time. An interrupt is generated if ERRIE=1. When the MMERR bit is set, the SPIEN and MSTEN bits cannot be set by software. The MMERR bit is cleared by read or write access to the SPI_STS register followed by write operation to the SPI_CTRL1 register.

In slave mode with CS being as an input, SWCSEN=1, the CS software control is enabled. The SPI judges the CS signal with the SWCSIL bit, instead of CS pin. When SWCSIL=0, the slave is selected for data reception and transmission.

SPI_SCK controller

The SPI protocol adopts synchronous transmission. In master mode with the SPI being as SPI, it is required to generate a communication clock for data reception and transmission on the SPI, and the communication clock should be output to the slave via IO for data reception and transmission. In slave mode, the communication clock is provided by peripherals, and is input to the SPI via IO. In all, the SPI_SCK controller is used for the generation and distribution of SPI_SCK, with the configuration procedure detailed as follows:

SPI_SCK controller configuration procedure:

- Clock polarity and clock phase selection: It is selected by setting the CLKPOL and CLKPHA bit.
- Clock prescaler selection: Select the desired PCLK frequency by setting the CRM bit. Select the desired prescaler by setting the MDIV[3:0] bit.
- Master/slave selection: Select SPI as master or slave by setting the MSTEN bit.

Note that the clock output is activated after the SPI is enabled in master reception-only mode, and it remains output until when the SPI is disabled and the reception is complete.

CRC

There is an independent transmission and reception CRC calculation unit in the SPI. When used as SPI through software configuration, the SPI enables CRC calculation and CRC check automatically while the user is reading or writing through DMA or CPU. During the transmission, if the received data is not consistent with, detected by hardware, the data in the SPI_RCRC register, and such data is exactly the CRC value, then the CCERR bit will be set. An interrupt is generated if ERRIE=1.

The CRC function and configuration procedure of the SPI are described as follows.

CRC configuration procedure

- CRC calculation polynominal is configured by setting the SPI_CPOLY register.
- CRC enable: The CRC calculation is enabled by setting the CCEN bit. This operation will reset the SPI_RCRC and SPI_TCRC registers.
- Select if or when the NTC bit is set, depending on DMA or CPU data register. See the following descriptions.

Trasmission using DMA

When DMA is used to write the data to be transmitted, if the CCEN bit is enabled, the hardware calculates the CRC value automatically according to the value in the SPI_CPOLY register and each transmitted data, and sends the CRC value at the end of the last data transmission. This result is regarded as the value of the SPI_TCRC register.

Reception using DMA

When DMA is used to read the data to be received, if the CCEN bit is enabled, the hardware calculates the CRC value automatically according to the value in the SPI_CPOLY register and each received data, and waits until the completion of CRC data reception at the end of the last data reception before comparing the received CRC value with the value of the SPI_RCRC register. If check error occurs, the CCERR flag is set. An interrupt is generated if the ERRIE bit is enabled.

Transmission using CPU

Unlike DMA mode, after writing the last data to be transmitted, the CPU mode requires the NTC bit to be set by software before the end of the last data transmission.

Reception using CPU

In two-wire unidirectional full-duplex mode, follow CPU transmission mode to operate the NTC bit, the CRC calculation and check in CPU reception mode will be completed automatically.

In single-wire unidirectional reception-only mode and single-wire bidirectional reception-only mode, it is required to set the NTC bit before the software receives the last data when the second-to-last data is received.

DMA transfer

The SPI supports write and read operations with DMA. Refer to the following configuration procedure.

Special attention should be paid to: when the CRC calculation and check is enabled, the number of data transferred by DMA is configured as the number of the data to be transferred. The number of data read with DMA is configured as the number of the data to be received. In this case, the hardware will send CRC automatically at the end of full transfer, and the receiver will also perform CRC check. Note that the received CRC data will be moved into the SPI_DT register by hardware, with the RDBF being set, and the DMA read request will be sent if then DAM transfer is enabled. Hence, it is recommended to read the SPI_DT register to get the CRC value at the end of CRC reception in order to avoid the upcoming transfer error.

Transmission with DMA

- Select DMA channel: Select a DMA channel for the current SPI from DMA channel map table described in DMA chapter.

- Configure the destination of DMA transfer: Configure the SPI_DT register address as the destination address bit of DMA transfer in the DMA control register. Datat will be sent to this address after transmit request is received by DMA.
- Configure the source of DMA transfer: Configure the memory address as the source of DMA transfer in the DMA control register. Data will be loaded into the SPI_DT register from the memory address after transmit request is received by DMA.
- Configure the total number of bytes to be transferred in the DMA control register.
- Configure the channel priority of DMA transfer in the DMA control register.
- Configure DMA interrupt generation after half or full transfer in the DMA control register.
- Enable DMA transfer channel in the DMA control register.

Reception with DMA

- Select DMA transfer channel: Select a DMA channel for the current SPI from DMA channel map table described in DMA chapter.
- Configure the destination of DMA transfer: Configure the memory address as the destination of DMA transfer in the DMA control register. Data will be loaded from the SPI_DT register to the programmed destination after reception request is received by DMA.
- Configure the source of DMA transfer: Configure the SPI_DT register address as the source of DMA transfer in the DMA control register. Data will be loaded from the SPI_DT register to the programmed destination after reception request is received by DMA.
- Configure the total number of bytes to be transferred in the DMA control register.
- Configure the total number of bytes to be transferred in the DMA control register.
- Configure DMA interrupt generation after half or full transfer in the DMA control registe
- Enable DMA transfer channel in the DMA control register.

Transmitter

The SPI transmitter is clocked by SPI_SCK controller. It can output different data frame formats, depending on software configuration. There is a SPI_DT register available in the SPI that is used to be written with the data to be transmitted. When the transmitter is clocked, the contents in the SPI_DT register are copied into the data buffer (Unlike SPI_DT, it is driven by SPI_SCK, and controlled by hardware,instead of software), and sent out in order based on the programmed frame format.

Both DMA and CPU can be used for write operation. For DMA transfer, refer to DMA transfer section for more details. For CPU transfer, attention should be paid to the TDBE bit. The reset value of this bit is 1, indicating that the SPI_DT register is empty. If the TDBEIE bit is set, an interrupt is generated. After the data is written, the TDBE is pulled low until the data is moved to the transmit data buffer before the TDBE is set once again. This means that the user can be allowed to write the data to be transmitted only when the TDBE is set.

After the transmitter is configured and the SPI is enabled, the SPI is ready for data transmission. Before going forward, it is necessary for the users to refer to full-duplex / half-duplex chapter to get detailed configuration information, go to the Chip select controller chapter for specific chip select mode, check the SPI_SCK controller chapter for information on communication clock, and refer to CRC and DMA transfer chapter to configure CRC and DMA (if necessary). The recommended configuration procedure are as follows.

Transmitter configuration procedure:

- Configure full-duplex/half-duplex selector
- Configure chip select controller
- Configure SPI_SCK controller
- Configure CRC (if necessary)
- Configure DMA transfer (if necessary)
- If the DMA transfer mode is not used, the software will check whether to enable transmit data interrupt (TDBEIE =1) through the TDBE bit.
- Configure frame format: select MSB/LSB mmode with the LTF bit, and select 8/16-bit data with the FBN bit
- Enable SPI by setting the SPIEN

Receiver

The SPI receiver is clocked by the SPI_SCK controller. It can output different data frame formats through software configuration. There is a receive data buffer register, driven by the SPI_SCK, in the SPI receiver.

At the last CLK of each transfer, the data is moved from the shift register to the receive data buffer register. Then the transmitter sets the receive data complete flag to the SPI logic. When the flag is detected by the SPI logic, the data in the receive data buffer is copied into the SPI_DT register, with the RDBF being set. This means that the data is received, and it is already stored into the SPI_DT. In this case, read access to the SPI_DT register will clear the RDBF bit.

Both DMA and CPU can be used for read operation. For DMA transfer, refer to DMA transfer section for more details. For CPU transfer, attention should be paid to the RDBE bit. The reset value of this bit is 0, indicating that the SPI_DT register is empty. If the data is received and moved into the SPI_DT, the RDBF is set, meaning that there are some data to be read in the SPI_DT register. An interrupt is generated if the RDBFIE bit is set.

When the next received data is ready to be moved to the SPI_DT register, if the previous received data is still not read (RDBF=1), then the data overflow occurs. The previous receive data is not lost, but the next received data will do. At this point, the ROERR is set. An interrupt is generated if the ERRIE is set. Read SPI_DT register and then the SPI_STS register will clear the ROERR bit. The recommended configuration procedure is as follows.

Receiver configuration procedure:

- Configure full-duplex/half-duplex selector
- Configure chip select controller
- Configure SPI_SCK controller
- Configure CRC (if necessary)
- Configure DMA transfer (if necessary)
- If the DMA transfer mode is not used, the software will check whether to enable receive data interrupt (RDBEIE =1) through the RDBE bit.
- Configure frame format: select MSB/LSB mmode with the LTF bit, and select 8/16-bit data with the FBN bit
- Enable SPI by setting the SPIEN

Interrupt

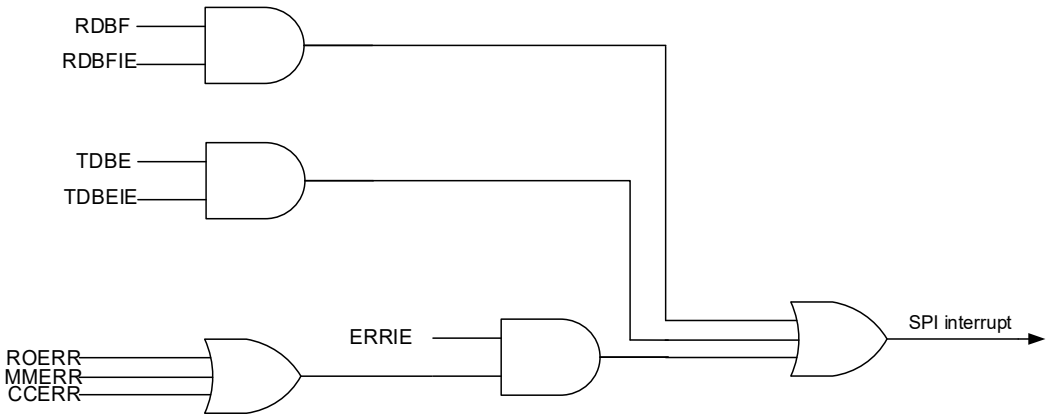


Figure 46. SPI interrupts

IO pin control

Usually, the SPI is connected to external devices through four pins:

- MISO: Master In/Slave Out. The pin receives data in master mode, and transmit data in slave moe.
- MOSI: Master Out/Slave In. The pin transmits data in master mode, and receives data in slave mode.
- SCK: SPI communication clock. The pin serves as output in master mode, and input in slave mode.
- CS: Chip Select. This is an optional pin which selects master/slave mode.

Note: Some of SPI1/I²S1 and SPI3/I²S3 are shared with JTAG pins (SPIx_CS/I2Sx_WS shared with JTDI, SPIx_SCK/I2Sx_CK with JTDO), so these pins are not cotrolled by IO controller, and they are used as JTAG by default after each reset. To configure them as SPIx/I²Sx, the JTAG should be disabled (during debugging) and switched to SWD interface, or both the JTAG and SWD are disabled (duing normal run)

Precautions

The software is required to read the DT register in order to get CRC value at the end of CRC reception.

I²S functional description

I²S introduction

The I²S can be configured by software as master repection/transmission, and slave reception/transmission, supporting foure kinds of audio protocols including Philips standard, MSB-aligned standard, LSB-aligned standard and PCM standard, respectively. The DMA transfer is also supported.

A single I²S supports half-duplex. However, it can work with the two additional instantiated I²S modules (I2S2EXT and I2S3EXT) to achieve full-duplex mode. In other words, combining the I²S2 with the I2S2EXT enables the I²S2 to support full-duplex mode. This is true for the I²S3 through the combination of the I²S3 with the I2S3EXT. Refer to I²S full-duplex section for more information.

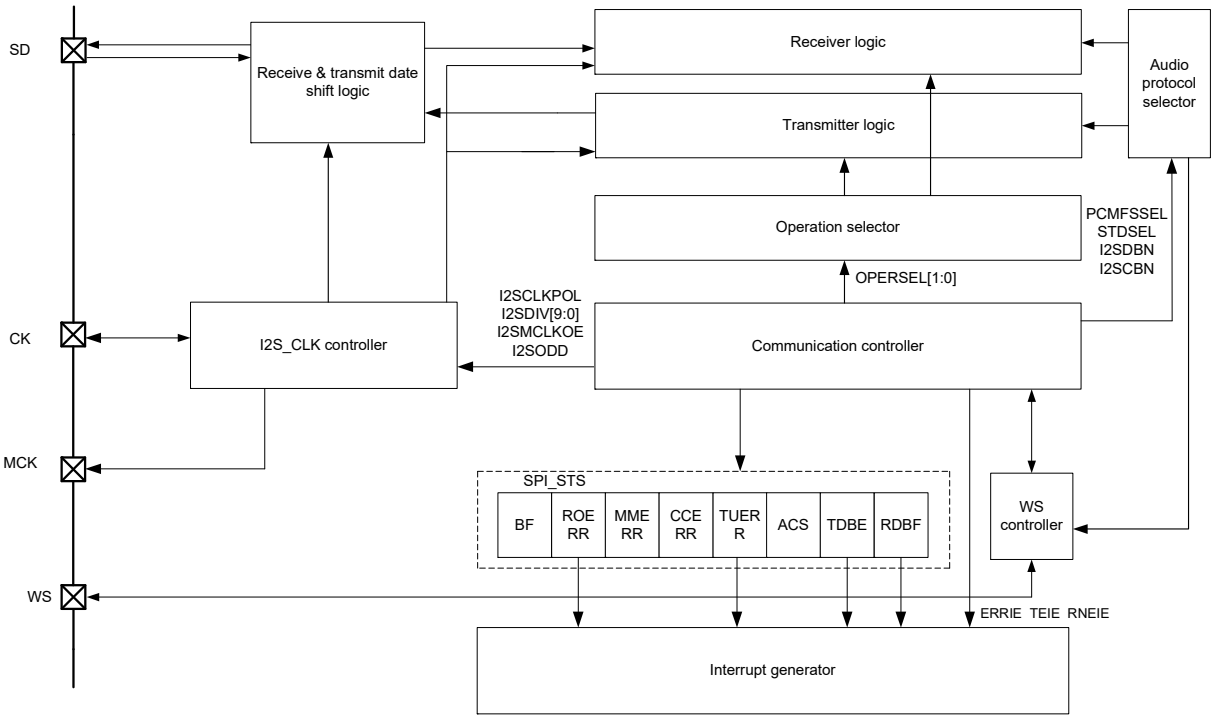


Figure 47. I²S block diagram

Main features when the SPI is used as I²S:

- Programmable operation mode
- Slave device transmission
  - Slave device reception
  - Master device transmission
  - Master device reception
- Programmable clock polarity
- Programmable clock frequency (8 KHz to 192 KHz)
- Prorammable data bits (16 bit, 24 bit, 32 bit)
- Programmable channel bits (16 bit, 24 bit)
- Programmable audio protocol
  - I²S Philips standard
  - MSB-aligned standard (left-aligned)
  - LSB-aligned standard (right-aligned)
  - PCM standard (long or short frame)
- I²S full-duplex
- DMA transfer
- Main peripheral clock with a fixed frequence of 256x Fs (audio sampling frequency)

I²S full-duplex

Two extra I²S modules (I2S2EXT and I2S3EXT) are used to support I²S full-duplex mode. Combine the I²S2 with the I2S2EXT, and I²S3 with I2S3EXT to support full-duplex mode.

Note: I2S2EXT and I2S3EXT only used for I²S full-duplex mode.

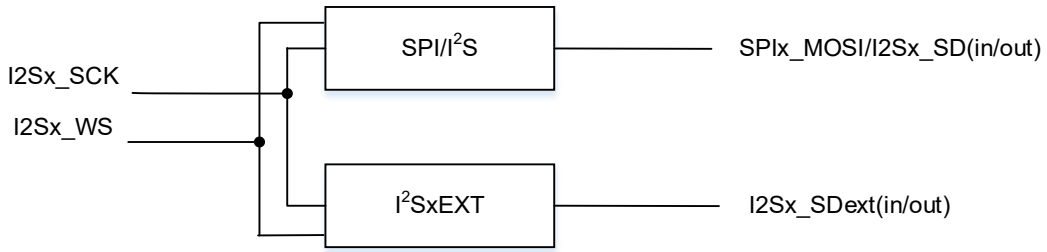


Figure 48. I²S full-duplex structure

I²Sx can be used as master, where x should be 2 or 3

- In half-duplex mode, only I²Sx can output SCK and WS
- In full-duplex mode, only I²Sx can output SCK and WS

I²SxEXT is only used for full-duplex mode, and I²SxEXT only for slave mode.

Both the I²Sx and I²SxEXT can be configured as transmit or receive mode.

Operation mode selector

The SPI, used as I²S selector, offers multiple operation modes for selection, namely, slave device transmission, slave device reception, master device transmission and master device reception. This is done by software configuration.

Slave device transmission:

Set the I2SMSEL bit, and OPERSEL[1:0] =00, the I²S will work in slave device transmission mode.

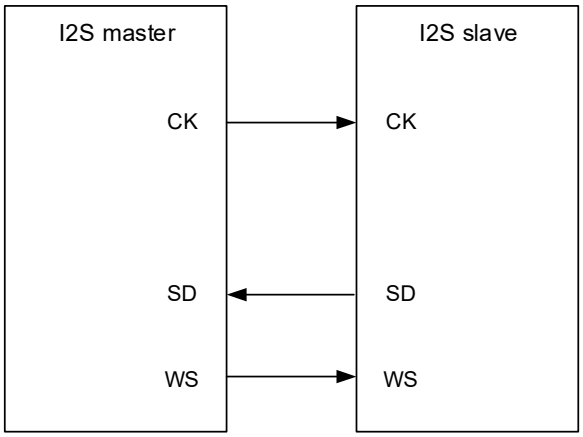


Figure 49. I²S slave device transmission

Slave device reception:

Set the I2SMSEL bit, and OPERSEL[1:0]=01, the I²S will work in slave device reception mode.

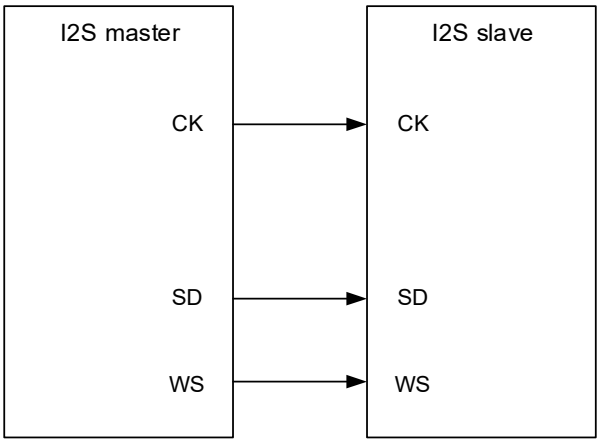


Figure 50. I²S slave device reception

Master device transmission:

Set the I2SMSEL bit, and OPERSEL[1:0]=10, the I²S will work in master device transmission mode.

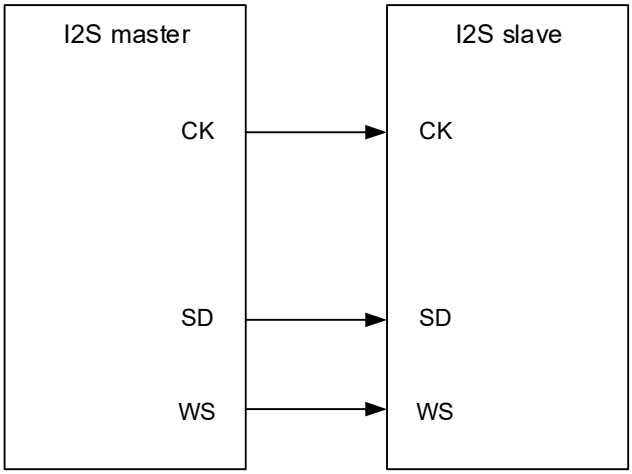


Figure 51. I²S master device transmission

Master device reception:

Set the I2SMSEL bit, and OPERSEL[1:0]=11, the I²S will work in master device repection mode.

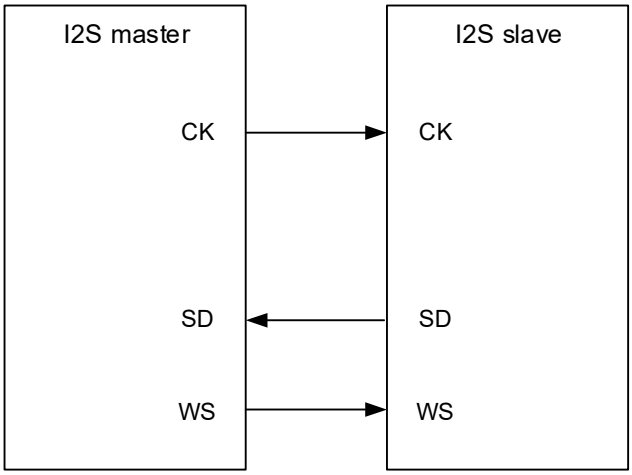


Figure 52. I²S master device reception

Audio protocol selector

While being used as I²S, the SPI supports multiple audio protocols. The user can control the audio protol selector through software configuration to select the desired audio protocol, with the data bits and channel bits being controlled by the audio protocol selector. Besides, the user can also select the data bits and channel bits through software configuration. Meanwhile, the audio protocol selector manages the WS controller, output or detect the WS siganal that meets the protocol requirements.

- Select audio protocol by seeting the STDSEL bit
  - STDSLE=00: Philips standard
  - STDSLE=01: MSB-aligned standard (left-aligned)
  - STDSLE=10: LSB-aligned standard (right-aligned)
  - STDSLE=11: PCM standard
- Select PCM frame synchronization format: PCMFSSEL=1 for PCM long frame synchronization, PCMFSSEL=0 for short frame synchronization (this step is required when selecting PCM protocol)
- Select data bits by setting the I2SDBN bit
  - I2SDBN=00: 16 bit
  - I2SDBN =01: 24 bit
  - I2SDBN =10: 32 bit
- Select channel bits by setting the I2SCBN bit
  - I2SDBN =0: 16 bit
  - I2SDBN =1: 32 bit

Note: Read/Write operation mode depends on the selected audio protocols, data bits and channel bits. The following lists all possible configuration combinations and their respective read and write operation mode.

- Philips standard, PCM standard, MSB-aligned or LSB-aligned standard, 16-bit data and 16-bit channel
  The data bit is the same as the channel bit. Each channel requires one read/write operation from/to the SPI_DT register, and the number of DMA transfer is 1.
- Philips standard, PCM standard or MSB-aligned standard, 16-bit data and 32-bit channel
  The data bit is different from the channel bit. Each channel requires one read/write operation from/to the SPI_DT register, and the number of DMA transfer is 1. The first 16 bits (MSB) are the significant bits, and the 16-bit LSB is forced to 0 by hardware.
- Philips standard, PCM standard or MSB-aligned standard, 24-bit data and 32-bit channel
  The data bit is different from the channgle bit. Each channel requires two read/write operations from/to the SPI_DT register, and the number of DMA transfer is 2. The 16-bit MSB transmits and receives the first 16-bit data, the 16-bit LSB transmits and receives the 8-bit MSB data, with 8-bit LSB data being forced to 0 by hardware.
- Philips standard, PCM standard, MSB-aligned or LSB-aligned standard, 32-bit data and 32-bit channel
  The data bit is the same as the channel bit. Each channel requires two read/write operations from/to the SPI_DT register, and the number of DMA transfer is 2. These 32-bit data are proceeded in two times, with 16-bit data each time.
- LSB-aligned standard, 16-bit data and 32-bit channel
  The data bit is different from the channel bit. Each channel requires one read/write operation from/to the SPI_DT register, and the number of DMA transfer is 1. The 16 bits (LSB) are the significant bits while the first 16-bit data (MSB) are forced to 0 by hardware.
- LSB-aligned standard, 24-bit data and 32-bit channel
  The data bit is different from the channel bit. Each channel requires two read/write operations from/to the SPI_DT register, and the number of DMA transfer is 2. For the first 16-bit data, its 8-bit LSB are the significant bits, with the 8-bit MSB forced to 0 by hardware; the subsequent 16 bits transmit and receive the second 16-bit data.

I2S_CLK controller

The audio protocols the SPI supports adopts synchronous transmission. In master mode, it is required to generate a communication clock for data reception and transmission on the SPI, and the communication clock should be output to the slave via IO for data reception and transmission. In slave mode, the communication clock is provided by master, and is input to the SPI via IO. In all, the I2S_SCK controller is used for the generation and distribution of I2S_SCK, with the configuration procedure detailed as follows:

When used as I^2S master, the SPI can provide communication clock (CK) and main peripheral clock (MCK) shown in the following diagram. The CK and MCK are generated by HCLK divider, with the prescaler of the MCK determined by I2SDIV and I2SODD. The calculation formula is seen in the following diagram.

The prescaler of the CK depends on whether to provide the main clock for peripherals. To ensure that the main clock is always 256 times larger than the audio sampling frequency, The channel bits should be taken into account. When the main clock is needed, the CK should be divided by 8 (I2SCBN=0) or 4 (I2SCBN=1), then divided again by the same prescaler as that of the MCK, that is the final communication clock; When the main clock is not needed, the the prescaler of the CK is determined by I2SDIV and I2SODD, shown in the following diagram.

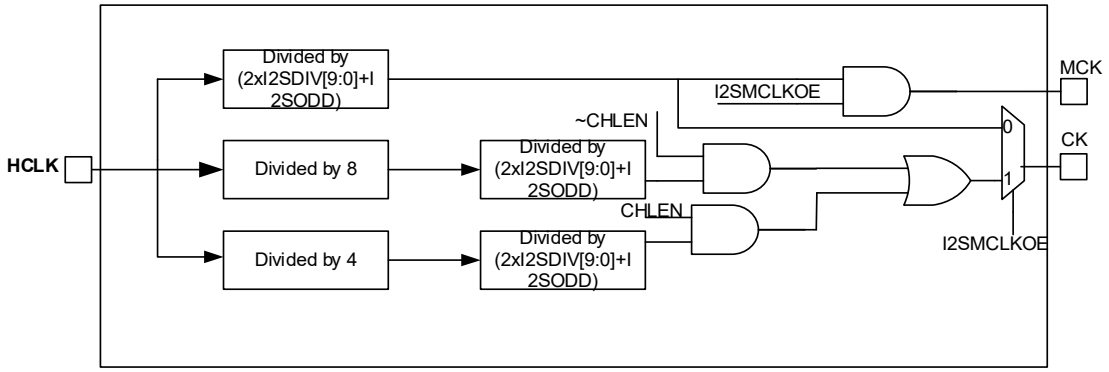


Figure 53. CK & MCK source in master mode

Apart from the above-mentioned configuration, the following table lists the values of I2SDIV and I2SODD corresponding to some specific frequencies, as well as their respective error for the users to configure the I2SDIV and I2SODD.

Table 43. Audio frequency precision using system clock

SysCLK (MHz)	MCLK	TargetFs (Hz)	16bit				32bit			
			I2S DIV	I2S_ ODD	RealFs	Error	I2S DIV	I2S_ ODD	RealFs	Error
240	NO	192000	19	1	192307	1.2%	10	0	187500	2.34%
240	NO	96000	39	0	96153	0.16%	19	1	96153	0.16%
240	NO	48000	78	0	48076	0.16%	39	0	48076	0.16%
240	NO	44100	85	0	44117	0.04%	42	1	44117	0.04%
240	NO	32000	117	0	32051	0.16%	58	1	32051	0.16%
240	NO	22050	170	0	22058	0.04%	85	0	22058	0.04%
240	NO	16000	234	1	15991	0.05%	117	0	16025	0.16%
240	NO	11025	340	0	11029	0.04%	170	0	11029	0.04%
240	NO	8000	469	0	7995	0.05%	234	1	7995	0.05%
240	YES	192000	2	1	187500	2.34%	2	1	187500	2.34%
240	YES	96000	5	0	93750	2.34%	5	0	93750	2.34%
240	YES	48000	10	0	46875	2.34%	10	0	46875	2.34%
240	YES	44100	10	1	44642	1.23%	10	1	44642	1.23%
240	YES	32000	14	1	32327	1.02%	14	1	32327	1.02%
240	YES	22050	21	1	21802	1.12%	21	1	21802	1.12%
240	YES	16000	29	1	15889	0.68%	29	1	15889	0.68%
240	YES	11025	42	1	11029	0.04%	42	1	11029	0.04%
240	YES	8000	58	1	8012	0.16%	58	1	8012	0.16%
200	No	192000	16	1	189393.9	1.36%	8	0	195312.5	1.73%
200	No	96000	32	1	96153.85	0.16%	16	1	94696.97	1.36%
200	No	48000	65	0	48076.92	0.16%	32	1	48076.92	0.16%
200	No	44100	71	0	44014.08	0.19%	35	1	44014.08	0.19%
200	No	32000	97	1	32051.28	0.16%	49	0	31887.76	0.35%
200	No	22050	141	1	22084.81	0.16%	71	0	22007.04	0.19%
200	No	16000	195	1	15984.65	0.10%	97	1	16025.64	0.16%

SysCLK (MHz)	MCLK	TargetFs (Hz)	16bit				32bit			
			I2S DIV	I2S_ ODD	RealFs	Error	I2S DIV	I2S_ ODD	RealFs	Error
200	No	11025	283	1	11022.93	0.02%	141	1	11042.4	0.16%
200	No	8000	390	1	8002.561	0.03%	195	1	7992.327	0.10%
200	Yes	192000	3	0	130208.3	32.18%	3	0	130208.3	32.18%
200	Yes	96000	4	0	97656.25	1.73%	4	0	97656.25	1.73%
200	Yes	48000	8	0	48828.13	1.73%	8	0	48828.13	1.73%
200	Yes	44100	9	0	43402.78	1.58%	9	0	43402.78	1.58%
200	Yes	32000	12	0	32552.08	1.73%	12	0	32552.08	1.73%
200	Yes	22050	17	1	22321.43	1.23%	17	1	22321.43	1.23%
200	Yes	16000	24	1	15943.88	0.35%	24	1	15943.88	0.35%
200	Yes	11025	35	1	11003.52	0.19%	35	1	11003.52	0.19%
200	Yes	8000	49	0	7971.939	0.35%	49	0	7971.939	0.35%
100	No	192000	8	0	195312.5	1.73%	4	0	195312.5	1.73%
100	No	96000	16	1	94696.97	1.36%	8	0	97656.25	1.73%
100	No	48000	32	1	48076.92	0.16%	16	1	47348.48	1.36%
100	No	44100	35	1	44014.08	0.19%	17	1	44642.86	1.23%
100	No	32000	49	0	31887.76	0.35%	24	1	31887.76	0.35%
100	No	22050	71	0	22007.04	0.19%	35	1	22007.04	0.19%
100	No	16000	97	1	16025.64	0.16%	49	0	15943.88	0.35%
100	No	11025	141	1	11042.4	0.16%	71	0	11003.52	0.19%
100	No	8000	195	1	7992.327	0.10%	97	1	8012.821	0.16%
100	Yes	96000	2	0	97656.25	1.73%	2	0	97656.25	1.73%
100	Yes	48000	4	0	48828.13	1.73%	4	0	48828.13	1.73%
100	Yes	44100	4	1	43402.78	1.58%	4	1	43402.78	1.58%
100	Yes	32000	6	0	32552.08	1.73%	6	0	32552.08	1.73%
100	Yes	22050	9	0	21701.39	1.58%	9	0	21701.39	1.58%
100	Yes	16000	12	0	16276.04	1.73%	12	0	16276.04	1.73%
100	Yes	11025	17	1	11160.71	1.23%	17	1	11160.71	1.23%
100	Yes	8000	24	1	7971.939	0.35%	24	1	7971.939	0.35%
72	No	192000	6	0	187500	2.34%	3	0	187500	2.34%
72	No	96000	11	1	97826.09	1.90%	6	0	93750	2.34%
72	No	48000	32	1	34615.38	27.88%	11	1	48913.04	1.90%
72	No	44100	25	1	44117.65	0.04%	13	0	43269.23	1.88%
72	No	32000	35	0	32142.86	0.45%	17	1	32142.86	0.45%
72	No	22050	51	0	22058.82	0.04%	25	1	22058.82	0.04%
72	No	16000	70	1	15957.45	0.27%	35	0	16071.43	0.45%
72	No	11025	102	0	11029.41	0.04%	51	0	11029.41	0.04%
72	No	8000	140	1	8007.117	0.09%	70	1	7978.723	0.27%
72	Yes	96000	2	0	70312.5	26.76%	2	0	70312.5	26.76%
72	Yes	48000	3	0	46875	2.34%	3	0	46875	2.34%
72	Yes	44100	3	0	46875	6.29%	3	0	46875	6.29%
72	Yes	32000	4	1	31250	2.34%	4	1	31250	2.34%
72	Yes	22050	6	1	21634.62	1.88%	6	1	21634.62	1.88%
72	Yes	16000	9	0	15625	2.34%	9	0	15625	2.34%

SysCLK (MHz)	MCLK	TargetFs (Hz)	16bit				32bit			
			I2S DIV	I2S_ ODD	RealFs	Error	I2S DIV	I2S_ ODD	RealFs	Error
72	Yes	11025	13	0	10817.31	1.88%	13	0	10817.31	1.88%
72	Yes	8000	17	1	8035.714	0.45%	17	1	8035.714	0.45%

DMA transfer

The SPI supports write and read operations with DMA. Whether used as SPI or I²S, read/write request using DMA comes from the same peripheral. As a result, their configuration procedure are the same, described as follows.

Transmission with DMA

- Select DMA channel: Select a DMA channel for the current SPI from DMA channel map table described in DMA chapter.
- Configure the destination of DMA transfer: Configure the SPI_DT register address as the destination address bit of DMA transfer in the DMA control register. Datat will be sent to this address after transmit request is received by DMA.
- Configure the source of DMA transfer: Configure the memory address as the source of DMA transfer in the DMA control register. Data will be loaded into the SPI_DT register from the memory address after transmit request is received by DMA.
- Configure the total number of bytes to be transferred in the DMA control register.
- Configure the channel priority of DMA transfer in the DMA control register.
- Configure DMA interrupt generation after half or full transfer in the DMA control register.
- Enable DMA transfer channel in the DMA control register.

Reception with DMA

- Select DMA transfer channel: Select a DMA channel for the current SPI from DMA channel map table described in DMA chapter.
- Configure the destination of DMA transfer: Configure the memory address as the destination of DMA transfer in the DMA control register. Data will be loaded from the SPI_DT register to the programmed destination after reception request is received by DMA.
- Configure the source of DMA transfer: Configure the SPI_DT register address as the source of DMA transfer in the DMA control register. Data will be loaded from the SPI_DT register to the programmed destination after reception request is received by DMA.
- Configure the total number of bytes to be transferred in the DMA control register.
- Configure the total number of bytes to be transferred in the DMA control register.
- Configure DMA interrupt generation after half or full transfer in the DMA control registe
- Enable DMA transfer channel in the DMA control register.

Transmitter/Receiver

Whether being used as SPI or I²S, there is no difference for CPU. The SPI (in whatever mode) shares the same base address, the same SPI_DT register, the same transmitter and receiver. The SPI transmitter and receiver is responsible fore sending and receiving the desired data frame according to the configuration of the communication controller. Thus their status flags such as TDBE, RDBF and ROERR, and their interrupt enable bits including TDBEIE, RDBFIE and ERRIE are identifical.

Special attention must be paid to:

- CRC check is not available on the I²S. Any operation linked to CRC, including CCERR flag and the corresponding interrupts, is not supported.
- I²S protocol needs decode the current channel status. The ACS bit is used to judge whether the current transfer occurs on the left channel (ACS=0) or the right channel (ACS=1).
- TUERR bit indicates whether an underrun occurs. TUERR=1 means an underrun error occurs on the transmitter. An interrupt is generated when the ERRIE is set.
- Read/write operation to the SPI_DT register is different under different audio protocols, data bits and channel bits. Refer to the audio protocol selector section for more information.
- Pay more attention to the I²S disable operation under different configurations, shown as follows:
  - I2SDBN=00, I2SCBN=1, STDSLE=10: wait for the second-to-last RDBF=1 and 17 CK periods before disabling the I²S.
  - I2SDBN=00, I2SCBN=1, STDSLE=00 or STDSLE=01 or STDSLE=11: wait for the last RDBF=1 and one CK period before the I²S.
  - I2SDBN, I2SCBN,STDSLE combination: wait for the second-to-last RDBF=1 and one CK period before disabling the I²S.

I²S transmitter configuration procedure:

- Configure operation mode selector
- Configure audio protocol selector
- Configure I2S_SCK controller
- Configure DMA transfer (if necessary)
- Set the I2SEN bit to enable I²S
- Follow above steps to configure the I²SxEXT (For I²S full-duplex mode)

I²S receiver configuration procedure:

- Configure operation mode selector
- Configure audio protocol selector
- Configure I2S_SCK controller
- Configure DMA transfer (if necessary)
- Set the I2SEN bit to enable I²S
- Follow above steps to configure the I²SxEXT (For I²S full-duplex mode)

Interrupts

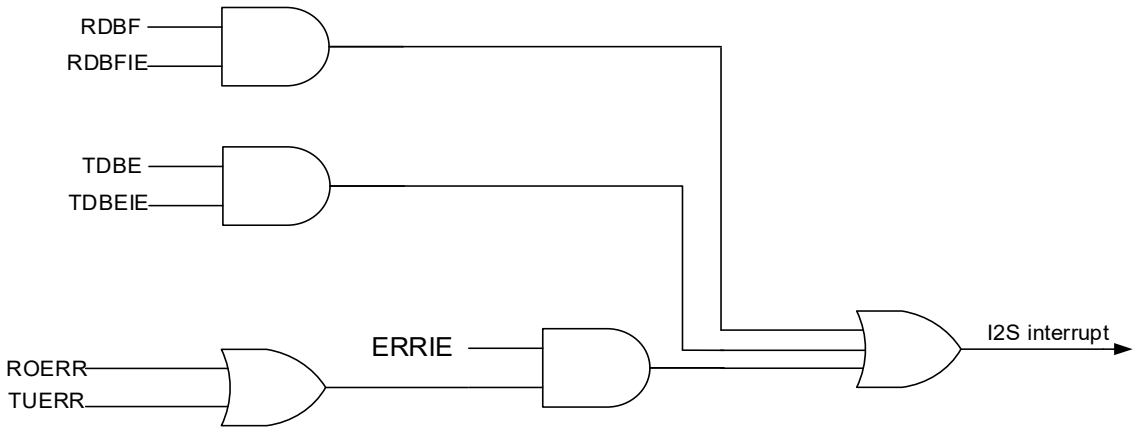


Figure 54. I²S interrupt

IO pin control

The I²S needs three pins for transfer operatioin, namely, the SD, WS and CK. The MCLK pin is also required if need to provide main clock for peripherals. The I²S shares some pins with the SPI, described as follows:

- SD: Serial data (mapped on the MOSI pin) for bidirectional data transmission and reception.
- WS: Word select (mapped on the CS pin) for data control signal output in master mode, and input in slave mode.
- CK: Communication clock (mapped on the SCK pin) as clock signal output in master mode,and input in slave mode.
- MCLK: Master clock (mapped independently) is used to provide main clock for peripherals. The frequency of output clock signal is set to 256x Fs (audio sampling frequency)

SPI registers

These peripheral registers must be accessed by half-word (16 bits) or word (32 bits).

Table 44. SPI register map and reset value

Register	Offset	Reset value
SPI_CTRL1	0x00	0x0000
SPI_CTRL2	0x04	0x0000
SPI_STS	0x08	0x0002
SPI_DT	0x0C	0x0000
SPI_CPOLY	0x10	0x0007
SPI_RCRC	0x14	0x0000
SPI_TCRC	0x18	0x0000
SPI_I2SCTRL	0x1C	0x0000
SPI_I2SCLKP	0x20	0x0002

SPI control register1 (SPI_CTRL1) (Not used in I²S mode)

Bit	Register	Reset value	Type	Description
Bit 15	SLBEN	0x0	rw	Single line bidirectional half-duplex enable 0: Disabled 1: Enabled
Bit 14	SLBTD	0x0	rw	Single line bidirectional half-duplex transmission direction This bit and the SLBEN bit together determine the data output direction in “Single line bidirectional half-duplex” mode. 0: Receive-only mode 1: Transmit-only mode
Bit 13	CCEN	0x0	rw	RC calculation enable 0: Disabled 1: Enabled
Bit 12	NTC	0x0	rw	Transmit CRC next When this bit is set, it indicates that the next data transferred is CRC value. 0: Next transmitted data is the normal value 1: Next transmitted data is CRC value
Bit 11	FBN	0x0	rw	Frame bit num This bit is used to configure the number of data frame bit for transmission/reception. 0: 8-bit data frame 1: 16-bit data frame
Bit 10	ORA	0x0	rw	Receive-only active In two-wire unidirectional mode, when this bit is set, it indicates that Receive-only is active, but the transmit is not allowed. 0: Transmission and reception 1: Receive-only mode
Bit 9	SWCSEN	0x0	rw	Software CS enable When this bit is set, the CS pin level is determined by the SWCSIL bit. The status of I/O level on the CK pin is invalid. 0: Disabled 1: Enabled
Bit 8	SWCSIL	0x0	rw	Software CS internal level This bit is valid only when the SWCSEN is set. It determines the level on the CS pin. In master mode, this bit must be set. 0: Low level 1: High level
Bit 7	LTF	0x0	rw	LSB transmit first This bit is used to select for MST transfer first or LSB transfer first. 0: MSB 1: LSB
Bit 6	SPIEN	0x0	rw	SPI enable 0: Disabled 1: Enabled

Bit	Register	Reset value	Type	Description
Bit 5:3	MDIV	0x0	rw	Master clock frequency division In master mode, the peripheral clock divided by the prescaler is used as SPI clock. The MDIV[3] bit is in the SPI_CTRL2 register, MDIV[3:0]: 0000: Divided by 2 0001: Divided by 4 0010: Divided by 8 0011: Divided by 16 0100: Divided by 32 0101: Divided by 64 0110: Divided by 128 0111: Divided by 256 1000: Divided by 512 1001: Divided by 1024
Bit 2	MSTEN	0x0	rw	Master enable 0: Disabled (Slave) 1: Enabled (Master)
Bit 1	CLKPOL	0x0	rw	Clock polarity Indicates the polarity of clock output in idle state. 0: Low level 1: High level
Bit 0	CLKPHA	0x0	rw	Clock phase 0: Data capture starts from the first clock edge 1: Data capture starts from the second clock edge

Note: The SPI_CTRL1 register must be 0 in I^2S mode.

SPI control register2 (SPI_CTRL2)

Bit	Register	Reset value	Type	Description
Bit 15:9	Reserved	0x00	resd	Forced to be 0 by hardware.
Bit 8	MDIV	0x0	rw	Master clock frequency division Refer to the MDIV[2:0] of the SPI_CTRL1 register.
Bit 7	TDBEIE	0x0	rw	Transmit data buffer empty interrupt enable 0: Disabled 1: Enabled
Bit 6	RDBFIE	0x0	rw	Receive data buffer full interrupt enable 0: Disabled 1: Enabled
Bit 5	ERRIE	0x0	rw	Error interrupt enable This bit controls interrupt generation when errors occur (CCERR, MMERR, ROERR and TUERR) 0: Disabled 1: Enabled
Bit 4:3	Reserved	0x0	resd	Kept at its default value
Bit 2	HWCSOE	0x0	rw	Hardware CS output enable This bit is valid only in master mode. When this bit is set, the I/O output on the CS pin is low; when this bit is 0, the I/O input on the CS pin must be set high. 0: Disabled 1: Enabled

Bit	Register	Reset value	Type	Description
Bit 1	DMATEN	0x0	rw	DMA transmit enable 0: Disabled 1: Enabled
Bit 0	DMAREN	0x0	rw	DMA receive enable 0: Disabled 1: Enabled

SPI status register (SPI_STS)

Bit	Register	Reset value	Type	Description
Bit 15:8	Reserved	0x00	resd	Forced to be 0 by hardware
Bit 7	BF	0x0	ro	Busy flag 0: SPI is not busy 1: SPI is busy
Bit 6	ROERR	0x0	ro	Receiver overflow error 0: No overflow error 1: Overflow error occurs.
Bit 5	MMERR	0x0	ro	Master mode error This bit is set by hardware and cleared by software (read/write access to the SPI_STS register, followed by write operation to the SPI_CTRL1 regitser) 0: No mode error 1: Mode error occurs
Bit 4	CCERR	0x0	rw0c	CRC error Set by hardware, and cleared by software. 0: No CRC error 1: CRC error occurs.
Bit 3	TUERR	0x0	ro	Transmitter underload error Set by hardware, and cleared by software (read the SPI_STS register). 0: No underload error 1: Underload error occurs. Note: This bit is only used in I^2S mode.
Bit 2	ACS	0x0	ro	Audio channel state This bit indicates the status of the current audio channel. 0: Left channel 1: Right channel Note: This bit is only used in I^2S mode.
Bit 1	TDBE	0x1	ro	Transmit data buffer empty 0: Transmit data buffer is not empty. 1: Transmit data buffer is empty.
Bit 0	RDBF	0x0	ro	Receive data buffer full 0: Transmit data buffer is not full. 1: Transmit data buffer is full.

SPI data register (SPI_DT)

Bit	Register	Reset value	Type	Description
Bit 15:0	DT	0x0000	rw	Data value This register controls read and write operations. When the data bit is set as 8 bit, only the 8-bit LSB[7:0] is valid.

SPICRC register (SPI_CPOLY) (Not used in I²S mode)

Bit	Register	Reset value	Type	Description
Bit 15:0	CPOLY	0x0007	rw	CRC polynomial This register contains the polynomial used for CRC calculation. Note: This register is valid only in SPI mode.

SPIRxCRC register (SPI_RCRC) (Not used in I²S mode)

Bit	Register	Reset value	Type	Description
Bit 15:0	RCRC	0x0000	ro	Receive CRC When CRC calculation is enabled, this register contains the CRC value computed based on the received data. This register is reset when the CCEN bit in the SPI_CTRL1 register is cleared. When the data frame format is set to 8-bit data, only the 8-bit LSB ([7:0]) are calculated based on CRC8 standard; when 16-bit data bit is selected, follow CRC16 standard. Note: This register is only used in SPI mode.

SPITxCRC register (SPI_TCRC)

Bit	Register	Reset value	Type	Description
Bit 15:0	TCRC	0x0000	ro	Transmit CRC When CRC calculation is enabled, this register contains the CRC value computed based on the transmitted data. This register is reset when the CCEN bit in the SPI_CTRL1 register is cleared. When the data frame format is set to 8-bit data, only the 8-bit LSB ([7:0]) are calculated based on CRC8 standard; when 16-bit data bit is selected, follow CRC16 standard. Note: This register is only used in SPI mode.

SPI_I2S register (SPI_I2SCTRL)

Bit	Register	Reset value	Type	Description
Bit 15:12	Reserved	0x0	resd	Forced to be 0 by hardware.
Bit 11	I2SMSEL	0x0	rw	I^2S mode select 0: SPI mode 1: I^2S mode
Bit 10	I2SEN	0x0	rw	I^2S enable 0: Disabled 1: Enabled
Bit 9:8	OPERSEL	0x0	rw	I^2S operation mode select 00: Slave transmission 01: Slave reception 10: Master transmission 11: Master reception
Bit 7	PCMFSSEL	0x0	rw	PCM frame synchronization This bit is valid only when the PCM standard is used. 0: Short frame synchronization 1: Long frame synchronization
Bit 6	Reserved	0x0	resd	Kept at its default value
Bit 5:4	STDSEL	0x0	rw	I^2S standard select 00: Philips standard 01: MSB-aligned standard (left-aligned) 10: LSB-aligned standard (right-aligned) 11: PCM standard
Bit 3	I2SCLKPOL	0x0	rw	I^2S clock polarity This bit indicates the clock polarity on the clock pin in idle state. 0: Low 1: High
Bit 2:1	I2SDBN	0x0	rw	I^2S data bit num 00: 16-bit data length 01: 24-bit data length 10: 32-bit data length 11: Not allowed.
Bit 0	I2SCBN	0x0	rw	I^2S channel bit num This bit can be configured only when the I^2S is set to 16-bit data; otherwise, it is fixed to 32-bit by hardware. 0: 16-bit wide 1: 32-bit wide

SPI_I2S prescaler register (SPI_I2SCLKP)

Bit	Register	Reset value	Type	Description
Bit 15:12	Reserved	0x0	resd	Forced to be 0
Bit 9	I2SMCLKOE	0x0	rw	I^2S Master clock output enable 0: Disabled 1: Enabled
Bit 8	I2SODD	0x0	rw	IOdd factor for I^2S division 0: Actual divider factor = I2SDIV × 2 1: Actual divider factor = (I2SDIV × 2) + 1
Bit 11:10 Bit 7:0	I2SDIV	0x02	rw	I^2S division It is not allowed to configure I2SDIV[9:0]=0 or I2SDIV[9:0]=1

14 Timer

HT32F49365/HT32F49395 timers include basic timers, general-purpose timers, and advanced-control timers.

Please refer to Section 14.1 ~ Section 14.4 for the detailed function modes. All functions of different timers are shown in the following tables.

Table 45. TMR functional comparison

Timer type	Timer	Counter bit	Count mode	Repetition	Prescaler	DMA requests	Capture/ compare channel	PWM input mode	EXT input	Break input
Advanced-control timer	TMR1 TMR8	16	Up Down Up/Down	8-bit	1~65535	O	4	O	O	O
General-purpose timer	TMR2 TMR5	16/32	Up Down Up/Down	X	1~65535	O	4	O	O	X
	TMR3 TMR4	16	Up Down Up/Down	X	1~65535	O	4	O	O	X
	TMR9 TMR12	16	Up	X	1~65535	X	2	O	X	X
	TMR10 TMR11 TMR13 TMR14	16	Up	X	1~65535	X	1	X	X	X
Basic timer	TMR6 TMR7	16	Up	X	1~65535	O	X	X	X	X

Timer type	Timer	Counter bit	Count mode	PWM output	Single pulse output	Complementary output	Dead-time	Encoder interface connection	Interfacing with hall sensors	Linkage peripheral
Advanced-control timer	TMR1 TMR8	16	Up Down Up/Down	O	O	O	O	O	O	Timer synchronization
General-purpose timer	TMR2 TMR5	16/32	Up Down Up/Down	O	O	X	X	O	O	Timer synchronization
	TMR3 TMR4	16	Up Down Up/Down	O	O	X	X	O	O	Timer synchronization
	TMR9 TMR12	16	Up	O	O	X	X	X	X	Timer synchronization ADC/DAC
	TMR10 TMR11 TMR13 TMR14	16	Up	O	O	X	X	X	X	NA
Basic timer	TMR6 TMR7	16	Up	X	X	X	X	X	X	DAC

Basic timer (TMR6 and TMR7)

TMR6 and TMR7 introduction

Each of the basic timers (TMR6 and TMR7) includes a 16-bit up counter and the corresponding control logic. without being connected to external I/Os. They can be used for a basic timing and providing clocks for the digital-to-analog converter (DAC).

TMR6 and TMR7 main features

- Internal clock used as counter clock
- 16-bit up counter
- Synchronization circuit to trigger DAC (Unique characteristics)
- Interrrupt on overflow event and DMA request

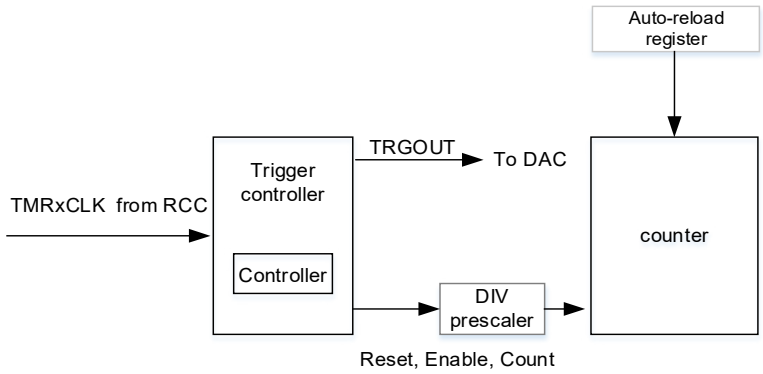


Figure 55. Basic timer block diagram

TMR6 and TMR7 functional descriptions

Count clock

The counter clock of TMR6 and TMR7 is provided by the internal clock source (CK_INT) divided by prescaler.

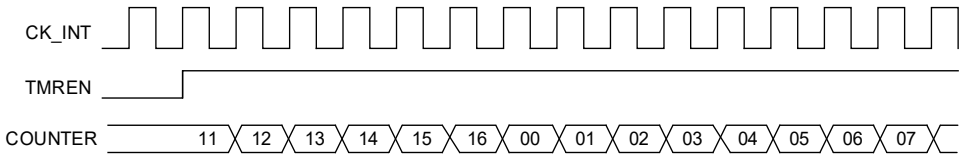


Figure 56. Control circuit with CK_INT divided by 1

Counting mode

The basic timer only supports upcounting mode. It has an internal 16-bit counter in which the value is loaded with the TMRx_PR register. The value in the TMRx_PR is immediately moved to the shadow register by deault. When the periodic buffer is enabled (PRBEN=1), the value in the TMRx_PR register is transferred to the shadow register only at an overflow event. The OVFEN and OVFS bits are used to configure the overflow event.

Settng the TMREN bit (TMREN=1) enables the timer to start counting. Base on synchronization logic, however, the actual enable signal TMR_EN is set 1 clock cycle after the TMREN is set.

In upcounting mode, the counter counts from 0 to the value programmed in the TMRx_PR register, then restarts from 0, and generates a counter overflow event, with the OVFIF bit being set to 1. If the overflow event is disabled, the counter is no longer reloaded with the preload value and period value at a counter overflow event, otherwise, the counter is updated with the preload value and period value on an overflow event.

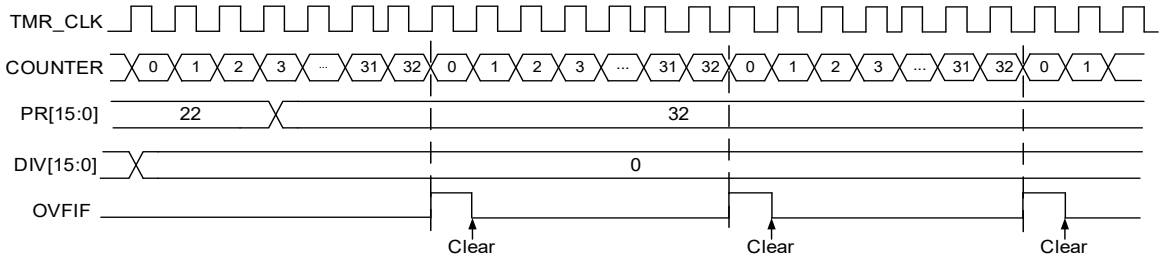


Figure 57. Overflow event when PRBEN=0

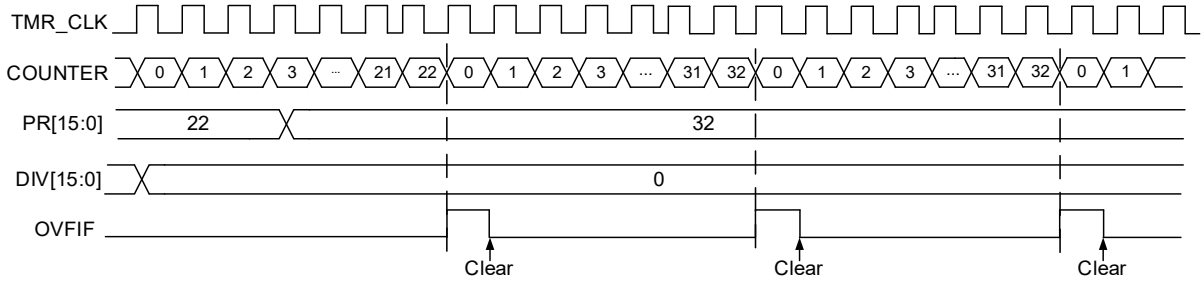


Figure 58. Overflow event when PRBEN=1

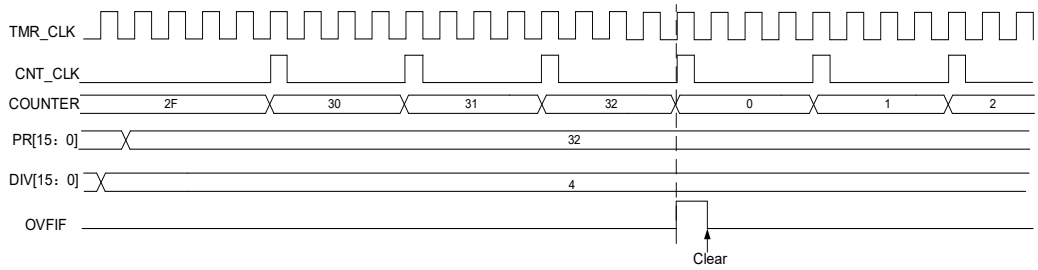


Figure 59. Counting timing diagram when the prescaler division is 4

Debug mode

When the microcontroller enters debug mode (Cortex®-M4F core halted), the TMRx counter stops counting when the TMRx_PAUSE bit is set.

TMR6 and TMR7 registers

These peripheral registers must be accessed by word (32 bits). In the following table, all the TMRx registers are mapped to a 16-bit addressable space.

Table 46. TMR6 and TMR7 – register table and reset value

Register	Offset	Reset value
TMRx_CTRL1	0x00	0x0000
TMRx_CTRL2	0x04	0x0000
TMRx_IDEN	0x0C	0x0000
TMRx_ISTS	0x10	0x0000
TMRx_SWEVT	0x14	0x0000
TMRx_CVAL	0x24	0x0000
TMRx_DIV	0x28	0x0000
TMRx_PR	0x2C	0x0000

TMR6 and TMR7 control register1 (TMRx_CTRL1)

Bit	Register	Reset value	Type	Description
Bit 15:8	Reserved	0x00	resd	Kept at its default value.
Bit 7	PRBEN	0x0	rw	Period buffer enable 0: Period buffer is disabled. 1: Period buffer is enabled.
Bit 6:4	Reserved	0x0	resd	Kept at its default value.
Bit 3	OCMEN	0x0	rw	One cycle mode enable This bit is used to select whether to stop the counter at update event. 0: Disabled 1: Enabled
Bit 2	OVFS	0x0	rw	Overflow event source This bit is used to configure overflow event or DMA request sources. 0: Counter overflow, setting the OVFSWTR bit or overflow event generated from the slave controller 1: Only counter overflow generates an overflow event.
Bit 1	OVFEN	0x0	rw	Overflow event enable This bit is used to enable or disable OEV event generation. 0: OEV event is enabled. An overflow event is generated by any of the following events: - Counter overflow - Setting the OVFSWTR bit - Overflow event generated from the slave controller 1: OEV event is disabled. If the OVFSWTR bit is set, or a hardware reset is generated from the slave controller, the counter and the prescaler are reinitialized. Note: This bit is set and cleared by software.
Bit 0	TMREN	0x0	rw	TMR enable 0: Disabled 1: Enabled

TMR6 and TMR7 control register2 (TMRx_CTRL2)

Bit	Register	Reset value	Type	Description
Bit 15:7	Reserved	0x000	resd	Kept at its default value.
Bit 6:4	PTOS	0x0	rw	Master TMR output selection This field is used to select the signals in master mode to be sent to slave timers. 000: Reset 001: Enable 010: Update
Bit 3:0	Reserved	0x0	resd	Kept at its default value.

TMR6 and TMR7 DMA/interrupt enable register (TMRx_IDEN)

Bit	Register	Reset value	Type	Description
Bit 15:9	Reserved	0x00	resd	Kept at its default value.
Bit 8	OVFDEN	0x0	rw	Overflow event DMA request enable 0: Disabled 1: Enabled
Bit 7:1	Reserved	0x00	resd	Kept at its default value.
Bit 0	OVFIEN	0x0	rw	Overflow interrupt enable 0: Disabled 1: Enabled

TMR6 and TMR7 interrupt status register (TMRx_ISTS)

Bit	Register	Reset value	Type	Description
Bit 15:1	Reserved	0x0000	resd	Kept at its default value.
Bit 0	OVFIF	0x0	rw0c	Overflow interrupt flag This bit is set by hardware at an update event. It is cleared by software. 0: No update event occurs. 1: Update event occurs, and OVFEN=0, and OVFS=0 in the TMRx_CTRL1 register: - An update event occurs when OVFG=1 in the TMRx_SWEVE register - An update event occurs when the counter value (CVAL) is reinitialized by a trigger event.

TMR6 and TMR7 software event register (TMRx_SWEVT)

Bit	Register	Reset value	Type	Description
Bit 15:1	Reserved	0x0000	resd	Kept at its default value.
Bit 0	OVFSWTR	0x0	rw0c	Overflow event triggered by software An overflow event is trigged by software. 0: No effect 1: Generate an overflow event by software write operation.

TMR6 and TMR7 counter value (TMRx_CVAL)

Bit	Register	Reset value	Type	Description
Bit 15:0	CVAL	0x0000	rw	Counter value

TMR6 and TMR7 division (TMRx_DIV)

Bit	Register	Reset value	Type	Description
Bit 15:0	DIV	0x0000	rw	Divider value The counter clock frequency $f_{CK_CNT} = f_{TMR_CLK}$ / (DIV[15:0]+1). At each overflow event, DIV value is sent to the DIV register.

TMR6 and TMR7 period register (TMRx_PR)

Bit	Register	Reset value	Type	Description
Bit 15:0	PR	0x0000	rw	Period value This indicates the period value of the TMRx counter. The timer stops working when the period value is 0.

General-purpose timer (TMR2 to TMR5)

TMRx introduction

The general-purpose timer (TMR2 to TMR5) consists of a 16-bit counter supporting up, down, up/down (bidirectional) counting modes, four capture/compare registers, and four independent channels to achieve input capture and programmable PWM output.

TMRx main features

- Source of count clock is selectable : internal clock, external clock and internal trigger
- 16-bit up, down, up/down and encoder mode counter (TMR2/5 can be extended to 32-bit)
- 4 independent channels for input capture, output compare, PWM generation and one-pulse mode output
- Synchronization control between master and slave timers
- Interrupt/DMA is generated at overflow event, trigger event and channel event
- Support TMR burst DMA transfer

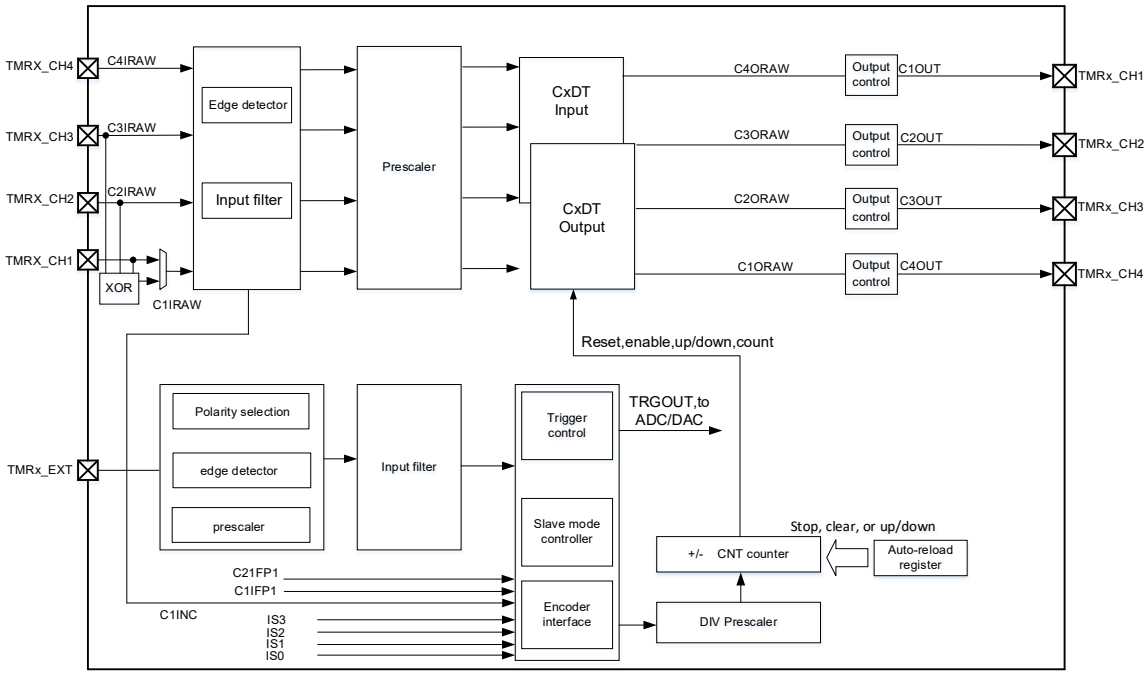


Figure 60. General-purpose timer block diagram

TMRx functional descriptions

Count clock

The count clock of TMR2~TMR5 can be provided by the internal clock (CK_INT), external clock (external clock mode A and B) and internal trigger input (ISx)

Internal clock (CK_INT)

By default, the CK_INT divided by the prescaler is used to drive the counter to start counting.

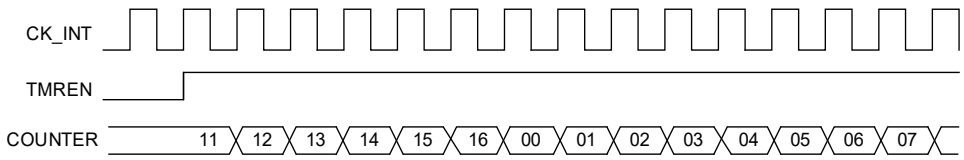


Figure 61. Control circuit with CK_INT divided by 1

External clock (TRGIN/EXT)

The counter clock can be provided by two external clock sources, namely, TRGIN and EXT signals.

When SMSEL=3'111, external clock mode A is selected. Set the STIS[2:0] bit to select TRGIN signal to drive the counter to start counting.

When ECMBEN=1, external clock mode B is selected. The counter starts counting driven by EXT signal.

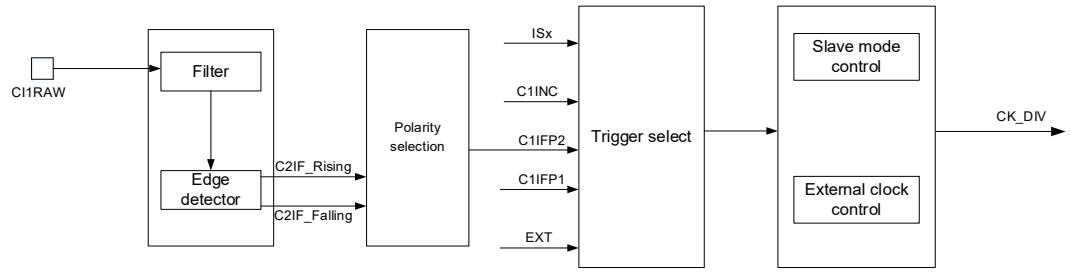


Figure 62. Block diagram of external clock mode A

Note: The delay between the signal on the input side and the actual clock of the counter is due to the synchronization circuit.

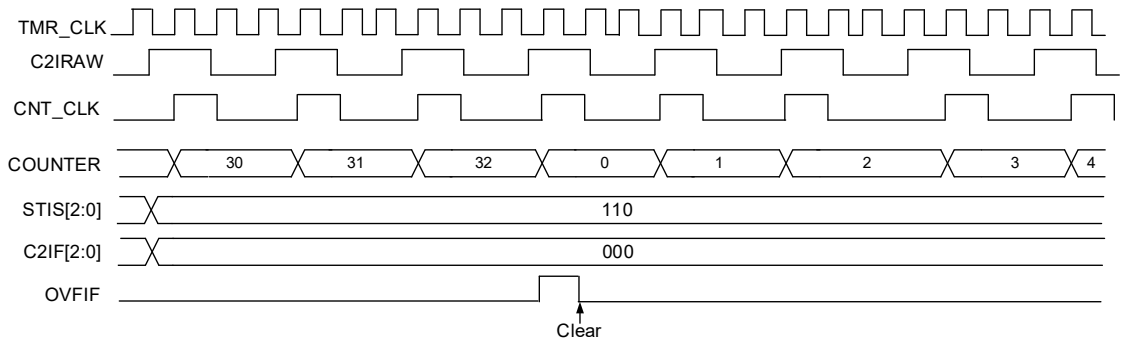


Figure 63. Counting in external clock mode A

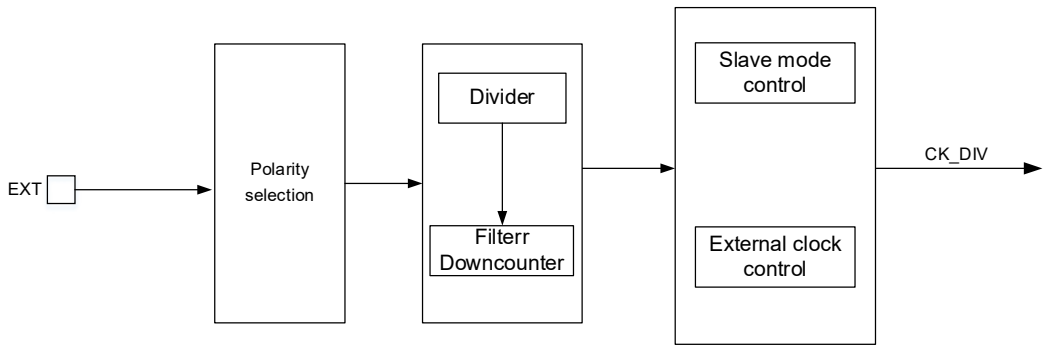


Figure 64. Block diagram of external clock mode B

Note: The delay between the EXT signal on the input side and the actual clock of the counter is due to the synchronization circuit.

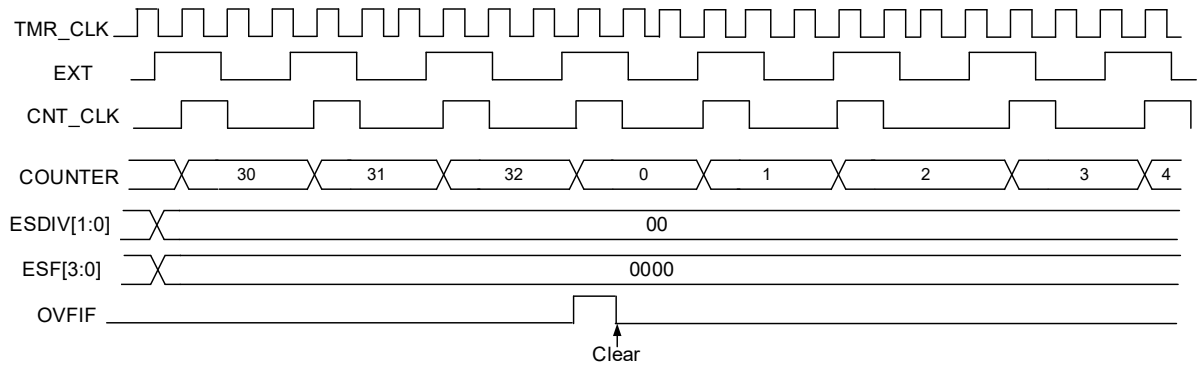


Figure 65. Counting in external clock mode B

Internal trigger input (ISx)

Timer synchronization allows interconnection between several timers. The TMR_CLK of one timer can be provided by the TRGOUT signal output by another timer. Set the STIS[2:0] bit to select internal trigger signal to enable counting.

Each timer (TMR2 to TMR5) consists of a 16-bit prescaler, which is used to generate the CK_CNT that enables the counter to count. The frequency division relationship between the CK_CNT and TMR_CLK can be adjusted by setting the value of the TMRx_DIV register. The prescaler value can be modified at any time, but it takes effect only when the next overflow event occurs.

Table 47. TMRx internal trigger connection

Slave controler	IS0 (STIS = 000)	IS1 (STIS = 001)	IS2 (STIS = 010)	IS3 (STIS = 011)
TMR2	TMR1	TMR8/USB_SOF(2)	TMR3	TMR4
TMR3	TMR1	TMR2	TMR5	TMR4
TMR4	TMR1	TMR2	TMR3	TMR8
TMR5	TMR2	TMR3	TMR4	TMR8

Note: 1. If there is no corresponding timer in a device, the corresponding trigger signal ISx is not present.
2. TMR8 or USB_SOF is available for IS1 to select, determined by the TMR2IS1_IRMP bit of the IOMUX_MAP4 register.

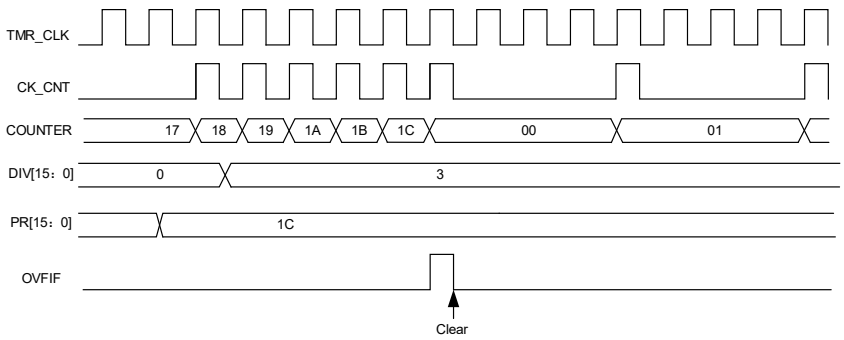


Figure 66. Counter timing with prescaler value changing from 1 to 4

Counting mode

The timer (TMR2 to TMR5) supports several counting modes to meet different application scenarios. Each timer has an internal 16-bit up, down, up/down counter. TMR2/5 can be extended to 32-bit by setting the PMEN bit. The TMRx_PR register is loaded with the counter value. The value in the TMRx_PR is immediately moved to the shadow register by deault. When the periodic buffer is enabled (PRBEN=1), the value in the TMRx_PR register is transferred to the shadow register only at an overflow event. The OVFEN and OVFS bits are used to configure the overflow event.

Settng the TMREN bit (TMREN=1) enables the timer to start counting. Base on synchronization logic, however, the actual enable signal TMR_EN is set 1 clock cycle after the TMREN is set.

Upcounting mode

In upcounting mode, the counter counts from 0 to the value programmed in the TMRx_PR register, then restarts from 0, and generates a counter overflow event, with the OVFIF bit being set to 1. If the overflow event is disabled, the counter is no longer reloaded with the preload value and period value at a counter overflow event, otherwise, the counter is updated with the preload value and period value on an overflow event.

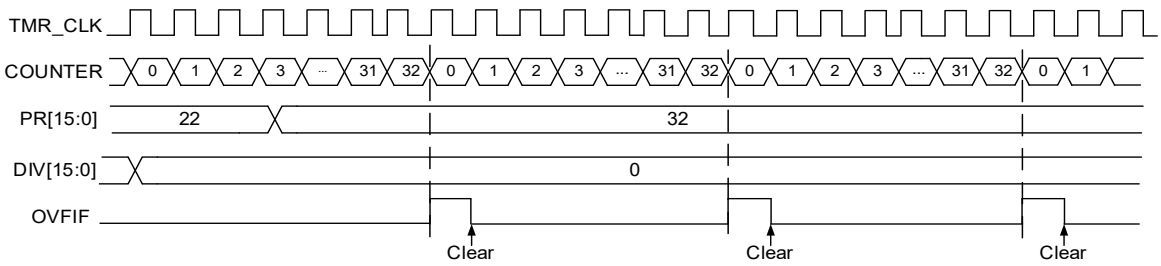


Figure 67. Overflow event when PRBEN=0

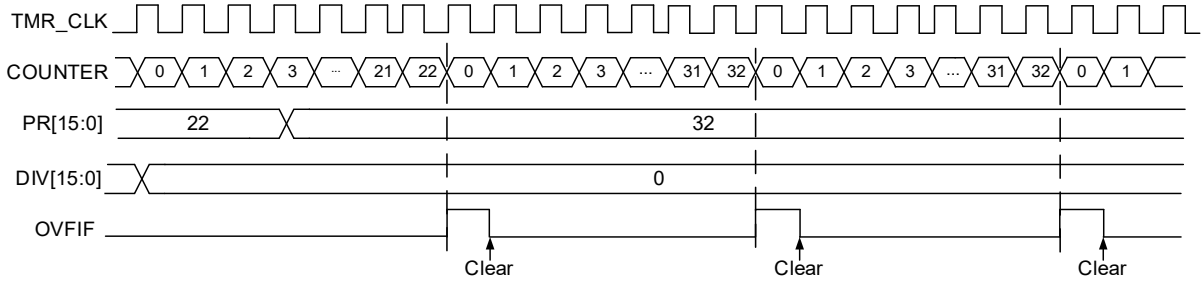


Figure 68. Overflow event when PRBEN=1

Downcounting mode

In downcounting mode, the counter counts from the value programmed in the TMRx_PR register down to 0, and restarts from the value programmed, and generates a counter underflow event.

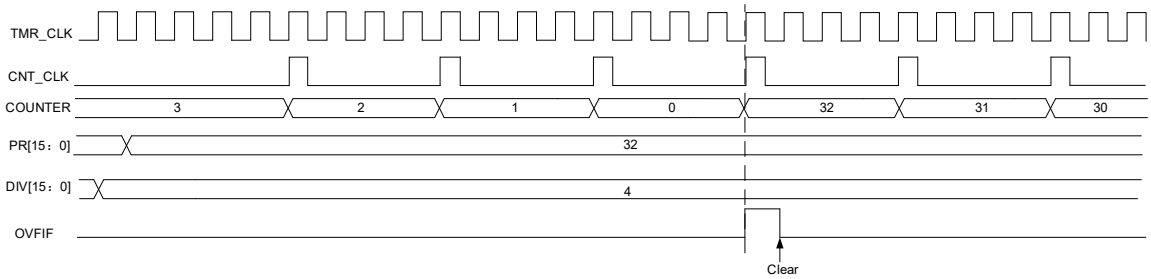


Figure 69. Counter timing diagram with internal clock divided by 4

Up/down counting mode

In up/down counting mode, the counter counts up/down alternatively. When the counter counts from the value programmed in the TMRx_PR register down to 1, an underflow event is generated, and then restarts counting from 0; When the counter counts from 0 to the value of the TMRx_PR register -1, an overflow event is generated, and then restarts counting from the value of the TMRx_PR register. The OWCDIR bit indicates the current counting direction.

Note: The OWCDIR is ready-only in up/down counting mode.

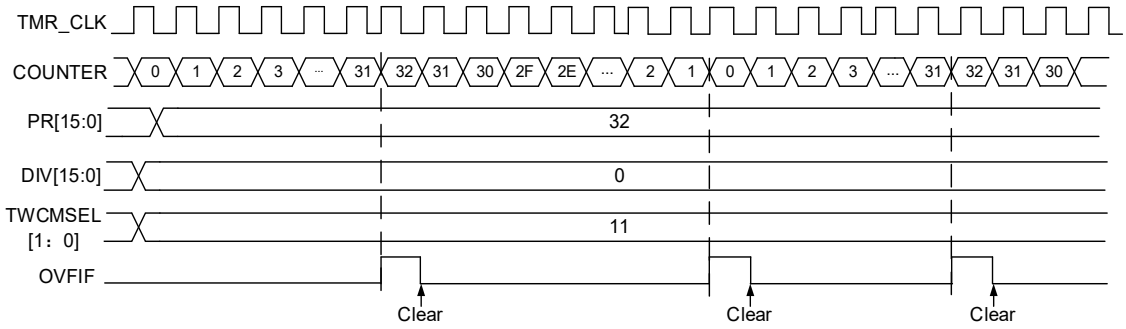


Figure 70. Counter timing diagram with internal clock divided by 1 and TMRx_PR=0x32

Encoder interface mode

To enble the encoder interface mode, write SMSEL[2:0] = 3'b001/3'b010/3'b011. In this mode, the two inputs (C1IN/C2IN) are required. Depending on the level on one input, the counter counts up or down on the edge of the other input. The OWCDIR bit indicates the direction of the counter, as shown in the table below:

Table 48. Counting direction versus encoder signals

Active edge	Level on opposite signal (C1INFP1 to C2IN, C2INFP2 to C1IN)	C1INFP1 signal		C2INFP2 signal	
		Rising	Falling	Rising	Falling
Count on C1IN only	High	Down	Up	No count	No count
	Low	Up	Down	No count	No count
Count on C2IN only	High	No count	No count	Up	Down
	Low	No count	No count	Down	Up
Count on both C1IN and C2IN	High	Down	Up	Up	Down
	Low	Up	Down	Down	Up

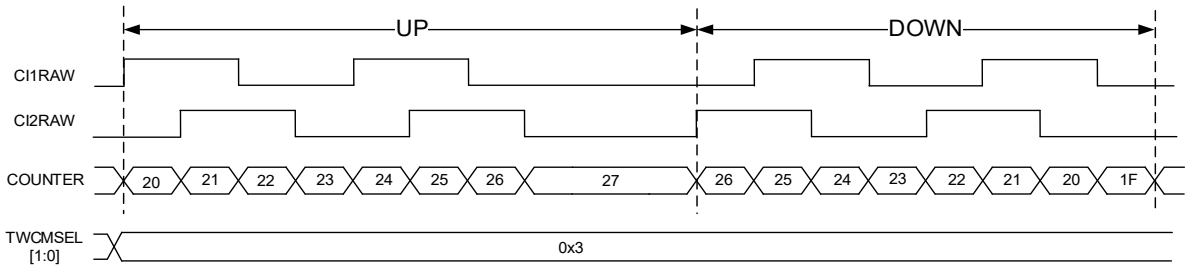


Figure 71. Example of counter behavior in encoder interface mode (encoder mode C)

TMR input function

Each of TMR2~TMR5 timers has four independent channels, with each channel being configured as input or output. As input, the channel can be used for the filtering, selection, division and input capture of the input signals.

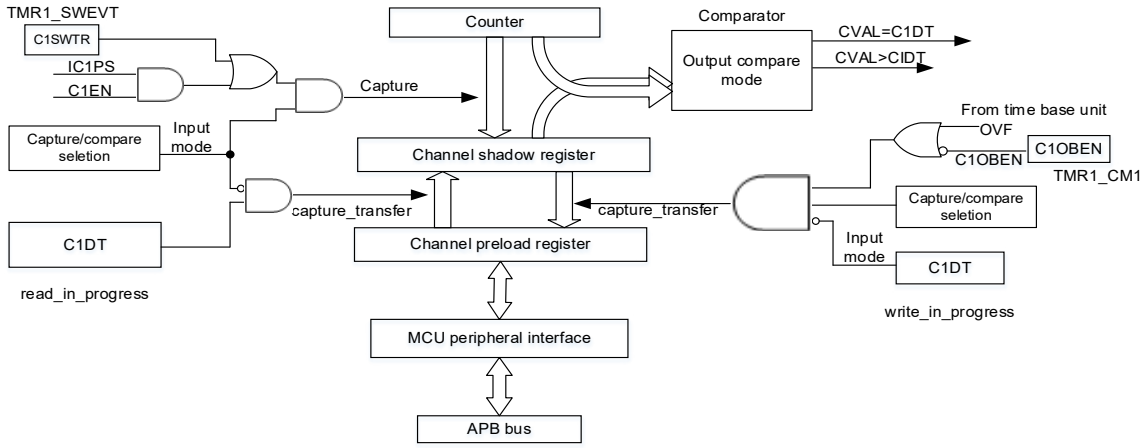


Figure 72. Input/output channel 1 main circuit

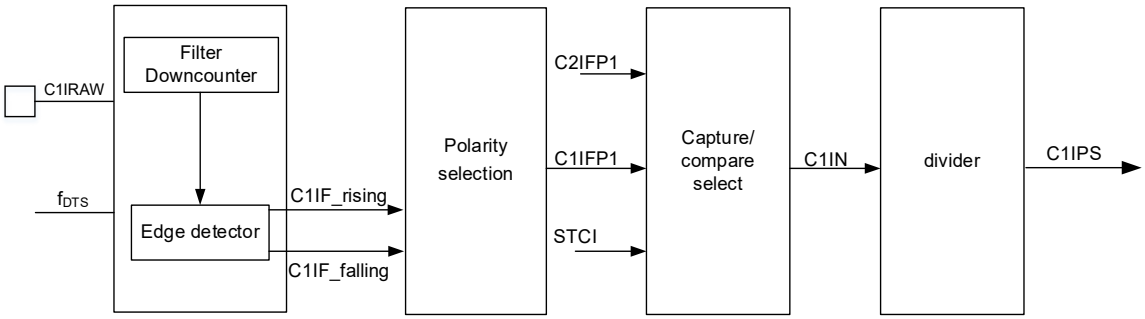


Figure 73. Channel 1 input stage

Input mode

In input mode, the TMRx_CxDT registers latch the current counter values after the selected triggle signal is detected, and the capture compare interrupt flag bit (CxIF) is set. An interrupt or a DMA request will be generated if the CxIEN and CxDEN bits are enabled. If the selected trigger signal is detected when the CxIF is set to 1, the CxRF is set to 1.

To capture the rising edge of C1IN input, following the configuration procedure mentioned below:

- Set C1C=01 in the TMRx_CxDT register to select the C1IN as channel 1 input
- Set the filter bandwidth of C1IN signal (CxDF[3:0])
- Set the active edge on the C1IN channel by writing C1P=0 (rising edge) in the TMRx_CCTR register
- Program the capture frequency of C1IN signal (C1DIV[1:0])
- Enable channel 1 input capture (C1EN=1)
- If needed, enable the relevant interrupt or DMA request by setting the C1IEN bit in the TMRx_IDEN register or the C1DEN bit in the TMRx_IDEN register

Timer Input XOR function

The 3 timer input pins (TMRx_CH1, TMRx_CH2 and TMRx_CH3) are connected to the channel 1 (selected by setting the C1INSE in the TMRx_CTRL2 register) through an XOR gate.

The XOR gate can be used to connect Hall sensors. For example, connect the three XOR inputs to the three Hall sensors respectively so as to calculate the position and speed of the rotation by analyzing three Hall sensor signals.

TMR output function

The TMR output consists of a comparator and an output controller. It is used to program the period, duty cycle and polarity of the output signal.

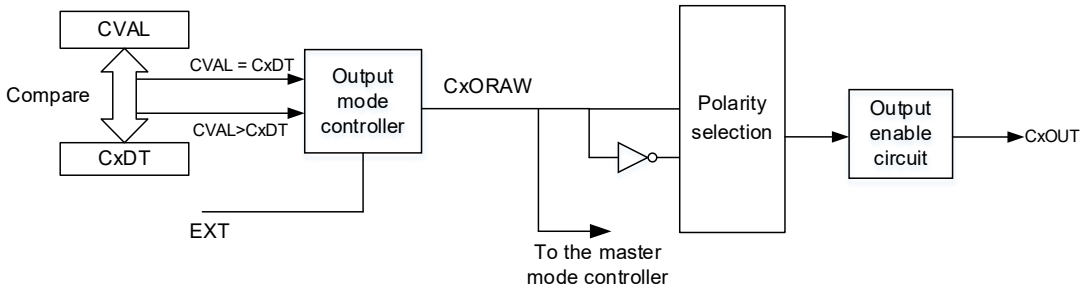


Figure 74. Capture/compare channel output stage (channel 1 to 4)

Output mode

Write CxC[2:0]≠2'b00 to configure the channel as output to implement multiple output modes. In this case, the counter value is compared with the value in the TMRx_CxDT register, and the intermediate signal CxORAW is generated according to the output mode selected by CxOCTRL[2:0], which is sent to IO after being processed by the output control circuit. The period of the output signal is configured by the TMRx_PR register, while the duty cycle by the TMRx_CxDT register.

Output compare modes include:

- PWM mode: Set CxOCTRL=3'b110/111 to enable PWM mode. Each channel can be independently configured to output one PWM signal. In this case, the period of the output signal is configured by the TMRx_PR register, and the duty cycle by the CxDT register. The counter value is compared with the value of the TMRx_CxDT register, and the corresponding level signal is sent according to the counting direction. For more information on PWM mode A/B, refer to the description of the CxOCTRL[2:0] bit. In up/down counting mode, the OWCDIR bit is used to indicate the counting direction.
- Forced output mode: Set CxOCTRL=3'b100/101 to enable forced output mode. In this case, the CxORAW is forced to be the programmed level, irrespective of the counter value. Despite this, the channel flag bit and DMA request still depend on the compare result.
- Output compare mode: Set CxOCTRL=3'b001/010/011 to enable output compare mode. In this
- case, when the counter value matches the value of the CxDT register, the CxORAW is forced high, low or toggling.
- One-pulse mode:This is a particular case of PWM mode. Set OCMEN=1 to enable one-pulse mode. In this mode, the comparison match is performed in the current counting period. The TMREN bit is cleared as soon as the current counting is completed. Therefore, only one pulse is output. When configured as in upcounting mode, the configureation must follow the rule: CVAL<CxDT≤PR; in downcounting mode, CVAL>CxDT is required.
- Fast output mode: Set CxOIEN=1 to enable this mode. If enabled, the CxORAW signal will not change when the counter value matches the CxDT, but at the beginning of the current counting period. In other words, the comparison result is advanced, so the comparison result between the counter value and the TMRx_CxDT register will determine the level of CxORAW in advance.

Figure 75 gives an example of output compare mode (toggle) with C1DT=0x3. When the counter value is equal to 0x3, C1OUT toggles.

Figure 76 gives an example of the combination between upcounting mode and PWM mode A. The output signal behaves when PR=0x32 but CxDT is configured with a different value.

Figure 77 gives an example of the combination between up/down counting mode and PWM mode A. The output signal behaves when PR=0x32 but CxDT is configured with a different value.

Figure 78 gives an example of the combination between upcounting mode and one-pulse PWM mode B. The counter only counts only one cycle, and the output signal sents only one pulse.

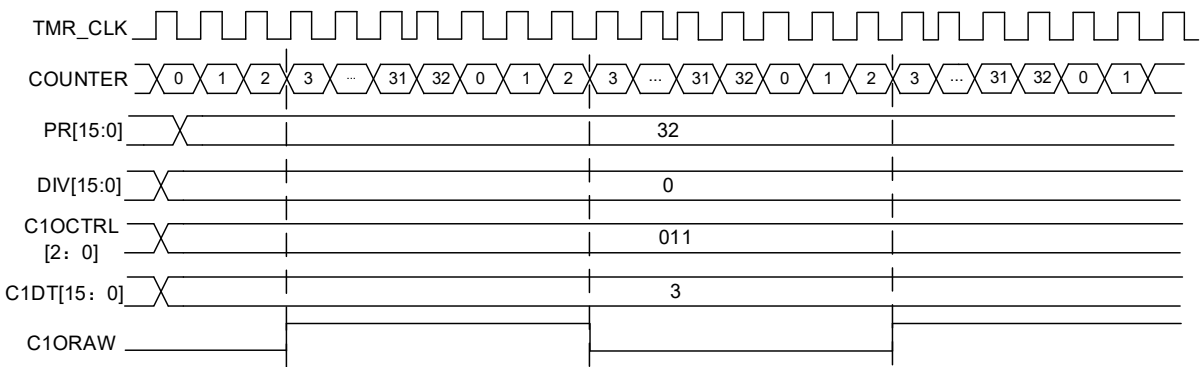


Figure 75. C1ORAW toggles when counter value matches the C1DT value

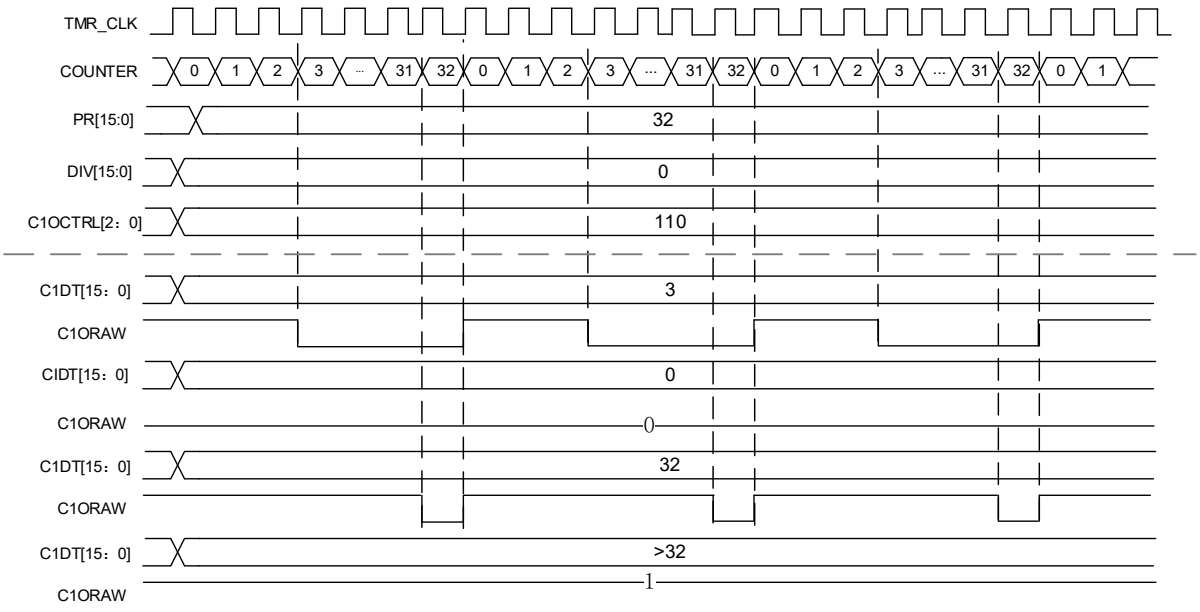


Figure 76. Upcounting mode and PWM mode A

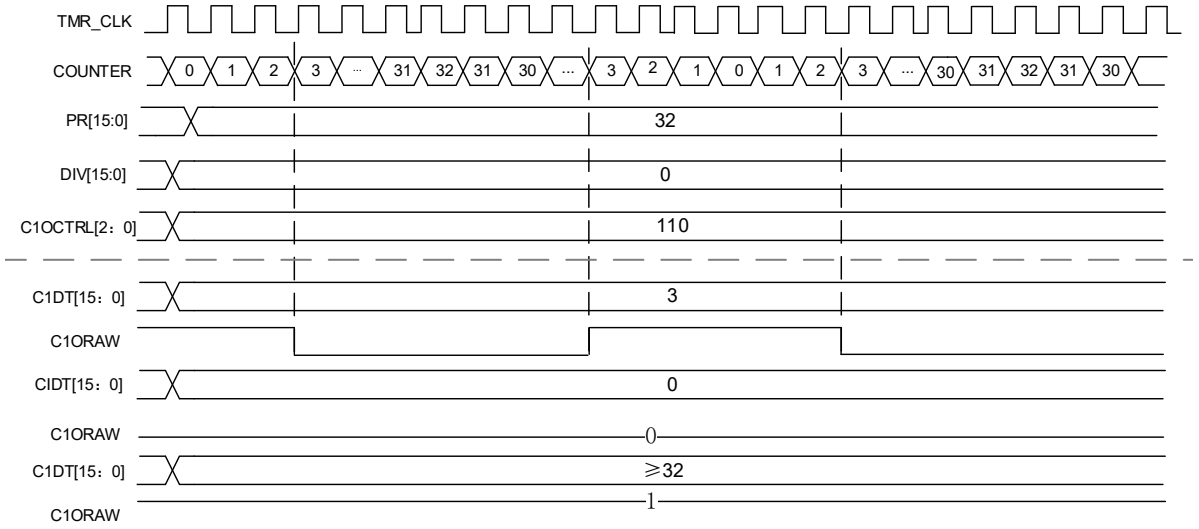


Figure 77. Up/down counting mode and PWM mode A

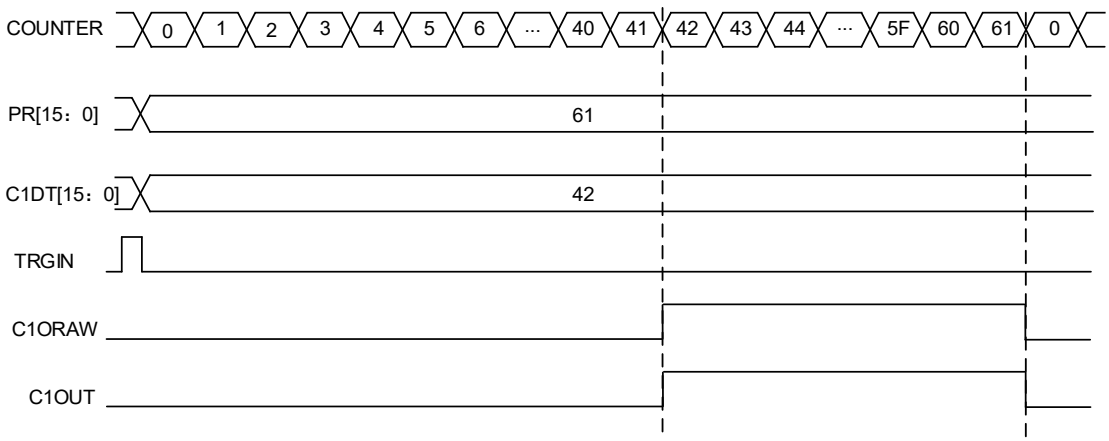


Figure 78. One-pulse mode

CxORAW clear

When the CxOSEN bit is set, the CxORAW signal for a given channel is cleared by applying a high level to the EXT input. The CxORAW signal remains unchanged until the next overflow event.

This function can only be used in output capture or PWM modes, and does not work in forced mode. The following diagram shows the example of clearing CxORAW signal. When the EXT input is high, the CxORAW signal, which was originally high, is driven low; when the EXT is low, the CxORAW signal outputs the corresponding level according to the comparison result between the counter value and CxDT value.

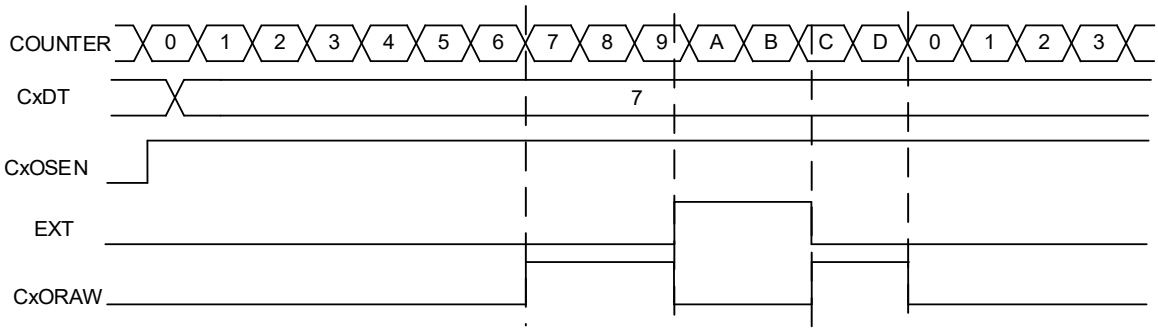


Figure 79. Clearing CxORAW(PWM mode A) by EXT input

TMR synchronization

The timers are linked together internnaly for timer synchronization. Master timer is selected by setting the PTOS[2:0] bit; Slave timer is selected by setting the SMSEL[2:0] bit.

Slave mode include:

Slave mode: Reset mode

The counter and its prescaler can be reset by a selected trigger signal. An overflow event is generated when OVFS=0.

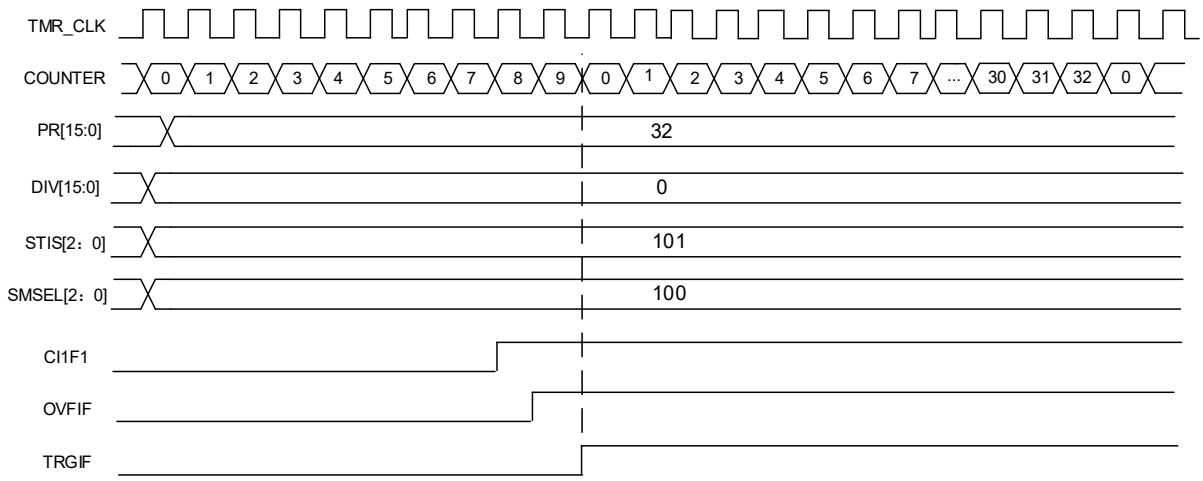


Figure 80. Example of reset mode

Slave mode: Suspend mode

In this mode, the counter is controlled by a selected trigger input. The counter starts counting when the trigger input is high and stops as soon as the trigger input is low.

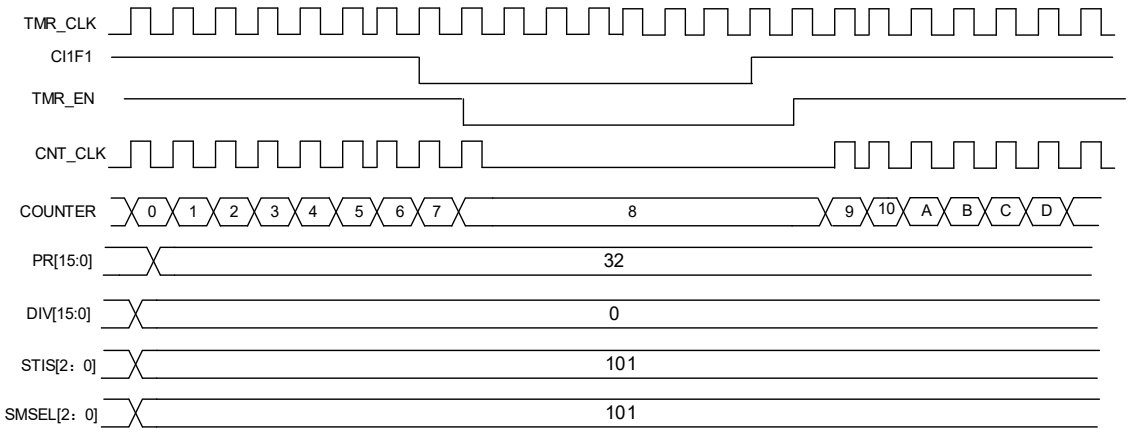


Figure 81. Example of suspend mode

Slave mode: Trigger mode

The counter can start counting on the rising edge of a selected trigger input (TMR_EN=1)

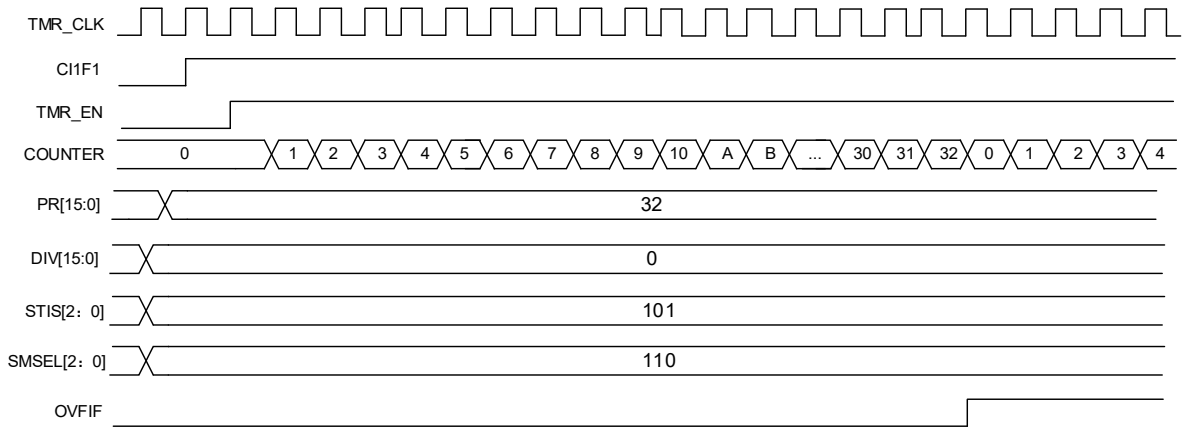


Figure 82. Example of trigger mode

Master/slave timer interconnection

Both Master and slave timer can be configured in different master and slave modes respectively. The combination of both them can be used for various purposes. The following diagram provides an example of interconnection between master timer and slave timer.

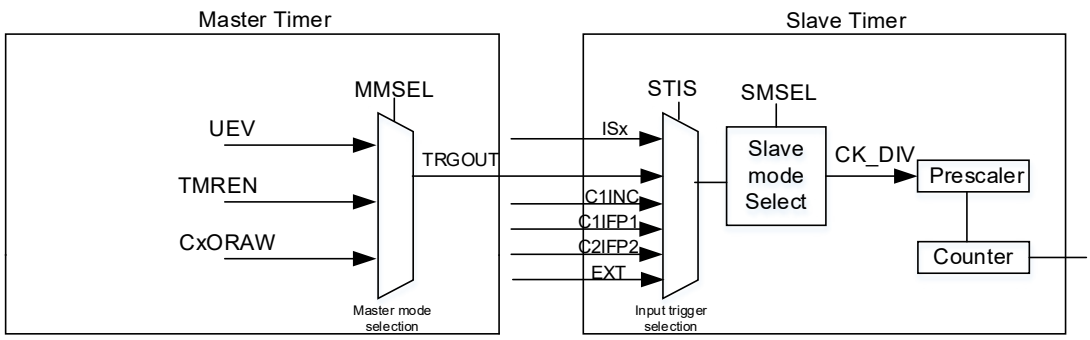


Figure 83. Master/slave timer connection

Using master timer to clock the slave timer:

- Configure master timer output signal TRGOUT as an overflow event (PTOS[2:0]=3’b010). The master timer outputs a pulse signal at each counter overflow event, which is used as the counting clock of the slave timer.
- Configure the master timer counting period (TMRx_PR registers)
- Configure the slave timer trigger input signal TRGIN as master timer output (STIS[2:0] in the TMRx_STCTRL register)
- Configure the slave timer to use external clock mode A (SMSEL[2:0]=3’b111 in the TMRx_STCTRL register)
- Set TMREN =1 in both master timer and slave timer to enable them

Using master timer to start slave timer:

- Configure master timer output signal TRGOUT as an overflow event (PTOS[2:0]=3'b010). The master timer outputs a pulse signal at each counter overflow event, which is used as the counting clock of the slave timer.
- Configure master timer counting period (TMRx_PR registers)
- Configure slave timer trigger input signal TRGIN as master timer input
- Configure slave timer as trigger mode (SMSEL=3'b110 in the TMR2_STCTRL register)
- Set TMREN=1 to enable master timer.

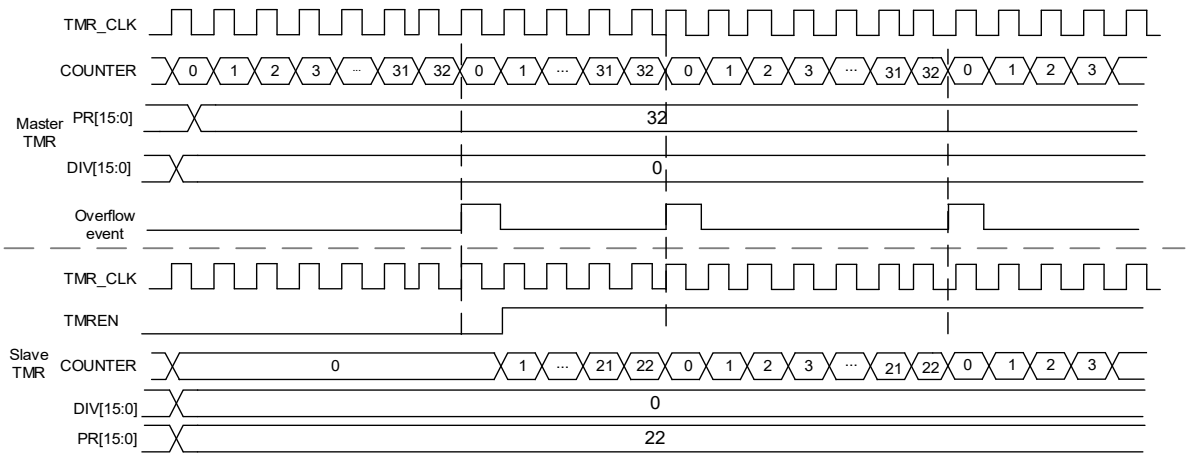


Figure 84. Using master timer to start slave timer

Starting master and slave timers synchronously by an external trigger:

In this example, configure the master timer as master/slave mode synchronously and enable its slave timer synchronization function. This mode is used for synchronization between master timer and slave timer.

- Set the STS bit of the master timer.
- Configure master timer output signal TRGOUT as an overflow event (PTOS[2:0]=3'b010). The master timer outputs a pulse signal at each counter overflow event, which is used as the counting clock of the slave timer.
- Configure the slave timer mode of the master timer as trigger mode, and select C1IN as trigger source
- Configure slave timer trigger input signal TRGIN as master timer output
- Configure slave timer as trigger mode (SMSEL=3'b110 in the TMR2_STCTRL register)

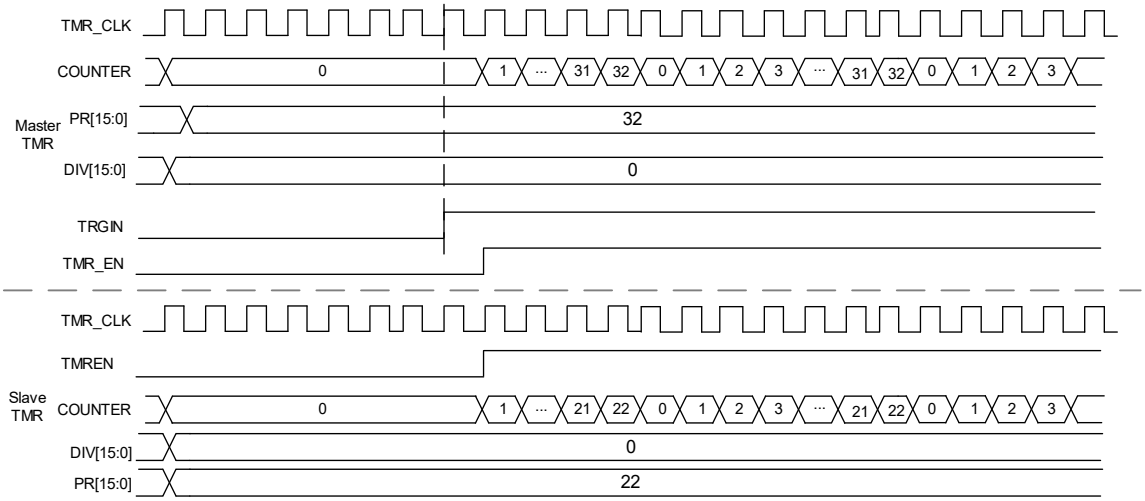


Figure 85. Starting master and slave timers synchronously by an external trigger

Debug mode

When the microcontroller enters debug mode (Cortex™-M4F core halted), the TMRx counter stops counting by setting the TMRx_PAUSE in the DEBUG module.

TMRx registers

These peripheral registers must be accessed by word (32 bits). All TMRx register are mapped into a 16-bit addressable space.

Table 49. TMRx register map and reset value

Register	Offset	Reset value
TMRx_CTRL1	0x00	0x0000
TMRx_CTRL2	0x04	0x0000
TMRx_STCTRL	0x08	0x0000
TMRx_IDEN	0x0C	0x0000
TMRx_ISTS	0x10	0x0000
TMRx_SWEVT	0x14	0x0000
TMRx_CM1	0x18	0x0000
TMRx_CM2	0x1C	0x0000
TMRx_CCTRL	0x20	0x0000
TMRx_CVAL	0x24	0x0000 0000
TMRx_DIV	0x28	0x0000
TMRx_PR	0x2C	0x0000 0000
TMRx_C1DT	0x34	0x0000 0000
TMRx_C2DT	0x38	0x0000 0000
TMRx_C3DT	0x3C	0x0000 0000
TMRx_C4DT	0x40	0x0000 0000
TMRx_DMACTRL	0x48	0x0000
TMRx_DMADT	0x4C	0x0000

Control register1 (TMRx_CTRL1)

Bit	Register	Reset value	Type	Description
Bit 15:11	Reserved	0x00	resd	Kept at its default value.
Bit 10	PMEN	0x0	rw	Plus Mode Enable This bit is used to enable TMRx plus mode. In this mode, TMRx_CVAL, TMRx_PR and TMRx_CxDT are extended from 16-bit to 32-bit. 0: Disabled 1: Enabled Note: This function is only valid for TMR2 and TMR5. It is not applicable to other TMRs.
Bit 9:8	CLKDIV	0x0	rw	Clock division 00: Normal 01: Divided by 2 10: Divided by 4 11: Reserved
Bit 7	PRBEN	0x0	rw	Period buffer enable 0: Period buffer is disabled 1: Period buffer is enabled
Bit 6:5	TWCMSEL	0x0	rw	Two-way counting mode selection 00: One-way counting mode, depending on the OWCDIR bit 01: Two-way counting mode1, count up and down alternately, the output flag bit is set only when the counter counts down 10: Two-way counting mode2, count up and down alternately, the output flag bit is set only when the counter counts up 11: Two-way counting mode3, count up and down alternately, the output flag bit is set when the counter counts up / down
Bit 4	OWCDIR	0x0	rw	One-way count direction 0: Up 1: Down
Bit 3	OCMEN	0x0	rw	One cycle mode enable This bit is use to select whether to stop counting at an update event 0: The counter does not stop at an update event 1: The counter stops at an update event
Bit 2	OVFS	0x0	rw	Overflow event source This bit is used to select overflow event or DMA request sources. 0: Counter overflow, setting the OVFSWTR bit or overflow event generated by slave timer controller 1: Only counter overflow generates an overflow event
Bit 1	OVFEN	0x0	rw	Overflow event enable 0: Enabled 1: Disabled
Bit 0	TMREN	0x0	rw	TMR enable 0: Disabled 1: Enabled

Control register2 (TMRx_CTRL2)

Bit	Register	Reset value	Type	Description
Bit 15:8	Reserved	0x00	resd	Kept at its default value.
Bit 7	C1INSEL	0x0	rw	C1IN selection 0: CH1 pin is connected to C1IRAW input 1: The XOR result of CH1, CH2 and CH3 pins is connected to C1IRAW input
Bit 6:4	PTOS	0x0	rw	Master TMR output selection This field is used to select the TMRx signal sent to the slave timer. 000: Reset 001: Enable 010: Update 011: Compare pulse 100: C1ORAW signal 101: C2ORAW signal 110: C3ORAW signal 111: C4ORAW signal
Bit 3	DRS	0x0	rw	DMA request source 0: Capture/compare event 1: Overflow event
Bit 2:0	Reserved	0x0	resd	Kept at its default value.

Slave timer control register (TMRx_STCTRL)

Bit	Register	Reset value	Type	Description
Bit 15	ESP	0x0	rw	External signal polarity 0: High or rising edge 1: Low or falling edge
Bit 14	ECMBEN	0x0	rw	External clock mode B enable This bit is used to enable external clock mode B 0: Disabled 1: Enabled
Bit 13:12	ESDIV	0x0	rw	External signal divide This field is used to select the frequency division of an external trigger 00: Normal 01: Divided by 2 10: Divided by 4 11: Divided by 8

Bit	Register	Reset value	Type	Description
Bit 11:8	ESF	0x0	rw	External signal filter This field is used to filter an external signal. The external signal can be sampled only after it has been generated N times. 0000: No filter, sampling by f_{DTS} 0001: $f_{SAMPLING} = f_{CK_INT}$, N=2 0010: $f_{SAMPLING} = f_{CK_INT}$, N=4 0011: $f_{SAMPLING} = f_{CK_INT}$, N=8 0100: $f_{SAMPLING} = f_{DTS}/2$, N=6 0101: $f_{SAMPLING} = f_{DTS}/2$, N=8 0110: $f_{SAMPLING} = f_{DTS}/4$, N=6 0111: $f_{SAMPLING} = f_{DTS}/4$, N=8 1000: $f_{SAMPLING} = f_{DTS}/8$, N=6 1001: $f_{SAMPLING} = f_{DTS}/8$, N=8 1010: $f_{SAMPLING} = f_{DTS}/16$, N=5 1011: $f_{SAMPLING} = f_{DTS}/16$, N=6 1100: $f_{SAMPLING} = f_{DTS}/16$, N=8 1101: $f_{SAMPLING} = f_{DTS}/32$, N=5 1110: $f_{SAMPLING} = f_{DTS}/32$, N=6 1111: $f_{SAMPLING} = f_{DTS}/32$, N=8
Bit 7	STS	0x0	rw	Subordinate TMR synchronization If enabled, master and slave timer can be synchronized. 0: Disabled 1: Enabled
Bit 6:4	STIS	0x0	rw	Subordinate TMR input selection This field is used to select the subordinate TMR input. 000: Internal selection 0 (IS0) 001: Internal selection 1 (IS1) 010: Internal selection 2 (IS2) 011: Internal selection 3 (IS3) 100: C1IRAW input detector (C1INC) 101: Filtered input 1 (C1IF1) 110: Filtered input 2 (C1IF2) 111: External input (EXT) Pleaser refer to Table 43 and 45 for more information on ISx for each timer.
Bit 3	Reserved	0x0	resd	Kept at its default value
Bit 2:0	SMSEL	0x0	rw	Subordinate TMR mode selection 000: Slave mode is disabled 001: Encoder mode A 010: Encoder mode B 011: Encoder mode C 100: Reset mode – Rising edge of the TRGIN input reinitializes the counter 101: Suspend mode – The counter starts counting when the TRGIN is high 110: Trigger mode – A trigger event is generated at the rising edge of the TRGIN input 111: External clock mode A – Rising edge of the TRGIN input clocks the counter Note: Please refer to count mode section for the details on encoder mode A/B/C.

DMA/interrupt enable register (TMRx_IDEN)

Bit	Register	Reset value	Type	Description
Bit 15	Reserved	0x0	resd	Kept at its default value
Bit 14	TDEN	0x0	rw	Trigger DMA request enable 0: Disabled 1: Enabled
Bit 13	Reserved	0x0	resd	Kept at its default value
Bit 12	C4DEN	0x0	rw	Channel 4 DMA request enable 0: Disabled 1: Enabled
Bit 11	C3DEN	0x0	rw	Channel 3 DMA request enable 0: Disabled 1: Enabled
Bit 10	C2DEN	0x0	rw	Channel 2 DMA request enable 0: Disabled 1: Enabled
Bit 9	C1DEN	0x0	rw	Channel 1 DMA request enable 0: Disabled 1: Enabled
Bit 8	OVFDEN	0x0	rw	Overflow event DMA request enable 0: Disabled 1: Enabled
Bit 7	Reserved	0x0	resd	Kept at its default value
Bit 6	TIEN	0x0	rw	Trigger interrupt enable 0: Disabled 1: Enabled
Bit 5	Reserved	0x0	resd	Kept at its default value
Bit 4	C4IEN	0x0	rw	Channel 4 interrupt enable 0: Disabled 1: Enabled
Bit 3	C3IEN	0x0	rw	Channel 3 interrupt enable 0: Disabled 1: Enabled
Bit 2	C2IEN	0x0	rw	Channel 2 interrupt enable 0: Disabled 1: Enabled
Bit 1	C1IEN	0x0	rw	Channel 1 interrupt enable 0: Disabled 1: Enabled
Bit 0	OVFIEN	0x0	rw	Overflow interrupt enable 0: Disabled 1: Enabled

Interrupt status register (TMRx_ISTS)

Bit	Register	Reset value	Type	Description
Bit 15:13	Reserved	0x0	resd	Kept at its default value
Bit 12	C4RF	0x0	rw0c	Channel 4 recapture flag Please refer to C1RF description.
Bit 11	C3RF	0x0	rw0c	Channel 3 recapture flag Please refer to C1RF description.
Bit 10	C2RF	0x0	rw0c	Channel 2 recapture flag Please refer to C1RF description.
Bit 9	C1RF	0x0	rw0c	Channel 1 recapture flag This bit indicates whether a recapture is detected when C1IF=1. This bit is set by hardware, and cleared by writing “0”. 0: No capture is detected 1: Capture is detected
Bit 8:7	Reserved	0x0	resd	Kept at its default value
Bit 6	TRGIF	0x0	rw0c	Trigger interrupt flag This bit is set by hardware on a trigger event. It is cleard by writing “0”. 0: No trigger event occurs 1: Trigger event is generated Trigger event: an active edge is detected on TRGIN input, or any edge in suspend mode.
Bit 5	Reserved	0x0	resd	Kept at its default value
Bit 4	C4IF	0x0	rw0c	Channel 4 interrupt flag Please refer to C1IF description.
Bit 3	C3IF	0x0	rw0c	Channel 3 interrupt flag Please refer to C1IF description.
Bit 2	C2IF	0x0	rw0c	Channel 2 interrupt flag Please refer to C1IF description.
Bit 1	C1IF	0x0	rw0c	Channel 1 interrupt flag If the channel 1 is configured as input mode: This bit is set by hardware on a capture event. It is cleared by software or read access to the TMRx_C1DT 0: No capture event occurs 1: Capture event is generated If the channel 1 is configured as output mode: This bit is set by hardware on a compare event. It is cleared by software. 0: No compare event occurs 1: Compare event is generated
Bit 0	OVFIF	0x0	rw0c	Overflow interrupt flag This bit is set by hardware on an overflow event. It is cleared by software. 0: No overflow event occurs 1: Overflow event is generated. If OVFEN=0 and OVFS=0 in the TMRx_CTRL1 register: − An overflow event is generated when OVFG= 1 in the TMRx_SWEVE register; − An overflow event is generated when the counter CVAL is reinitialized by a trigger event.

Software event register (TMRx_SWEVT)

Bit	Register	Reset value	Type	Description
Bit 15:7	Reserved	0x000	resd	Kept at its default value.
Bit 6	TRGSWTR	0x0	rw	Trigger event triggered by software This bit is set by software to generate a trigger event. 0: No effect 1: Generate a trigger event.
Bit 5	Reserved	0x0	resd	Kept at its default value.
Bit 4	C4SWTR	0x0	wo	Channel 4 event triggered by software Please refer to C1M description.
Bit 3	C3SWTR	0x0	wo	Channel 3 event triggered by software Please refer to C1M description.
Bit 2	C2SWTR	0x0	wo	Channel 2 event triggered by software Please refer to C1M description
Bit 1	C1SWTR	0x0	wo	Channel 1 event triggered by software This bit is set by software to generate a channel 1 event. 0: No effect 1: Generate a channel 1 event.
Bit 0	OVFSWTR	0x0	wo	Overflow event triggered by software This bit is set by software to generate an overflow event. 0: No effect 1: Generate an overflow event.

Channel mode register1 (TMRx_CM1)

Output compare mode:

Bit	Register	Reset value	Type	Description
Bit 15	C2OSEN	0x0	rw	Channel 2 output switch enable
Bit 14:12	C2OCTRL	0x0	rw	Channel 2 output control
Bit 11	C2OBEN	0x0	rw	Channel 2 output buffer enable
Bit 10	C2OIEN	0x0	rw	Channel 2 output enable immediately
Bit 9:8	C2C	0x0	rw	Channel 2 configuration This field is used to define the direction of the channel 2 (input or output), and the selection of input pin when C2EN='0': 00: Output 01: Input, C2IN is mapped on C2IRAW 10: Input, C2IN is mapped on C1IRAW 11: Input, C2IN is mapped on STCI. This mode works only when the internal trigger input is selected by STIS register.
Bit 7	C1OSEN	0x0	rw	Channel 1 output switch enable 0: C1ORAW is not affected by EXT 1: Once high level is detect on EXT input, clear C1ORAW.

Bit	Register	Reset value	Type	Description
Bit 6:4	C1OCTRL	0x0	rw	Channel 1 output control This field defines the behavior of the original signal C1ORAW. 000: Disconnected. C1ORAW is disconnected from C1OUT; 001: C1ORAW is high when TMRx_CVAL=TMRx_C1DT 010: C1ORAW is low when TMRx_CVAL=TMRx_C1DT 011: Switch C1ORAW level when TMRx_CVAL=TMRx_C1DT 100: C1ORAW is forced low 101: C1ORAW is forced high. 110: PWM mode A - OWCDIR=0, C1ORAW is high once TMRx_C1DT > TMRx_CVAL, else low; - OWCDIR=1, C1ORAW is low once TMRx_C1DT < TMRx_CVAL, else high; 111: PWM mode B - OWCDIR=0, C1ORAW is low once TMRx_C1DT > TMRx_CVAL, else high; - OWCDIR=1, C1ORAW is high once TMRx_C1DT < TMRx_CVAL, else low. Note: In the configurations othern than 000', the C1OUT is connected to C1ORAW. The C1OUT output level is not only subject to the changes of C1ORAW, but also the output polarity set by CCTRL.
Bit 3	C1OBEN	0x0	rw	Channel 1 output buffer enable 0: Buffer function of TMRx_C1DT is disabled. The new value written to the TMRx_C1DT takes effect immediately. 1: Buffer function of TMRx_C1DT is enabled. The value to be written to the TMRx_C1DT is stored in the buffer register, and can be sent to the TMRx_C1DT register only on an overflow event.
Bit 2	C1OIEN	0x0	rw	Channel 1 output enable immediately In PWM mode A or B, this bit is used to accelerate the channel 1 output's response to the trigger event. 0: Need to compare the CVAL with C1DT before generating an output 1: No need to compare the CVAL and C1DT. An output is generated immediately when a trigger event occurs.
Bit 1:0	C1C	0x0	rw	Channel 1 configuration This field is used to define the direction of the channel 1 (input or output), and the selection of input pin when C1EN='0': 00: Output 01: Input, C1IN is mapped on C1IRAW 10: Input, C1IN is mapped on C2IRAW 11: Input, C1IN is mapped on STCI. This mode works only when the internal trigger input is selected by STIS.

Input capture mode:

Bit	Register	Reset value	Type	Description
Bit 15:12	C2DF	0x0	rw	Channel 2 digital filter
Bit 11:10	C2IDIV	0x0	rw	Channel 2 input divider
Bit 9:8	C2C	0x0	rw	Channel 2 configuration This field is used to define the direction of the channel 2 (input or output), and the selection of input pin when C2EN='0': 00: Output 01: Input, C2IN is mapped on C2IRAW 10: Input, C2IN is mapped on C1IRAW 11: Input, C2IN is mapped on STCI. This mode works only when the internal trigger input is selected by STIS.
Bit 7:4	C1DF	0x0	rw	Channel 1 digital filter This field defines the digital filter of the channel 1. N stands for the number of filtering, indicating that the input edge can pass the filter only after N sampling events. 0000: No filter, sampling is done at f_{DTS} 1000: $f_{SAMPLING} = f_{DTS}/8$, N=6 0001: $f_{SAMPLING} = f_{CK_INT}$, N=2 1001: $f_{SAMPLING} = f_{DTS}/8$, N=8 0010: $f_{SAMPLING} = f_{CK_INT}$, N=4 1010: $f_{SAMPLING} = f_{DTS}/16$, N=5 0011: $f_{SAMPLING} = f_{CK_INT}$, N=8 1011: $f_{SAMPLING} = f_{DTS}/16$, N=6 0100: $f_{SAMPLING} = f_{DTS}/2$, N=6 1100: $f_{SAMPLING} = f_{DTS}/16$, N=8 0101: $f_{SAMPLING} = f_{DTS}/2$, N=8 1101: $f_{SAMPLING} = f_{DTS}/32$, N=5 0110: $f_{SAMPLING} = f_{DTS}/4$, N=6 1110: $f_{SAMPLING} = f_{DTS}/32$, N=6 0111: $f_{SAMPLING} = f_{DTS}/4$, N=8 1111: $f_{SAMPLING} = f_{DTS}/32$, N=8
Bit 3:2	C1IDIV	0x0	rw	Channel 1 input divider This field defines Channel 1 input divider. 00: No divider. An input capture is generated at each active edge. 01: An input compare is generated every 2 active edges 10: An input compare is generated every 4 active edges 11: An input compare is generated every 8 active edges Note: the divider is reset once C1EN='0'
Bit 1:0	C1C	0x0	rw	Channel 1 configuration This field is used to define the direction of the channel 1 (input or output), and the selection of input pin when C1EN='0': 00: Output 01: Input, C1IN is mapped on C1IRAW 10: Input, C1IN is mapped on C2IRAW 11: Input, C1IN is mapped on STCI. This mode works only when the internal trigger input is selected by STIS.

Channel mode register2 (TMRx_CM2)

Output compare mode:

Bit	Register	Reset value	Type	Description
Bit 15	C4OSEN	0x0	rw	Channel 4 output switch enable
Bit 14:12	C4OCTRL	0x0	rw	Channel 4 output control
Bit 11	C4OBEN	0x0	rw	Channel 4 output buffer enable
Bit 10	C4OIEN	0x0	rw	Channel 4 output enable immediately
Bit 9:8	C4C	0x0	rw	Channel 4 configuration This field is used to define the direction of the channel 1 (input or output), and the selection of input pin when C4EN='0': 00: Output 01: Input, C4IN is mapped on C4IRAW 10: Input, C4IN is mapped on C3IRAW 11: Input, C4IN is mapped on STCI. This mode works only when the internal trigger input is selected by STIS.
Bit 7	C3OSEN	0x0	rw	Channel 3 output switch enable
Bit 6:4	C3OCTRL	0x0	rw	Channel 3 output control
Bit 3	C3OBEN	0x0	rw	Channel 3 output buffer enable
Bit 2	C3OIEN	0x0	rw	Channel 3 output enable immediately
Bit 1:0	C3C	0x0	rw	Channel 3 configuration This field is used to define the direction of the channel 1 (input or output), and the selection of input pin when C3EN='0': 00: Output 01: Input, C3IN is mapped on C3IRAW 10: Input, C3IN is mapped on C4IRAW 11: Input, C3IN is mapped on STCI. This mode works only when the internal trigger input is selected by STIS.

Input capture mode:

Bit	Register	Reset value	Type	Description
Bit 15:12	C4DF	0x0	rw	Channel 4 digital filter
Bit 11:10	C4IDIV	0x0	rw	Channel 4 input divider
Bit 9:8	C4C	0x0	rw	Channel 4 configuration This field is used to define the direction of the channel 1 (input or output), and the selection of input pin when C4EN='0': 00: Output 01: Input, C4IN is mapped on C4IRAW 10: Input, C4IN is mapped on C3IRAW 11: Input, C4IN is mapped on STCI. This mode works only when the internal trigger input is selected by STIS.
Bit 7:4	C3DF	0x0	rw	Channel 3 digital filter
Bit 3:2	C3IDIV	0x0	rw	Channel 3 input divider
Bit 1:0	C3C	0x0	rw	Channel 3 configuration This field is used to define the direction of the channel 1 (input or output), and the selection of input pin when C3EN='0': 00: Output 01: Input, C3IN is mapped on C3IRAW 10: Input, C3IN is mapped on C4IRAW 11: Input, C3IN is mapped on STCI. This mode works only when the internal trigger input is selected by STIS.

Channel control register (TMRx_CCTRL)

Bit	Register	Reset value	Type	Description
Bit 15:14	Reserved	0x0	resd	Kept at its default value.
Bit 13	C4P	0x0	rw	Channel 4 polarity Pleaser refer to C1P description.
Bit 12	C4EN	0x0	rw	Channel 4 enable Pleaser refer to C1EN description.
Bit 11:10	Reserved	0x0	resd	Default value
Bit 9	C3P	0x0	rw	Channel 3 polarity Pleaser refer to C1P description.
Bit 8	C3EN	0x0	rw	Channel 3 enable Pleaser refer to C1EN description.
Bit 7:6	Reserved	0x0	resd	Kept at its default value.
Bit 5	C2P	0x0	rw	Channel 2 polarity Pleaser refer to C1P description.
Bit 4	C2EN	0x0	rw	Channel 2 enable Pleaser refer to C1EN description.
Bit 3:2	Reserved	0x0	resd	Kept at its default value.
Bit 1	C1P	0x0	rw	Channel 1 polarity When the channel 1 is configured as output mode: 0: C1OUT is active high 1: C1OUT is active low When the channel 1 is configured as input mode: 0: C1IN active edge is on its rising edge. When used as external trigger, C1IN is not inverted. 1: C1IN active edge is on its falling edge. When used as external trigger, C1IN is inverted.
Bit0	C1EN	0x0	rw	Channel 1 enable 0: Input or output is disabled 1: Input or output is enabled

Table 50. Standard CxOUT channel output control bit

CxEN bit	CxOUT output state
0	Output disabled (CxOUT=0, Cx_EN=0)
1	CxOUT = CxORAW + polarity, Cx_EN=1

Note: The state of the external I/O pins connected to the standard CxOUT channel depends on the CxOUT channel state and the GPIO and IOMUX registers.

Counter value (TMRx_CVAL)

Bit	Register	Reset value	Type	Description
Bit 31:16	CVAL	0x0000	rw	Counter value When TMR2 or TMR5 enables plus mode (the PMEN bit in the TMR_CTRL1 register), the CVAL is expanded to 32 bits.
Bit 15:0	CVAL	0x0000	rw	Counter value

Division value (TMRx_DIV)

Bit	Register	Reset value	Type	Description
Bit 15:0	DIV	0x0000	rw	Divider value The counter clock frequency $f_{CK_CNT} = f_{TMR_CLK}/(DIV[15:0]+1)$. DIV contains the value written at an overflow event.

Period register (TMRx_PR)

Bit	Register	Reset value	Type	Description
Bit 31:16	PR	0x0000	rw	Period value When TMR2 or TMR5 enables plus mode (the PMEN bit in the TMR_CTRL1 register), the PR is expanded to 32 bits.
Bit 15:0	PR	0x0000	rw	Period value This defines the period value of the TMRx counter. The timer stops working when the period value is 0.

Channel 1 data register (TMRx_C1DT)

Bit	Register	Reset value	Type	Description
Bit 31:16	C1DT	0x0000	rw	Channel 1 data register When TMR2 or TMR5 enables plus mode (the PMEN bit in the TMR_CTRL1 register), the C1DT is expanded to 32 bits.
Bit 15:0	C1DT	0x0000	rw	Channel 1 data register When the channel 1 is configured as input mode: The C1DT is the CVAL value stored by the last channel 1 input event (C1IN) When the channel 1 is configured as output mode: C1DT is the value to be compared with the CVAL value. Whether the written value takes effective immediately depends on the C1OBEN bit, and the corresponding output is generated on C1OUT as configured.

Channel 2 data register (TMRx_C2DT)

Bit	Register	Reset value	Type	Description
Bit 31:16	C2DT	0x0000	rw	Channel 2 data register When TMR2 or TMR5 enables plus mode (the PMEN bit in the TMR_CTRL1 register), the C2DT is expanded to 32 bits.
Bit 15:0	C2DT	0x0000	rw	Channel 2 data register When the channel 2 is configured as input mode: The C2DT is the CVAL value stored by the last channel 2 input event (C1IN) When the channel 2 is configured as output mode: C2DT is the value to be compared with the CVAL value. Whether the written value takes effective immediately depends on the C2OBEN bit, and the corresponding output is generated on C2OUT as configured.

Channel 3 data register (TMRx_C3DT)

Bit	Register	Reset value	Type	Description
Bit 31:16	C3DT	0x0000	rw	Channel 3 data register When TMR2 or TMR5 enables plus mode (the PMEN bit in the TMR_CTRL1 register), the C3DT is expanded to 32 bits.
Bit 15:0	C3DT	0x0000	rw	Channel 3 data register When the channel 3 is configured as input mode: The C3DT is the CVAL value stored by the last channel 3 input event (C1IN) When the channel 3 is configured as output mode: C3DT is the value to be compared with the CVAL value. Whether the written value takes effective immediately depends on the C3OBEN bit, and the corresponding output is generated on C3OUT as configured.

Channel 4 data register (TMRx_C4DT)

Bit	Register	Reset value	Type	Description
Bit 31:16	C4DT	0x0000	rw	Channel 4 data register When TMR2 or TMR5 enables plus mode (the PMEN bit in the TMR_CTRL1 register), the C4DT is expanded to 32 bits.
Bit 15:0	C4DT	0x0000	rw	Channel 4 data register When the channel 4 is configured as input mode: The C4DT is the CVAL value stored by the last channel 4 input event (C1IN) When the channel 4 is configured as output mode: C4DT is the value to be compared with the CVAL value. Whether the written value takes effective immediately depends on the C4OBEN bit, and the corresponding output is generated on C4OUT as configured.

DMA control register (TMRx_DMACTRL)

Bit	Register	Reset value	Type	Description
Bit 15:13	Reserved	0x0	resd	Kept at its default value.
Bit 12:8	DTB	0x00	rw	DMA transfer bytes This field defines the number of DMA transfers: 00000: 1 byte 00001: 2 bytes 00010: 3 bytes 00011: 4 bytes .... 10000: 17 bytes 10001: 18 bytes
Bit 7:5	Reserved	0x0	resd	Kept at its default value.
Bit 4:0	ADDR	0x00	rw	DMA transfer address offset ADDR is defined as an offset starting from the address of the TMRx_CTRL1 register. 00000: TMRx_CTRL1 00001: TMRx_CTRL2 00010: TMRx_STCTRL .....

DMA data register (TMRx_DMADT)

Bit	Register	Reset value	Type	Description
Bit 15:0	DMADT	0x0000	rw	DMA data register A read or write operation to the DMADT register accesses the TMR registers at the following address: TMRx peripheral address + ADDR × 4 to TMRx peripheral address + ADDR × 4 + DTB × 4.

General-purpose timer (TMR9 to TMR14)

TMRx introduction

The general-purpose timer (TMR9 to TMR14) consists of a 16-bit counter supporting upcounting mode. These timers can be synchronized.

TMRx main features

TMR9 and TMR12 main features

The main functions of general-purpose TMR9 and TMR12 include:

- Souce of counter clock: internal clock and external clock
- 16-bit up counter
- 2 independent channels for input capture, output compare, PWM generation and one-pulse mode output
- Synchronization control between master and slave timers
- Interrrupt is generated at overflow event, trigger event and channel event

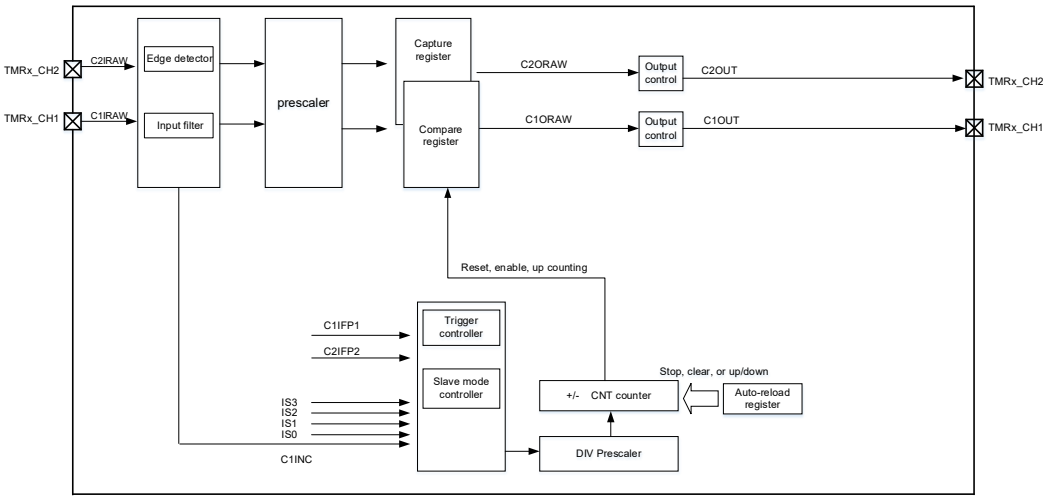


Figure 86. Block diagram of general-purpose TMR9/12

TMR10, TMR11, TMR13 and TMR14 main features

The main functions of general-purpose TMRx (TMR10, TMR11, TMR13 and TMR14) include:

- Souce of counter clock: internal clock
- 16-bit up counter
- 1 independent channel for input capture, output compare, PWM generation
- Synchronization control between master and slave timers
- Interrrupt is generated at overflow event and channel event

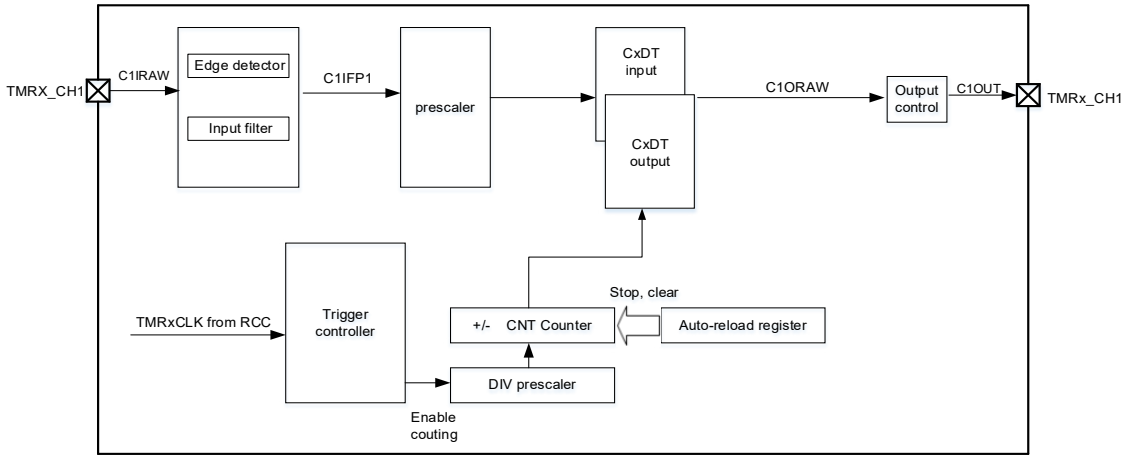


Figure 87. Block diagram of general-purpose TMR10/11/13/14

TMRx functional descriptions

Count clock

The count clock of general-purpose timers can be provided by the internal clock (CK_INT), external clock (external clock mode A) and internal trigger input (ISx)

Internal clock (CK_INT)

By default, the CK_INT divided by the prescaler is used to drive the counter to start counting.

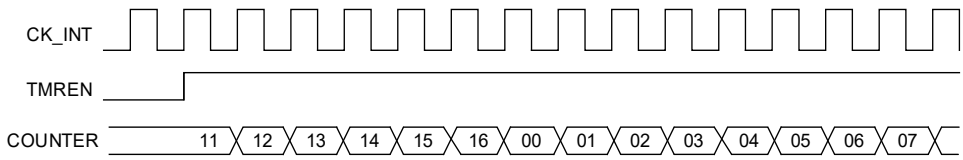


Figure 88. Control circuit with CK_INT divided by 1

External clock (TMR9/12 only)

The counter clock can be provided by TRGIN signal.

When SMSEL=3'111, external clock mode A is selected. Set the STIS[2:0] bit to select TRGIN signal to drive the counter to start counting.

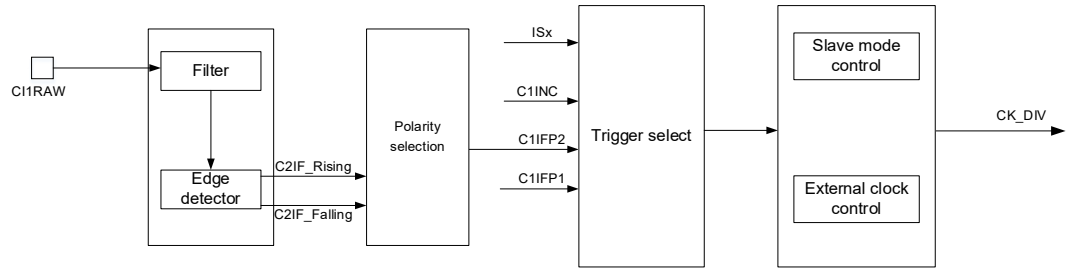


Figure 89. Block diagram of external clock mode A

Note: The delay between the signal on the input side and the actual clock of the counter is due to the synchronization circuit.

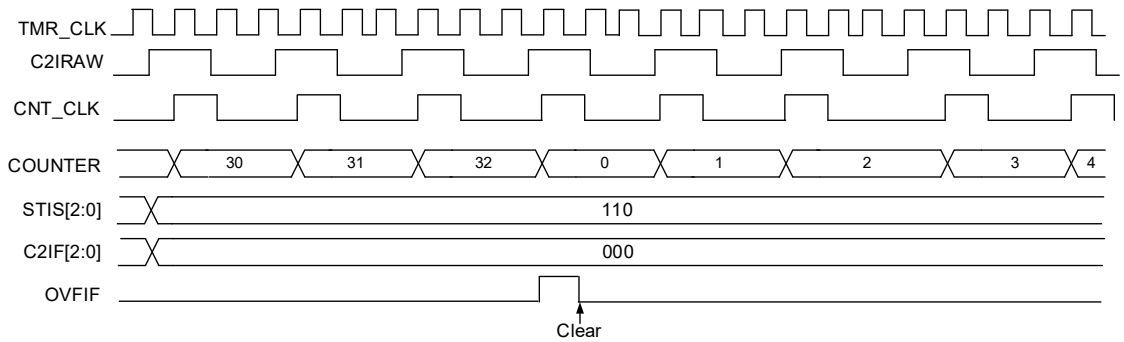


Figure 90. Counting in external clock mode A

Internal trigger input (ISx)

Timer synchronization allows interconnection between several timers. The TMR_CLK of one timer can be provided by the TRGOUT signal output by another timer. Set the STIS[2:0] bit to select internal trigger signal to enable counting.

Each timer consists of a 16-bit prescaler, which is used to generate the CK_CNT that enables the counter to count. The frequency division relationship between the CK_CNT and TMR_CLK can be adjusted by setting the value of the TMRx_DIV register. The prescaler value can be modified at any time, but it takes effect only when the next overflow event occurs.

Table 51. TMRx internal trigger connection

Slave controler	IS0 (STIS=000)	IS1 (STIS = 001)	IS2 (STIS = 010)	IS3 (STIS = 011)
TMR9	TMR2_TRGOUT	TMR3_TRGOUT	TMR10_OC	TMR11_OC
TMR12	TMR4_TRGOUT	TMR5_TRGOUT	TMR13_OC	TMR14_OC

Note: If there is no corresponding timer in a device, the corresponding trigger signal ISx is not present.

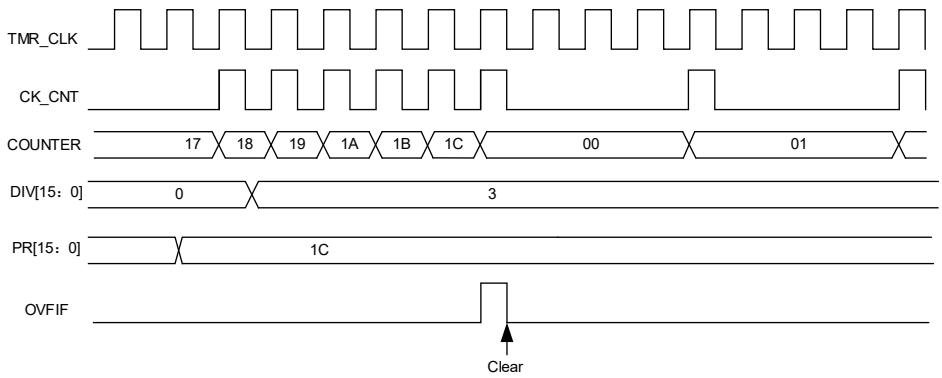


Figure 91. Counter timing with prescaler value changing from 1 to 4

Counting mode

The general-purpose timer consists of a 16-bit counter supporting upcounting mode. The TMRx_PR register is loaded with the counter value. The value in the TMRx_PR is immediately moved to the shadow register by deault. When the periodic buffer is enabled (PRBEN=1), the value in the TMRx_PR register is transferred to the shadow register only at an overflow event. The OVFEN and OVFS bits are used to configure the overflow event.

Settng TMREN=1 to enable the timer to start counting. Base on synchronization logic, however, the actual enable signal TMR_EN is set 1 clock cycle after the TMREN is set.

Upcounting mode

In upcounting mode, the counter counts from 0 to the value programmed in the TMRx_PR register, then restarts from 0, and generates a counter overflow event, with the OVFIF bit being set to 1. If the overflow event is disabled, the counter is no longer reloaded with the preload value and period value at a counter overflow event, otherwise, the counter is updated with the preload value and period value on an overflow event.

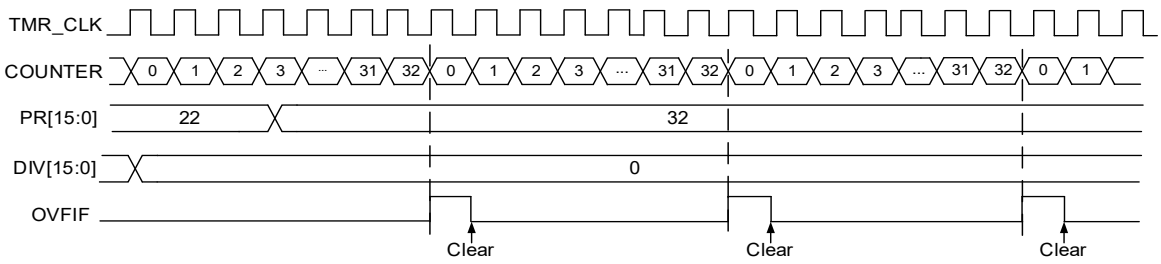


Figure 92. Overflow event when PRBEN=0

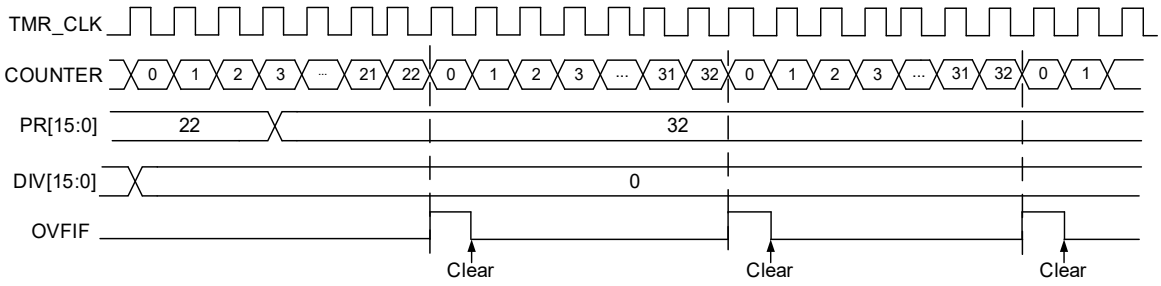


Figure 93. Overflow event when PRBEN=1

TMR input function

Each timer of TMR9 and TMR12 has two independent channels, while each of TMR10, TMR11, TMR13 and TMR14 has an independent channel. Each channel can be configured as input or output. As input, the channel can be used for the filtering, selection, division and input capture of the input signals.

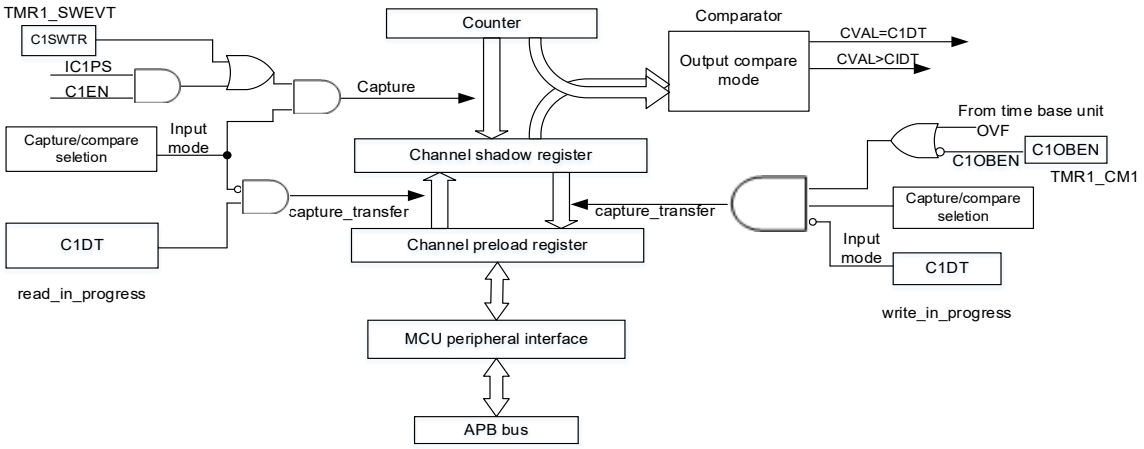


Figure 94. Input/output channel 1 main circuit

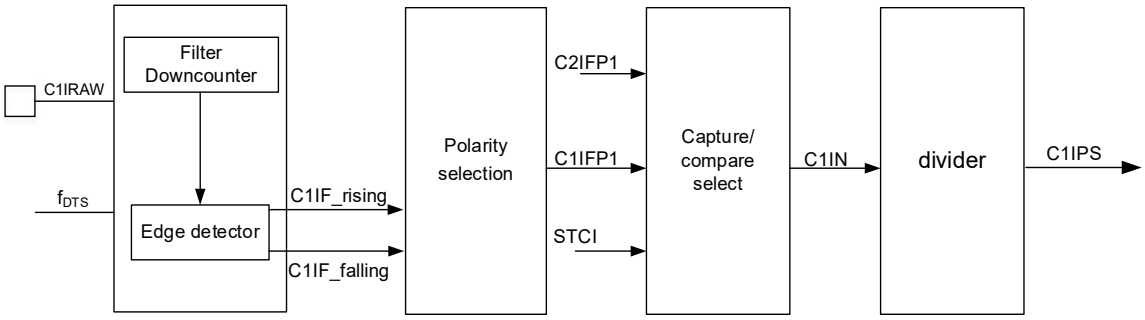


Figure 95. Channel 1 input stage

Input mode

In input mode, the TMRx_CxDT registers latch the current counter values after the selected triggle signal is detected, and the capture compare interrupt flag bit (CxIF) is set to 1. An interrupt will be generated if the CxIEN bit is enabled. If the selected trigger signal is detected when CxIF=1, the CxRF is set to 1.

To capture the rising edge of C1IN input, following the configuration procedure mentioned below:

- Set C1C=01 in the TMRx_CxDT register to select the C1IN as channel 1 input
- Set the filter bandwidth of C1IN signal (CxDF[3:0])
- Set the active edge on the C1IN channel by writing C1P=0 (rising edge) in the TMRx_CCTR register
- Program the capture frequency of C1IN signal (C1DIV[1:0])
- Enable channel 1 input capture (C1EN=1)
- If needed, enable the relevant interrupt by setting the C1IEN bit in the TMRx_IDEN register

TMR output function

The TMR output consists of a comparator and an output controller. It is used to program the period, duty cycle and polarity of the output signal.

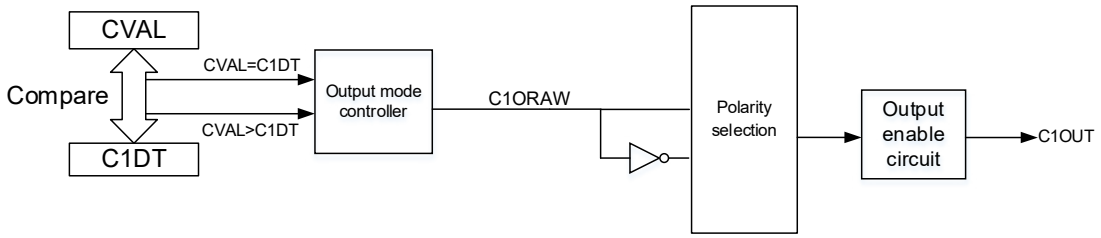


Figure 96. Capture/compare channel output stage (channel 1)

Output mode

Write CxC[2:0]≠2'b00 to configure the channel as output to implement multiple output modes. In this case, the counter value is compared with the value in the TMRx_CxDT register, and the intermediate signal CxORAW is generated according to the output mode selected by CxOCTRL[2:0], which is sent to IO after being processed by the output control circuit. The period of the output signal is configured by the TMRx_PR register, while the duty cycle by the TMRx_CxDT register.

Output compare modes include:

- PWM mode: Set CxOCTRL=3'b110/111 to enable PWM mode. Each channel can be independently configured to output one PWM signal. In this case, the period of the output signal is configured by the TMRx_PR register, and the duty cycle by the CxDT register. The counter value is compared with the value of the TMRx_CxDT register, and the corresponding level signal is sent according to the counting direction. For more information on PWM mode A/B, refer to the description of the CxOCTRL[2:0] bit. In up/down counting mode, the OWCDIR bit is used to indicate the counting direction.

- Forced output mode: Set CxOCTRL=3'b100/101 to enable forced output mode. In this case, the CxORAW is forced to be the programmed level, irrespective of the counter value. Despite this, the channel flag bit and DMA request still depend on the compare result.
- Output compare mode: Set CxOCTRL=2'b001/010/011 to enable output compare mode. In this case, when the counter value matches the value of the CxDT register, the CxORAW is forced high, low or toggling.
- One-pulse mode (TMR9/12 only):This is a particular case of PWM mode. Set OCMEN=1 to enable one-pulse mode. In this mode, the comparison match is performed in the current counting period. The TMREN bit is cleared as soon as the current counting is completed. Therefore, only one pulse is output. When configured as in upcounting mode, the configureation must follow the rule: CVAL<CxDT≤PR; in downcounting mode, CVAL>CxDT is required.
- Fast output mode (TMR9/12 only): Set CxOIEN=1 to enable this mode. If enabled, the CxORAW signal will not change when the counter value matches the CxDT, but at the beginning of the current counting period. In other words, the comparison result is advanced, so the comparison result between the counter value and the TMRx_CxDT register will determine the level of CxORAW in advance.

The following diagram gives an example of output compare mode (toggle) with C1DT=0x3. When the counter value is equal to 0x3, C1OUT toggles.

Figure 97 gives an example of the combination between upcounting mode and PWM mode A. The output signal behaves when PR=0x32 but CxDT is configured with a different value.

Figure 98 gives an example of the combination between upcounting mode and one-pulse PWM mode B. The counter only counts only one cycle, and the output signal sents only one pulse.

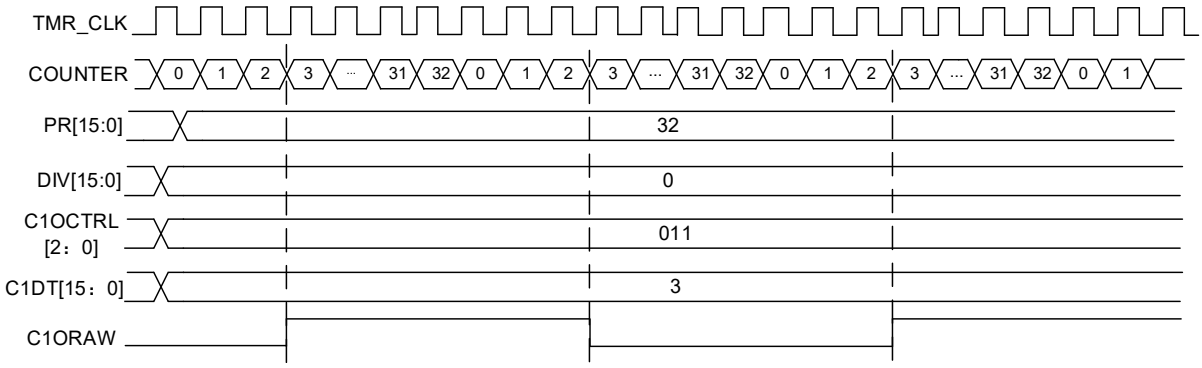


Figure 97. C1ORAW toggles when counter value matches the C1DT value

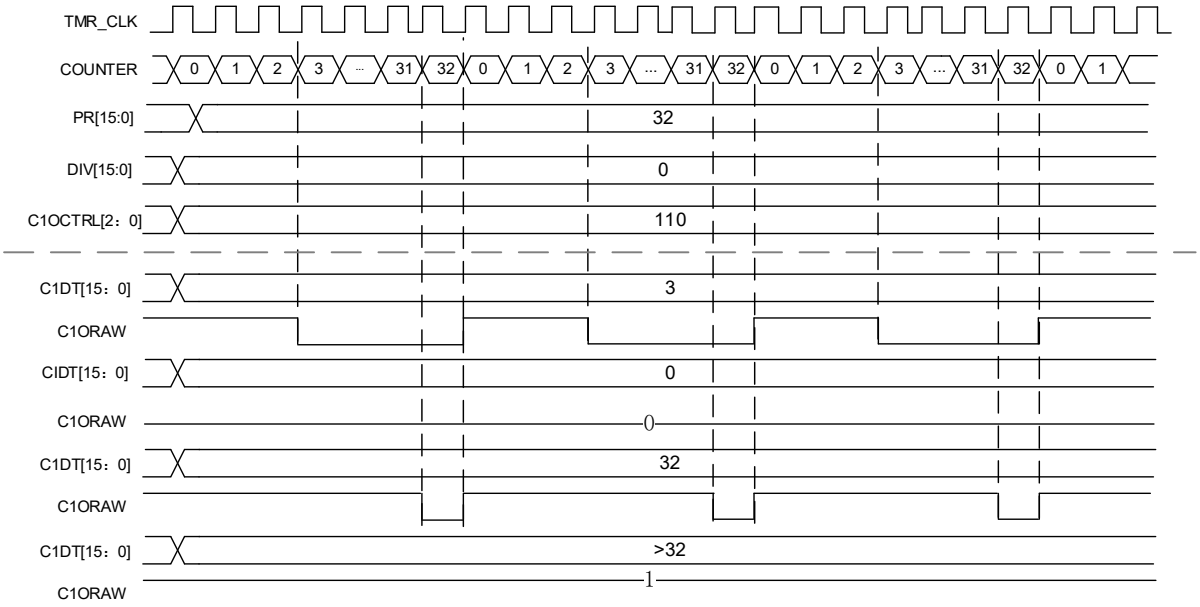


Figure 98. Upcounting mode and PWM mode A

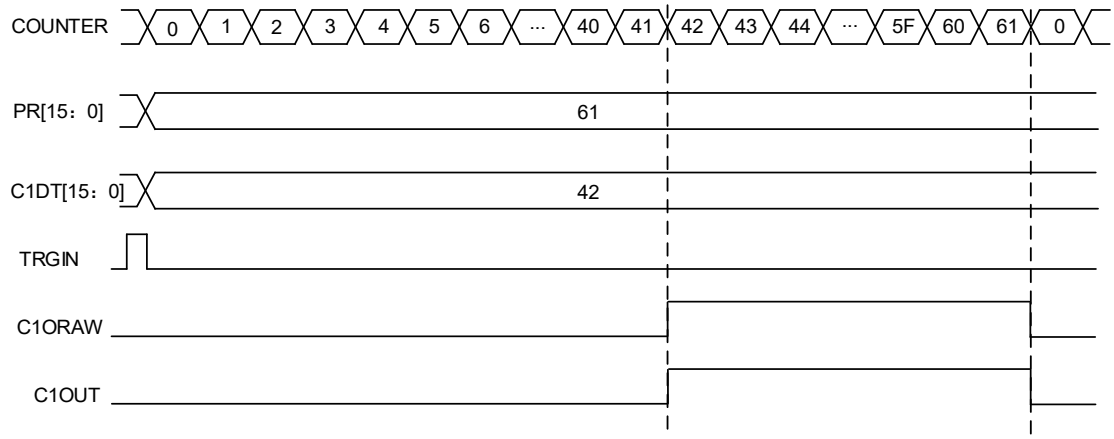


Figure 99. One-pulse mode

TMR synchronization

TMR9 and TMR12 are linked together internally for timer synchronization. Slave timer is selected by setting the SMSEL[2:0] bit.

Slave mode: Reset mode

The counter and its prescaler can be reset by a selected trigger signal. An overflow event is generated when OVFS=0.

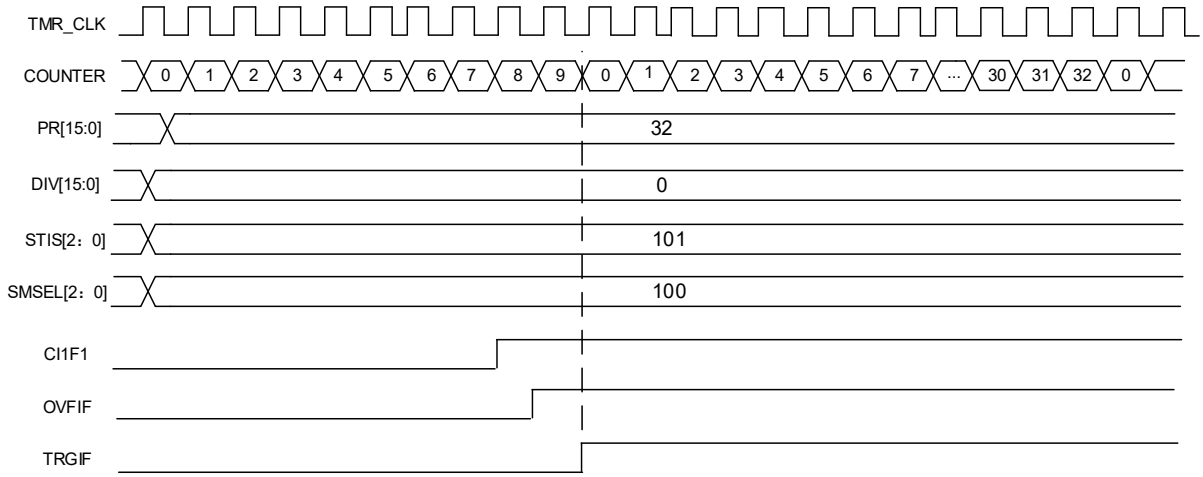


Figure 100. Example of reset mode

Slave mode: Suspend mode

In this mode, the counter is controlled by a selected trigger input. The counter starts counting when the trigger input is high and stops as soon as the trigger input is low.

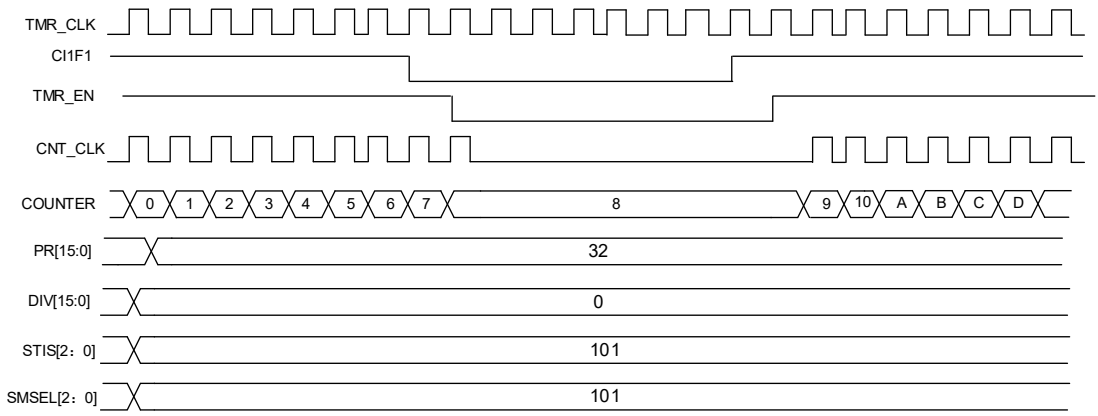


Figure 101. Example of suspend mode

Slave mode: Trigger mode

The counter can start counting on the rising edge of a selected trigger input (TMR_EN=1)

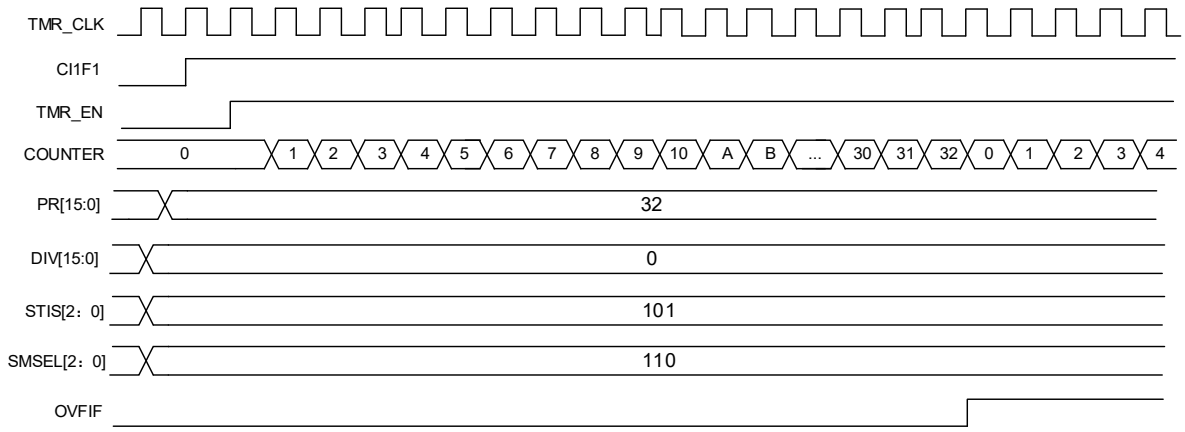


Figure 102. Example of trigger mode

Please refer to section 14.2.3.5 for more information on timer synchronization.

Debug mode

When the microcontroller enters debug mode (Cortex™-M4F core halted), the TMRx counter stops counting by setting the TMRx_PAUSE in the DEBUG module.

TMR9 and TMR12 registers

These peripheral registers must be accessed by word (32 bits). All TMRx register are mapped into a 16-bit addressable space.

Table 52. TMRx register map and reset value

Bit	Register	Reset value
TMRx_CTRL1	0x00	0x0000
TMRx_STCTRL	0x08	0x0000
TMRx_IDEN	0x0C	0x0000
TMRx_ISTS	0x10	0x0000
TMRx_SWEVT	0x14	0x0000
TMRx_CM1	0x18	0x0000
TMRx_CCTRL	0x20	0x0000
TMRx_CVAL	0x24	0x0000
TMRx_DIV	0x28	0x0000
TMRx_PR	0x2C	0x0000
TMRx_C1DT	0x34	0x0000 0000
TMRx_C2DT	0x38	0x0000 0000

Control register1 (TMRx_CTRL1)

Bit	Register	Reset value	Type	Description
Bit 15:10	Reserved	0x00	resd	Kept at its default value
Bit 9:8	CLKDIV	0x0	rw	Clock divider 00: Normal 01: Divided by 2 10: Divided by 4 11: Reserved
Bit 7	PRBEN	0x0	rw	Period buffer enable 0: Period buffer is disabled 1: Period buffer is enabled
Bit 6:4	Reserved	0x0	resd	Kept at its default value
Bit 3	OCMEN	0x0	rw	One cycle mode enable This bit is use to select whether to stop counting at an update event 0: The counter does not stop at an update event 1: The counter stops at an update event
Bit 2	OVFS	0x0	rw	Overflow event source This bit is used to select overflow event or DMA request sources. 0: Counter overflow, setting the OVFSWTR bit or overflow event generated by slave timer controller 1: Only counter overflow generates an overflow event
Bit 1	OVFEN	0x0	rw	Overflow event enable 0: Enabled 1: Disabled
Bit 0	TMREN	0x0	rw	TMR enable 0: Enabled 1: Disabled

Slave timer control register (TMRx_STCTRL)

Bit	Register	Reset value	Type	Description
Bit 15:7	Reserved	0x000	resd	Kept at its default value
Bit 6:4	STIS	0x0	rw	Subordinate TMR input selection This field is used to select the subordinate TMR input. 000: Internal selection 0 (IS0) 001: Internal selection 1 (IS1) 010: Internal selection 2 (IS2) 011: Internal selection 3 (IS3) 100: C1IRAW input detector (C1INC) 101: Filtered input 1 (C1IF1) 110: Filtered input 2 (C1IF2) 111: Reserved Pleaser refer to Table 47 for more information on ISx for each timer.
Bit 3	Reserved	0x0	resd	Kept at its default value

Bit	Register	Reset value	Type	Description
Bit 2:0	SMSEL	0x0	rw	Subordinate TMR mode selection 000: Slave mode is disabled 001: Encoder mode A 010: Encoder mode B 011: Encoder mode C 100: Reset mode – Rising edge of the TRGIN input reinitializes the counter 101: Suspend mode – The counter starts counting when the TRGIN is high 110: Trigger mode – A trigger event is generated at the rising edge of the TRGIN input 111: External clock mode A – Rising edge of the TRGIN input clocks the counter Note: Please refer to count mode section for details on encoder mode A/B/C.

DMA/interrupt enable register (TMRx_IDEN)

Bit	Register	Reset value	Type	Description
Bit 15:7	Reserved	0x0	resd	Kept at its default value.
Bit 6	TIEN	0x0	rw	Trigger interrupt enable 0: Disabled 1: Enabled
Bit 5:3	Reserved	0x0	resd	Kept at its default value.
Bit 2	C2IEN	0x0	rw	Channel 2 interrupt enable 0: Disabled 1: Enabled
Bit 1	C1IEN	0x0	rw	Channel 1 interrupt enable 0: Disabled 1: Enabled
Bit 0	OVFIEN	0x0	rw	Overflow interrupt enable 0: Disabled 1: Enabled

Interrupt status register (TMRx_ISTS)

Bit	Register	Reset value	Type	Description
Bit 15:11	Reserved	0x0	resd	Kept at its default value.
Bit 10	C2RF	0x0	rw0c	Channel 2 recapture flag Please refer to C1RF description.
Bit 9	C1RF	0x0	rw0c	Channel 1 recapture flag This bit indicates whether a recapture is detected when C1IF = 1. This bit is set by hardware, and cleared by writing “0”. 0: No capture is detected 1: Capture is detected
Bit 8:7	Reserved	0x0	resd	Kept at its default value.

Bit	Register	Reset value	Type	Description
Bit 6	TRGIF	0x0	rw0c	Trigger interrupt flag This bit is set by hardware on a trigger event. It is cleard by writing “0”. 0: No trigger event occurs 1: Trigger event is generated Trigger event: an active edge is detected on TRGIN input, or any edge in suspend mode.
Bit 5:3	Reserved	0x0	resd	Kept at its default value.
Bit 2	C2IF	0x0	rw0c	Channel 2 interrupt flag Please refer to C1IF description.
Bit 1	C1IF	0x0	rw0c	Channel 1 interrupt flag If the channel 1 is configured as input mode: This bit is set by hardware on a capture event. It is cleared by software or read access to the TMRx_C1DT 0: No capture event occurs 1: Capture event is generated If the channel 1 is configured as output mode: This bit is set by hardware on a compare event. It is cleared by software. 0: No compare event occurs 1: Compare event is generated
Bit 0	OVFIF	0x0	rw0c	Overflow interrupt flag This bit is set by hardware on an overflow event. It is cleared by software. 0: No overflow event occurs 1: Overflow event is generated

Software event register (TMRx_SWEVT)

Bit	Register	Reset value	Type	Description
Bit 15:7	Reserved	0x000	resd	Kept at its default value.
Bit 6	TRGSWTR	0x0	rw	Trigger event triggered by software This bit is set by software to generate a trigger event. 0: No effect 1: Generate a trigger event
Bit 5:3	Reserved	0x0	resd	Kept at its default value.
Bit 2	C2SWTR	0x0	wo	Channel 2 event triggered by software Please refer to C1M description
Bit 1	C1SWTR	0x0	wo	Channel 1 event triggered by software This bit is set by software to generate a channel 1 event. 0: No effect 1: Generate a channel 1 event
Bit 0	OVFSWTR	0x0	wo	Overflow event triggered by software This bit is set by software to generate an overflow event. 0: No effect 1: Generate an overflow event.

Channel mode register1 (TMRx_CM1)

The channel can be used in input (capture mode) or output (compare mode). The direction of a channel is defined by the corresponding CxC bits. All the other bits of this register have different functons in input and output modes. The CxOx describes its function in output mode when the channel is in output mode, while the CxIx describes its function in output mode when the channel is in input mode. Attention must be given to the fact that the same bit can have different functions in input mode and output mode.

Output compare mode:

Bit	Register	Reset value	Type	Description
Bit 15	Reserved	0x0	resd	Kept at its default value.
Bit 14:12	C2OCTRL	0x0	rw	Channel 2 output control
Bit 11	C2OBEN	0x0	rw	Channel 2 output buffer enable
Bit 10	C2OIEN	0x0	rw	Channel 2 output enable immediately
Bit 9:8	C2C	0x0	rw	Channel 2 configuration This field is used to define the direction of the channel 2 (input or output), and the selection of input pin when C2EN='0': 00: Output 01: Input, C2IN is mapped on C2IRAW 10: Input, C2IN is mapped on C1IRAW 11: Input, C2IN is mapped on STCI. This mode works only when the internal trigger input is selected by STIS register.
Bit 7	Reserved	0x0	resd	Kept at its default value.
Bit 6:4	C1OCTRL	0x0	rw	Channel 1 output control This field defines the behavior of the original signal C1ORAW. 000: Disconnected. C1ORAW is disconnected from C1OUT; 001: C1ORAW is high when TMRx_CVAL=TMRx_C1DT 010: C1ORAW is low when TMRx_CVAL=TMRx_C1DT 011: Switch C1ORAW level when TMRx_CVAL=TMRx_C1DT 100: C1ORAW is forced low 101: C1ORAW is forced high. 110: PWM mode A - OWCDIR=0, C1ORAW is high once TMRx_C1DT > TMRx_CVAL, else low; - OWCDIR=1, C1ORAW is low once TMRx_C1DT < TMRx_CVAL, else high; 111: PWM mode B - OWCDIR=0, C1ORAW is low once TMRx_C1DT > TMRx_CVAL, else high; - OWCDIR=1, C1ORAW is high once TMRx_C1DT < TMRx_CVAL, else low. Note: In the configurations othern than 000', the C1OUT is connected to C1ORAW. The C1OUT output level is not only subject to the changes of C1ORAW, but also the output polarity set by CCTRL.

Bit	Register	Reset value	Type	Description
Bit 3	C1OBEN	0x0	rw	Channel 1 output buffer enable 0: Buffer function of TMRx_C1DT is disabled. The new value written to the TMRx_C1DT takes effect immediately. 1: Buffer function of TMRx_C1DT is enabled. The value to be written to the TMRx_C1DT is stored in the buffer register, and can be sent to the TMRx_C1DT register only on an overflow event.
Bit 2	C1OIEN	0x0	rw	Channel 1 output enable immediately In PWM mode A or B, this bit is used to accelerate the channel 1 output's response to the trigger event. 0: Need to compare the CVAL with C1DT before generating an output 1: No need to compare the CVAL and C1DT. An output is generated immediately when a trigger event occurs.
Bit 1:0	C1C	0x0	rw	Channel 1 configuration This field is used to define the direction of the channel 1 (input or output), and the selection of input pin when C1EN='0': 00: Output 01: Input, C1IN is mapped on C1IRAW 10: Input, C1IN is mapped on C2IRAW 11: Input, C1IN is mapped on STCI. This mode works only when the internal trigger input is selected by STIS.

Input capture mode:

Bit	Register	Reset value	Type	Description
Bit 15:12	C2DF	0x0	rw	Channel 2 digital filter
Bit 11:10	C2IDIV	0x0	rw	Channel 2 input divider
Bit 9:8	C2C	0x0	rw	Channel 2 configuration This field is used to define the direction of the channel 2 (input or output), and the selection of input pin when C2EN='0': 00: Output 01: Input, C2IN is mapped on C2IRAW 10: Input, C2IN is mapped on C1IRAW 11: Input, C2IN is mapped on STCI. This mode works only when the internal trigger input is selected by STIS.
Bit 7:4	C1DF	0x0	rw	Channel 1 digital filter This field defines the digital filter of the channel 1. N stands for the number of filtering, indicating that the input edge can pass the filter only after N sampling events. 0000: No filter, sampling is done at f_{DTS} 1000: $f_{SAMPLING} = f_{DTS}/8$, N=6 0001: $f_{SAMPLING} = f_{CK_INT}$, N=2 1001: $f_{SAMPLING} = f_{DTS}/8$, N=8 0010: $f_{SAMPLING} = f_{CK_INT}$, N=4 1010: $f_{SAMPLING} = f_{DTS}/16$, N=5 0011: $f_{SAMPLING} = f_{CK_INT}$, N=8 1011: $f_{SAMPLING} = f_{DTS}/16$, N=6 0100: $f_{SAMPLING} = f_{DTS}/2$, N=6 1100: $f_{SAMPLING} = f_{DTS}/16$, N=8 0101: $f_{SAMPLING} = f_{DTS}/2$, N=8 1101: $f_{SAMPLING} = f_{DTS}/32$, N=5 0110: $f_{SAMPLING} = f_{DTS}/4$, N=6 1110: $f_{SAMPLING} = f_{DTS}/32$, N=6 0111: $f_{SAMPLING} = f_{DTS}/4$, N=8 1111: $f_{SAMPLING} = f_{DTS}/32$, N=8
Bit 3:2	C1IDIV	0x0	rw	Channel 1 input divider This field defines Channel 1 input divider. 00: No divider. An input capture is generated at each active edge. 01: An input compare is generated every 2 active edges 10: An input compare is generated every 4 active edges 11: An input compare is generated every 8 active edges Note: the divider is reset once C1EN='0'
Bit 1:0	C1C	0x0	rw	Channel 1 configuration This field is used to define the direction of the channel 1 (input or output), and the selection of input pin when C1EN='0': 00: Output 01: Input, C1IN is mapped on C1IRAW 10: Input, C1IN is mapped on C2IRAW 11: Input, C1IN is mapped on STCI. This mode works only when the internal trigger input is selected by STIS.

Channel control register (TMRx_CCTRL)

Bit	Register	Reset value	Type	Description
Bit 15:6	Reserved	0x0	resd	Kept at its default value.
Bit 5	C2P	0x0	rw	Channel 2 polarity Pleaser refer to C1P description.
Bit 4	C2EN	0x0	rw	Channel 2 enable Pleaser refer to C1EN description.
Bit 3:2	Reserved	0x0	resd	Kept at its default value.
Bit 1	C1P	0x0	rw	Channel 1 polarity When the channel 1 is configured as output mode: 0: C1OUT is active high 1: C1OUT is active low When the channel 1 is configured as input mode: 0: C1IN active edge is on its rising edge. When used as external trigger, C1IN is not inverted. 1: C1IN active edge is on its falling edge. When used as external trigger, C1IN is inverted.
Bit0	C1EN	0x0	rw	Channel 1 enable 0: Input or output is disabled 1: Input or output is enabled

Table 53. Standard CxOUT channel output control bit

CxEN bit	CxOUT output state
0	Output disabled (CxOUT=0)
1	CxOUT = CxORAW + polarity

Note: The state of the external I/O pins connected to the standard CxOUT channel depends on the CxOUT channel state and the GPIO and IOMUX registers.

Counter value (TMRx_CVAL)

Bit	Register	Reset value	Type	Description
Bit 15:0	CVAL	0x0000	rw	Counter value

Division value (TMRx_DIV)

Bit	Register	Reset value	Type	Description
Bit 15:0	DIV	0x0000	rw	Divider value The counter clock frequency $f_{CK_CNT} = f_{TMR_CLK}$ /(DIV[15:0]+1). DIV contains the value written at an overflow event.

Period register (TMRx_PR)

Bit	Register	Reset value	Type	Description
Bit 15:0	PR	0x0000	rw	Period value This defines the period value of the TMRx counter. The timer stops working when the period value is 0.

Channel 1 data register (TMRx_C1DT)

Bit	Register	Reset value	Type	Description
Bit 31:16	Reserved	0x0000	resd	Kept at its default value.
Bit 15:0	C1DT	0x0000	rw	Channel 1 data register When the channel 1 is configured as input mode: The C1DT is the CVAL value stored by the last channel 1 input event (C1IN) When the channel 1 is configured as output mode: C1DT is the value to be compared with the CVAL value. Whether the written value takes effective immediately depends on the C1OBEN bit, and the corresponding output is generated on C1OUT as configured.

Channel 2 data register (TMRx_C2DT)

Bit	Register	Reset value	Type	Description
Bit 31:16	C2DT	0x0000	resd	Kept at its default value.
Bit 15:0	C2DT	0x0000	rw	Channel 2 data register When the channel 2 is configured as input mode: The C2DT is the CVAL value stored by the last channel 2 input event (C1IN) When the channel 2 is configured as output mode: C2DT is the value to be compared with the CVAL value. Whether the written value takes effective immediately depends on the C2OBEN bit, and the corresponding output is generated on C2OUT as configured.

TMR10,TMR11, TMR13 and TMR14 registers

These peripheral registers must be accessed by words (32 bits). All TMRx register are mapped into a 1-bit addressable space.

Table 54. TMRx register map and reset value

Register	Offset	Reset value
TMRx_CTRL1	0x00	0x0000
TMRx_IDEN	0x0C	0x0000
TMRx_ISTS	0x10	0x0000
TMRx_SWEVT	0x14	0x0000
TMRx_CM1	0x18	0x0000
TMRx_CCTRL	0x20	0x0000
TMRx_CVAL	0x24	0x0000
TMRx_DIV	0x28	0x0000
TMRx_PR	0x2C	0x0000
TMRx_C1DT	0x34	0x0000

Control register1 (TMRx_CTRL1)

Bit	Register	Reset value	Type	Description
Bit 15:10	Reserved	0x00	resd	Kept at its default value
Bit 9:8	CLKDIV	0x0	rw	Clock divider 00: Normal 01: Divided by 2 10: Divided by 4 11: Reserved
Bit 7	PRBEN	0x0	rw	Period buffer enable 0: Period buffer is disabled 1: Period buffer is enabled
Bit 6:4	Reserved	0x0	resd	Default value
Bit 3	OCMEN	0x0	rw	One cycle mode enable This bit is use to select whether to stop counting at an update event 0: The counter does not stop at an update event 1: The counter stops at an update event
Bit 2	OVFS	0x0	rw	Overflow event source This bit is used to select overflow event or DMA request sources. 0: Counter overflow, setting the OVFSWTR bit or overflow event generated by slave timer controller 1: Only counter overflow generates an overflow event
Bit 1	OVFEN	0x0	rw	Overflow event enable 0: Enabled 1: Disabled
Bit 0	TMREN	0x0	rw	TMR enable 0: Enabled 1: Disabled

DMA/interrupt enable register (TMRx_IDEN)

Bit	Register	Reset value	Type	Description
Bit 15:2	Reserved	0x0	resd	Kept at its default value
Bit 1	C1IEN	0x0	rw	Channel 1 interrupt enable 0: Disabled 1: Enabled
Bit 0	OVFIEN	0x0	rw	Overflow interrupt enable 0: Disabled 1: Enabled

Interrupt status register (TMRx_ISTS)

Bit	Register	Reset value	Type	Description
Bit 15:10	Reserved	0x0	resd	Kept at its default value.
Bit 9	C1RF	0x0	rw0c	Channel 1 recapture flag This bit indicates whether a recapture is detected when C1IF=1. This bit is set by hardware, and cleared by writing "0". 0: No capture is detected 1: Capture is detected
Bit 8:2	Reserved	0x0	resd	Kept at its default value.
Bit 1	C1IF	0x0	rw0c	Channel 1 interrupt flag If the channel 1 is configured as input mode: This bit is set by hardware on a capture event. It is cleared by software or read access to the TMRx_C1DT 0: No capture event occurs 1: Capture event is generated If the channel 1 is configured as output mode: This bit is set by hardware on a compare event. It is cleared by software. 0: No compare event occurs 1: Compare event is generated
Bit 0	OVFIF	0x0	rw0c	Overflow interrupt flag This bit is set by hardware on an overflow event. It is cleared by software. 0: No overflow event occurs 1: Overflow event is generated

Software event register (TMRx_SWEVT)

Bit	Register	Reset value	Type	Description
Bit 15:2	Reserved	0x000	resd	Kept at its default value.
Bit 1	C1SWTR	0x0	wo	Channel 1 event triggered by software This bit is set by software to generate a channel 1 event. 0: No effect 1: Generate a channel 1 event.
Bit 0	OVFSWTR	0x0	wo	Overflow event triggered by software This bit is set by software to generate an overflow event. 0: No effect 1: Generate an overflow event.

Channel mode register1 (TMRx_CM1)

The channel can be used in input (capture mode) or output (compare mode). The direction of a channel is defined by the corresponding CxC bits. All the other bits of this register have different functons in input and output modes. The CxOx describes its function in output mode when the channel is in output mode, while the CxIx describes its function in output mode when the channel is in input mode. Attention must be given to the fact that the same bit can have different functions in input mode and output mode.

Output compare mode:

Bit	Register	Reset value	Type	Description
Bit 15:7	Reserved	0x0	resd	Kept at its default value.
Bit 6:4	C1OCTRL	0x0	rw	Channel 1 output control This field defines the behavior of the original signal C1ORAW. 000: Disconnected. C1ORAW is disconnected from C1OUT; 001: C1ORAW is high when TMRx_CVAL=TMRx_C1DT 010: C1ORAW is low when TMRx_CVAL=TMRx_C1DT 011: Switch C1ORAW level when TMRx_CVAL=TMRx_C1DT 100: C1ORAW is forced low 101: C1ORAW is forced high. 110: PWM mode A - OWCDIR=0, C1ORAW is high once TMRx_C1DT > TMRx_CVAL, else low; - OWCDIR=1, C1ORAW is low once TMRx_C1DT < TMRx_CVAL, else high; 111: PWM mode B - OWCDIR=0, C1ORAW is low once TMRx_C1DT > TMRx_CVAL, else high; - OWCDIR=1, C1ORAW is high once TMRx_C1DT < TMRx_CVAL, else low. Note: In the configurations othern than 000', the C1OUT is connected to C1ORAW. The C1OUT output level is not only subject to the changes of C1ORAW, but also the output polarity set by CCTRL.
Bit 3	C1OBEN	0x0	rw	Channel 1 output buffer enable 0: Buffer function of TMRx_C1DT is disabled. The new value written to the TMRx_C1DT takes effect immediately. 1: Buffer function of TMRx_C1DT is enabled. The value to be written to the TMRx_C1DT is stored in the buffer register, and can be sent to the TMRx_C1DT register only on an overflow event.
Bit 2	C1OIEN	0x0	rw	Channel 1 output enable immediately In PWM mode A or B, this bit is used to accelerate the channel 1 output's response to the trigger event. 0: Need to compare the CVAL with C1DT before generating an output 1: No need to compare the CVAL and C1DT. An output is generated immediately when a trigger event occurs.
Bit 1:0	C1C	0x0	rw	Channel 1 configuration This field is used to define the direction of the channel 1 (input or output), and the selection of input pin when C1EN='0': 00: Output 01: Input, C1IN is mapped on C1IRAW 10: Input, C1IN is mapped on C2IRAW 11: Input, C1IN is mapped on STCI. This mode works only when the internal trigger input is selected by STIS.

Input capture mode:

Bit	Register	Reset value	Type	Description
Bit 15:8	Reserved	0x0	resd	Kept at its default value.
Bit 7:4	C1DF	0x0	rw	Channel 1 digital filter This field defines the digital filter of the channel 1. N stands for the number of filtering, indicating that the input edge can pass the filter only after N sampling events. 0000: No filter, sampling is done at f_{DTS} 1000: $f_{SAMPLING} = f_{DTS}/8$, N=6 0001: $f_{SAMPLING} = f_{CK_INT}$, N=2 1001: $f_{SAMPLING} = f_{DTS}/8$, N=8 0010: $f_{SAMPLING} = f_{CK_INT}$, N=4 1010: $f_{SAMPLING} = f_{DTS}/16$, N=5 0011: $f_{SAMPLING} = f_{CK_INT}$, N=8 1011: $f_{SAMPLING} = f_{DTS}/16$, N=6 0100: $f_{SAMPLING} = f_{DTS}/2$, N=6 1100: $f_{SAMPLING} = f_{DTS}/16$, N=8 0101: $f_{SAMPLING} = f_{DTS}/2$, N=8 1101: $f_{SAMPLING} = f_{DTS}/32$, N=5 0110: $f_{SAMPLING} = f_{DTS}/4$, N=6 1110: $f_{SAMPLING} = f_{DTS}/32$, N=6 0111: $f_{SAMPLING} = f_{DTS}/4$, N=8 1111: $f_{SAMPLING} = f_{DTS}/32$, N=8
Bit 3:2	C1IDIV	0x0	rw	Channel 1 input divider This field defines Channel 1 input divider. 00: No divider. An input capture is generated at each active edge. 01: An input compare is generated every 2 active edges 10: An input compare is generated every 4 active edges 11: An input compare is generated every 8 active edges Note: the divider is reset once C1EN=’0’
Bit 1:0	C1C	0x0	rw	Channel 1 configuration This field is used to define the direction of the channel 1 (input or output), and the selection of input pin when C1EN=’0’: 00: Output 01: Input, C1IN is mapped on C1IRAW 10: Input, C1IN is mapped on C2IRAW 11: Input, C1IN is mapped on STCI. This mode works only when the internal trigger input is selected by STIS.

Channel control register (TMRx_CCTRL)

Bit	Register	Reset value	Type	Description
Bit 15:2	Reserved	0x0	resd	Kept at its default value.
Bit 1	C1P	0x0	rw	Channel 1 polarity When the channel 1 is configured as output mode: 0: C1OUT is active high 1: C1OUT is active low When the channel 1 is configured as input mode: 0: C1IN active edge is on its rising edge. When used as external trigger, C1IN is not inverted. 1: C1IN active edge is on its falling edge. When used as external trigger, C1IN is inverted.
Bit0	C1EN	0x0	rw	Channel 1 enable 0: Input or output is disabled 1: Input or output is enabled

Table 55. Standard CxOUT channel output control bit

CxEN bit	CxOUT output state
0	Output disabled (CxOUT=0)
1	CxOUT = CxORAW + polarity

Note: The state of the external I/O pins connected to the standard CxOUT channel depends on the CxOUT channel state and the GPIO and IOMUX registers.

Counter value (TMRx_CVAL)

Bit	Register	Reset value	Type	Description
Bit 15:0	CVAL	0x0000	rw	Counter value

Division value (TMRx_DIV)

Bit	Register	Reset value	Type	Description
Bit 15:0	DIV	0x0000	rw	Divider value The counter clock frequency $f_{CK_CNT} = f_{TMR_CLK}$ /(DIV[15:0]+1). DIV contains the value written at an overflow event.

Period register (TMRx_PR)

Bit	Register	Reset value	Type	Description
Bit 15:0	PR	0x0000	rw	Period value This defines the period value of the TMRx counter. The timer stops working when the period value is 0.

Channel 1 data register (TMRx_C1DT)

Bit	Register	Reset value	Type	Description
Bit 15:0	C1DT	0x0000	rw	Channel 1 data register When the channel 1 is configured as input mode: The C1DT is the CVAL value stored by the last channel 1 input event (C1IN) When the channel 1 is configured as output mode: C1DT is the value to be compared with the CVAL value. Whether the written value takes effective immediately depends on the C1OBEN bit, and the corresponding output is generated on C1OUT as configured.

Advanced-control timers (TMR1 and TMR8)

TMR1 and TMR8 introduction

Each of the advanced-control timer (TMR1 and TMR8) consists of a 16-bit counter supporting up, down, up/down (bidirectional) counting modes, four capture/compare registers, and four independent channels to achieve embedded dead-time, input capture and programmable PWM output.

TMR1 and TMR8 main features

The main functions of general-purpose TMR1 and TMR8 include:

- Souce of counter clock: internal clock, external clock an internal trigger input
- 16-bit up, down, up/down, repetition and encoder mode counter
- 4 independent channels for input capture, output compare, PWM generation, one-pulse mode output and embedded dead-time
- 3 independent channesl for complementary output
- TMR break function
- Synchronization control between master and slave timers
- Interrrupt/DMA is generated at overflow event, trigger event, break signal input and channel event
- Support TMR burst DMA transfer

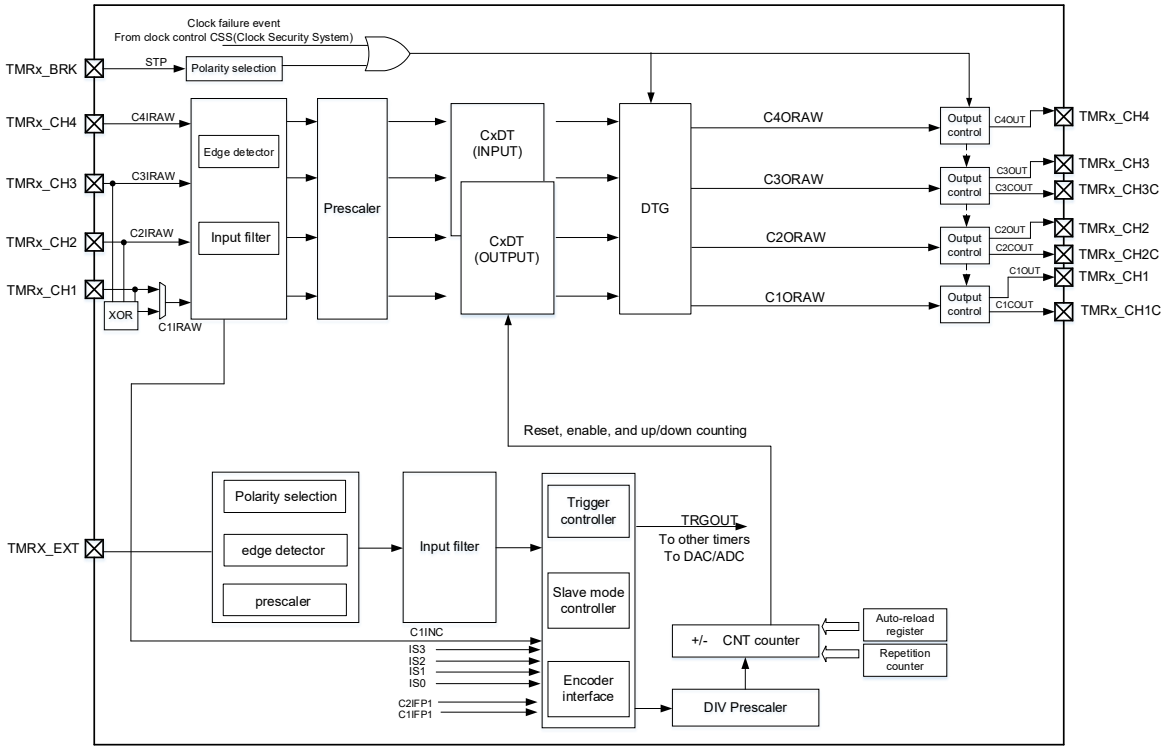


Figure 103. Block diagram of advanced-control timer

TMR1 and TMR8 functional descriptions

Count clock

The count clock of TMR1 and TMR8 can be provided by the internal clock (CK_INT), external clock (external clock mode A and B) and internal trigger input (ISx)

Internal clock (CK_INT)

By default, the CK_INT divided by the prescaler is used to drive the counter to start counting.

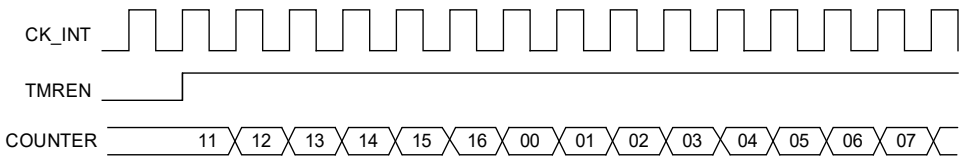


Figure 104. Control circuit with CK_INT divided by 1

External clock (TRGIN/EXT)

The counter clock can be provided by two external clock sources, namely, TRGIN and EXT signals.

When SMSEL=3'111, external clock mode A is selected. Set the STIS[2:0] bit to select TRGIN signal to drive the counter to start counting.

When ECMBEN=1, external clock mode B is selected. The counter starts counting driven by EXT signal.

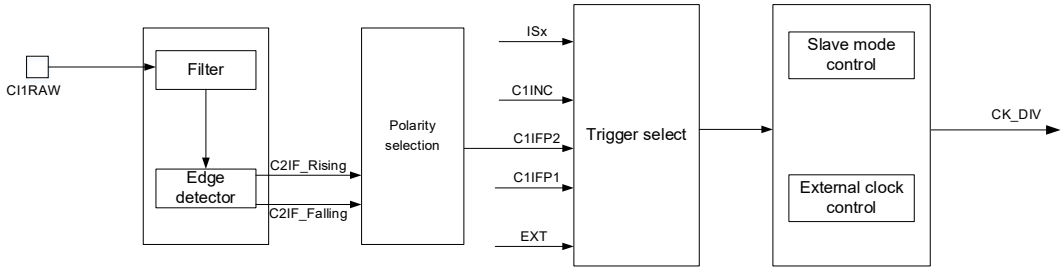


Figure 105. Block diagram of external clock mode A

Note: The delay between the signal on the input side and the actual clock of the counter is due to the synchronization circuit.

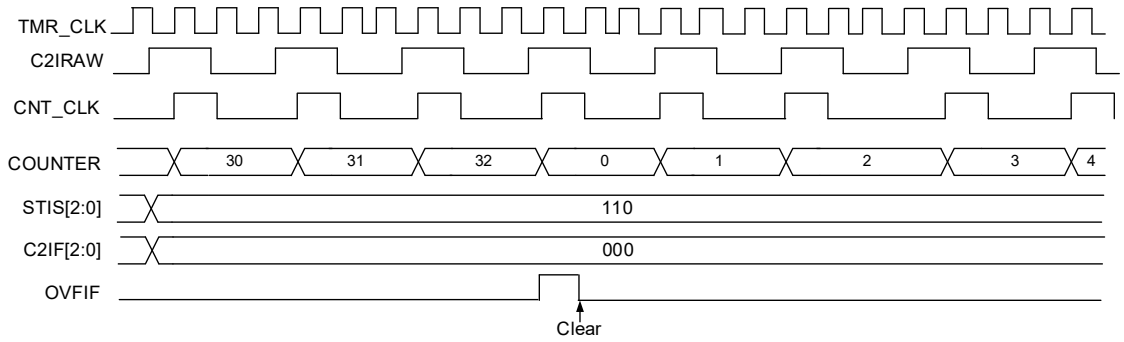


Figure 106. Counting in external clock mode A

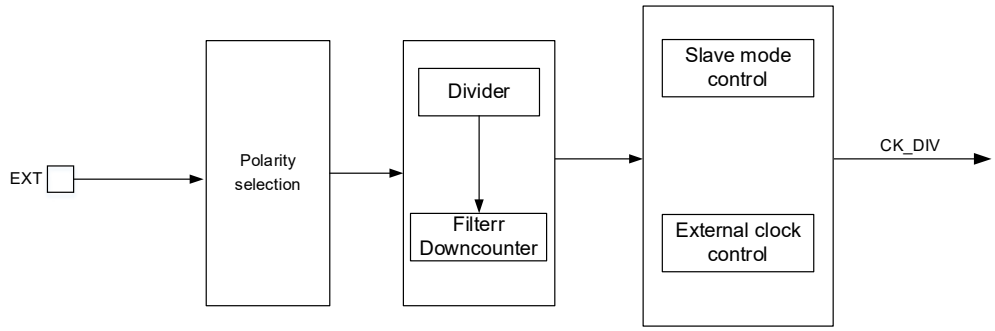


Figure 107. Block diagram of external clock mode B

Note: The delay between the ext signal on the input side and the actual clock of the counter is due to the synchronization circuit.

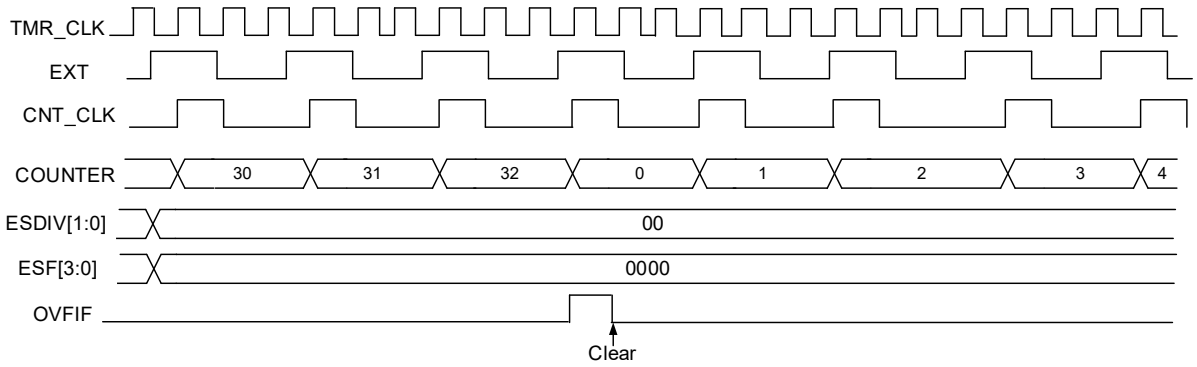


Figure 108. Counting in external clock mode B

Internal trigger input (ISx)

Timer synchronization allows interconnection between several timers. The TMR_CLK of one timer can be provided by the TRGOUT signal output by another timer. Set the STIS[2:0] bit to select internal trigger signal to enable counting.

Each timer consists of a 16-bit prescaler, which is used to generate the CK_CNT that enables the counter to count. The frequency division relationship between the CK_CNT and TMR_CLK can be adjusted by setting the value of the TMRx_DIV register. The prescaler value can be modified at any time, but it takes effect only when the next overflow event occurs.

Table 56. TMRx internal trigger connection

Slave timer	IS0 (STIS=000)	IS1 (STIS=001)	IS2 (STIS=010)	IS3 (STIS=011)
TMR1	TMR5	TMR2	TMR3	TMR4
TMR8	TMR1	TMR2	TMR4	TMR5
Reserved	TMR8	TMR2	TMR4	TMR5

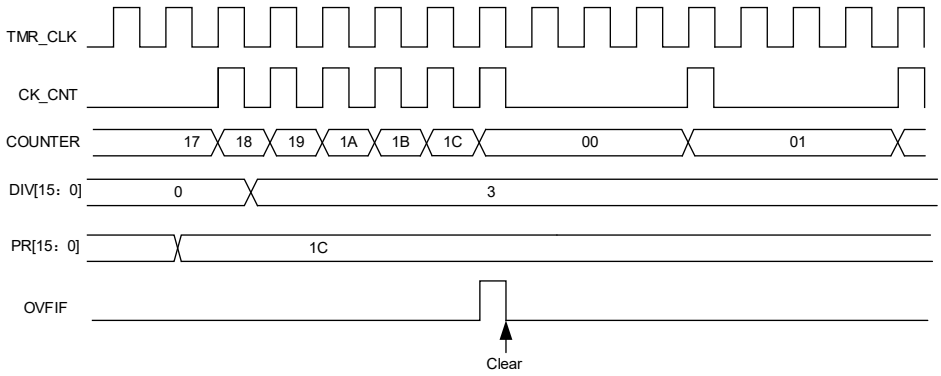


Figure 109. Counter timing with prescaler value changing from 1 to 4

Counting mode

The advanced-control timer consists of a 16-bit counter supporting up, down, up/down counting modes. The TMRx_PR register is loaded with the counter value. The value in the TMRx_PR is immediately moved to the shadow register by deault. When the periodic buffer is enabled (PRBEN=1), the value in the TMRx_PR register is transferred to the shadow register only at an overflow event. The OVFEN and OVFS bits are used to configure the overflow event.

Settng TMREN=1 to enable the timer to start counting. Base on synchronization logic, however, the actual enable signal TMR_EN is set 1 clock cycle after the TMREN is set.

Upcounting mode

In upcounting mode, the counter counts from 0 to the value programmed in the TMRx_PR register, then restarts from 0, and generates a counter overflow event, with the OVFIF bit being set to 1. If the overflow event is disabled, the counter is no longer reloaded with the preload value and period value at a counter overflow event, otherwise, the counter is updated with the preload value and period value on an overflow event.

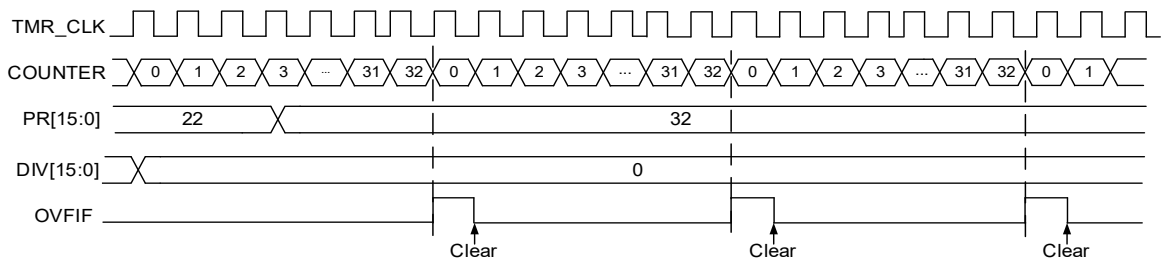


Figure 110. Overflow event when PRBEN=0

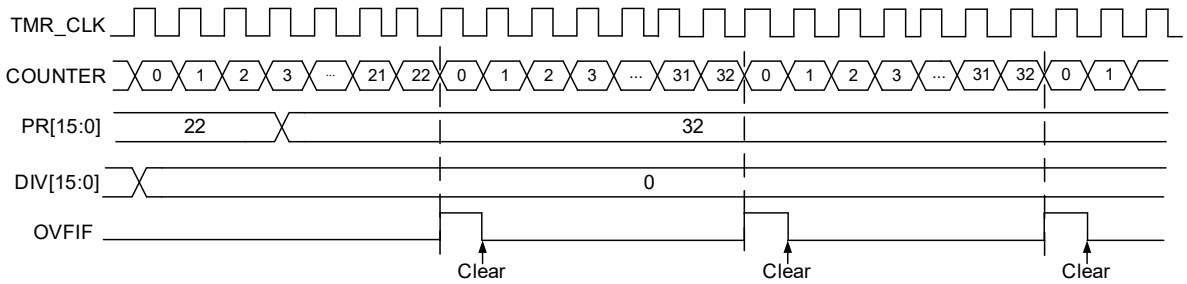


Figure 111. Overflow event when PRBEN=1

Downcounting mode

In downcounting mode, the counter counts from the value programmed in the TMRx_PR register down to 0, and restarts from the value programmed in the TMRx_PR register, and generates a counter underflow event.

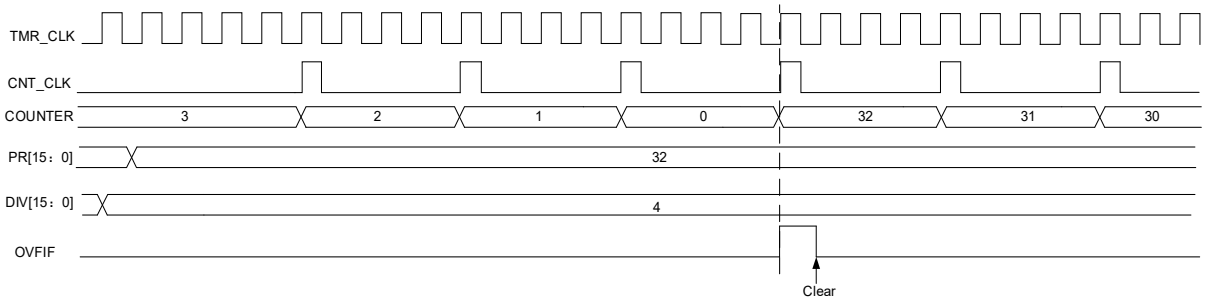


Figure 112. Counter timing diagram with internal clock divided by 4

Up/down counting mode

In up/down counting mode, the counter counts up/down alternatively. When the counter counts from the value programmed in the TMRx_PR register down to 1, an underflow event is generated, and then restarts counting from 0; when the counter counts from 0 to the value of the TMRx_PR register, an overflow event is generated, and then restarts counting from the value of the TMRx_PR register. The OWCDIR bit indicates the current counting direction.

Note: The OWCDIR is ready-only in up/down counting mode.

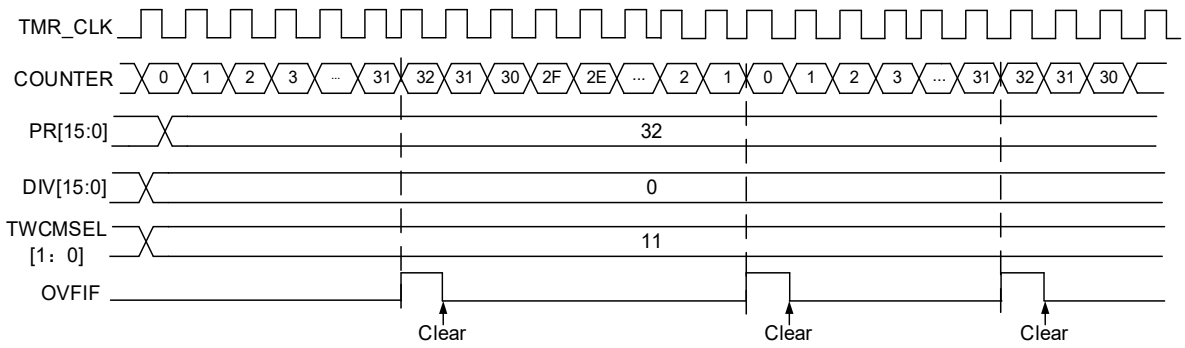


Figure 113. Counter timing diagram with internal clock divided by 1 and TMRx_PR=0x32

Repetition counter mode:

The repletion counter mode is enabled when the repetition counter value is not equal to 0. In this mode, the repetition counter is decremented at each counter overflow. An overflow event is generated when the repetition counter reaches 0. The frequency of the overflow event can be adjusted by setting the repettion counter value.

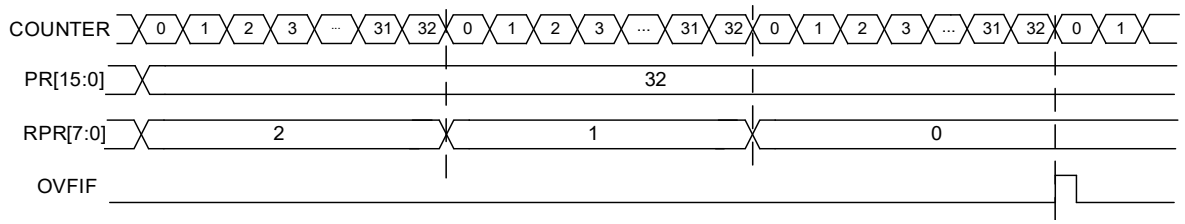


Figure 114. OVFIF when RPR=2

Encoder interface mode

To enble the encoder interface mode, write SMSEL[2:0]= 3’b001/3’b010/3’b011. In this mode, the two inputs (C1IN/C2IN) are required. Depending on the level on one input, the counter counts up or down on the edge of the other input. The OWCDIR bit indicates the direction of the counter, as shown in the table below:

Table 57. Couting direction versus encoder signals

Active edge	Level on opposite signal (C1INFP1 to C2IN, C2INFP2 to C1IN)	C1INFP1 signal		C2INFP2 signal	
		Rising	Falling	Rising	Falling
Count on C1IN only	High	Down	Up	No count	No count
	Low	Up	Down	No count	No count
Count on C2IN only	High	No count	No count	Up	Down
	Low	No count	No count	Down	Up
Count on both C1IN and C2IN	High	Down	Up	Up	Down
	Low	Up	Down	Down	Up

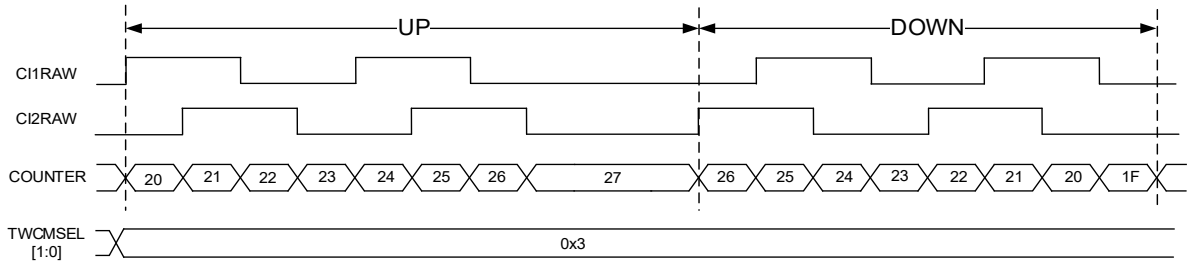


Figure 115. Example of encoder interface mode C

TMR input function

Each timer of TMR1 and TMR8 has four independent channels. Each channel can be configured as input or output. As input, the channel can be used for the filtering, selection, division and input capture of the input signals.

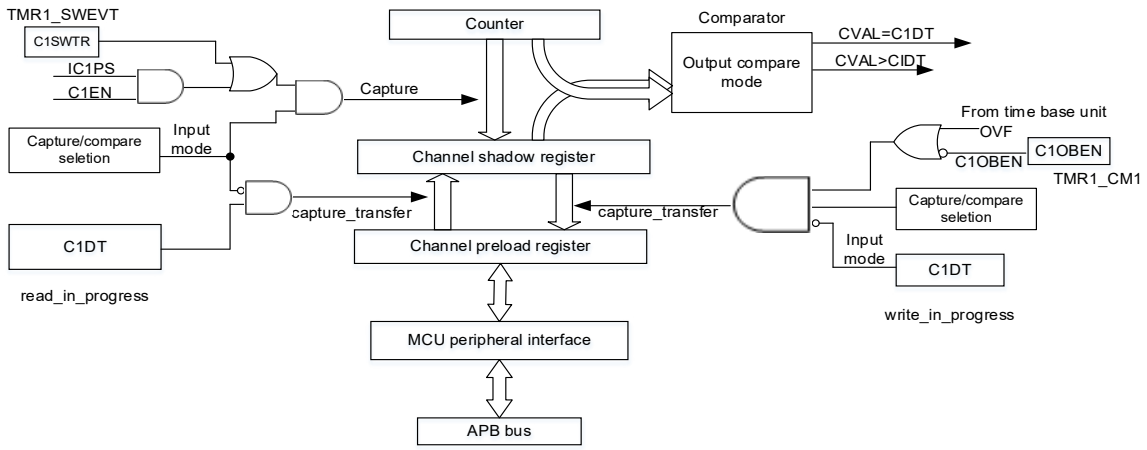


Figure 116. Input/output channel 1 main circuit

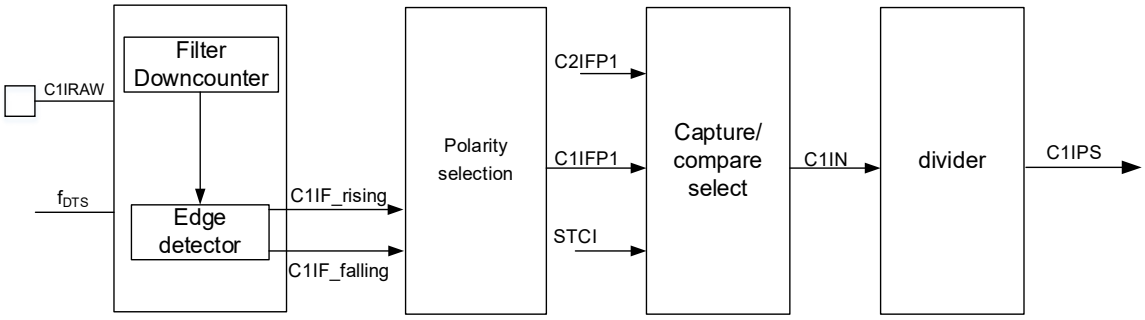


Figure 117. Channel 1 input stage

Input mode

In input mode, the TMRx_CxDT registers latch the current counter values after the selected triggle signal is detected, and the capture compare interrupt flag bit (CxIF) is set. An interrupt/DMA request will be generated if the CxIEN bit and CxDEN bit are enabled. If the selected trigger signal is detected when the CxIF is set to 1, the CxRF is set to 1.

To capture the rising edge of C1IN input, following the configuration procedure mentioned below:

- Set C1C=01 in the TMRx_CxDT register to select the C1IN as channel 1 input
- Set the filter bandwidth of C1IN signal (CxDF[3:0])
- Set the active edge on the C1IN channel by writing C1P=0 (rising edge) in the TMRx_CCTR register
- Program the capture frequency of C1IN signal (C1DIV[1:0])
- Enable channel 1 input capture (C1EN=1)
- If needed, enable the relevant interrupt or DMA request by setting the C1IEN bit in the TMRx_IDEN register or the C1DEN bit in the TMRx_IDEN register

Timer Input XOR function

The timer input pins (TMRx_CH1, TMRx_CH2 and TMRx_CH3) are connected to the channel 1 (selected by setting the C1INSE in the TMRx_CTRL2 register) through an XOR gate.

The XOR gate can be used to connect Hall sensors. For example, connect the three XOR inputs to the three Hall sensors respectively so as to calculate the position and speed of the rotation by analyzing three Hall sensor signals.

TMR output function

The TMR output consists of a comparator and an output controller. It is used to program the period, duty cycle and polarity of the output signal. The advanced-control timer output function varies from one channel to one channel.

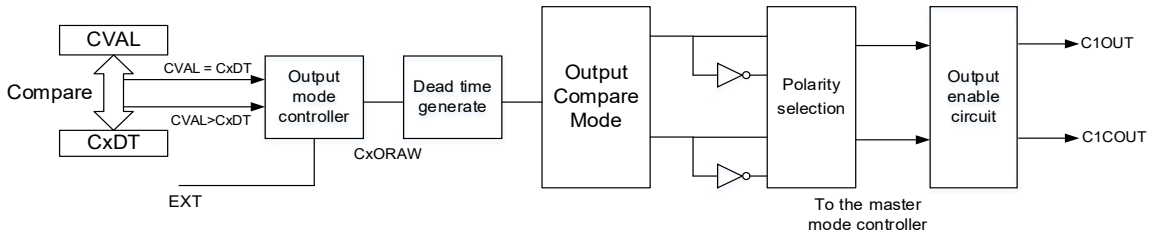


Figure 118. Channel output stage (channel 1 to 3)

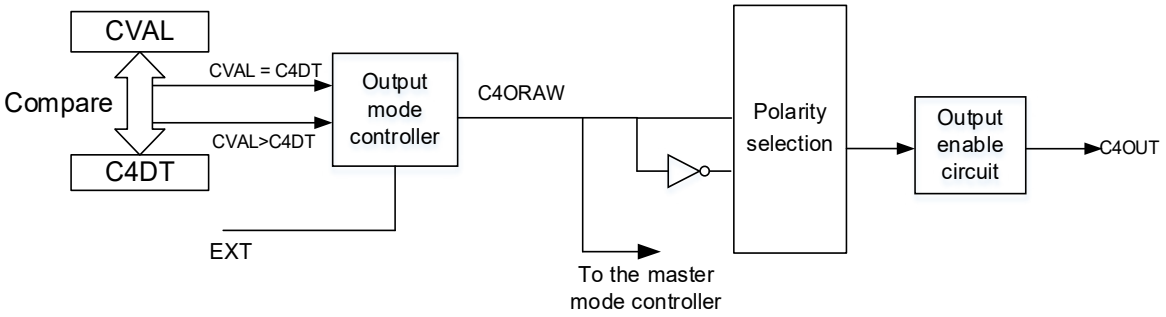


Figure 119. Channel 4 output stage

Output mode

Write CxC[2:0]≠2'b00 to configure the channel as output to implement multiple output modes. In this case, the counter value is compared with the value in the TMRx_CxDT register, and the intermediate signal CxORAW is generated according to the output mode selected by CxOCTRL[2:0], which is sent to IO after being processed by the output control circuit. The period of the output signal is configured by the TMRx_PR register, while the duty cycle by the TMRx_CxDT register.

Output compare modes include:

- PWM mode: Set CxOCTRL=2'b110/111 to enable PWM mode. Each channel can be independently configured to output one PWM signal. In this case, the period of the output signal is configured by the TMRx_PR register, and the duty cycle by the CxDT register. The counter value is compared with the value of the TMRx_CxDT register, and the corresponding level signal is sent according to the counting direction. For more information on PWM mode A/B, refer to the description of the CxOCTRL[2:0] bit. In up/down counting mode, the OWCDIR bit is used to indicate the counting direction.
- Forced output mode: Set CxOCTRL=2'b100/101 to enable forced output mode. In this case, the CxORAW is forced to be the programmed level, irrespective of the counter value. Despite this, the channel flag bit and DMA request still depend on the compare result.
- Output compare mode: Set CxOCTRL=2'b001/010/011 to enable output compare mode. In this case, when the counter value matches the value of the CxDT register, the CxORAW is forced high, low or toggling.
- One-pulse mode: This is a particular case of PWM mode. Set OCMEN=1 to enable one-pulse mode. In this mode, the comparison match is performed in the current counting period. The TMREN bit is cleared as soon as the current counting is completed. Therefore, only one pulse is output. When configured as in upcounting mode, the configureation must follow the rule: CVAL<CxDT≤PR; in downcounting mode, CVAL>CxDT is required.
- Fast output mode: Set CxOIEN=1 to enable this mode. If enabled, the CxORAW signal will not change when the counter value matches the CxDT, but at the beginning of the current counting period. In other words, the comparison result is advanced, so the comparison result between the counter value and the TMRx_CxDT register will determine the level of CxORAW in advance.

Figure 120 gives an example of output compare mode (toggle) with C1DT=0x3. When the counter value is equal to 0x3, C1OUT toggles.

Figure 121 gives an example of the combination between upcounting mode and PWM mode A. The output signal behaves when PR=0x32 but CxDT is configured with a different value.

Figure 122 gives an example of the combination between up/down counting mode and PWM mode A. The output signal behaves when PR=0x32 but CxDT is configured with a different value.

Figure 123 gives an example of the combination between upcounting mode and one-pulse PWM mode B. The counter only counts only one cycle, and the output signal sents only one pulse.

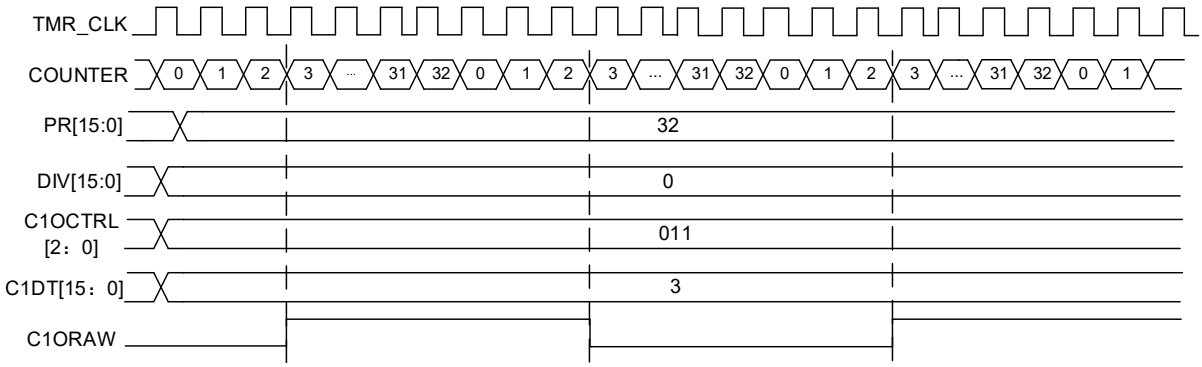


Figure 120. C1ORAW toggles when counter value matches the C1DT value

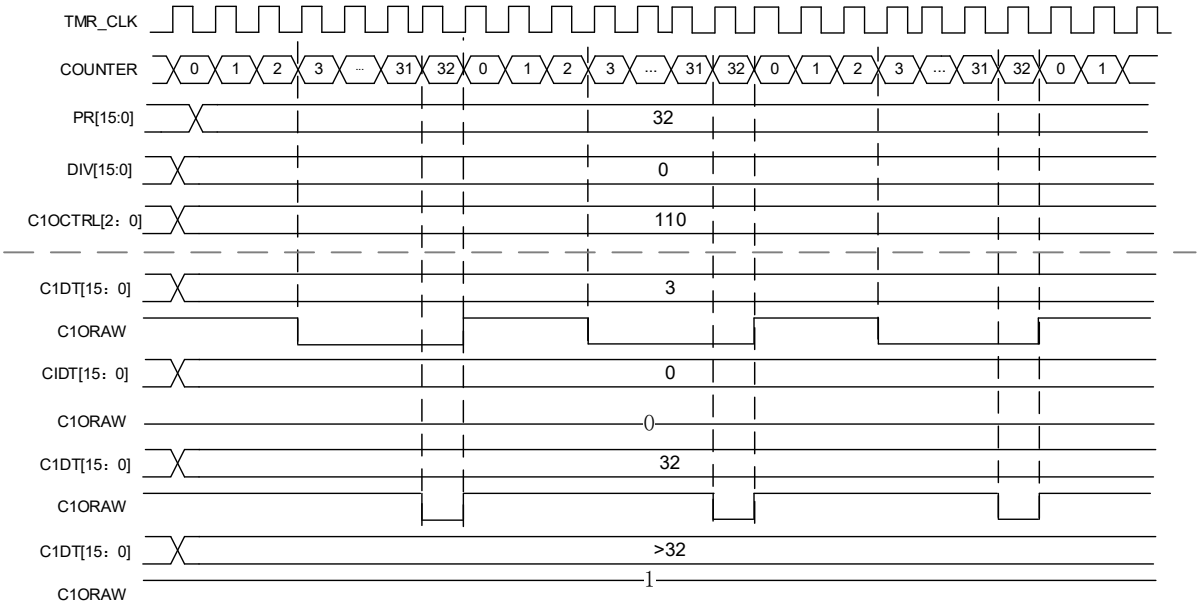


Figure 121. Upcounting mode and PWM mode A

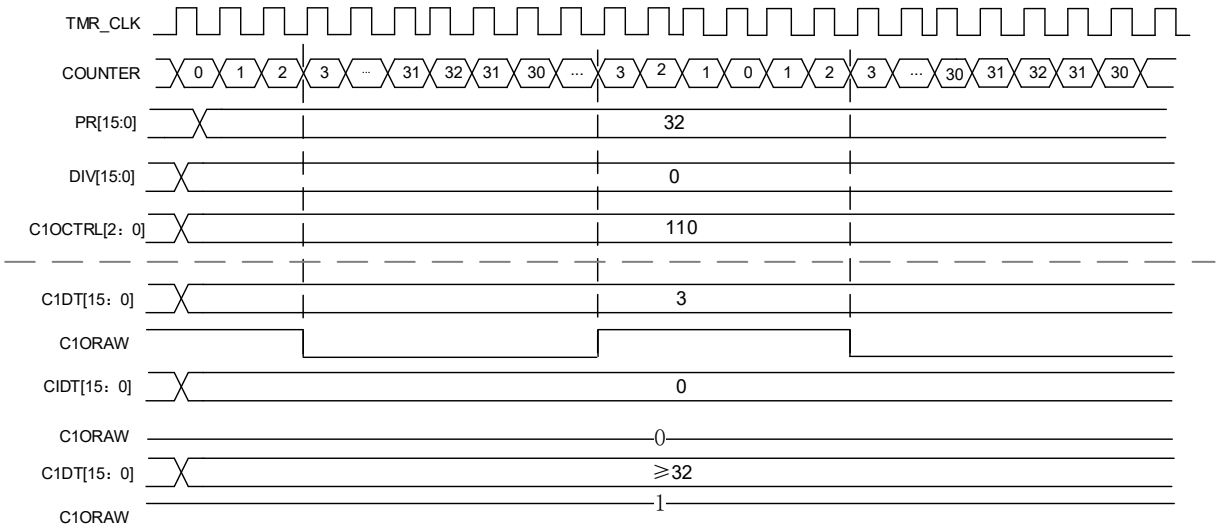


Figure 122. Up/down counting mode and PWM mode A

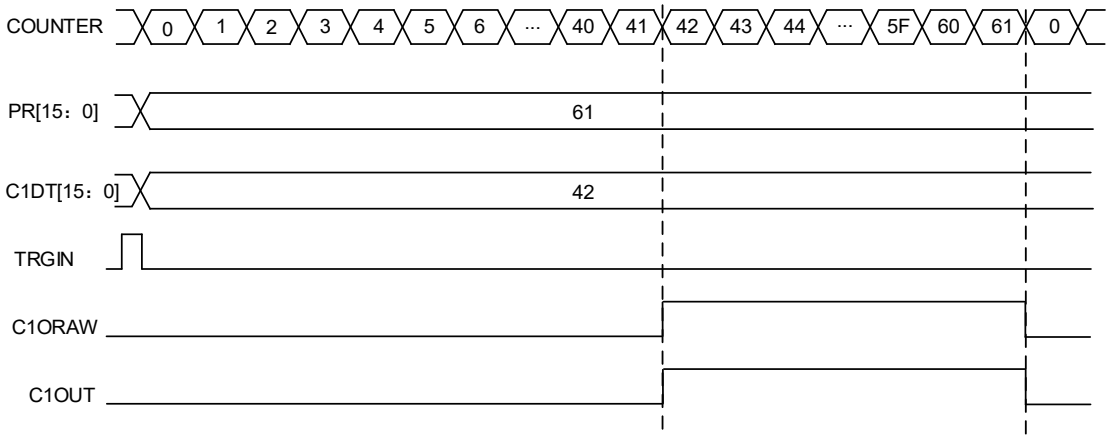


Figure 123. One-pulse mode

CxORAW clear

When the CxOSEN bit is set, the CxORAW signal for a given channel is cleared by applying a high level to the EXT input. The CxORAW signal remains unchanged until the next overflow event.

This function can only be used in output capture or PWM modes, and does not work in forced mode. The following diagram shows the example of clearing CxORAW. When the EXT input is high, the CxORAW signal, which was originally high, is driven low; when the EXT is low, the CxORAW signal outputs the corresponding level according to the comparison result between the counter value and CxDT value.

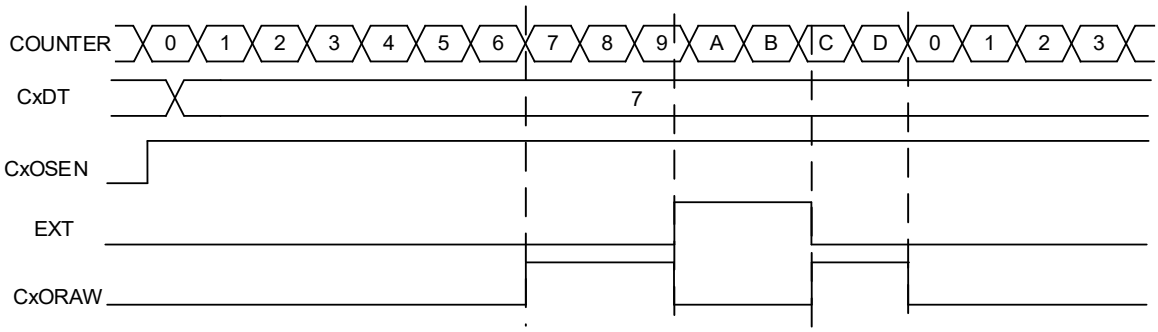


Figure 124. Clearing CxORAW(PWM mode A) by EXT input

Dead-time insertion

The channel 1 to 3 of the advanced-control timers contains a set of reverse channel output. This function is enabled by the CxCEN bit and its polarity is defined by CxCP. Refer to Table 54 for more information about the output state of CxOUT and CxCOUT.

The dead-time is activated when switching to IDLEF state (OEN falling down to 0).

Setting both CxEN and CxCEN bits, and using DTC[7:0] bit to insert dead-time of different durations. After the dead-time insertion, the rising edge of the CxOUT is delayed compared to the rising edge of the reference signal; the rising edge of the CxCOU is delayed compared to the falling edge of the reference signal.

If the delay is greater than the width of the active output, and if C1OUT and C1COUT are to generate corresponding pulses, the dead-time should be less than the width of the active output.

The following diagram gives an example of dead-time insertion when CxP=0, CxCP=0, OEN=1, CxEN=1 and CxCEN=1.

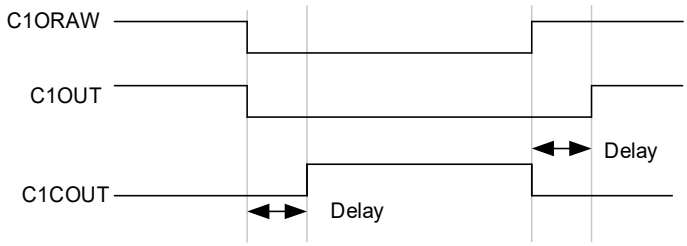


Figure 125. Complementary output with dead-time insertion

TMR break function

When the break function is enabled (BRKEN=1), the CxOUT and CxCOUT are jointly controlled by OEN, FCSODIS, FCSOEN, CxIOS and CxCIOS. But, CxOUT and CxCOUT cannot be set both to active level at the same time. Please refer to 1415 for more details.

The break souce can be the break input pin or a clock failure event. The polarity is controlled by the BRKV bit.

When a break event occurs, there are the following actions:

- The OEN bit is cleared asynchronously, and the channel output state is selected by setting the FCSODIS bit. This function works even if the MCU oscillator is off.
- Once OEN=0, the channel output level is defined by the CxIOS bit. If FCSODIS=0, the timer output is disabled, otherwise, the output enable remains high.
- When complementary outputs are used:
  - The outputs are first put in reset state, that is, inactive state (depending on the polarity). This is done asynchronously so that it works even if no clock is provided to the timer.
  - If the timer clock is still active, then the dead-time generator is activated. The CxIOS and CxCIOS bits are used to program the level after dead-time. Even in this case, the CxIOS and CxCIOS cannot be driven to their actival level a the same time. It should be note that because of synchronization on OEN, the dead-time duration is usually longer than usual (around 2 clk_tmr clock cycles)
  - If FCSODIS=0, the timer releases the enable output, otherwise, it keeps the enable output; the enable output becomes high as soon as one of the CxEN and CxCEN bits becomes high.
- If the break interrupt or DMA request is enabled, the break statue flag is set, and a break interrupt or DMA request can be generated.
- If AOEN=1, the OEN bit is automatically set again at the next overflow event.

Note: When the break input is active, the OEN cannot be set, nor the status flag, BRKIF can be cleared.

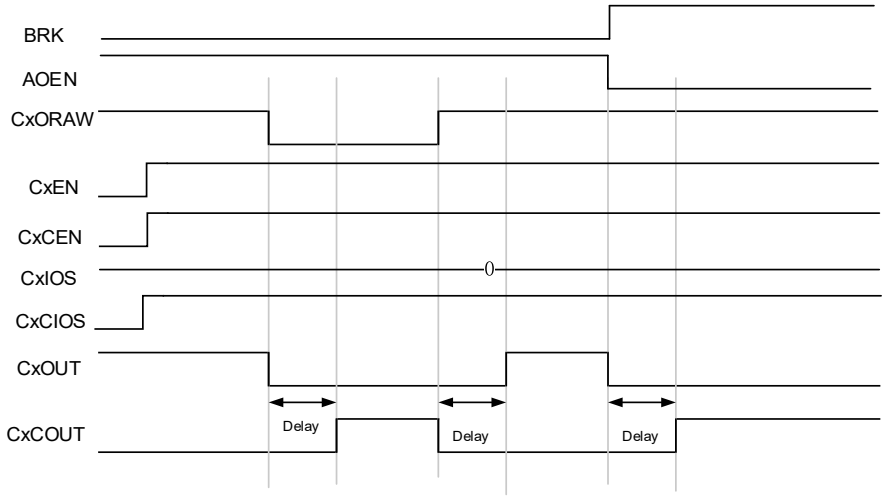


Figure 126. Example of TMR break function

TMR synchronization

The timers are linked together internnaly for timer synchronization. Master timer is selected by setting the PTOS[2:0] bit; Slave timer is selected by setting the SMSEL[2:0] bit.

Slave modes include:

Slave mode: Reset mode

The counter and its prescaler can be reset by a selected trigger signal. An overflow event can be generated when OVFS=0.

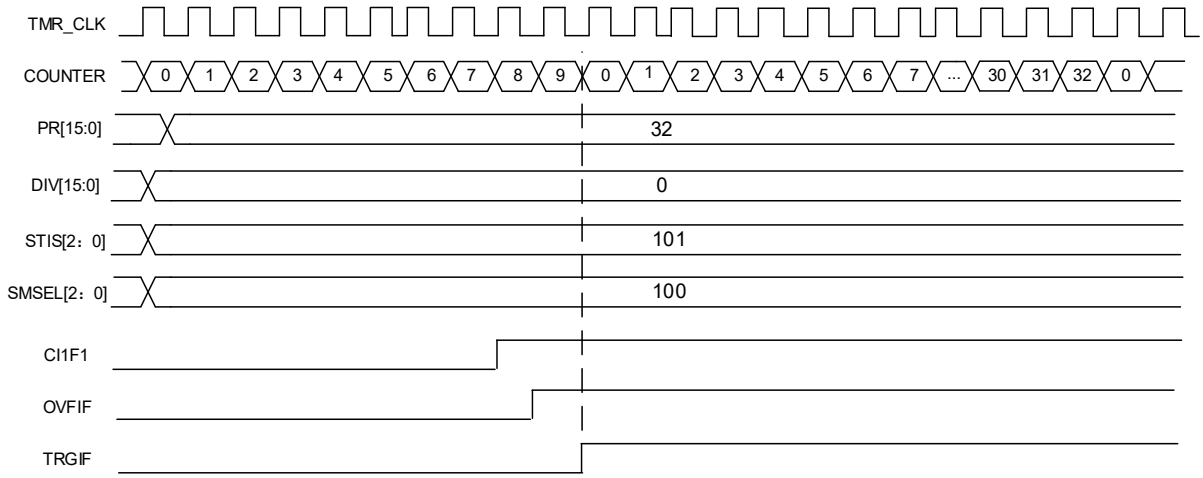


Figure 127. Example of reset mode

Slave mode: Suspend mode

In this mode, the counter is controlled by a selected trigger input. The counter starts counting when the trigger input is high and stops as soon as the trigger input is low.

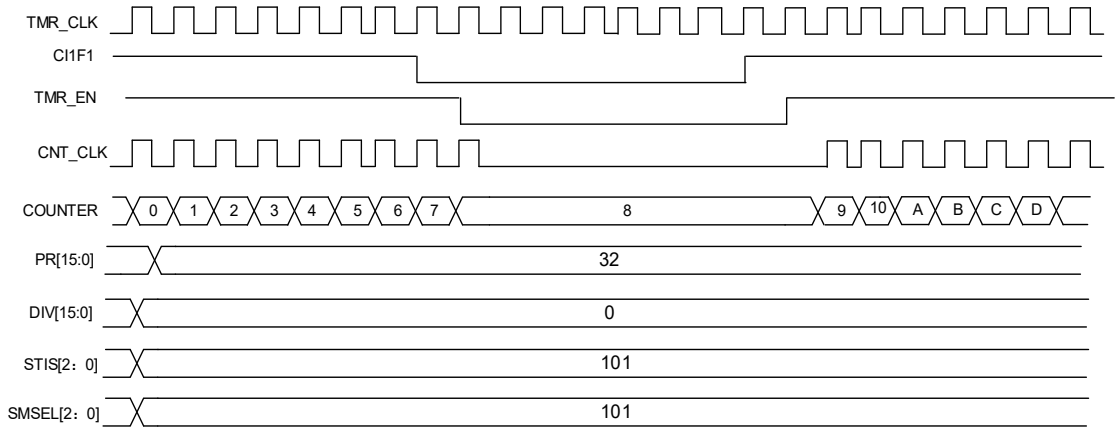


Figure 128. Example of suspend mode

Slave mode: Trigger mode

The counter can start counting on the rising edge of a selected trigger input (TMR_EN=1)

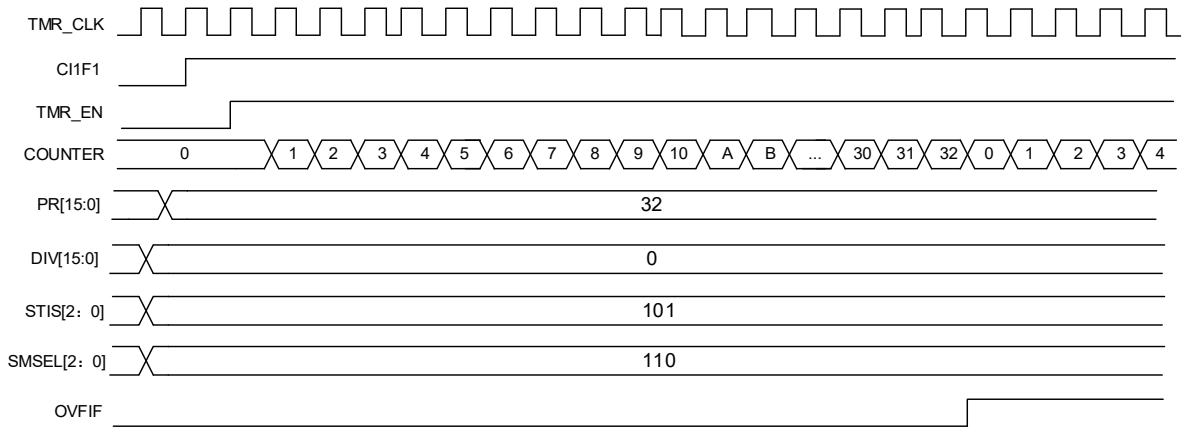


Figure 129. Example of trigger mode

Please refer to Section 14.2.3.5 fore more details.

Debug mode

When the microcontroller enters debug mode (CortexTM-M4F core halted), the TMRx counter stops counting by setting the TMRx_PAUSE in the DEBUG module.

TMR1 and TMR8 registers

These peripheral registers must be accessed by word (32 bits). All TMR1 and TMR8 register are mapped into a 16-bit addressable space.

Table 58. TMR1 and TMR8 register map and reset value

Register	Offset	Reset value
TMRx_CTRL1	0x00	0x0000
TMRx_CTRL2	0x04	0x0000
TMRx_STCTRL	0x08	0x0000
TMRx_IDEN	0x0C	0x0000
TMRx_ISTS	0x10	0x0000
TMRx_SWEVT	0x14	0x0000
TMRx_CM1	0x18	0x0000
TMRx_CM2	0x1C	0x0000
TMRx_CCTRL	0x20	0x0000
TMRx_CVAL	0x24	0x0000
TMRx_DIV	0x28	0x0000
TMRx_PR	0x2C	0x0000
TMRx_RPR	0x30	0x0000
TMRx_C1DT	0x34	0x0000
TMRx_C2DT	0x38	0x0000
TMRx_C3DT	0x3C	0x0000
TMRx_C4DT	0x40	0x0000

Register	Offset	Reset value
TMRx_BRK	0x44	0x0000
TMRx_DMACTRL	0x48	0x0000
TMRx_DMADT	0x4C	0x0000

TMR1 and TMR8 control register1 (TMRx_CTRL1)

Bit	Register	Reset value	Type	Description
Bit 15:10	Reserved	0x00	resd	Kept at its default value.
Bit 9:8	CLKDIV	0x0	rw	Clock division 00: Normal 01: Divided by 2 10: Divided by 4 11: Reserved
Bit 7	PRBEN	0x0	rw	Period buffer enable 0: Period buffer is disabled 1: Period buffer is enabled
Bit 6:5	TWCMSEL	0x0	rw	Two-way counting mode selection 00: One-way counting mode, depending on the OWCDIR bit 01: Two-way counting mode1, count up and down alternately, the output flag bit is set only when the counter counts down 10: Two-way counting mode2, count up and down alternately, the output flag bit is set only when the counter counts up 11: Two-way counting mode3, count up and down alternately, the output flag bit is set when the counter counts up / down
Bit 4	OWCDIR	0x0	rw	One-way count direction 0: Up 1: Down
Bit 3	OCMEN	0x0	rw	One cycle mode enable This bit is use to select whether to stop counting at an update event 0: The counter does not stop at an update event 1: The counter stops at an update event
Bit 2	OVFS	0x0	rw	Overflow event source This bit is used to select overflow event or DMA request sources. 0: Counter overflow, setting the OVFSWTR bit or overflow event generated by slave timer controller 1: Only counter overflow generates an overflow event
Bit 1	OVFEN	0x0	rw	Overflow event enable 0: Enabled 1: Disabled
Bit 0	TMREN	0x0	rw	TMR enable 0: Disabled 1: Enabled

TMR1 and TMR8 control register2 (TMRx_CTRL2)

Bit	Register	Reset value	Type	Description
Bit 15	Reserved	0x00	resd	Kept at its default value.
Bit 14	C4IOS	0x0	rw	Channel 4 idle output state
Bit 13	C3CIOS	0x0	rw	Channel 3 complementary idle output state
Bit 12	C3IOS	0x0	rw	Channel 3 idle output state
Bit 11	C2CIOS	0x0	rw	Channel 2 complementary idle output state
Bit 10	C2IOS	0x0	rw	Channel 2 idle output state
Bit 9	C1CIOS	0x0	rw	Channel 1 complementary idle output state OEN = 0 after dead-time: 0: C1OUTL=0 1: C1OUTL=1
Bit 8	C1IOS	0x0	rw	Channel 1 idle output state OEN = 0 after dead-time: 0: C1OUT=0 1: C1OUT=1
Bit 7	C1INSEL	0x0	rw	C1IN selection 0: CH1 pin is connected to C1IRAW input 1: The XOR result of CH1, CH2 and CH3 pins is connected to C1IRAW input
Bit 6:4	PTOS	0x0	rw	Master TMR output selection This field is used to select the TMRx signal sent to the slave timer. 000: Reset 001: Enable 010: Update 011: Compare pulse 100: C1ORAW signal 101: C2ORAW signal 110: C3ORAW signal 111: C4ORAW signal
Bit 3	DRS	0x0	rw	DMA request source 0: Capture/compare event 1: Overflow event
Bit 2	CCFS	0x0	rw	Channel control bit flash selection This bit only acts on channels that have complementary output. If the channel contro bits are buffered: 0: Control bits are updated by setting the HALL bit 1: Control bits are updated by setting the HALL bit or a rising edge on TRGIN.
Bit 1	Reserved	0x0	resd	Kept at its default value.
Bit 0	CBCTRL	0x0	rw	Channel buffer control This bit acts on channels that have complementary output. 0: CxEN, CxCEN and CxOCTRL bits are not buffered. 1: CxEN, CxCEN and CxOCTRL bits are not buffered.

TMR1 and TMR8 slave timer control register (TMRx_STCTRL)

Bit	Register	Reset value	Type	Description
Bit 15	ESP	0x0	rw	External signal polarity 0: High or rising edge 1: Low or falling edge
Bit 14	ECMBEN	0x0	rw	External clock mode B enable This bit is used to enable external clock mode B 0: Disabled 1: Enabled
Bit 13:12	ESDIV	0x0	rw	External signal divide This field is used to select the frequency division of an external trigger 00: Normal 01: Divided by 2 10: Divided by 4 11: Divided by 8
Bit 11:8	ESF	0x0	rw	External signal filter This field is used to filter an external signal. The external signal can be sampled only after it has been generated N times. 0000: No filter, sampling by f_{DTS} 0001: $f_{SAMPLING} = f_{CK_INT}$, N=2 0010: $f_{SAMPLING} = f_{CK_INT}$, N=4 0011: $f_{SAMPLING} = f_{CK_INT}$, N=8 0100: $f_{SAMPLING} = f_{DTS}/2$, N=6 0101: $f_{SAMPLING} = f_{DTS}/2$, N=8 0110: $f_{SAMPLING} = f_{DTS}/4$, N=6 0111: $f_{SAMPLING} = f_{DTS}/4$, N=8 1000: $f_{SAMPLING} = f_{DTS}/8$, N=6 1001: $f_{SAMPLING} = f_{DTS}/8$, N=8 1010: $f_{SAMPLING} = f_{DTS}/16$, N=5 1011: $f_{SAMPLING} = f_{DTS}/16$, N=6 1100: $f_{SAMPLING} = f_{DTS}/16$, N=8 1101: $f_{SAMPLING} = f_{DTS}/32$, N=5 1110: $f_{SAMPLING} = f_{DTS}/32$, N=6 1111: $f_{SAMPLING} = f_{DTS}/32$, N=8
Bit 7	STS	0x0	rw	Subordinate TMR synchronization If enabled, master and slave timer can be synchronized. 0: Disabled 1: Enabled
Bit 6:4	STIS	0x0	rw	Subordinate TMR input selection This field is used to select the subordinate TMR input. 000: Internal selection 0 (IS0) 001: Internal selection 1 (IS1) 010: Internal selection 2 (IS2) 011: Internal selection 3 (IS3) 100: C1IRAW input detector (C1INC) 101: Filtered input 1 (C1IF1) 110: Filtered input 2 (C1IF2) 111: External input (EXT) Pleaser refer to Table 43 and 45 for more information on ISx for each timer.
Bit 3	Reserved	0x0	resd	Kept at its default value.

Bit	Register	Reset value	Type	Description
Bit 2:0	SMSEL	0x0	rw	Subordinate TMR mode selection 000: Slave mode is disabled 001: Encoder mode A 010: Encoder mode B 011: Encoder mode C 100: Reset mode – Rising edge of the TRGIN input reinitializes the counter 101: Suspend mode – The counter starts counting when the TRGIN is high 110: Trigger mode – A trigger event is generated at the rising edge of the TRGIN input 111: External clock mode A – Rising edge of the TRGIN input clocks the counter Note: Please refer to count mode section for the details on encoder mode A/B/C.

TMR1 and TMR8 DMA/interrupt enable register (TMRx_IDEN)

Bit	Register	Reset value	Type	Description
Bit 15	Reserved	0x0	resd	Kept at its default value.
Bit 14	TDEN	0x0	rw	Trigger DMA request enable 0: Disabled 1: Enabled
Bit 13	HALLDE	0x0	rw	HALL DMA request enable 0: Disabled 1: Enabled
Bit 12	C4DEN	0x0	rw	Channel 4 DMA request enable 0: Disabled 1: Enabled
Bit 11	C3DEN	0x0	rw	Channel 3 DMA request enable 0: Disabled 1: Enabled
Bit 10	C2DEN	0x0	rw	Channel 2 DMA request enable 0: Disabled 1: Enabled
Bit 9	C1DEN	0x0	rw	Channel 1 DMA request enable 0: Disabled 1: Enabled
Bit 8	OVFDEN	0x0	rw	Overflow event DMA request enable 0: Disabled 1: Enabled
Bit 7	BRKIE	0x0	rw	Break interrupt enable 0: Disabled 1: Enabled
Bit 6	TIEN	0x0	rw	Trigger interrupt enable 0: Disabled 1: Enabled
Bit 5	HALLIEN	0x0	rw	HALL interrupt enable 0: Disabled 1: Enabled

Bit	Register	Reset value	Type	Description
Bit 4	C4IEN	0x0	rw	Channel 4 interrupt enable 0: Disabled 1: Enabled
Bit 3	C3IEN	0x0	rw	Channel 3 interrupt enable 0: Disabled 1: Enabled
Bit 2	C2IEN	0x0	rw	Channel 2 interrupt enable 0: Disabled 1: Enabled
Bit 1	C1IEN	0x0	rw	Channel 1 interrupt enable 0: Disabled 1: Enabled
Bit 0	OVFIEN	0x0	rw	Overflow interrupt enable 0: Disabled 1: Enabled

TMR1 and TMR8 interrupt status register (TMRx_ISTS)

Bit	Register	Reset value	Type	Description
Bit 15:13	Reserved	0x0	resd	Kept at its default value.
Bit 12	C4RF	0x0	rw0c	Channel 4 recapture flag Please refer to C1RF description.
Bit 11	C3RF	0x0	rw0c	Channel 3 recapture flag Please refer to C1RF description.
Bit 10	C2RF	0x0	rw0c	Channel 2 recapture flag Please refer to C1RF description.
Bit 9	C1RF	0x0	rw0c	Channel 1 recapture flag This bit indicates whether a recapture is detected when C1IF=1. This bit is set by hardware, and cleared by writing “0”. 0: No capture is detected 1: Capture is detected
Bit 8	Reserved	0x0	resd	Default value
Bit 7	BRKIF	0x0	rw0c	Break interrupt flag This bit indicates whether the break input is active or not. It is set by hardware and cleared by writing “0” 0: Inactive level 1: Active level
Bit 6	TRGIF	0x0	rw0c	Trigger interrupt flag This bit is set by hardware on a trigger event. It is cleard by writing “0”. 0: No trigger event occurs 1: Trigger event is generated Trigger event: an active edge is detected on TRGIN input, or any edge in suspend mode.
Bit 5	HALLIF	0x0	rw0c	HALL interrupt flag This bit is set by hardware on HALL event. It is cleared by writing “0”. 0: No Hall event occurs 1: Hall event is detected HALL even: CxEN, CxCEN and CxOCTRL are updated.

Bit	Register	Reset value	Type	Description
Bit 4	C4IF	0x0	rw0c	Channel 4 interrupt flag Please refer to C1IF description.
Bit 3	C3IF	0x0	rw0c	Channel 3 interrupt flag Please refer to C1IF description.
Bit 2	C2IF	0x0	rw0c	Channel 2 interrupt flag Please refer to C1IF description.
Bit 1	C1IF	0x0	rw0c	Channel 1 interrupt flag If the channel 1 is configured as input mode: This bit is set by hardware on a capture event. It is cleared by software or read access to the TMRx_C1DT 0: No capture event occurs 1: Capture event is generated If the channel 1 is configured as output mode: This bit is set by hardware on a compare event. It is cleared by software. 0: No compare event occurs 1: Compare event is generated
Bit 0	OVFIF	0x0	rw0c	Overflow interrupt flag This bit is set by hardware on an overflow event. It is cleared by software. 0: No overflow event occurs 1: Overflow event is generated. If OVFEN=0 and OVFS=0 in the TMRx_CTRL1 register: − An overflow event is generated when OVFG= 1 in the TMRx_SWEVE register; − An overflow event is generated when the counter CVAL is reinitialized by a trigger event.

Software event register (TMRx_SWEVT)

Bit	Register	Reset value	Type	Description
Bit 15:8	Reserved	0x000	resd	Kept at its default value.
Bit 7	BRKSWTR	0x0	wo	Break event triggered by software This bit is set by software to generate a break event. 0: No effect 1: Generate a break event.
Bit 6	TRGSWTR	0x0	rw	Trigger event triggered by software This bit is set by software to generate a trigger event. 0: No effect 1: Generate a trigger event.
Bit 5	HALLSWTR	0x0	wo	HALL event triggered by software This bit is set by software to generate a HALL event. 0: No effect 1: Generate a HALL event. Note: This bit acts only on channels that have complementary output.
Bit 4	C4SWTR	0x0	wo	Channel 4 event triggered by software Please refer to C1M description.
Bit 3	C3SWTR	0x0	wo	Channel 3 event triggered by software Please refer to C1M description.

Bit	Register	Reset value	Type	Description
Bit 2	C2SWTR	0x0	wo	Channel 2 event triggered by software Please refer to C1M description
Bit 1	C1SWTR	0x0	wo	Channel 1 event triggered by software This bit is set by software to generate a channel 1 event. 0: No effect 1: Generate a channel 1 event.
Bit 0	OVFSWTR	0x0	wo	Overflow event triggered by software This bit is set by software to generate an overflow event. 0: No effect 1: Generate an overflow event.

TMR1 and TMR8 channel mode register1 (TMRx_CM1)

The channel can be used in input (capture mode) or output (compare mode). The direction of a channel is defined by the corresponding CxC bits. All the other bits of this register have different functons in input and output modes. The CxOx describes its function in output mode when the channel is in output mode, while the CxIx describes its function in output mode when the channel is in input mode. Attention must be given to the fact that the same bit can have different functions in input mode and output mode.

Output compare mode:

Bit	Register	Reset value	Type	Description
Bit 15	C2OSEN	0x0	rw	Channel 2 output switch enable
Bit 14:12	C2OCTRL	0x0	rw	Channel 2 output control
Bit 11	C2OBEN	0x0	rw	Channel 2 output buffer enable
Bit 10	C2OIEN	0x0	rw	Channel 2 output enable immediately
Bit 9:8	C2C	0x0	rw	Channel 2 configuration This field is used to define the direction of the channel 2 (input or output), and the selection of input pin when C2EN='0': 00: Output 01: Input, C2IN is mapped on C2IRAW 10: Input, C2IN is mapped on C1IRAW 11: Input, C2IN is mapped on STCI. This mode works only when the internal trigger input is selected by STIS register.
Bit 7	C1OSEN	0x0	rw	Channel 1 output switch enable 0: C1ORAW is not affected by EXT input. 1: Once a high level is detect on EXT input, clear C1ORAW

Bit	Register	Reset value	Type	Description
Bit 6:4	C1OCTRL	0x0	rw	Channel 1 output control This field defines the behavior of the original signal C1ORAW. 000: Disconnected. C1ORAW is disconnected from C1OUT; 001: C1ORAW is high when TMRx_CVAL=TMRx_C1DT 010: C1ORAW is low when TMRx_CVAL=TMRx_C1DT 011: Switch C1ORAW level when TMRx_CVAL=TMRx_C1DT 100: C1ORAW is forced low 101: C1ORAW is forced high 110: PWM mode A - OWCDIR=0, C1ORAW is high once TMRx_C1DT > TMRx_CVAL, else low; - OWCDIR=1, C1ORAW is low once TMRx_C1DT < TMRx_CVAL, else high; 111: PWM mode B - OWCDIR=0, C1ORAW is low once TMRx_C1DT > TMRx_CVAL, else high; - OWCDIR=1, C1ORAW is high once TMRx_C1DT < TMRx_CVAL, else low. Note: In the configurations othern than 000', the C1OUT is connected to C1ORAW. The C1OUT output level is not only subject to the changes of C1ORAW, but also the output polarity set by CCTRL.
Bit 3	C1OBEN	0x0	rw	Channel 1 output buffer enable 0: Buffer function of TMRx_C1DT is disabled. The new value written to the TMRx_C1DT takes effect immediately. 1: Buffer function of TMRx_C1DT is enabled. The value to be written to the TMRx_C1DT is stored in the buffer register, and can be sent to the TMRx_C1DT register only on an overflow event.
Bit 2	C1OIEN	0x0	rw	Channel 1 output enable immediately In PWM mode A or B, this bit is used to accelerate the channel 1 output's response to the trigger event. 0: Need to compare the CVAL with C1DT before generating an output 1: No need to compare the CVAL and C1DT. An output is generated immediately when a trigger event occurs.
Bit 1:0	C1C	0x0	rw	Channel 1 configuration This field is used to define the direction of the channel 1 (input or output), and the selection of input pin when C1EN='0': 00: Output 01: Input, C1IN is mapped on C1IRAW 10: Input, C1IN is mapped on C2IRAW 11: Input, C1IN is mapped on STCI. This mode works only when the internal trigger input is selected by STIS.

Input capture mode:

Bit	Register	Reset value	Type	Description
Bit 15:12	C2DF	0x0	rw	Channel 2 digital filter
Bit 11:10	C2IDIV	0x0	rw	Channel 2 input divider
Bit 9:8	C2C	0x0	rw	Channel 2 configuration This field is used to define the direction of the channel 2 (input or output), and the selection of input pin when C2EN='0': 00: Output 01: Input, C2IN is mapped on C2IRAW 10: Input, C2IN is mapped on C1IRAW 11: Input, C2IN is mapped on STCI. This mode works only when the internal trigger input is selected by STIS.
Bit 7:4	C1DF	0x0	rw	Channel 1 digital filter This field defines the digital filter of the channel 1. N stands for the number of filtering, indicating that the input edge can pass the filter only after N sampling events. 0000: No filter, sampling is done at f_{DTS} 1000: $f_{SAMPLING} = f_{DTS}/8$, N=6 0001: $f_{SAMPLING} = f_{CK_INT}$, N=2 1001: $f_{SAMPLING} = f_{DTS}/8$, N=8 0010: $f_{SAMPLING} = f_{CK_INT}$, N=4 1010: $f_{SAMPLING} = f_{DTS}/16$, N=5 0011: $f_{SAMPLING} = f_{CK_INT}$, N=8 1011: $f_{SAMPLING} = f_{DTS}/16$, N=6 0100: $f_{SAMPLING} = f_{DTS}/2$, N=6 1100: $f_{SAMPLING} = f_{DTS}/16$, N=8 0101: $f_{SAMPLING} = f_{DTS}/2$, N=8 1101: $f_{SAMPLING} = f_{DTS}/32$, N=5 0110: $f_{SAMPLING} = f_{DTS}/4$, N=6 1110: $f_{SAMPLING} = f_{DTS}/32$, N=6 0111: $f_{SAMPLING} = f_{DTS}/4$, N=8 1111: $f_{SAMPLING} = f_{DTS}/32$, N=8
Bit 3:2	C1IDIV	0x0	rw	Channel 1 input divider This field defines Channel 1 input divider. 00: No divider. An input capture is generated at each active edge. 01: An input compare is generated every 2 active edges 10: An input compare is generated every 4 active edges 11: An input compare is generated every 8 active edges Note: the divider is reset once C1EN='0'
Bit 1:0	C1C	0x0	rw	Channel 1 configuration This field is used to define the direction of the channel 1 (input or output), and the selection of input pin when C1EN='0': 00: Output 01: Input, C1IN is mapped on C1IRAW 10: Input, C1IN is mapped on C2IRAW 11: Input, C1IN is mapped on STCI. This mode works only when the internal trigger input is selected by STIS.

Channel mode register2 (TMRx_CM2)

The channel can be used in input (capture mode) or output (compare mode). The direction of a channel is defined by the corresponding CxC bits. All the other bits of this register have different functons in input and output modes. The CxOx describes its function in output mode when the channel is in output mode, while the CxIx describes its function in output mode when the channel is in input mode. Attention must be given to the fact that the same bit can have different functions in input mode and output mode.

Output compare mode:

Bit	Register	Reset value	Type	Description
Bit 15	C4OSEN	0x0	rw	Channel 4 output switch enable
Bit 14:12	C4OCTRL	0x0	rw	Channel 4 output control
Bit 11	C4OBEN	0x0	rw	Channel 4 output buffer enable
Bit 10	C4OIEN	0x0	rw	Channel 4 output enable immediately
Bit 9:8	C4C	0x0	rw	Channel 4 configuration This field is used to define the direction of the channel 1 (input or output), and the selection of input pin when C4EN='0': 00: Output 01: Input, C4IN is mapped on C4IRAW 10: Input, C4IN is mapped on C3IRAW 11: Input, C4IN is mapped on STCI. This mode works only when the internal trigger input is selected by STIS.
Bit 7	C3OSEN	0x0	rw	Channel 3 output switch enable
Bit 6:4	C3OCTRL	0x0	rw	Channel 3 output control
Bit 3	C3OBEN	0x0	rw	Channel 3 output buffer enable
Bit 2	C3OIEN	0x0	rw	Channel 3 output enable immediately
Bit 1:0	C3C	0x0	rw	Channel 3 configuration This field is used to define the direction of the channel 1 (input or output), and the selection of input pin when C3EN='0': 00: Output 01: Input, C3IN is mapped on C3IRAW 10: Input, C3IN is mapped on C4IRAW 11: Input, C3IN is mapped on STCI. This mode works only when the internal trigger input is selected by STIS.

Input capture mode:

Bit	Register	Reset value	Type	Description
Bit 15:12	C4DF	0x0	rw	Channel 4 digital filter
Bit 11:10	C4IDIV	0x0	rw	Channel 4 input divider
Bit 9:8	C4C	0x0	rw	Channel 4 configuration This field is used to define the direction of the channel 1 (input or output), and the selection of input pin when C4EN=’0’: 00: Output 01: Input, C4IN is mapped on C4IRAW 10: Input, C4IN is mapped on C3IRAW 11: Input, C4IN is mapped on STCI. This mode works only when the internal trigger input is selected by STIS.
Bit 7:4	C3DF	0x0	rw	Channel 3 digital filter
Bit 3:2	C3IDIV	0x0	rw	Channel 3 input divider
Bit 1:0	C3C	0x0	rw	Channel 3 configuration This field is used to define the direction of the channel 1 (input or output), and the selection of input pin when C3EN=’0’: 00: Output 01: Input, C3IN is mapped on C3IRAW 10: Input, C3IN is mapped on C4IRAW 11: Input, C3IN is mapped on STCI. This mode works only when the internal trigger input is selected by STIS.

Channel control register (TMRx_CCTRL)

Bit	Register	Reset value	Type	Description
Bit 15:14	Reserved	0x0	resd	Kept its default value.
Bit 13	C4P	0x0	rw	Channel 4 polarity Pleaser refer to C1P description.
Bit 12	C4EN	0x0	rw	Channel 4 enable Pleaser refer to C1EN description.
Bit 11	C3CP	0x0	rw	Channel 3 complementary polarity Please refer to C1P description.
Bit 10	C3CEN	0x0	rw	Channel 3 complementary enable Please refer to C1EN description.
Bit 9	C3P	0x0	rw	Channel 3 polarity Pleaser refer to C1P description.
Bit 8	C3EN	0x0	rw	Channel 3 enable Pleaser refer to C1EN description.
Bit 7	C2CP	0x0	rw	Channel 2 complementary polarity Please refer to C1P description.
Bit 6	C2CEN	0x0	rw	Channel 2 complementary enable Please refer to C1EN description.
Bit 5	C2P	0x0	rw	Channel 2 polarity Pleaser refer to C1P description.
Bit 4	C2EN	0x0	rw	Channel 2 enable Pleaser refer to C1EN description.

Bit	Register	Reset value	Type	Description
Bit 3	C1CP	0x0	rw	Channel 1 complementary polarity 0: C1COUT is active high 1: C1COUT is active low
Bit 2	C1CEN	0x0	rw	Channel 1 complementary enable 0: Output is disabled 1: Output is enabled
Bit 1	C1P	0x0	rw	Channel 1 polarity When the channel 1 is configured as output mode: 0: C1OUT is active high 1: C1OUT is active low When the channel 1 is configured as input mode: 0: C1IN active edge is on its rising edge. When used as external trigger, C1IN is not inverted. 1: C1IN active edge is on its falling edge. When used as external trigger, C1IN is inverted.
Bit0	C1EN	0x0	rw	Channel 1 enable 0: Input or output is disabled 1: Input or output is enabled

Table 59. Complementary output channel CxOUT and CxCOUT control bits with break function

Control bit					Output state (1)	
OEN bit	FCSODIS bit	FCSOEN bit	CxEN bit	CxCEN bit	CxOUT output state	CxCOUT output state
1	X	0	0	0	Output disabled (no driven by the timer) CxOUT=0, Cx_EN=0	Output disabled (no driven by the timer) CxCOUT=0, CxCEN=0
		0	0	1	Output disabled (no driven by the timer) CxOUT=0, Cx_EN=0	CxORAW + polarity, CxCOUT= CxORAW xor CxCP, CxCEN=1
		0	1	0	CxORAW+ polarity CxOUT= CxORAW xor CxP, Cx_EN=1	Output disabled (no driven by the timer) CxCOUT=0, CxCEN=0
		0	1	1	CxORAW+polarity+dead-time, Cx_EN=1	CxORAW inverted+polarity+ dead-time, CxCEN=1
		1	0	0	Output disabled (no driven by the timer) CxOUT=CxP, Cx_EN=0	Output disabled (no driven by the timer) CxCOUT=CxCP, CxCEN=0
		1	0	1	Off-state (Output enabled with inactive level) CxOUT=CxP, Cx_EN=1	CxORAW + polarity, CxCOUT= CxORAW xor CxCP, CxCEN=1
		1	1	0	CxORAW + polarity, CxOUT= CxORAW xor CxP, Cx_EN=1	Off-state (Output enabled with inactive level) CxCOUT=CxCP, CxCEN=1
		1	1	1	CxORAW+ polarity+dead-time, Cx_EN=1	CxORAW inverted+polarity+ dead-time, CxCEN=1

Control bit					Output state (1)	
OEN bit	FCSODIS bit	FCSOEN bit	CxEN bit	CxCEN bit	CxOUT output state	CxCOUT output state
0	0	X	0	0	Output disabled (no driven by the timer) Asynchronously: CxOUT=CxP, Cx_EN=0, CxCOUT=CxCP, CxCEN=0; If the clock is present: after a dead-time, CxOUT=CxIOS, CxCOUT=CxCIOS, assuming that CxIOS and CxCIOS do not correspond to CxOUT and CxCOUT active level.	
	0		0	1		
	0		1	0		
	0		1	1		
	1		0	0	Off-state (Output enabled with inactive level) Asynchronously: CxOUT =CxP, Cx_EN=1, CxCOUT=CxCP, CxCEN=1; If the clock is present: after a dead-time, CxOUT=CxIOS, CxCOUT=CxCIOS, assuming that CxIOS and CxCIOS do not correspond to CxOUT and CxCOUT active level.	
	1		0	1		
	1		1	0		
	1		1	1		

Note: 1. If the two outputs of a channel are not used (CxEN = CxCEN = 0), CxIOS, CxCIOS, CxP and CxCP must be cleared.
2. The state of the external I/O pins connected to the complementary CxOUT and CxCOUT channels depends on the CxOUT and CxCOUT channel state and the GPIO and the IOMUX registers.

TMR1 and TMR8 counter value (TMRx_CVAL)

Bit	Register	Reset value	Type	Description
Bit 15:0	CVAL	0x0000	rw	Counter value

TMR1 and TMR8 division value (TMRx_DIV)

Bit	Register	Reset value	Type	Description
Bit 15:0	DIV	0x0000	rw	Divider value The counter clock frequency $f_{CK_CNT} = f_{TMR_CLK}$ / (DIV[15:0]+1). The value of this register is transferred to the actual prescaler register when an overflow event occurs.

TMR1 and TMR8 period register (TMRx_PR)

Bit	Register	Reset value	Type	Description
Bit 15:0	PR	0x0000	rw	Period value This defines the period value of the TMRx counter. The timer stops working when the period value is 0.

TMR1 and TMR8 repetition period register (TMRx_RPR)

Bit	Register	Reset value	Type	Description
Bit 15:8	Reserved	0x00	resd	Kept at its default value.
Bit 7:0	RPR	0x00	rw	Repetition of period value This field is used to reduce the generation rate of overflow events. An overflow event is generated when the repetition counter reaches 0.

TMR1 and TMR8 channel 1 data register (TMRx_C1DT)

Bit	Register	Reset value	Type	Description
Bit 15:0	C1DT	0x0000	rw	Channel 1 data register When the channel 1 is configured as input mode: The C1DT is the CVAL value stored by the last channel 1 input event (C1IN) When the channel 1 is configured as output mode: C1DT is the value to be compared with the CVAL value. Whether the written value takes effective immediately depends on the C1OBEN bit, and the corresponding output is generated on C1OUT as configured.

TMR1 and TMR8 channel 2 data register (TMRx_C2DT)

Bit	Register	Reset value	Type	Description
Bit 15:0	C2DT	0x0000	rw	Channel 2 data register When the channel 2 is configured as input mode: The C2DT is the CVAL value stored by the last channel 2 input event (C1IN) When the channel 2 is configured as output mode: C2DT is the value to be compared with the CVAL value. Whether the written value takes effective immediately depends on the C2OBEN bit, and the corresponding output is generated on C2OUT as configured.

TMR1 and TMR8 channel 3 data register (TMRx_C3DT)

Bit	Register	Reset value	Type	Description
Bit 15:0	C3DT	0x0000	rw	Channel 3 data register When the channel 3 is configured as input mode: The C3DT is the CVAL value stored by the last channel 3 input event (C1IN) When the channel 3 is configured as output mode: C3DT is the value to be compared with the CVAL value. Whether the written value takes effective immediately depends on the C3OBEN bit, and the corresponding output is generated on C3OUT as configured.

TMR1 and TMR8 channel 4 data register (TMRx_C4DT)

Bit	Register	Reset value	Type	Description
Bit 15:0	C3DT	0x0000	rw	Channel 4 data register When the channel 4 is configured as input mode: The C4DT is the CVAL value stored by the last channel 4 input event (C1IN) When the channel 3 is configured as output mode: C4DT is the value to be compared with the CVAL value. Whether the written value takes effective immediately depends on the C4OBEN bit, and the corresponding output is generated on C4OUT as configured.

TMR1 and TMR8 break register (TMRx_BRK)

Bit	Register	Reset value	Type	Description
Bit 15	OEN	0x0	rw	Output enable This bit acts on the channels as output. It is used to enable CxOUT and CxCOUT outputs. 0: Disabled 1: Enabled
Bit 14	AOEN	0x0	rw	Automatic output enable OEN is set automatically at an overflow event. 0: Disabled 1: Enabled
Bit 13	BRKV	0x0	rw	Break input validity This bit is used to select the active level of a break input. 0: Break input is active low. 1 Break input is active high.
Bit 12	BRKEN	0x0	rw	Break enable This bit is used to enable break input. 0: Break input is disabled. 1: Break input is enabled.
Bit 11	FCSOEN	0x0	rw	Frozen channel status when holistic output enable This bit acts on the channels that have complementary output. It is used to set the channel state when the timer is inactive and MOEN=1. 0: CxOUT/CxCOUT outputs are disabled. 1: CxOUT/CxCOUT outputs are enabled. Output inactive level.
Bit 10	FCSODIS	0x0	rw	Frozen channel status when holistic output disable This bit acts on the channels that have complementary output. It is used to set the channel state when the timer is inactive and MOEN=0. 0: CxOUT/CxCOUT outputs are disabled. 1: CxOUT/CxCOUT outputs are enabled. Output idle level.
Bit 9:8	WPC	0x0	rw	Write protection configuration his field is used to enable write protection. 00: Write protection is OFF 01: Write protection level 3, and the following bits are write protected: TMRx_BRK: DTC, BRKEN, BRKV and AOEN TMRx_CTRL2: CxIOS and CxCIOS 10: Write protection level 2. The following bits and all bits in leve 3 are write protected: TMRx_CCTRL: CxP and CxCP TMRx_BRK: FCSODIS and FCSOEN 11: Write protection level 1. The following bits and all bits in level 2 are write protected: TMRx_CMx: C2OCTRL and C2OBEN Note: Once WPC>0, its content remains frozen until the next system reset.

Bit	Register	Reset value	Type	Description
Bit 7:0	DTC	0x00	rw	Dead-time configuration This field defines the duration of the dead-time insertation. The 3-bit MSB of DTC[7:0] is used for function selection: 0xx: DT = DTC[7:0] × TDTS 10x: DT = (64 + DTC[5:0]) × TDTS × 2 110: DT = (32 + DTC[4:0]) × TDTS × 8 111: DT = (32 + DTC[4:0]) × TDTS × 16

Note: Based on lock configuration, AOEN, BRKV, BRKEN, FCSODIS, FCSOEN and DTC[7:0] can all be write protected. Thus it is necessary to configure write protection when writing to the TMRx_BRK register for the first time.

TMR1 and TMR8 DMA control register (TMRx_ DMACTRL)

Bit	Register	Reset value	Type	Description
Bit 15:13	Reserved	0x0	resd	Kept at its default value.
Bit 12:8	DTB	0x00	rw	DMA transfer bytes This field defines the number of DMA transfers: 00000: 1 byte 00001: 2 bytes 00010: 3 bytes 00011: 4 bytes .... 10000: 17 bytes 10001: 18 bytes
Bit 7:5	Reserved	0x0	resd	Kept at its default value.
Bit 4:0	ADDR	0x00	rw	DMA transfer address offset ADDR is defined as an offset starting from the address of the TMRx_CTRL1 register: 00000: TMRx_CTRL1 00001: TMRx_CTRL2 00010: TMRx_STCTRL .....

TMR1 and TMR8 DMA data register (TMRx_ DMADT)

Bit	Register	Reset value	Type	Description
Bit 15:0	DMADT	0x0000	rw	DMA data register A write/read operation to the DMADT register accesses any TMR register located at the following address: TMRx peripheral address + ADDR × 4 to TMRx peripheral address + ADDR ×4 + DTB × 4

15 Window watchdog timer (WWDT)

WWDT introduction

The window watchdog downcounter must be reloaded in a limited time window to prevent the watchdog circuits from generating a system reset. The window watch dog is used to detect the occurrence of system malfunctions.

The window watchdog timer is clocked by a divided APB1_CLK. The presion of the APB1_CLK enables the window watchdog to take accurate control of the limited window.

Main features

- 7-bit downcounter
- If the watchdog is enabled, a system reset is generated when the value of the downcounter is less than 0x40 or when the downcounter is reloaded outside the window.
- The downcounter can be reloaded by enabling the counter interrupt.

Functional descriptions

If the watchdog is enabled, a system reset is generated at the following contions:

When the 7-bit downcounter scrolls from 0x40 to 0x3F;

When the counter is reloaded while the 7-bit downcounter is greater than the value programmed in the window register.

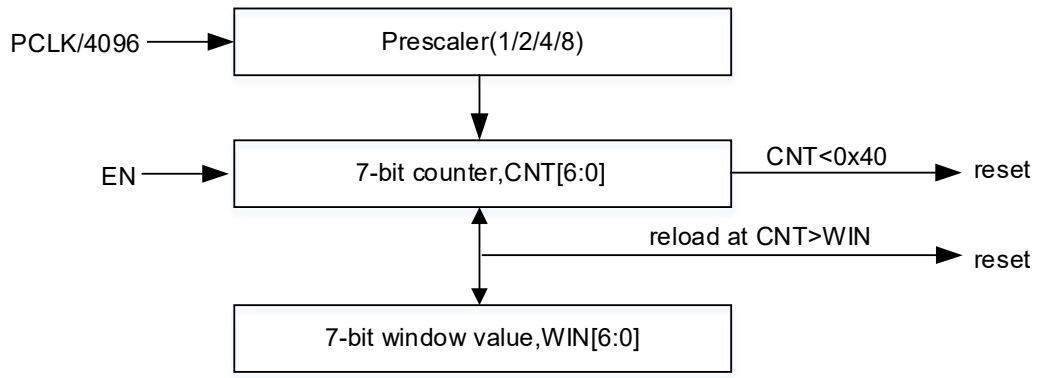


Figure 130. Window watchdog block diagram

To prevent sytem reset, the counter must be reloaded only when its value is less than the value stored in the window register and greater than 0x40.

The WWDT counter is clocked by a divided APB1_CLK, with the division factor being defined by the DIV[1:0] bit in the WWDT_CFG register. The counter value determines the maximum counter period before the watchdow generates a reset. The WIN[6:0] bit can be used to configure the window value.

WWDT offers reload counter interrupt feature. If enabled, the WWDT will set the RLDF flag when the counter value reaches 0x40h, and an interrupt is generated accordingly. The interrupt service routine (ISTS) can be used to reload the counter to prevent a system reset. Note that if CNT[6]=0, setting the WWDTEN bit will generate a system reset, so the CNT[6] bit must be always set (CNT[6]=1) while writing to the WWDT_CTRL register to prevent the occurrence of an immediate reset once the window watchdog is enabled.

The formula to calculate the window watchdog time out:

$$T_{WWDT} = T_{PCLK1} \times 4096 \times 2^{DIV[1:0]} \times (CNT[5:0] + 1); \text{(ms)}$$

Where: T_{PCLK1} refers to APB1 clock period, in ms.

Table 60. Minimum and maximum timeout value when PCLK1=72 MHz

Prescaler	Min. Timeout value	Max. Timeout value
0	56.5μs	3.64ms
1	113.5μs	7.28ms
2	227.5μs	14.56ms
3	455μs	29.12ms

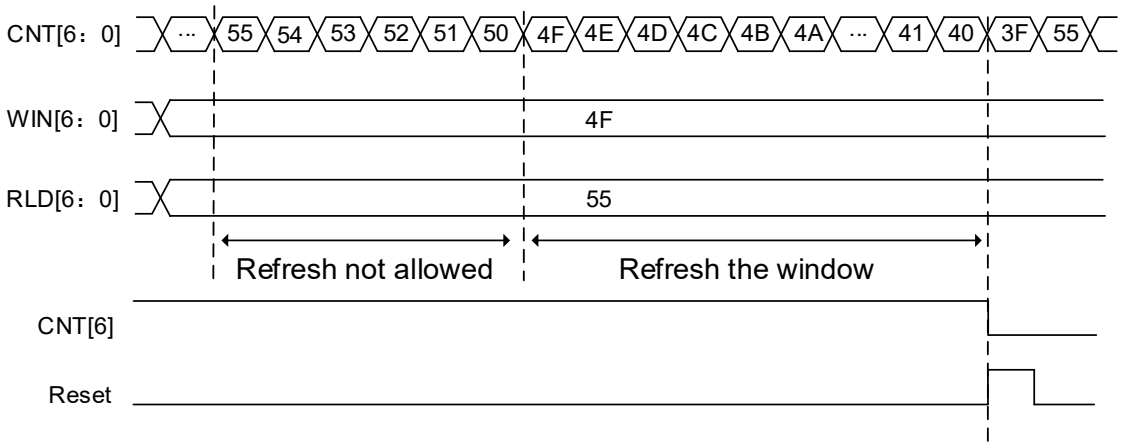


Figure 131. Window watchdog timing diagram

Debug mode

When the microcontroller enters debug mode (CortexTM-M4F core halted), the WWDT counter stops counting by setting the WWDT_PAUSE in the DEBUG module.

WWDT registers

These peripheral registers must be accessed by word (32 bits).

Table 61. WWDT register map and reset value

Register	Offset	Reset value
WWDT_CTRL	0x00	0x7F
WWDT_CFG	0x04	0x7F
WWDT_STS	0x08	0x00

Control register (WWDT_CTRL)

Bit	Register	Reset value	Type	Description
Bit 31:8	Reserved	0x000000	resd	Kept at its default value.
Bit 7	WWDTEN	0x0	rw1s	Window watchdog enable 0: Disabled 1: Enabled This bit is set by software, but can be cleared only after reset.
Bit 6:0	CNT	0x7F	rw	Downcounter When the counter counts down to 0x3F, a reset is generated.

Configuration register (WWDT_CFG)

Bit	Register	Reset value	Type	Description
Bit 31:10	Reserved	0x000000	resd	Kept at its default value.
Bit 9	RLDIEN	0x0	rw	Reload counter interrupt 0: Disabled 1: Enabled
Bit 8:7	DIV	0x0	rw	Clock division value 00: PCLK1 divided by 4096 01: PCLK1 divided by 8192 10: PCLK1 divided by 16384 11: PCLK1 divided by 32768
Bit 6:0	WIN	0x7F	rw	Window value if the counter is reloaded while its value is greater than the window register value, a reset is generated. The counter must be reloaded between 0x40 and WIN[6:0].

Status register (WWDT_STS)

Bit	Register	Reset value	Type	Description
Bit 31:1	Reserved	0x0000 0000	resd	Kept at its default value.
Bit 0	RLDF	0x0	rw0c	Reload counter interrupt flag This flag is set when the downcounter reaches 0x40. This bit is set by hardware and cleared by software.

16 Watchdog timer (WDT)

WDT introduction

The WDT is driven by a dedicated low-speed clock (LICK). Due to the lower clock accuracy of LICK, the WDT is best suited to the applications that have lower timing accuracy and can run independently outside the main application.

Main features

- 12-bit downcounter
- The counter is clocked by LICK (can work in Stop and Standby modes)
- If the WDT is enabled, a system reset is generated when the counter value reaches 0

Functional descriptions

WDT enable

Both software and hardware operations can be used to enable WDT. In other words, the WDT can be enabled by writing 0xCCCC to the WDT_CMD register; or when the user enables the hardware watchdog through user system data area, the WDT will be automatically enabled after power-on reset.

WDT reset

When the counter value of the WDT counts down to 0, a WDT reset be generated. Thus the WDT_CMD register must be written with the value 0xAAAA at regular intervals to reload the counter value to avoid the WDT reset.

WDT write-protected

The WDT_DIV and WDT_RLD registers are write-protected. Writing the value 0x5555 to the WDT_CMD register will unlock write protection. The update status of these two registers are indicated by the DIVF and RLDF bits in the WDT_STS register. If a different value is written to the WDT_CMD register, these two registers will be re-protected. Writing the value 0xAAAA to the WDT_CMD register also enables write protection.

WDT clock

The WDT counter is clocked by the LICK. The LICK is an internal RC clock, with its range falling between 30kHz and 60kHz. The timeout period is also within a certain range, so a margin should be taken into account when configuring timeout period. The LICK can be calibrated to obtain the WDT timeout with a relatively accuracy. For more details, please refer to section 4.1.1.

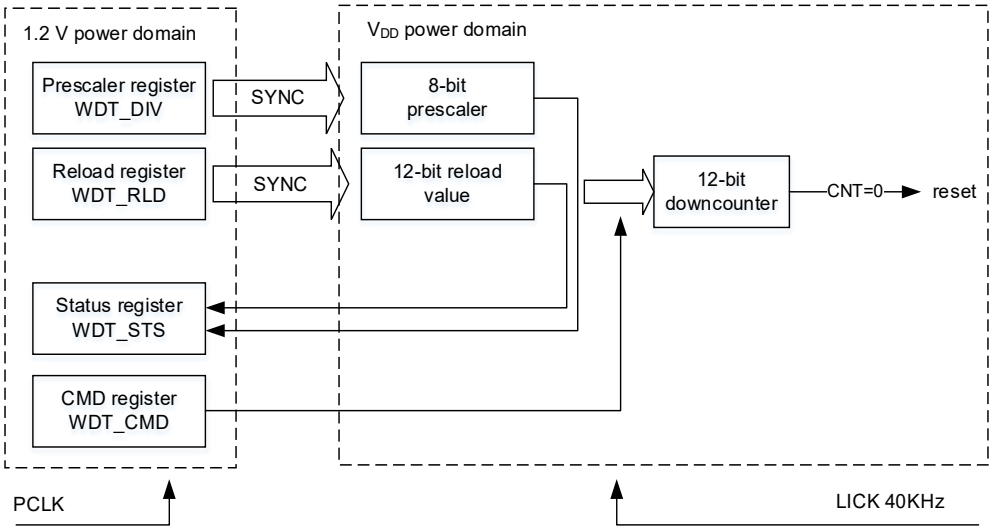


Figure 132. WDT block diagram

Table 62. WDT timeout period (LICK=40kHz)

Prescaler divider	DIV[2:0] bits	Min.timeout (ms) RLD[11:0] = 0x000	Max. timeout (ms) RLD[11:0] = 0xFFF
/4	0	0.1	409.6
/8	1	0.2	819.2
/16	2	0.4	1638.4
/32	3	0.8	3276.8
/64	4	1.6	6553.6
/128	5	3.2	13107.2
/256	(6 or 7)	6.4	26214.4

Debug mode

When the microcontroller enters debug mode (Cortex™-M4F core halted), the WDT counter stops counting by setting the WDT_PAUSE in the DEBUG module.

WDT registers

These peripheral registers must be accessed by words (32 bits).

Table 63. WDT register and reset value

Register	Offset	Reset value
WDT_CMD	0x00	0x0000 0000
WDT_DIV	0x04	0x0000 0000
WDT_RLD	0x08	0x0000 0FFF
WDT_STS	0x0C	0x0000 0000

Command register (WDT_CMD)

(Reset in Standby mode)

Bit	Register	Reset value	Type	Description
Bit 31:16	Reserved	0x0000	resd	Kept at its default value.
Bit 15:0	CMD	0x0000	wo	Command register 0xAAAA: Reload counter 0x5555: Unlock write-protected WDT_DIV and WDT_RLD 0xCCCC: Enable WDT. If the hardware watchdog has been enabled, ignore this operation.

Divider register (WDT_DIV)

Bit	Register	Reset value	Type	Description
Bit 31:3	Reserved	0x0000 0000	resd	Kept at its default value.
Bit 2:0	DIV	0x0	rw	Clock division value 000: LICK divided by 4 001: LICK divided by 8 010: LICK divided by 16 011: LICK divided by 32 100: LICK divided by 64 101: LICK divided by 128 110: LICK divided by 256 111: LICK divided by 256 The write protection must be unlocked in order to enable write access to the register. The register can be read only when DIVF=0.

Reload register (WDT_RLD)

(Reset in Standby mode)

Bit	Register	Reset value	Type	Description
Bit 31:12	Reserved	0x00000	resd	Kept at its default value.
Bit 11:0	RLD	0xFFF	rw	Reload value The write protection must be unlocked in order to enable write access to the register. The register can be read only when RLDF=0.

Status register (WDT_STS)

Bit	Register	Reset value	Type	Description
Bit 31:2	Reserved	0x0000 0000	resd	Kept at its default value.
Bit 1	RLDF	0x0	ro	Reload value update complete flag 0: Reload value update complete 1: Reload value update is in process. The reload register WDT_RLD can be written only when RLDF=0
Bit 0	DIVF	0x0	ro	Division value update complete flag 0: Division value update complete 1: Division value update is in process. The divider register WDT_DIV can be written only when DIVF=0

17 Real-time clock (RTC)

RTC introduction

The real-time clock provides a calendar clock function. It has an internal 32-bit incremental couner that is increased by one at each second. In other words, this counter serves as a second clock. The current seond value can be converted into time and date to provide a calendar function. The time and date can be modified by modifying the counter value.

The RTC module is in battery powered domain, which means that it keeps running and free from the influence of system reset and V_{DD} power off as long as V_{BAT} is powered.

Main features

- 20-bit prescaler
- 32-bit counter
- Three RTC clock sources: HEXT/128, LEXT and LICK
- Three interrupts: Second interrupt, Alarm interrupt and Overflow interrrupt

Note: The frequency of the RTC clock must be one forth slower than the PCLK1 clock.

RTC structure

RTC consists of an APB1 interface and a RTC counter logic, as shown in the following diagram.

APB1 interface: It is used to interface with the APB1 bus and battery powered domain for the configuration and read operation of the RTC registers.

RTC counter logic: It consists of a 20-bit divider and a 32-bit programmable counter. The prescaler is used to generate the RTC count clock, LN_CLK, which is usually set to 1 second in order to convert the counter value into a calendar. As the RTC counter logic is in the V_{BAT} domain and driven by the RTC_CLK, the RTC still keeps running even if the APB1 interface is disabled. If the RTC_CLK frequency is 32.768kHz, write the value 0x7FFF in the prescaler load register can produce a LN_CLK of 1Hz.

The RTC counter logic is independent from the APB1 interface. The RTC registers can be configured through the APB1 interface and synchronized to the RTC counter module through the RTC_CLK; The associated flag bits arising from the RTC counter module is synchronized to the RTC registers by the PCLK1. The RTC counter module is driven by the RTC_CLK. Set RTCEN=1 to enable RTC_CLK. Configure the RTC_CLK clock source by setting the RTCSEL[1:0]. To re-configure the RTC_CLK, it must wait until the reset of the battery powered domain before configuration.

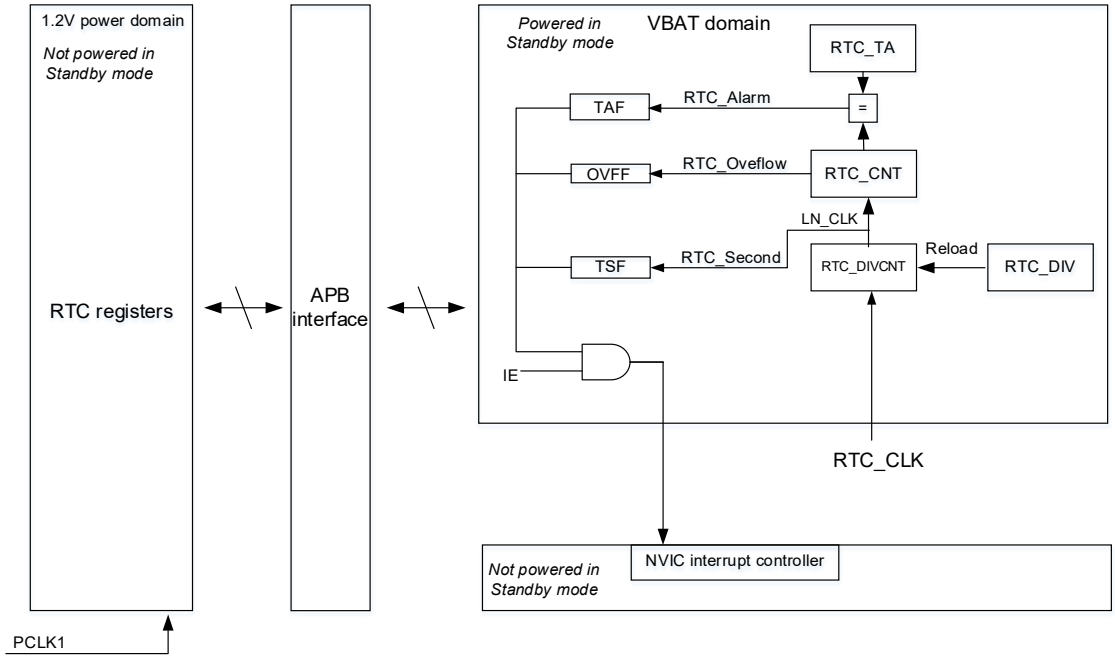


Figure 133. Simplified RTC block diagram

Functional descriptions

Configuring RTC registers

After power-on reset, all RTC registers are write protected. Write access to the RTC registers is allowed only when the write protection is unlocked.

Configuration procedure:

- Enable power and battery powered domain interface clock by setting PWCEN =1 and BPREN=1 in the CRM_APB1EN register
- Unlock write protection in the battery powered domain by setting BPWEN=1 in the PWC_CTRL register

Configuring DIV, CNT and ALA registers:

To enable write operation to these registers, the first step is to enter configuration mode (CFGEN = 1). Setting CFGEN = 0 to exit configuration mode, the values in these registers are actually written to the battery powered domain, which takes at least three RTCCLK cycles to complete.

Based on synchronization circuit, a new value can be written to the RTC registers onlh when the previous RTC configuration is completed (CFGF=1).

Configuration procedure:

1. Wait until the end of register configuration (CFGF=1)
2. Enter configuration mode (CFGEN=1)
3. Configure the corresponding RTC registers

4. Exit configuration mode (CFGEN=0)

5. Wait until the end of register configuration (CFGF=1)

The registers including DIV, ALA, CNT and DIVCNT are reset only by the reset signals in the battery powered domain. The rest of the registers are asynchronously reset by system reset or power reset.

Reading RTC registers

Based on synchronization circuit, when reading the RTC registers, the correct values have yet been uploaded from the battery powered domain to the APB1 interface if one of the following events occurred:

A system reset or power reset has occurred;

The microcontroller has woken up from Standby or Stop modes.

At this time, the software must wait until UPDF=1 before read operation, otherwise, an error value is returned.

RTC interrupts

RTC supports the following interrupt requests:

- Second interrupt: If Second interrupt is enabled (TSIEN=1), a second interrupt is generated at each LN_CLK period.
- Alarm interrupt: If Alarm interrupt is enabled (TAIEN=1), an alarm interrupt is generated when the value in the TA register is equal to the CTN value.
- Overflow interrupt: If Overflow interrupt is enabled (OVFIEN=1), an overflow interrupt is generated when the counter reaches the value 0xFFFFFFFF.

The RTC global interrupt vector (RTC_IRQn) and alarm interrupt vector (RTCAlarm_IRQn) are both supported. To wake up from DEEPSLEEP mode using the RTC alarm interrupt, the RTC alarm interrupt must be enabled to use the RTCAlarm_IRQn vector, and the EXINT 17 is configured as interrupt mode at the same time; To wake from DEEPSLEEP mode using the RTC alarm event, the EXINT 17 must be configured as event mode, but without the need of enabling the RTC alarm interrupt. When the RTC alarm event is used to wake up from Standby mode, it is unnecessary to enable alarm interrupt and EXINT 17.

RTC flag bits are described as follows:

- RTC Second flag (TSF): It indicates the update of RTC counter. The Second flag is set one RTC_CLK period before the update of the RTC counter
- RTC Alarm flag (TAF): The flag is set one RTC_CLK period before the counter value reaches the RTC alarm value in the alarm register increased by one (TA+1)
- RTC Overflow flag (OVFF): The flag is set one RTC_CLK period before the RTC counter value reaches the value 0x00000000

When the RTC interrupts are generated, clearing the corresponding flag bits means that the interrupt requests have been received. The flag bits can only be set by hardware, and cleared by software. After reset, all interrupts will be disabled. The flag bits will no longer be updated when the APB1 clock stops running.

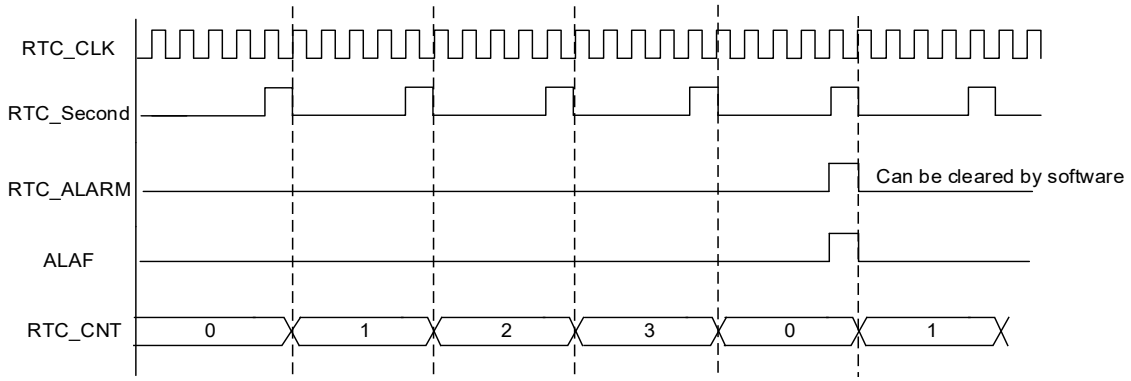


Figure 134. RTC second and alarm waveform example with DIV=0004 and TA=00003

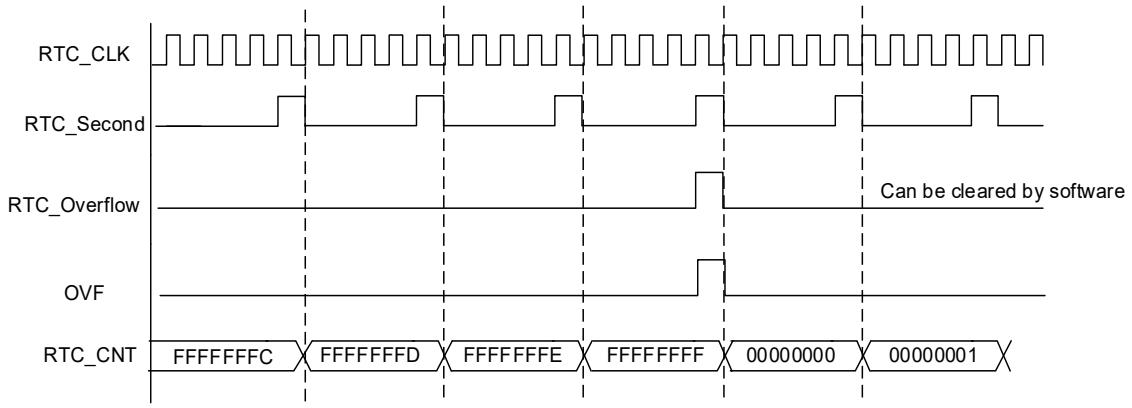


Figure 135. RTC overflow waveform example with DIV=0004

RTC registers

These peripheral registers must be accessed by word (32 bits). RTC registers are 16-bit addressable registers.

Table 64. RTC register map and reset values

Register	Offset	Reset value
RTC_CTRLH	0x00	0x0000
RTC_CTRLL	0x04	0x0020
RTC_DIVH	0x08	0x0000
RTC_DIVL	0x0C	0x8000
RTC_DIVCNTH	0x10	0x0000
RTC_DIVCNTL	0x14	0x8000
RTC_CNTH	0x18	0x0000
RTC_CNTL	0x1C	0x0000
RTC_TAH	0x20	0xFFFF
RTC_TAL	0x24	0xFFFF

RTC control register high (RTC_CTRLH)

Bit	Register	Reset value	Type	Description
Bit 15:3	Reserved	0x0000	resd	Kept at its default value.
Bit 2	OVFIEN	0x0	rw	Overflow interrupt enable This bit is used to enable overflow interrupt. 0: Disabled 1: Enabled
Bit 1	TAIEN	0x0	rw	Time alarm interrupt enable This bit is used to enable alarm interrupt. 0: Disabled 1: Enabled
Bit 0	TSIEN	0x0	rw	Time second interrupt enable This bit is used to enable second interrupt. 0: Disabled 1: Enabled

Note: This register is reset after system reset. Refer to Section 17.4.1 for more details.

RTC control register low (RTC_CTRLL)

Bit	Register	Reset value	Type	Description
Bit 15:6	Reserved	0x000	resd	Kept at its default value.
Bit 5	CFGF	0x1	ro	RTC configuration finish Indicates whether the last write operation on the RTC registers has been completed or not. Write access to the RTC registers is allowed only when this bit is set. 0: Last write operation on RTC registers is ongoing 1: Last write operation on RTC registers ends
Bit 4	CFGEN	0x0	rw	RTC Configuration enable This bit is set to enter configuration mode in order to enable write access to the CNT, ALA, DIVCNT registers. 0: Exit configuration mode 1: Enter configuration mode
Bit 3	UPDF	0x0	rw0c	RTC update finish flag This bit indicates wheter the update of the RTC registers has been completed or not. This bit is set by hardware when the CNT and DIVCNT are updated. Before any read operation, this bit must be cleared by software, and the user must wait until this bit is set. 0: RTC registers not updated 1: RTC registers updated
Bit 2	OVFF	0x0	rw0c	Overflow flag This bit is set when the counter overflows. An interrupt is generarted if OVFIEN =1. 0: Overflow not detected 1: Overflow occurred
Bit 1	TAF	0x0	rw0c	Time alarm flag This bit is set when an alarm event is detected. An interrupt is generated if TAIEN =1. 0: Alarm not detected 1: Alarm detected

Bit	Register	Reset value	Type	Description
Bit 0	TSF	0x0	rw0c	Time second flag This bit is set when a second event is detected. An interrupt is generated if TSIEN =1. 0: Second event not detected 1: Second event detected

RTC divider register (RTC_ DIVH/RTC_DIVL)

RTC divider register high (RTC_DIVH)

Bit	Register	Reset value	Type	Description
Bit 15:4	Reserved	0x000	resd	Kept at its default value.
Bit 3:0	DIV	0x0	wo	RTC divider This field is used to define the counter clock frequency according to the formula: $f_{LN_CLK} = f_{RTCCLK}/(DIV[19:0]+1)$

RTC divider register high (RTC_DIVL)

Bit	Register	Reset value	Type	Description
Bit 15:0	DIV	0x8000	wo	RTC divider This field is used to define the counter clock frequency according to the formula: $f_{LN_CLK} = f_{RTCCLK}/(DIV[19:0]+1)$ Note: the zero value is not recommended.

RTC divider counter register (RTC_ DIVCNTH/RTC_DIVCNTL)

RTC divider counter register high (RTC_DIVCNTH)

Bit	Register	Reset value	Type	Description
Bit 15:4	Reserved	0x000	resd	Kept at its default value.
Bit 3:0	DIVCNT	0x0	ro	RTC clock divider counter

RTC divider counter register low (RTC_DIVCNTL)

Bit	Register	Reset value	Type	Description
Bit 15:0	DIVCNT	0x8000	ro	RTC clock divider counter

RTC counter value register (RTC_CNTH/RTC_CNTL)

RTC counter value register high (RTC_CNTH)

Bit	Register	Reset value	Type	Description
Bit 15:0	CNT	0x0000	rw	RTC counter value This field is used to configure or read the high part of the RTC counter value.

RTC counter value register low (RTC_CNTL)

Bit	Register	Reset value	Type	Description
Bit 15:0	CNT	0x0000	rw	RTC counter value This field is used to configure or read the low part of the RTC counter value.

RTC alarm register (RTC_TAH/RTC_TAL)

RTC alarm register high (RTC_TAH)

Bit	Register	Reset value	Type	Description
Bit 15:0	TA	0xFFFF	wo	Time alarm clock value This field is used to define the high part of the alarm value.

RTC alarm register low (RTC_TAL)

Bit	Register	Reset value	Type	Description
Bit 15:0	TA	0xFFFF	wo	Time alarm clock value This field is used to define the low part of the alarm value.

18 Battery powered registers (BPR)

BPR introduction

The battery powered registers are located in the battery powered domain and powered by V_{DD}/V_{BAT}. These registers are forty two 16-bit registers. Upon a tamper event or when battery powered domain reset occurs, the contents in these registers are cleared so as to ensure the highest level of data security.

Main features

- Forty two 16-bit registers
- Reset at a tamper event
- Configurable PC13 pin multiplexed function output

Functional descriptions

To enable access to the battery powered registers, the PWCEN, BPREN and BPWEN must be set.

The BPR provides tamper detection function for the purpose of data security. If enabled, the polarity of the TAMPER pin is configured with TPP bit. Once a tamper event is detected, the TPEF bit will be set, and the battery powered registers will be cleared accordingly; If the tamper interrupt is enabled, an interrupt will be generated, with the TPIF bit being set.

In addition, the BPR also has RTC calibration feature that can be slowed down up to 121 ppm by setting the CALVAL[6:0] bits. If the RTC calibration output is enabled, the RTC clock with a frequency divided by 64 can be output on the TAMPER pin (CCOS=1).

Note: When TPP=0 or 1, if the TAMPER pin is already high or low before it is enabled by setting the TPEN bit, an extra tamper event is generated when TPEN=1, despite the fact that there was no rising or falling edge on the TAMPER pin.

BPR registers

These peripheral registers must be accessed by words (32 bits). BPR registers are 16-bit addressable registers.

Table 65. BPR register map and reset values

Register	Offset	Reset value
BPR_DT1	0x04	0x0000 0000
BPR_DT2	0x08	0x0000 0000
BPR_DT3	0x0C	0x0000 0000
BPR_DT4	0x10	0x0000 0000
BPR_DT5	0x14	0x0000 0000
BPR_DT6	0x18	0x0000 0000
BPR_DT7	0x1C	0x0000 0000
BPR_DT8	0x20	0x0000 0000
BPR_DT9	0x24	0x0000 0000

Register	Offset	Reset value
BPR_DT10	0x28	0x0000 0000
BPR_RTCCAL	0x2C	0x0000 0000
BPR_CTRL	0x30	0x0000 0000
BPR_CTRLSTS	0x34	0x0000 0000
BPR_DT11	0x40	0x0000 0000
BPR_DT12	0x44	0x0000 0000
BPR_DT13	0x48	0x0000 0000
BPR_DT14	0x4C	0x0000 0000
BPR_DT15	0x50	0x0000 0000
BPR_DT16	0x54	0x0000 0000
BPR_DT17	0x58	0x0000 0000
BPR_DT18	0x5C	0x0000 0000
BPR_DT19	0x60	0x0000 0000
BPR_DT20	0x64	0x0000 0000
BPR_DT21	0x68	0x0000 0000
BPR_DT22	0x6C	0x0000 0000
BPR_DT23	0x70	0x0000 0000
BPR_DT24	0x74	0x0000 0000
BPR_DT25	0x78	0x0000 0000
BPR_DT26	0x7C	0x0000 0000
BPR_DT27	0x80	0x0000 0000
BPR_DT28	0x84	0x0000 0000
BPR_DT29	0x88	0x0000 0000
BPR_DT30	0x8C	0x0000 0000
BPR_DT31	0x90	0x0000 0000
BPR_DT32	0x94	0x0000 0000
BPR_DT33	0x98	0x0000 0000
BPR_DT34	0x9C	0x0000 0000
BPR_DT35	0xA0	0x0000 0000
BPR_DT36	0xA4	0x0000 0000
BPR_DT37	0xA8	0x0000 0000
BPR_DT38	0xAC	0x0000 0000
BPR_DT39	0xB0	0x0000 0000
BPR_DT40	0xB4	0x0000 0000
BPR_DT41	0xB8	0x0000 0000
BPR_DT42	0xBC	0x0000 0000

Battery powered data register x (BPR_DTx) (x = 1 … 42)

Bit	Register	Reset value	Type	Description
Bit 15:0	DT	0x0000	rw	Battery powered domain data This field is used for data storage. The BPR_DTx registers can be set only by a battery powered domain reset or by a tamper event.

RTC calibration register (BPR_ RTCCAL)

Bit	Register	Reset value	Type	Description
Bit 15:12	Reserved	0x0	resd	Kept at its default value.
Bit 11	OUTM	0x0	rw	Output mode This bit is used to select the output mode of alarm output or second output: 0: Pulse output (The output pulse width is one RTC clock period) 1: Toggle output (The corresponding pin output level changes at each time when an alarm event or second event is detected) Note: This bit is reset only by a battery powered domain reset.
Bit 10	CCOS	0x0	rw	Calibration clock output selection 0: Calibration clock output is disabled 1: Calibration clock output is enabled Note: This bit is cleared only by a battery powered domain reset.
Bit 9	OUTSEL	0x0	rw	Output selection This bit is used to select either the RTC alarm event or the second event. 0: RTC alarm event output 1: Second event output Note: This bit is cleared only by a battery powered domain reset.
Bit 8	OUTEN	0x0	rw	Output enable 0: Disabled 1: Enabled Note: This bit is cleared only by a battery powered domain reset. It is used to enable the event that is output on the TAMPER pin. The TAMPER function can not be used if the output is enabled.
Bit 7	CALOUT	0x0	rw	Calibration clock output 0: No effect 1: Output the RTC clock with a frequency divided by 64 on the TAMPER pin. The TAMPER function can not be used when the calibration clock output is enabled. Note: This bit is cleared when the V_{DD} supply is powered off.
Bit 6:0	CALVAL	0x00	rw	Calibration value This value indicates the number of clock filtered in one cycle (2^{20} clocks). The clock frequency is reduced with a minium accuracy of $1000000/2^{20}$ppm. The RTC clock can be slowed down from 0 to 121 ppm.

BPR control register (BPR_ CTRL)

Bit	Register	Reset value	Type	Description
Bit 15:2	Reserved	0x0000	resd	Kept at its default value.
Bit 1	TPP	0x0	rw	TAMPER pin polarity This bit defines the polarity of the TAMPER pin. The contents in the data registers are cleared when an active level is detected. 0: Active high 1: Active low Note: To avoid unwanted tamper event, it is recommended to modify the polarity of the TAMPER pin when it is disabled.
Bit 0	TPEN	0x0	rw	TAMPER pin enable 0: Disabled. The TAMPER pin can be used as GPIO. 1: Enabled

BPR control/status register (BPR_ CTRLSTS)

Bit	Register	Reset value	Type	Description
Bit 15:10	Reserved	0x00	resd	Kept at its default value.
Bit 9	TPIF	0x0	ro	Tamper interrupt flag This bit is set when a tamper event is detected and the TPIEN is set. 0: No tamper event 1: A tamper event is detect Note: This bit is reset only a system reset or exit Standby mode.
Bit 8	TPEF	0x0	ro	Tamper event flag This bit is set when a tamper event is detected. 0: No tamper event 1: A tamper event is detected Note: A tamper event will reset the BPR_DTx registers. Do not write the BPR_DTx registers when TPEF=1.
Bit 7:3	Reserved	0x00	resd	Kept at its default value.
Bit 2	TPIEN	0x0	rw	Tamper pin interrupt enable 0: Disabled 1: Enabled Note: A tamper interrupt does not wake up the core from low-power modes.
Bit 1	TPIFCLR	0x0	wo	Tamper interrupt flag clear Setting this bit clears the TAMPER interrupt. 0: No effect 1: Clear the tamper interrupt
Bit 0	TPEFCLR	0x0	wo	Tamper event flag clear Setting this bit clears the TAMPER event flag. 0: No effect 1: Clear the tamper event flag

19 Analog-to-digital converter (ADC)

ADC introduction

The ADC is a peripheral that converts an analog input signal into a 12-bit digital signal. Its sampling rate is as high as 2 M_{SPS}. It has up to 18 channels for sampling and conversion.

Main features

In terms of analog part:

- 12-bit resolution
- Self-calibration time: 154 ADC clock cycles
- ADC conversion time
- ADC conversion time is 0.5 μs at 28 MHz (If the system clock is at 240 MHz, then the ADC clock maximum frequency is at 20 M, and the conversion time is 0.7 μs)
- ADC supply requirement: 2.6 V to 3.6 V
- ADC input range: $V_{REF-} \leq V_{IN} \leq V_{REF+}$

In terms of digital control:

- Regular channels and injected channels with different priority
- Regular channels and injected channels both have their own trigger detection circuit
- Each channel can independently define its own sampling time
- Conversion sequence supports various conversion modes
- Optional data alignment mode
- Programmable voltage monitor threshold
- Regual channels with DMA transfers
- Interrupt generation at one of the following events:
  - End of the conversion of preempted group
  - End of the conversion of channels
  - Voltage outside the threshold programmed
  - ADC Master/slave mode

ADC structure

The following diagram shows the block diagram of ADC1.

Differences between ADC2 and ADC1:

1. ADC2 is not connected to the internal temperature sensor and internal reference voltage
2. ADC2 has no DMA request. Refer to Section 19.4.4.2 for more details

Differences between ADC3 and ADC1:

1. ADC3 is not connected to the internal temperature sensor and internal reference voltage
2. ADC3 has different external analog input channel pins from ADC1. Refer to Section 19.4.1 for details
3. ADC3 has different trigger sources from ADC1. Refer to Section 19.4.2.2 for more details

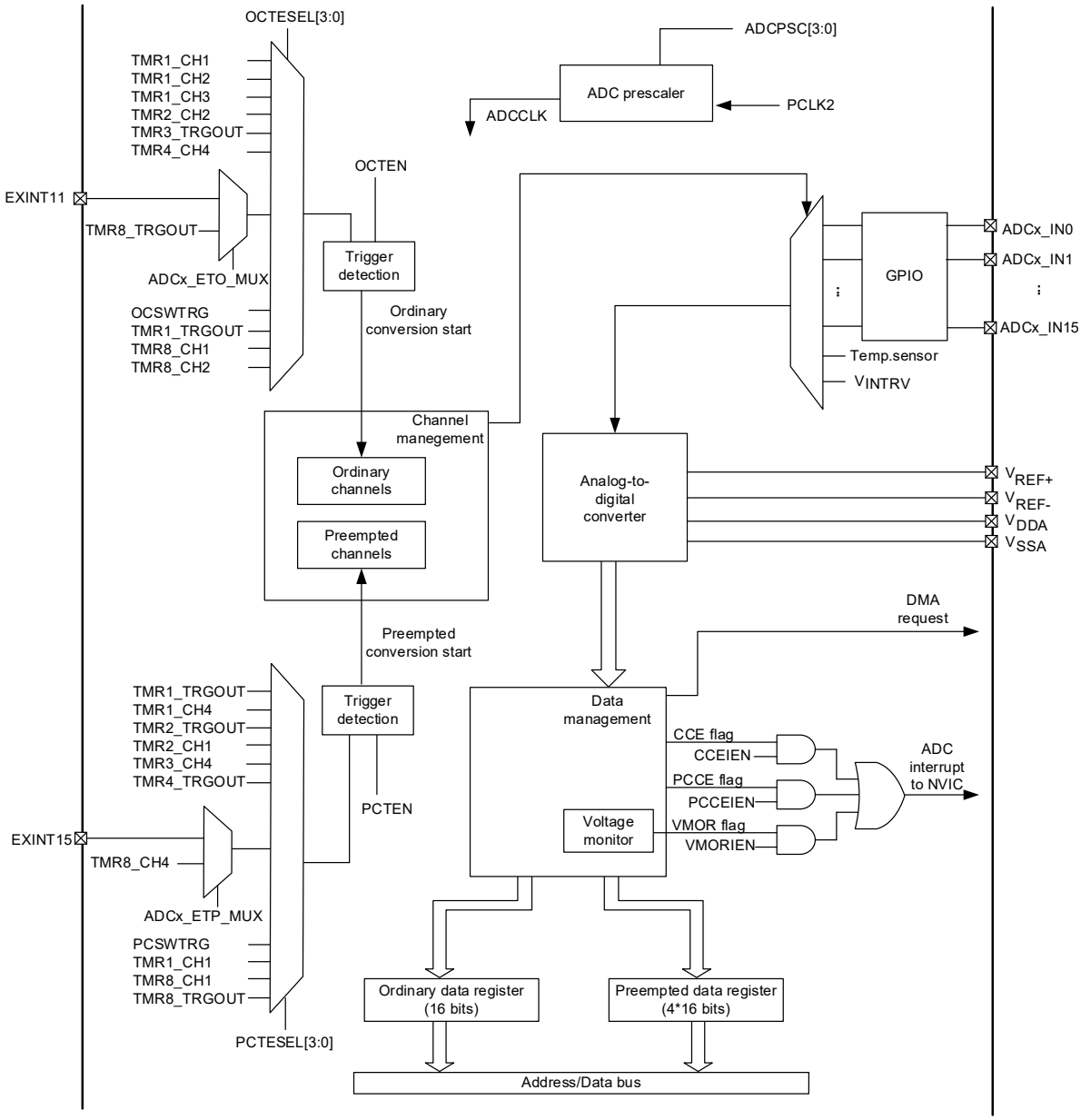


Figure 136. ADC1 block diagram

Input pin description:

- V_{DDA}: Analog supply, ADC analog supply, can be tied to V_{DD}, or $2.6V \leq V_{DDA} \leq V_{DD}$ (3.6V)
- V_{SSA}: Analog supply ground, ADC analog supply ground, must be tied to V_{SS}
- V_{REF+}: Analog reference positive, high/positive reference voltage for ADC, $2.0V \leq V_{REF+} \leq V_{DDA}$
- V_{REF-}: Analog reference negative, low/negative reference voltage for ADC, must be tied to V_{SS}
- ADCx_IN: Analog input signal channels

Functional descriptions

Channel management

Analog signal channel input

There are 18 analog signal channel inputs for each of the ADCs, expressed by ADC_INx (x = 0 to 17).

- ADC1_IN0 to ADC1_IN15 are referred to as the external analog input, ADC1_IN16 as the internal temperature sensor, and ADC1_IN17 as the internal reference voltage.
- ADC2_IN0 to ADC2_IN15 are referred to as the external analog input, and ADC2_IN16 and ADC2_IN17 as V_{SS}
- ADC3_IN0 to ADC3_IN3, and ADC3_IN10 to ADC3_IN13 are referred to as the external analog input, the rest of them are V_{SS}.

Channel conversion

The conversions are divided into two groups: ordinary and preempted. The preempted group has priority over the ordinary group.

If the preempted channel trigger occurs during the ordinary channel conversion, then the ordinary channel conversion is interrupted, giving the priority to the preempted channel, and the ordinary channel continues its conversion at the end of the preempted channel conversion. If the ordinary channel trigger occurs during the preempted channel conversion, the ordinary channel conversion won't start until the end of the preempted channel conversion.

Program the ADC_INx into the ordinary channel sequence (ADC_OSQx) and the preempted channel sequence (ADC_PSQ), and the same channel can be repeated, the total number of sequences is determined by OCLEN and PCLEN, then it is ready to enable the ordinary channel or preempted channel conversion.

Internal temperature sensor

The temperature sensor is connected to ADC1_IN16. Before the temperature sensor channel conversion, it is required to enable the ITSRVEN bit in the ADC_CTRL2 register and wait after power-on time.

Obtain the temperature

Obtain the temperature using the following formula:

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

Where,

$V_{25} = V_{SENSE}$ value for 25° C and

Avg_Slope = Average Slope for curve between Temperature vs. V_{SENSE} (given in mV/° C).

Internal reference voltage

The internal reference voltage of the typical value 1.2 V is connected to ADC1_IN17. It is required to enable the ITSRVEN bit in the ADC_CTRL2 register before the internal reference channel conversion. The converted data of such channel can be used to calculate the external reference voltage.

ADC operation process

The following diagram shows the basic operation process of the ADC. It is recommended to do the calibration after the initial power-on in order to improve the accuracy of sampling and conversion. After the calibration, trigger is used to enable ADC sampling conversion. Read data at the end of the conversion.

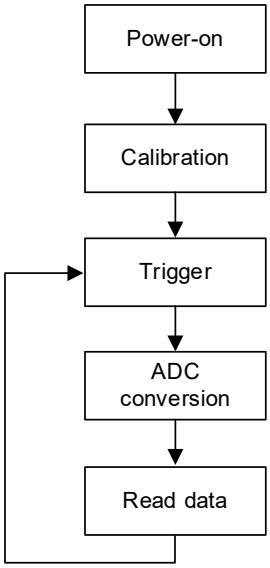


Figure 137. ADC basic operation process

Power-on and calibration

Power-on

Set the ADCxEN bit in the CRM_APB2EN register to enable ADC clocks: PCLK2 and ADCCLK.

Program the desired ADCCLK frequency by setting the ADCDIV bit in the CRM_CFG register. The ADCCLK is derived from PCLK2 frequency division.

Note: ADCCLK must be less than 28 MHz.

Then set the ADCEN bit in the ADC_CTRL2 register to supply the ADC, and wait until the t_{STAB} is reached before subsequent operations. Clear the ADCEN bit will stop the ADC conversion and result in a reset. In the meantime, the ADC is switched off to save power.

Calibration

After power-on, the calibration is enabled by setting the ADCAL bit in the ADC_CTRL2 register. When the calibration is over, the ADCAL bit is cleared by hardware and the conversion is performed by software trigger.

After each calibration, the calibration value is stored in the ADC_ODT register, and then value is automatically sent back to the ADC so as to eliminate capacitance errors. The storage of the calibration value will not set the CCE flag, or generate interrupts or DMA requests.

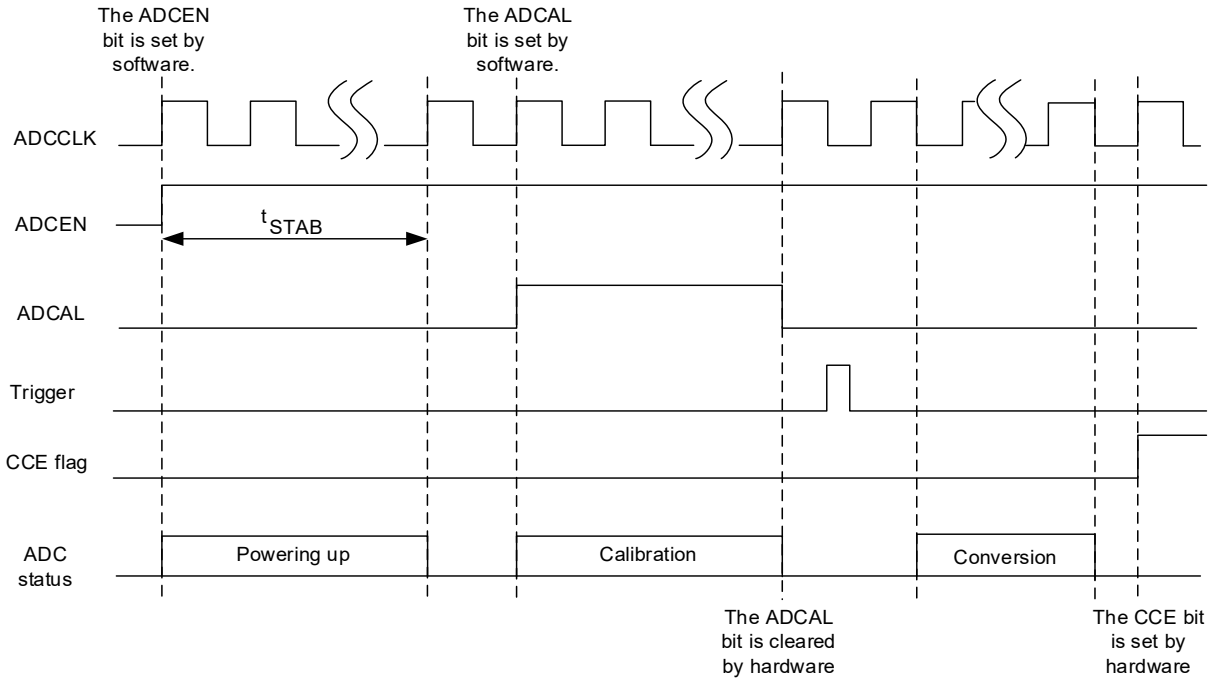


Figure 138. ADC power-on and calibration

Trigger

The ADC triggers contain ordinary channel trigger and preempted channel trigger. The ordinary channel conversion is triggered by ordinary channel triggers while the preempted channel conversion is triggered by preempted ones. When the OCTEN or PCTEN bit is set in the ADC_CTRL2 register, only the rising edge of the trigger source can start the conversion.

The conversion can be triggered by software write operation to the OCSWTRG and PCSWTRG bits in the ADC_CTRL2 register, or by an external event. The external events include timer and pin triggers. The OCTESEL and PCTESEL bits in the ADC_CTRL2 register are used to select specific trigger sources, as shown in the following tables.

The ordinary channel has another special trigger source, that is, enable the ADCEN bit repeatedly to trigger the conversion. In this case, the ordinary channel conversion can be triggered without the need of the OCTEN enable bit in the ADC_CTRL2 register.

Table 66. Trigger sources for ADC1 and ADC2

OCTESEL	Source	PCTESEL	Source
0000	TMR1_CH1 event	0000	TMR1_TRGOUT event
0001	TMR1_CH2 event	0001	TMR1_CH4 event
0010	TMR1_CH3 event	0010	TMR2_TRGOUT event
0011	TMR2_CH2 event	0011	TMR2_CH1 event
0100	TMR3_TRGOUT event	0100	TMR3_CH4 event
0101	TMR4_CH4 event	0101	TMR4_TRGOUT event

OCTESEL		Source	PCTESEL		Source
0110	ADCx_ETO_MUX=0	EXINT line11 external pin	0110	ADCx_ ETP_MUX=0	EXINT line15 external pin
	ADCx_ETO_MUX=1	TMR8_TRGOUT event		ADCx_ ETP_MUX=1	TMR8_CH4 event
0111		OCSWTRG bit	0111		PCSWTRG bit
1000		Reserved	1000		Reserved
1001		Reserved	1001		Reserved
1010		Reserved	1010		Reserved
1011		Reserved	1011		Reserved
1100		Reserved	1100		Reserved
1101		TMR1_TRGOUT event	1101		TMR1_CH1 event
1110		TMR8_CH1 event	1110		TMR8_CH1 event
1111		TMR8_CH2 event	1111		TMR8_TRGOUT event

Table 67. Trigger sources for ADC3

OCTESEL	Source	PCTESEL	Source
0000	TMR3_CH1 event	0000	TMR1_TRGOUT event
0001	TMR2_CH3 event	0001	TMR1_CH4 event
0010	TMR1_CH3 event	0010	TMR4_CH3 event
0011	TMR8_CH1 event	0011	TMR8_CH2 event
0100	TMR8_TRGOUT event	0100	TMR8_CH4 event
0101	TMR5_CH1 event	0101	TMR5_TRGOUT event
0110	TMR5_CH3 event	0110	TMR5_CH4 event
0111	OCSWTRG bit	0111	PCSWTRG bit
1000	Reserved	1000	Reserved
1001	Reserved	1001	Reserved
1010	Reserved	1010	Reserved
1011	Reserved	1011	Reserved
1100	Reserved	1100	Reserved
1101	TMR1_TRGOUT event	1101	TMR1_CH1 event
1110	TMR1_CH1 event	1110	TMR1_CH2 event
1111	TMR8_CH3 event	1111	TMR8_TRGOUT event

Sampling and conversion sequence

The sampling period can be configured by setting the CSPTx bit in the ADC_SPT1 and ADC_SPT2 registers. A single one conversion time is calculated with the following formula:

A single one conversion tiem (ADCCLK period) = sampling time + 12.5

Example:

If the CSPTx selects 1.5 period, one conversion requires 1.5+12.5=14 ADCCLK periods

If the CSPTx selects 7.5 period, one conversion requires 7.5+12.5=20 ADCCLK periods.

Conversion sequence management

Only one channel is converted at each trigger event by default, that is, OSN1-defined channel or PSN4-defined channel.

The detailed conversion sequence modes are described in the following sections. With this, the channels can be converted in a specific order.

Sequence mode

The sequence mode is enabled by setting the SQEN bit in the ADC_CTRL1 register. The ADC_OSQx registers are used to configure the sequence and total number of the ordinary channels while the ADC_PSQ register is used to define the sequence and total number of the preempted channels. If the sequence mode is enabled, a single trigger event enables the conversion of a group of channels in order. The ordinary channels start converting from the QSN1 while the preempted channels starts from the PSNx, where x=4-PCLEN. The following diagram shows an example of the behavior in sequence mode.

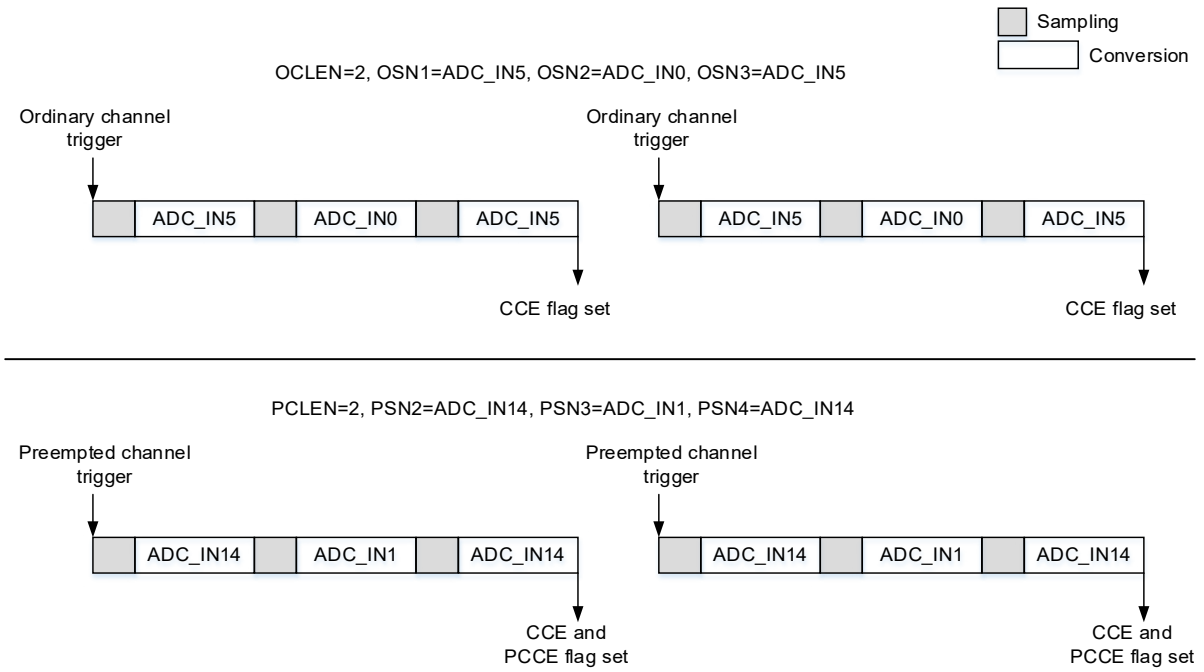


Figure 139. Sequence mode

Automatic preempted group conversion mode

The automatic preempted group conversion mode is enabled by setting the PCAUTOEN bit in the ADC_CTRL1 register. Once the ordinary channel conversion is over, the preempted group will automatically continue its conversion. This mode can work with the sequence mode. The preempted group conversion starts automatically at the end of the conversion of the ordinary group. The following diagram shows an example of the behavior when the automatic preempted group conversion mode works with the ordinary group.

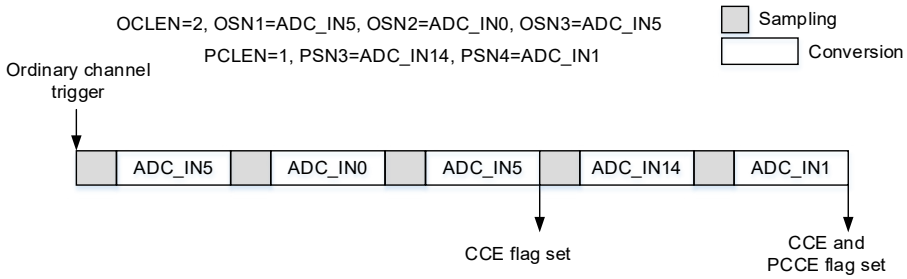


Figure 140. Preempted group auto conversion mode

Repetition mode

The repetition mode is enabled by setting the RPEN bit in the ADC_CTRL2 register. When a trigger signal is detected, the ordinary channels will be converting repeatedly. This mode can work with the ordinary channel conversion in the sequence mode to enable the repeated conversion of the ordinary group. Such mode can also work with the preempted group auto conversion mode to repeatedly convert the ordinary group and preempted group in sequence. The following diagram shows an example of the behavior when the repetition mode works with the sequence mode and preempted group auto conversion mode.

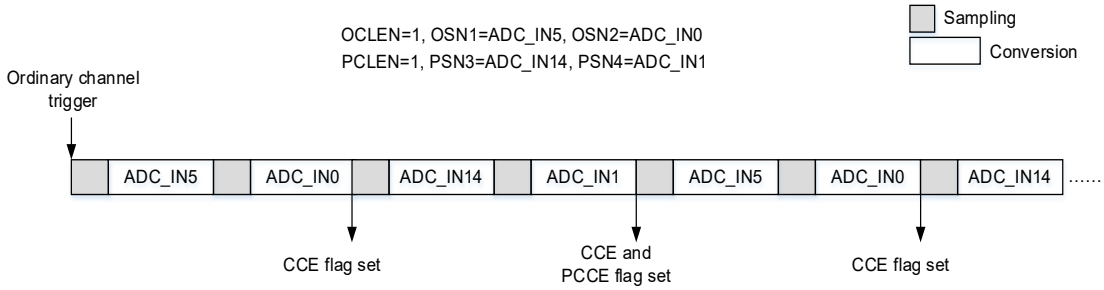


Figure 141. Repetition mode

Partition mode

The partition mode of the ordinary group can be enabled by setting the OCPEN bit in the ADC_CTRL1 register. In this mode, the ordinary group conversion sequence length (OCLEN bit in the ADC_OSQ1 register) is divided into a smaller sub-group, in which the number of the channels is programmed with the OCPCNT bit in the ADC_CTRL1 register. A single trigger event will enable the conversion of all the channels in the sub-group. Each trigger event selects different sub-group in order.

Set the PCPEN bit in the ADC_CTRL1 register will enable the partition mode of the preempted group. In this mode, the ordinary group conversion sequence length (OCLEN bit in the ADC_OSQ1 register) is divided into a sub-group with only one channel. A single one trigger event will convert the channel in the sub-group. Each trigger event select different sub-group in order.

The partition mode cannot be used with the repetition mode at the same time. The partition mode of the ordinary group cannot be used with that of the preempted group at the same time. The following diagram shows an example of the behavior in partition mode for ordinary group and preempted group.

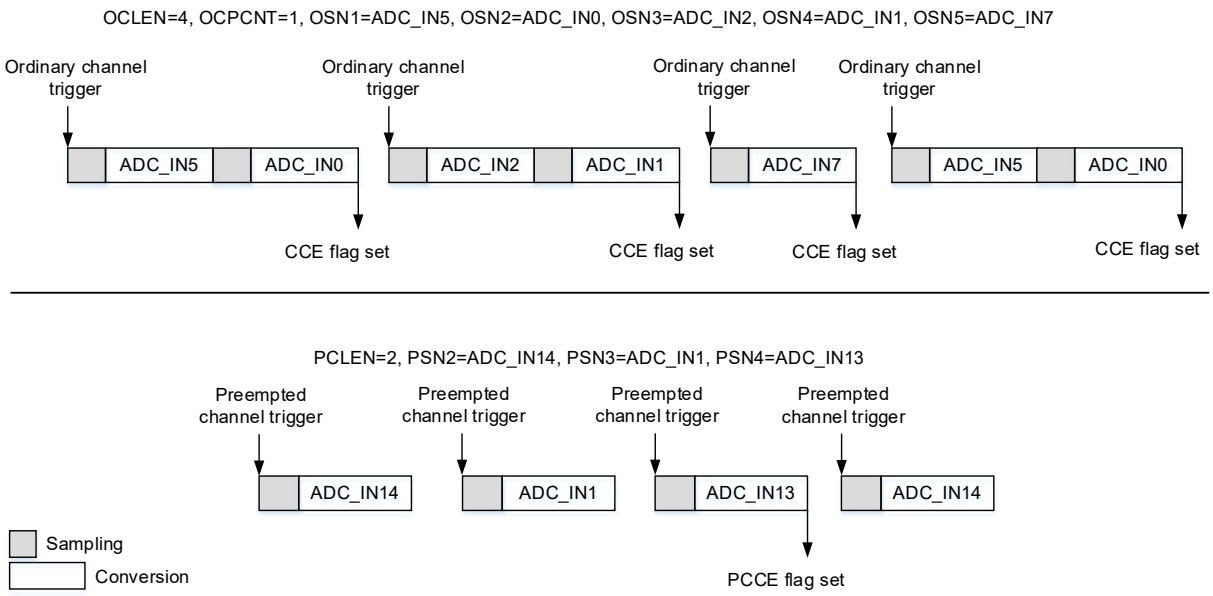


Figure 142. Partition mode

Data management

At the end of the conversion of the ordinary group, the converted value is stored in the ADC_ODT register. Once the preempted group conversion ends, the converted data of the preempted group is stored in the ADC_PDTx register.

Data alignment

DTALIGN bit in the ADC_CTRL2 register selects the alignment of data (right-aligned or left-aligned). Apart from this, the converted data of the preempted group is decreased by the offset written in the ADC_PCDTOx register. Thus the result may be a negative value, marked by SIGN, as shown in the following diagram.

Ordinary channel data 12 bits

Right-alignment

0	0	0	0	DT[11]	DT[10]	DT[9]	DT[8]	DT[7]	DT[6]	DT[5]	DT[4]	DT[3]	DT[2]	DT[1]	DT[0]

Left-alignment

DT[11]	DT[10]	DT[9]	DT[8]	DT[7]	DT[6]	DT[5]	DT[4]	DT[3]	DT[2]	DT[1]	DT[0]	0	0	0	0

Preempted channel data 12 bits

Right-alignment

SIGN	SIGN	SIGN	SIGN	DT[11]	DT[10]	DT[9]	DT[8]	DT[7]	DT[6]	DT[5]	DT[4]	DT[3]	DT[2]	DT[1]	DT[0]

Left-alignment

SIGN	DT[11]	DT[10]	DT[9]	DT[8]	DT[7]	DT[6]	DT[5]	DT[4]	DT[3]	DT[2]	DT[1]	DT[0]	0	0	0

Figure 143. Data alignment

Data read

Read access to the ADC_ODT register using CPU or DMA gets the converted data of the ordinary group. Read access to the ADC_PDTx register using CPU gets the converted data of the preempted group.

When the OCDMAEN is set in the ADC_CTRL2 register, the ADC will issue DMA requests each time when the ADC_OTD register is updated.

ADC1 and ADC3 both have their respective DMA channels. In Master/Slave mode, the ADC2 used as slave is read by DMA through the master ADC1.

Voltage monitor

OCVMEN bit or PCVMEN bit in the ADC_CTRL1 register is used to enable voltage monitor. The VMOR bit will be set if the converted result is outside the high threshold (ADC_VMHB register) or is less than the low threshold (ADC_VMLB register).

VMSGEN bit in the ADC_CTRL1 register is used to enable voltage monitor on either a single specific channel or all the channels. The VMCSEL bit is used to select the specific channel that requires voltage monitoring.

Voltage monitor is based on the comparison result between the original converted data and the 12-bit voltage monitor boundary register, irrespective of the PCDTOx and DTALIGN bits.

Status flag and interrupts

Each of the ADCs has its dedicated ADCx_STS reisters, that is, OCCS (ordinary channel conversion start flag), PCCS (preempted channel conversion start flag), PCCE (preempted channel conversion end flag), CCE (channel conversion end flag) and VMOR (voltage monitor out of range).

PCCE, CCE and VMOR have their respective interrupt enable bits. Once the interrupt bits are enabled, the corresponding flag is set and an interrupt is sent to CPU. ADC1 shares an interrupt vector with ADC2. ADC3 has a separate interrupt vector.

Master/Slave mode

If Master/Slave mode is enabled, the master is triggered to work with the slave to do the channel conversion. The ADC_ODT register is used as a single interface obtaining the ordinary channel converted data of master/salve ADC.

In this mode, ADC1 acts as a master while ADC2 as a slave.

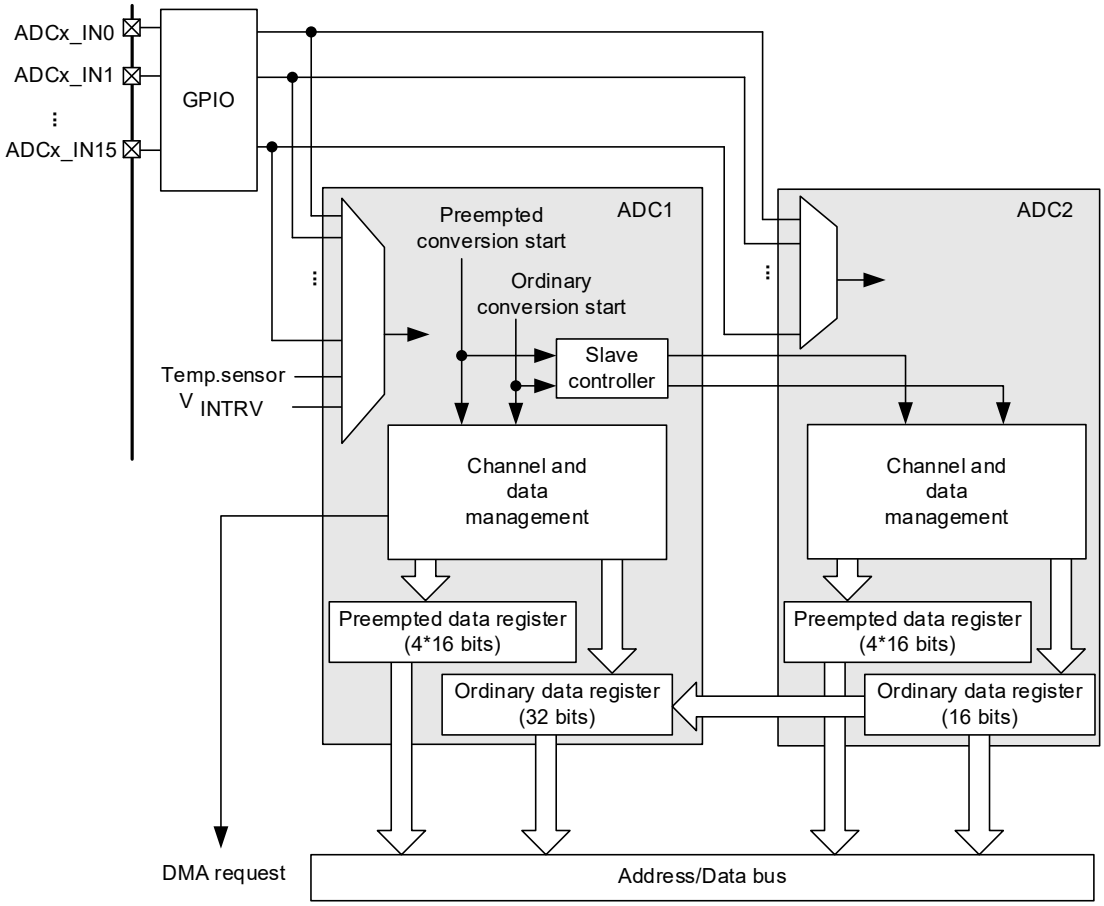


Figure 144. Block diagram of master/salve mode

Data management

In Master/Slave mode, the data of ordinary channels is also stored in the ADC_ODT register of ADC1. As long as the OCDMAEN is set in the ADC1_CTRL2 register, the ADC1 DMA channel is used to generate a DMA request each time when the data is ready.

Regular simultaneous mode

Regular simultaneous mode

MSSEL bit in the ADC_CTRL1 register is used to select regular simultaneous mode. If this mode is enabled, the regular channels of the master is triggered so that the master works with the slave to convert the regular channels simultaneously. In this mode, it is required to configure the same sampling time and the same sequence length for the master and slave to avoid the loss of data due to the lack of synchronization.

The following diagram shows an example of the regular simultaneous mode.

Note: The same channel is not allowed to be sampled by several ADCs simultaneously. Do not put the same channel in the same sequence location of different ADCs.

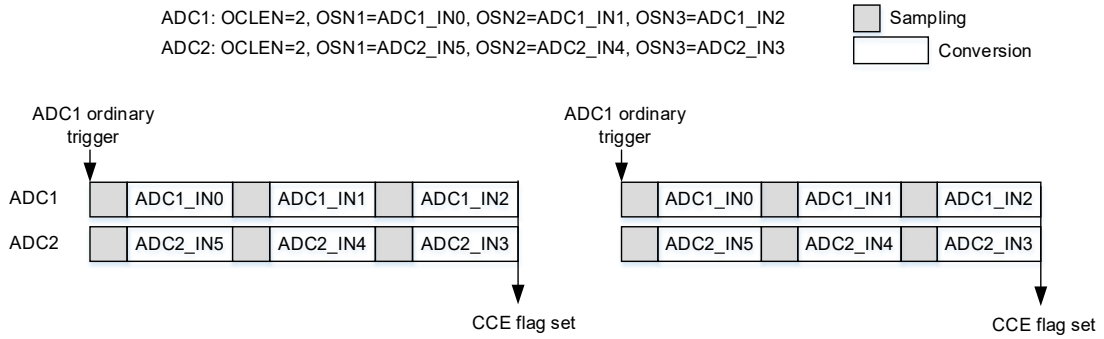


Figure 145. Regular simultaneous mode

Preempted simultaneous mode

MSSEL bit in the ADC_CTRL1 register is used to select preempted simultaneous mode. If this mode is enabled, the preempted channels of the master is triggered so that the master works with the slave to convert the preempted channels simultaneously. The following diagram shows an example of the preempted simultaneous mode

Note: The same channel is not allowed to be sampled by several ADCs simultaneously. Do not put the same channel in the same sequence location of different ADCs.

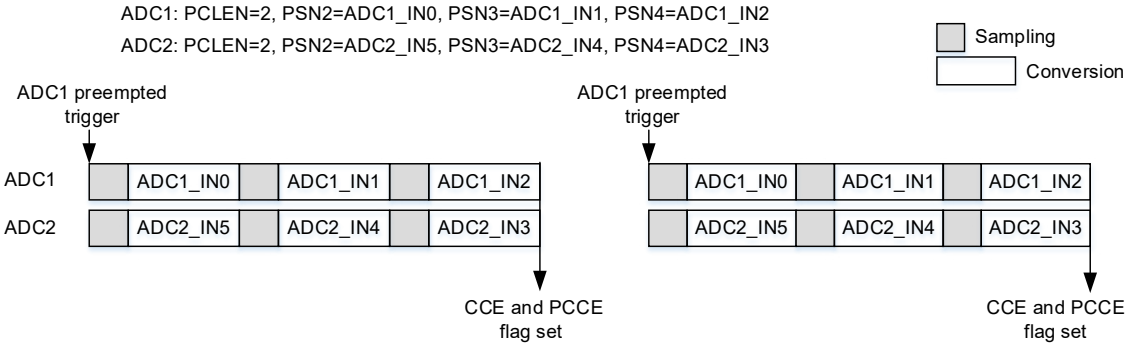


Figure 146. Regular simultaneous mode

Combined regular/preempted simultaneous mode

MSSEL bit in the ADC_CTRL1 register is used to select combined regular/preempted simultaneous mode. If this mode is enabled, the regular channels of the master is triggered so that the master works with the slave to convert the regular channels simultaneously, or the preempted channels of the master is triggered to enable the master and slave to convert the preempted channels simultaneously.

Alternate preempted trigger mode

Alternate preempted trigger mode

MSSEL bit in the ADC_CTRL1 register selects the alternate preempted trigger mode. If this mode is enabled, the preempted channels of the master are triggered continuously so that the master/slave ADCs convert the preempted channels alternately. The following diagram shows an example of the alternate preempted trigger mode.

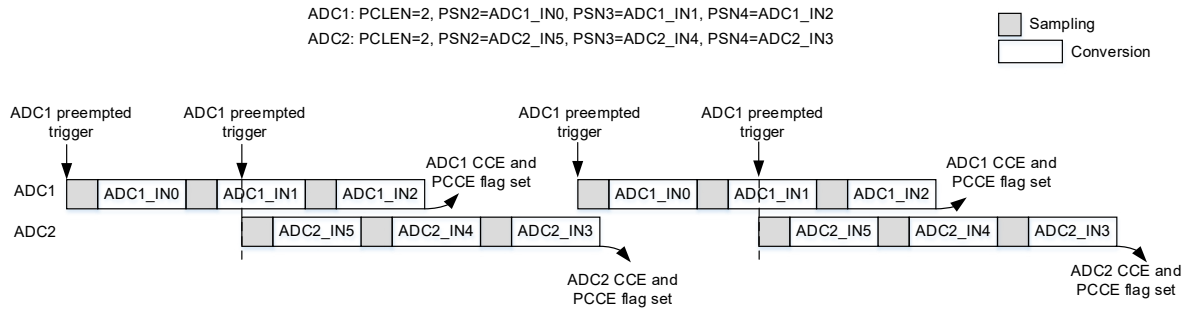


Figure 147. Alternate preempted trigger mode

Combined regular simultaneous + alternate preempted trigger mode

MSSEL bit in the ADC_CTRL1 register is used to select combined regular simultaneous + alternate preempted trigger mode. In this mode, trigger the regular group of the master to start regular simultaneous conversion of master/slave, or trigger the preempted group of the master continuously to allow the master/slave ADCs to convert the preempted group alternately.

If the regular conversion is interrupted by the preempted trigger, the regular conversion of all ADCs is stopped, and one of the ADCs starts the preempted conversion. At this point, the master will ignore the preempted trigger until the regular conversion is resumed.

Regular switch mode

Fast switch mode on regular group

MSSEL bit in the ADC_CTRL1 register is used to select fast switch mode on regular group. After a master trigger occurs, the conversion interval between ADCs is 7 ADCCLK cycles, that is, ADC2 starts immediately, ADC1 starts after 7 ADCCLK cycles, and then ADC2 starts after another 7 ADCCLK cycles, and so non. In this mode, the sampling time allowed is 1.5 ADCCLK cycles, as shown in the following diagram.

Note: The preempted trigger is not allowed in this mode.

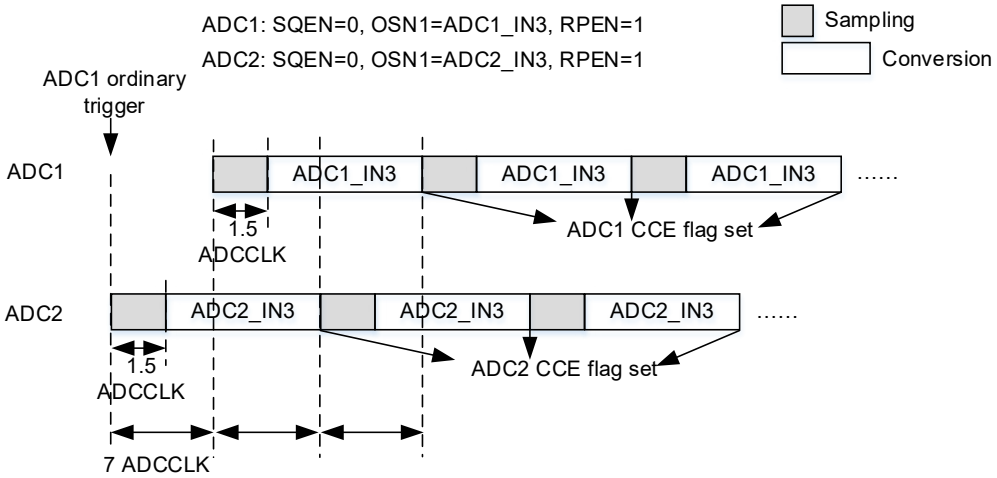


Figure 148. Fast switch mode

Combined preempted simultaneous + fast switch mode on regular group

MSSEL bit in the ADC_CTRL1 register is used to select combined preempted simultaneous + fast switch mode. After a regular group of the master is triggered, the conversion interval between ADCs is 7 ADCCLK cycles. Or trigger the preempted group of th master to allow a simultaneous conversion of the preempted group by master/slave.

If the regular group conversion is interrupted by the preempted group, the regular group conversion is stopped and resumes from ADC2 at the end of the preempted conversion.

Slow switch mode on regular group

MSSEL bit in the ADC_CTRL1 register is used to select slow switch mode. After a master trigger occurs, the conversion interval between ADCs is 14 ADCCLK cycles, that is, ADC2 starts immdediately, ADC2 starts after a delay of 14 ADCCLK cycles, and then the ADC2 starts after another delay of 14 ADCCLK cycles, and so on. In this mode, the sampling time allowed is less than 14 ADCCLK cycles, as shown in the following diagram.

Note: The preempted trigger is not allowed in this mode.

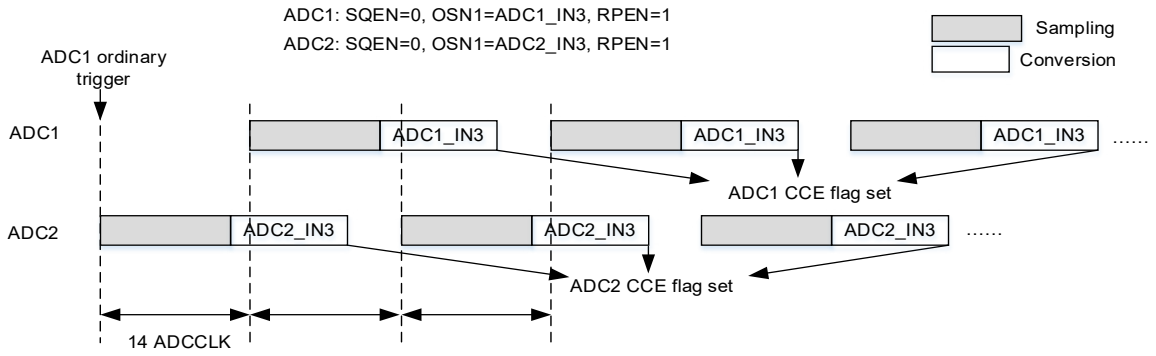


Figure 149. Fast slow mode

Combined preempted simultaneous + slow switch mode

MSSEL bit in the ADC_CTRL1 register is used to select combined preempted simultaneous + slow switch mode. After a master trigger occurs, the conversion interval between ADCs is 14 ADCCLK cycles. Or r trigger the preempted group of th master to allow a simultaneous conversion of the preempted group by master/slave.

If the regular group conversion is interrupted by the preempted group, the regular group conversion is stopped and resumes from ADC2 at the end of the preempted conversion.

ADC registers

The following table lists ADC register map and their reset values. These peripheral registers must be accessed by word (32 bits).

Table 68. ADC register map and reset values

Register	Offset	Reset value
ADC_STS	0x000	0x0000 0000
ADC_CTRL1	0x004	0x0000 0000
ADC_CTRL2	0x008	0x0000 0000
ADC_SPT1	0x00C	0x0000 0000
ADC_SPT2	0x010	0x0000 0000
ADC_PCDTO1	0x014	0x0000 0000
ADC_PCDTO2	0x018	0x0000 0000
ADC_PCDTO3	0x01C	0x0000 0000
ADC_PCDTO4	0x020	0x0000 0000
ADC_VMHB	0x024	0x0000 0FFF
ADC_VMLB	0x028	0x0000 0000
ADC_OSQ1	0x02C	0x0000 0000
ADC_OSQ2	0x030	0x0000 0000
ADC_OSQ3	0x034	0x0000 0000
ADC_PSQ	0x038	0x0000 0000
ADC_PDT1	0x03C	0x0000 0000
ADC_PDT2	0x040	0x0000 0000
ADC_PDT3	0x044	0x0000 0000
ADC_PDT4	0x048	0x0000 0000
ADC_ODT	0x04C	0x0000 0000

ADC status register (ADC_STS)

Bit	Register	Reset value	Type	Description
Bit 31:5	Reserved	0x0000000	resd	Kept at its default value.
Bit 4	OCCS	0x0	rw0c	Ordinary channel conversion start flag This bit is set by hardware and cleared by software (writing 0). 0: No ordinary channel conversion started 1: Ordinary channel conversion has started
Bit 3	PCCS	0x0	rw0c	Preempted channel conversion start flag This bit is set by hardware and cleared by software (writing 0). 0: No preempted channel conversion started 1: Preempted channel conversion has started
Bit 2	PCCE	0x0	rw0c	Preempted channel end of conversion flag This bit is set by hardware and cleared by software (writing 0). 0: Conversion is not complete 1: Conversion is complete
Bit 1	CCE	0x0	rw0c	End of conversion flag This bit is set by hardware. It is cleared by software (writing 0) or by reading the ADC_ODT register. 0: Conversion is not complete 1: Conversion is complete Note: This bit is set at the end of the ordinary or preempted group.

Bit	Register	Reset value	Type	Description
Bit 0	VMOR	0x0	rw0c	Voltage monitoring out of range flag This bit is set by hardware and cleared by software (writing 0). 0: Voltage is within the value programmed 1: Voltage is outside the value programmed

ADC control register1 (ADC_CTRL1)

Bit	Register	Reset value	Type	Description
Bit 31:24	Reserved	0x00	resd	Kept at its default value.
Bit 23	OCVMEN	0x0	rw	Voltage monitoring enable on ordinary channels 0: Voltage monitoring disabled on ordinary channels 1: Voltage monitoring enabled on ordinary channels
Bit 22	PCVMEN	0x0	rw	Voltage monitoring enable on preempted channels 0: Voltage monitoring disabled on preempted channels 1: Voltage monitoring enabled on preempted channels
Bit 21:20	Reserved	0x0	resd	Kept at its default value.
Bit 19:16	MSSEL	0x0	rw	Master/slave mode select 0000: Independnet mode 0001: Combined regular simultaneous + preempted simultaneous mode 0010: Combined regular simultaneous + alternate preempted trigger mode 0011: Combined preempted simultaneous + fast switch mode on regular group 0100: Combined preempted simultaneous + slow switch mode on regular group 0101: Preempted simultaneous mode 0110: Regular simultaneous mode 0111: Fast switch mode on regular group 1000: Slow switch mode on regular group 1001: Alternate preempted trigger mode 1010~1111: Unused, configuration is not allowed. Note: These bits are reserved in ADC2 and ADC3. In master/slave mode, a change of configurmation will result in a loss of synchronization. It is recommended to disable master/slave mode before any change.
Bit 15:13	OCPCNT	0x0	rw	Partitioned mode conversion count of ordinary channels 000: 1 channel 001: 2 channels …… 111: 8 channels Note: In this mode, the preempted group converts only one channel at each trigger.
Bit 12	PCPEN	0x0	rw	Partitioned mode enable on preempted channels 0: Partitioned mode disabled on preempted channels 1: Partitioned mode enabled on preempted channels
Bit 11	OCPEN	0x0	rw	Partitioned mode enable on ordinary channels This is set and cleared by software to enable or disable partitioned mode on ordinary channels. 0: Partitioned mode disabled on ordinary channels 1: Partitioned mode enabled on ordinary channels

Bit	Register	Reset value	Type	Description
Bit 10	PCAUTOEN	0x0	rw	Preempted group automatic conversion enable after ordinary group 0: Preempted group automatic conversion disabled 1: Preempted group automatic conversion enabled
Bit 9	VMSGEN	0x0	rw	Voltage monitoring enable on a single channel 0: Disabled (Voltage monitoring enabled on all channels) 1: Enabled (Voltage monitoring enabled a single channel)
Bit 8	SQEN	0x0	rw	Sequence mode enable 0: Sequence mode disabled (a single channel is converted) 1: Sequence mode enabled (the selected multiple channels are converted) Note: If this mode is enabled and the CCEIEN/PCCEIEN is set, a CCE or PCCE interrupt is generated only at the end of conversion of the last channel.
Bit 7	PCCEIEN	0x0	rw	Conversion end interrupt enable on Preempted channels 0: Conversion end interrupt disabled on Preempted channels 1: Conversion end interrupt enabled on Preempted channels
Bit 6	VMORIEN	0x0	rw	Voltage monitoring out of range interrupt enable 0: Voltage monitoring out of range interrupt disabled 1: Voltage monitoring out of range interrupt enabled
Bit 5	CCEIEN	0x0	rw	Channel conversion end interrupt enable 0: Channel conversion end interrupt disabled 1: Channel conversion end interrupt enabled
Bit 4:0	VMCSEL	0x00	rw	Voltage monitoring channel select This filed is valid only when the VMSGEN is enabled. 00000: ADC_IN0 channel 00001: ADC_IN1 channel …… 01111: ADC_IN15 channel 10000: ADC_IN16 channel 10001: ADC_IN17 channel 10010~11111: Unused, configuration is not allowed.
Bit 0	VMOR	0x0	rw0c	Voltage monitoring out of range flag This bit is set by hardware and cleared by software (writing 0). 0: Voltage is within the value programmed 1: Voltage is outside the value programmed

ADC control register2 (ADC_CTRL2)

Bit	Register	Reset value	Type	Description
Bit 30: 26	Reserved	0x00	resd	Kept at its default value
Bit 23	ITSRVEN	0x0	rw	Internal temperature sensor and VINTRV enable 0: Internal temperature sensor and VINTRV disabled 1: Internal temperature sensor and VINTRV enabled Note: These bits are reserved in ADC2 and ADC3, and must be kept at its default value.
Bit 22	OCSWTRG	0x0	rw	Conversion of ordinary channels triggered by software 0: Conversion of ordinary channels not triggered 1: Conversion of ordinary channels triggered (This bit is cleared by software or by hardware as soon as the conversion starts)
Bit 21	PCSWTRG	0x0	rw	Conversion of preempted channels triggered by software 0: Conversion of preempted channels not triggered 1: Conversion of preempted channels triggered (This bit is cleared by software or by hardware as soon as the conversion starts)
Bit 20	OCTEN	0x0	rw	Trigger mode enable for ordinary channels conversion 0: Trigger mode disabled for ordinary channels conversion 1: Trigger mode enabled for ordinary channels conversion
Bit 25 Bit 19:17	OCTESEL	0x0	rw	Trigger event select for ordinary channels conversion For ADC1 and ADC2, the trigger events are configured as follows: 0000: Timer 1 CH1 event 0001: Timer 1 CH2 event 0010: Timer 1 CH3 event 0011: Timer 2 CH2 event 0100: Timer 3 TRGOUT event 0101: Timer 4 CH4 event 0110: EXINT line 11/ TMR8_TRGOUT event 0111: OCSWTRG 1000~1100: Unused. Configuration is not allowed. 1101: Timer 1 TRGOUT event 1110: Timer 8 CH1 event 1111: Timer 8 CH2 event For ADC3, the trigger events are configured as follows: 0000: Timer 3 CH1 event 0001: Timer 2 CH3 event 0010: Timer 1 CH3 event 0011: Timer 8 CH1 event 0100: Timer 8 TRGOUT event 0101: Timer 5 CH1 event 0110: Timer 5 CH3 event 0111: OCSWTRG 1000~1100: Unused. Configuration is not allowed. 1101: Timer 1 TRGOUT event 1110: Timer 1 CH1 event 1111: Timer 8 CH3 event
Bit 16	Reserved	0x0	resd	Kept at its default value
Bit 15	PCTEN	0x0	rw	Trigger mode enable for preempted channels conversion 0: Disabled 1: Enabled

Bit	Register	Reset value	Type	Description
Bit 24 Bit 14:12	PCTESEL	0x0	rw	Trigger event select for preempted channels conversion For ADC1 and ADC2, the trigger events are configured as follows: 0000: Timer 1 TRGOUT event 0001: Timer 1 CH4 event 0010: Timer 2 TRGOUT event 0011: Timer 2 CH1 event 0100: Timer 3 CH4 event 0101: Timer 4 TRGOUT event 0110: EXINT line 15/TMR8_CH4 event 0111: PCSWTRG 1000~1100: Unused. Configuration is not allowed. 1101: Timer 1 CH1 event 1110: Timer 8 CH1 event 1111: Timer 8 TRGOUT event For ADC3, the trigger events are configured as follows: 0000: Timer 1 TRGOUT event 0001: Timer 1 CH4 event 0010: Timer 4 CH3 event 0011: Timer 8 CH2 event 0100: Timer 8 CH4 event 0101: Timer 5 TRGOUT event 0110: Timer 5 CH4 event 0111: PCSWTRG 1000~1100: Unused. Configuration is not allowed. 1101: Timer 1 CH1 event 1110: Timer 1 CH2 event 1111: Timer 8 TRGOUT event
Bit 11	DTALIGN	0x0	rw	Data alignment 0: Right alignment 1: Left alignment
Bit 10:9	Reserved	0x0	resd	Kept at its default value.
Bit 8	OCDMAEN	0x0	rw	DMA transfer enable of ordinary channels 0: DMA transfer disabled 1: DMA transfer enabled Note: ADC2 has no DMA function, and thus it cannot generate a DMA request itself
Bit 7:4	Reserved	0x0	resd	Kept at its default value.
Bit 3	ADCALINIT	0x0	rw	Initialize A/D calibration This bit is set by software and cleared by hardware. It is cleared after the calibration registers are initialized. 0: No initialization occurred or initialization completed 1: Enable initialization or initializationis is ongoing
Bit 2	ADCAL	0x0	rw	A/D Calibration 0: No calibration occurred or calibration completed 1: Enable calibration or calibration is in process

Bit	Register	Reset value	Type	Description
Bit 1	RPEN	0x0	rw	Repition mode enable 0: Repition mode disabled When SQEN=0, a single conversion is done each time when a trigger event arrives; when SQEN=1, a group of conversion is done each timer when a trigger event arrives. 1: Repition mode enabled When SQEN =0, continuous conversion mode on a single channel is enabled at each trigger event; when SQEN =1, continuous conversion mode on a group of channels is enabled at each trigger event.
Bit 0	ADCEN	0x0	rw	A/D converter enable 0: A/D converter disabled (ADC goes to power-down mode) 1: A/D converter enabled Note: When this bit is in OFF state, write an ON command can wake up The ADC from power-down mode. When this bit in ON state, write another ON command can start a regular group conversion. The application should pay attention to the fact that there is a delay of tSTAB between power on and start of conversion.
Bit 0	VMOR	0x0	rw0c	Voltage monitoring out of range flag This bit is set by hardware and cleared by software (writing 0). 0: Voltage is within the value programmed 1: Voltage is outside the value programmed

ADC sampling time register 1 (ADC_SPT1)

Bit	Register	Reset value	Type	Description
Bit 31; 24	Reserved	0x00	resd	Kept at its default value.
Bit 23:21	CSPT17	0x0	rw	Sample time slection of channel ADC_IN17 000: 1.5 c 001: 7.5 cycles 010: 13.5 cycles 011: 28.5 cycles 100: 41.5 cycles 101: 55.5 cycles 110: 71.5 cycles 111: 239.5 cycles
Bit 20:18	CSPT16	0x0	rw	Sample time selection of channel ADC_IN16 000: 1.5 cycles 001: 7.5 cycles 010: 13.5 cycles 011: 28.5 cycles 100: 41.5 cycles 101: 55.5 cycles 110: 71.5 cycles 111: 239.5 cycles

Bit	Register	Reset value	Type	Description
Bit 17:15	CSPT15	0x0	rw	Sample time selection of channel ADC_IN15 000: 1.5 cycles 001: 7.5 cycles 010: 13.5 cycles 011: 28.5 cycles 100: 41.5 cycles 101: 55.5 cycles 110: 71.5 cycles 111: 239.5 cycles
Bit 14:12	CSPT14	0x0	rw	Sample time selection of channel ADC_IN14 000: 1.5 cycles 001: 7.5 cycles 010: 13.5 cycles 011: 28.5 cycles 100: 41.5 cycles 101: 55.5 cycles 110: 71.5 cycles 111: 239.5 cycles
Bit 11:9	CSPT13	0x0	rw	Sample time selection of channel ADC_IN13 000: 1.5 cycles 001: 7.5 cycles 010: 13.5 cycles 011: 28.5 cycles 100: 41.5 cycles 101: 55.5 cycles 110: 71.5 cycles 111: 239.5 cycles
Bit 8:6	CSPT12	0x0	rw	Sample time selection of channel ADC_IN12 000: 1.5 cycles 001: 7.5 cycles 010: 13.5 cycles 011: 28.5 cycles 100: 41.5 cycles 101: 55.5 cycles 110: 71.5 cycles 111: 239.5 cycles
Bit 5:3	CSPT11	0x0	rw	Sample time selection of channel ADC_IN11 000: 1.5 cycles 001: 7.5 cycles 010: 13.5 cycles 011: 28.5 cycles 100: 41.5 cycles 101: 55.5 cycles 110: 71.5 cycles 111: 239.5 cycles
Bit 2:0	CSPT10	0x0	rw	Sample time selection of channel ADC_IN10 000: 1.5 cycles 001: 7.5 cycles 010: 13.5 cycles 011: 28.5 cycles 100: 41.5 cycles 101: 55.5 cycles 110: 71.5 cycles 111: 239.5 cycles

ADC sampling time register 2 (ADC_SPT2)

Bit	Register	Reset value	Type	Description
Bit 31:30	Reserved	0x0	resd	Kept at its default value
Bit 29:27	CSPT9	0x0	rw	Sample time selection of channel ADC_IN9 000: 1.5 cycles 001: 7.5 cycles 010: 13.5 cycles 011: 28.5 cycles 100: 41.5 cycles 101: 55.5 cycles 110: 71.5 cycles 111: 239.5 cycles
Bit 26:24	CSPT8	0x0	rw	Sample time selection of channel ADC_IN8 000: 1.5 cycles 001: 7.5 cycles 010: 13.5 cycles 011: 28.5 cycles 100: 41.5 cycles 101: 55.5 cycles 110: 71.5 cycles 111: 239.5 cycles
Bit 23:21	CSPT7	0x0	rw	Sample time selection of channel ADC_IN7 000: 1.5 cycles 001: 7.5 cycles 010: 13.5 cycles 011: 28.5 cycles 100: 41.5 cycles 101: 55.5 cycles 110: 71.5 cycles 111: 239.5 cycles
Bit 20:18	CSPT6	0x0	rw	Sample time selection of channel ADC_IN6 000: 1.5 cycles 001: 7.5 cycles 010: 13.5 cycles 011: 28.5 cycles 100: 41.5 cycles 101: 55.5 cycles 110: 71.5 cycles 111: 239.5 cycles
Bit 17:15	CSPT5	0x0	rw	Sample time selection of channel ADC_IN5 000: 1.5 cycles 001: 7.5 cycles 010: 13.5 cycles 011: 28.5 cycles 100: 41.5 cycles 101: 55.5 cycles 110: 71.5 cycles 111: 239.5 cycles

Bit	Register	Reset value	Type	Description
Bit 14:12	CSPT4	0x0	rw	Sample time selection of channel ADC_IN4 000: 1.5 cycles 001: 7.5 cycles 010: 13.5 cycles 011: 28.5 cycles 100: 41.5 cycles 101: 55.5 cycles 110: 71.5 cycles 111: 239.5 cycles
Bit 11:9	CSPT3	0x0	rw	Sample time selection of channel ADC_IN3 000: 1.5 cycles 001: 7.5 cycles 010: 13.5 cycles 011: 28.5 cycles 100: 41.5 cycles 101: 55.5 cycles 110: 71.5 cycles 111: 239.5 cycles
Bit 8:6	CSPT2	0x0	rw	Sample time selection of channel ADC_IN2 000: 1.5 cycles 001: 7.5 cycles 010: 13.5 cycles 011: 28.5 cycles 100: 41.5 cycles 101: 55.5 cycles 110: 71.5 cycles 111: 239.5 cycles
Bit 5:3	CSPT1	0x0	rw	Sample time selection of channel ADC_IN1 000: 1.5 cycles 001: 7.5 cycles 010: 13.5 cycles 011: 28.5 cycles 100: 41.5 cycles 101: 55.5 cycles 110: 71.5 cycles 111: 239.5 cycles
Bit 2:0	CSPT0	0x0	rw	Sample time selection of channel ADC_IN0 000: 1.5 cycles 001: 7.5 cycles 010: 13.5 cycles 011: 28.5 cycles 100: 41.5 cycles 101: 55.5 cycles 110: 71.5 cycles 111: 239.5 cycles

ADC preempted channel data offset register x (ADC_PCDTOx) (x=1..4)

Bit	Register	Reset value	Type	Description
Bit 31:12	Reserved	0x00000	resd	Kept at its default value
Bit 11:0	PCDTOx	0x000	rw	Data offset for Preempted channel x Converted data stored in the ADC_PDTx = Raw converted data – ADC_PCDTOx

ADC voltage monitor high threshold register (ADC_VWHB)

Bit	Register	Reset value	Type	Description
Bit 31:12	Reserved	0x00000	resd	Kept at its default value
Bit 11:0	VMHB	0xFFF	rw	Voltage monitoring high boundary

ADC voltage monitor low threshold register (ADC_VWLB)

Bit	Register	Reset value	Type	Description
Bit 31:12	Reserved	0x00000	resd	Kept at its default value
Bit 11:0	VMLB	0xFFF	rw	Voltage monitoring low boundary

ADC ordinary sequence register 1 (ADC_OSQ1)

Bit	Register	Reset value	Type	Description
Bit 31:24	Reserved	0x00	resd	Kept at its default value
Bit 23:20	OCLEN	0x0	rw	Ordinary conversion sequence length 0000: 1 conversion 0001: 2 conversions …… 1111: 16 conversions
Bit 19:15	OSN16	0x00	rw	Number of 16th conversion in ordinary sequence
Bit 14:10	OSN15	0x00	rw	Number of 15th conversion in ordinary sequence
Bit 9:5	OSN14	0x00	rw	Number of 14th conversion in ordinary sequence
Bit 4:0	OSN13	0x00	rw	Number of 13th conversion in ordinary sequence Note: The number can be from 0 to 17. For example, if the number is set to 3, it means that the 13th conversion is ADC_IN3 channel.

ADC ordinary sequence register 2 (ADC_OSQ2)

Bit	Register	Reset value	Type	Description
Bit 31:30	Reserved	0x0	resd	Kept at its default value
Bit 29:25	OSN12	0x00	rw	Number of 12th conversion in ordinary sequence
Bit 24:20	OSN11	0x00	rw	Number of 11th conversion in ordinary sequence
Bit 19:15	OSN10	0x00	rw	Number of 10th conversion in ordinary sequence
Bit 14:10	OSN9	0x00	rw	Number of 9th conversion in ordinary sequence
Bit 9:5	OSN8	0x00	rw	Number of 8th conversion in ordinary sequence
Bit 4:0	OSN7	0x00	rw	Number of 7th conversion in ordinary sequence Note: The number can be from 0 to 17. For example, if the number is set to 8, it means that the 7th conversion is ADC_IN8 channel.

ADC ordinary sequence register 3 (ADC_OSQ3)

Bit	Register	Reset value	Type	Description
Bit 31:30	Reserved	0x0	resd	Kept at its default value
Bit 29:25	OSN6	0x00	rw	Number of 6th conversion in ordinary sequence
Bit 24:20	OSN5	0x00	rw	Number of 5th conversion in ordinary sequence
Bit 19:15	OSN4	0x00	rw	Number of 4th conversion in ordinary sequence
Bit 14:10	OSN3	0x00	rw	Number of 3rd conversion in ordinary sequence
Bit 9:5	OSN2	0x00	rw	Number of 2nd conversion in ordinary sequence
Bit 4:0	OSN1	0x00	rw	Number of 1st conversion in ordinary sequence Note: The number can be from 0 to 17. For example, if the number is set to 8, it means that the 1st conversion is ADC_IN17 channel.

ADC preempted sequence register (ADC_PSQ)

Bit	Register	Reset value	Type	Description
Bit 31:30	Reserved	0x0	resd	Kept at its default value
Bit 21:20	PCLEN	0x0	rw	Preempted conversion sequence length 00: 1 conversion 01: 2 conversions 10: 3 conversions 11: 4 conversions
Bit 19:15	PSN4	0x00	rw	Number of 4th conversion in preempted sequence
Bit 14:10	PSN3	0x00	rw	Number of 3rd conversion in preempted sequence
Bit 9:5	PSN2	0x00	rw	Number of 2nd conversion in preempted sequence
Bit 4:0	PSN1	0x00	rw	Number of 1st conversion in preempted sequence Note: The number can be from 0 to 17. For example, if the number is set to 3, it refers to the ADC_IN3 channel. If PCLEN is less than 4, the conversion sequence starts from 4-PCLEN. For example, when ADC_PSQ ([21:0]) =10 00110 00101 00100 00011, it indicates that the scan conversion follows the sequence : 4, 5, 6, not 3, 4,5.

ADC preempted data register x (ADC_PDTx) (x = 1..4)

Bit	Register	Reset value	Type	Description
Bit 31:16	Reserved	0x0000	resd	Kept at its default value
Bit 15:0	PDTx	0x0000	rw	Conversion data from preempted channel

ADC ordinary data register (ADC_ODT)

Bit	Register	Reset value	Type	Description
Bit 31:16	ADC2ODT	0x0000	ro	ADC2 conversion data of ordinary channel Note: These bits are reserved in ADC2 and ADC3. In ADC1, these bits are valid only in master/slave mode, and they contain the conversion result from the ADC2 ordinary cahnnels.
Bit 15:0	ODT	0x0000	ro	Conversion data of ordinary channel

20 Digital-to-analog converter (DAC)

DAC introduction

The DAC uses a 12-bit digital input to generate an analog output between 0 and reference voltage. The digital part of the DAC can be configured in 8-bit or 12-bit mode and can be used in conjunction with the DMA. It supports left or right alignment in a single /dual DAC modes. It has two output channels, DAC1 and DCA2, with its own converter each. Each DAC1/DAC2 can be converted independently or simulatenously in dual DAC mode. The input reference voltage V_{REF+} makes conversion more accuracy.

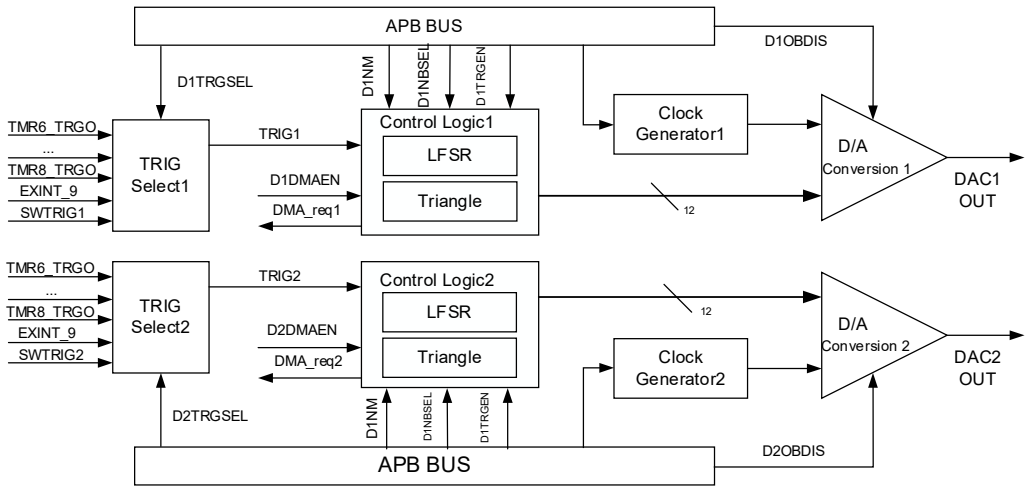


Figure 150. DAC1/DAC2 block diagram

Main features

- A single/dual DAC 8-bit or 12-bit digital input
- Left or right data alignment
- Noise-wave/Triangular-wave generation
- Dual DAC or single DAC1/DAC2 independent conversions
- DMA mode for DAC1/DAC2
- Software or external triggers for conversion
- Input reference voltage V_{REF+}

Design tips

The following information can be used as DAC design reference:

- Analog module configuration
  The analog part of the DAC1/DAC2 can be enabled by setting the ENx bit in the DAC_CTRL register, but its digital part is not subject to this bit. The DAC integrates two output gains that

can be used to reduce the output impedance, and to drive external loads directly without the need of an external operational amplifer. The DAC1/DAC2 output gain can be enabled and disabled through the DxOBDIS bit in the DAC_CTRL register.

- DMA capability
  The DAC1/DAC2 both have a DMA capability that can be enabled by setting the DxDMAEN bit in the DAC_CTRL register. A DMA request is generated when a trigger signal is active while the DxTRGEN bit is set. The DAC DMA request is not added up, meaning the new DAM request will be ignored and no error is reported.
- In dual DAC mode, the application can handle two channels (DAC1/DAC2) by using only one DMA request and a DMA channel.
- Input/output configuration
  The digital inputs are linearly converted to analog voltage outputs by the DAC, and it is between 0 and V_{REF+}. The analog DAC module is supplied by V_{DDA}. The positive analog reference voltage input falls between 2.0 V and V_{DDA}. To avoid parasitic interruption and excessive consumption, the PA4 or PA5 should be configured to analog input.
  DAC output = V_{REF+} × (DxODT[11:0] / 4095)

Functional descriptions

Trigger events

If the DxTRGEN bit in the DAC_CTRL register is set, the DAC conversion can then be triggered by an external event (timer counter, external interrupt line) or by software. The DxTRGSEL[2:0] is used to select trigger sources.

Table 69. Trigger source selection

Source	DxTRGSEL[2:0]	Description
TMR6_TRGOUT	000	On-chip signals
TMR8_TRGOUT	001	
TMR7_TRGOUT	010	
TMR5_TRGOUT	011	
TMR2_TRGOUT	100	
TMR4_TRGOUT	101	
EXINT_9	110	External signals
DxSWTRG	111	Software trigger

When the DxTRGEN bit is set, each time a DAC detects an active trigger event, the data stored into the HDRx register is transferred into the DAC_DxODT register. If the software trigger is selected, the DxSWTRG flag is cleared by hardware after being set. The DAC output becomes active after a period of time once the data is loaded into the DAC_DxODT register.

When the DxTRGEN bit is cleared, each data written to the data register is immediately transferred into the DAC_DxODT register without the need of a trigger event.

Noise/Triangular-wave generation

The DAC can output a variable-amplitude pseudonise and a triangular wave, which is done by the LENinear Feedback Shift Register and triangle wave generator respectively. The DAC noise generation is selected by setting DxNM[1:0]=01 in the LFSR, while the DAC triangular-wave generation is selected by setting the DxNM[1:0]=1x.

LFSR logic

The preloaded value in the LFSR is 0xAAA. This register is updated after each trigger event based on a specific calculation algorithm.

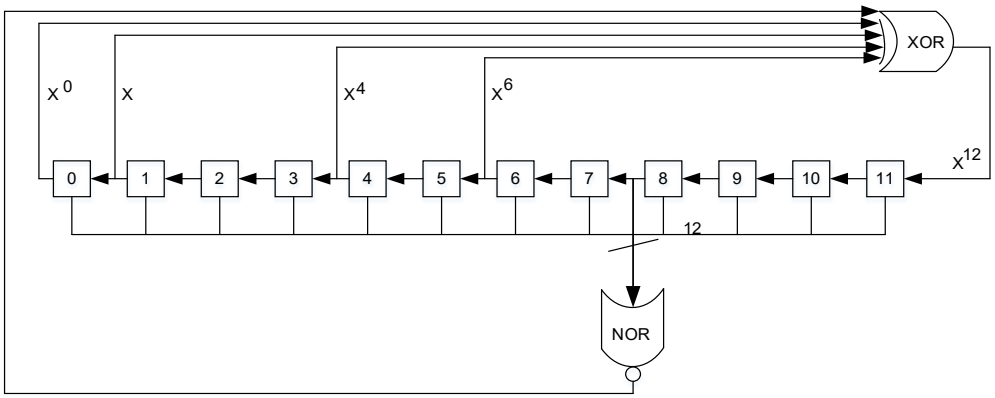


Figure 151. LFSR register calculation algorithm

The DxNBSEL[3:0] bit in the DAC_CTRL register is set to mark partially or totally the LFSR data. The resulting value is then added up to the DHRx value without overflow and this value is loaded into the DAC_DxODT register. It is possible to disable LFSR function and reset LFSR wave generation algorithm by setting DxNM[1:0]=00.

Triangular wave logic

The DAC triangular-wave generation is selected by setting DxNM[1:0]=1x. The amplitude is configured through the DxNBSEL[3:0] bit in the DAC_CTRL register. An internal triangular-wave counter is incremented at each trigger event. Once the maximum amplitude programmed in the DxNBSEL[3:0] is reached, the value of this counter is decremented down to 0, then incremented again, and so on.

Meanwhile, the value of this counter is then added up to the DHRx register without overflow and the resulting value is loaded into the DAC_DxODT register. It is possible to disable/reset the triangular-wave generation by setting DxNM[1:0]=00.

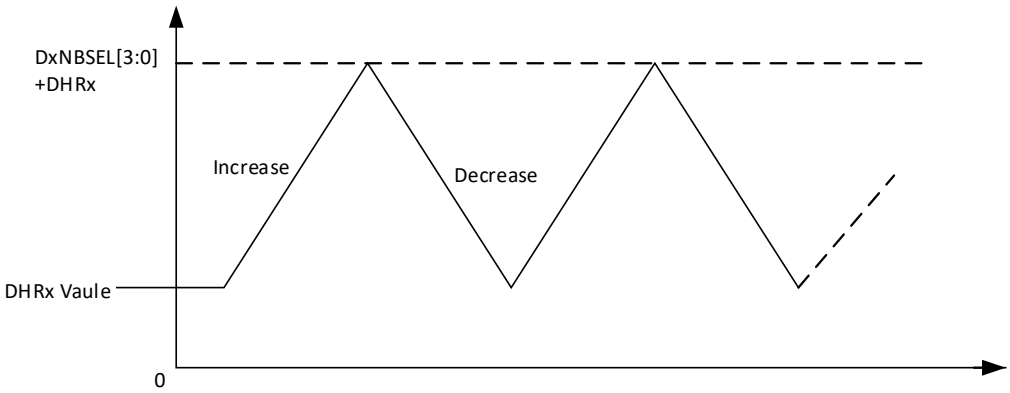


Figure 152. Trianaular-wave generation

DAC data alignment

The DAC supports a single DAC and dual DA mode. The data format is dependent on the selected configuration mode.

Single DAC data format:

8-bit right alignment: load data into the DAC_DxDTH8R[7:0]

12-bit left alignment: load data into the DAC_DxDTH12L[15:4]

12-bit right alignment: load data in the DAC_DxDTH12R[11:0]

Dual DAC data format:

8-bit right alignment: load data into the DAC_DDTH8R[7:0] and DAC_DDTH8R[15:8]

12-bit left alignment: load data into the DAC_DDTH12L[15:4] and DAC_DDTH12L[31:20]

12-bit right alignment: load data into the DAC_DDTH12R[11:0] and DAC_DDTH12R[27:16]

The loaded 8-bit data corresponds to the DHRx[11:4] and the loaded 12-bit data corresponds to the DHRx[11:0]

DAC registers

These peripheral registers must be accessed by word (32 bits).

Table 70. DAC register map and reset values

Register	Offset	Reset value
DAC_CTRL	000h	0x0000 0000
DAC_SWTRG	004h	0x0000 0000
DAC_D1DTH12R	008h	0x0000 0000
DAC_D1DTH12L	00Ch	0x0000 0000
DAC_D1DTH8R	010h	0x0000 0000
DAC_D2DTH12R	014h	0x0000 0000
DAC_D2DTH12L	018h	0x0000 0000
DAC_D2DTH8R	01Ch	0x0000 0000
DAC_DDTH12R	020h	0x0000 0000
DAC_DDTH12L	024h	0x0000 0000
DAC_DDTH8R	028h	0x0000 0000
DAC_D1ODT	02Ch	0x0000 0000
DAC_D2ODT	030h	0x0000 0000

DAC control register (DAC_CTRL)

Bit	Register	Reset value	Type	Description
Bit 31:29	Reserved	0x0	resd	Kept at its default value
Bit 28	D2DMAEN	0x0	rw	DAC2 DMA transfer enable This bit is set and cleared by software. 0: DAC2 DMA mode disabled 1: DAC2 DMA mode enabled
Bit 27:24	D2NBSEL	0x0	rw	DAC2 noise bit select These bits are used to select the mark bit in noise generation mode or amplitude in triangular-wave generation mode. 0000: Unmask LSFR bit0 / Triangle amplitude is equal to 1 0001: Unmask LSFR bit[1:0] / Triangle amplitude is equal to 3 0010: Unmask LSFR bit[2:0] / Triangle amplitude is equal to 7 0011: Unmask LSFR bit[3:0] / Triangle amplitude is equal to 15 0100: Unmask LSFR bit[4:0] / Triangle amplitude is equal to 31 0101: Unmask LSFR bit[5:0] / Triangle amplitude is equal to 63 0110: Unmask LSFR bit[6:0] / Triangle amplitude is equal to 127 0111: Unmask LSFR bit[7:0] / Triangle amplitude is equal to 255 1000: Unmask LSFR bit[8:0] / Triangle amplitude is equal to 511 1001: Unmask LSFR bit[9:0] / Triangle amplitude is equal to 1023 1010: Unmask LSFR bit[10:0] / Triangle amplitude is equal to 2047 ≥1011: Unmask LSFR bit[11:0] / Triangle amplitude is equal to 4095
Bit 23:22	D2NM	0x0	rw	DAC2 noise mode 00: Wave generation disabled 01: Noise wave generation enabled 1x: Triangular wave generation enabled
Bit 21:19	D2TRGSEL	0x0	rw	DAC2 trigger select 000: TMR6 TRGOUT event 001: TMR8 TRGOUT event 010: TMR7 TRGOUT event 011: TMR5 TRGOUT event 100: TMR2 TRGOUT event 101: TMR4 TRGOUT event 110: External interrupt line 9 111: Software trigger Note: These bits can be valid only when D2TRGEN = 1.

Bit	Register	Reset value	Type	Description
Bit 18	D2TRGEN	0x0	rw	DAC2 trigger enable 0: DAC2 trigger disabled 1: DAC2 trigger enabled Note: When the DAC2 trigger is disabled, the data written into the DAC_D2DTHx register is transferred into the DAC_D2ODT register after one APB1 clock cycle. When the DAC2 trigger is enabled, the data written into the DAC_D2DTHx register is transferred into the DAC_ D2ODT register after three APB1 clock cycles If the software trigger is selected, it takes one APB1 clock cycle to have the data written into the DAC_D2DTHx register transferred into the DAC_ D2ODT register.
Bit 17	D2OBDIS	0x0	rw	DAC2 output buffer disable 0: DAC2 output buffer enabled 1: DAC2 output buffer disabled
Bit 16	D2EN	0x0	rw	DAC2 enable 0: DAC2 disabled 1: DAC2 enabled
Bit 15:13	Reserved	0x0	resd	Kept at its default value
Bit 12	D1DMAEN	0x0	rw	DAC1 DMA transfer enable 0: DAC1 DMA transfer disabled 1: DAC1 DMA transfer enabled
Bit 11:8	D1NBSEL	0x0	rw	DAC1 noise bit select These bits are used to select the mark bit in noise generation mode or amplitude in triangular-wave generation mode. 0000: Unmask LSFR bit0 / Triangle amplitude is equal to 1 0001: Unmask LSFR bit[1:0] / Triangle amplitude is equal to 3 0010: Unmask LSFR bit[2:0] / Triangle amplitude is equal to 7 0011: Unmask LSFR bit[3:0] / Triangle amplitude is equal to 15 0100: Unmask LSFR bit[4:0] / Triangle amplitude is equal to 31 0101: Unmask LSFR bit[5:0] / Triangle amplitude is equal to 63 0110: Unmask LSFR bit[6:0] / Triangle amplitude is equal to 127 0111: Unmask LSFR bit[7:0] / Triangle amplitude is equal to 255 1000: Unmask LSFR bit[8:0] / Triangle amplitude is equal to 511 1001: Unmask LSFR bit[9:0] / Triangle amplitude is equal to 1023 1010: Unmask LSFR bit[10:0] / Triangle amplitude is equal to 2047 ≥1011: Unmask LSFR bit[11:0] / Triangle amplitude is equal to 4095
Bit 7:6	D1NM	0x0	rw	DAC1 noise mode 00: Wave generation disabled 01: Noise wave generation enabled 1x: Triangular wave generation enabled

Bit	Register	Reset value	Type	Description
Bit 5:3	D1TRGSEL	0x0	rw	DAC1 trigger select 000: TMR6 TRGOUT event 001: TMR8 TRGOUT event 010: TMR7 TRGOUT event 011: TMR5 TRGOUT event 100: TMR2 TRGOUT event 101: TMR4 TRGOUT event 110: External interrupt line 9 111: Software trigger Note: These bits can be valid only when D1TRGEN = 1.
Bit 2	D1TRGEN	0x0	rw	DAC1 trigger enable 0: DAC1 trigger disabled 1: DAC1 trigger enabled Note: When the DAC1 trigger is disabled, the data written into the DAC_D1DTHx register is transferred into the DAC_D1ODT register after one APB1 clock cycle. When the DAC1 trigger is enabled, the data written into the DAC_D1DTHx register is transferred into the DAC_ D1ODT register after three APB1 clock cycles If the software trigger is selected, it takes one APB1 clock cycle to have the data written into the DAC_D1DTHx register transferred into the DAC_ D1ODT register.
Bit 1	D1OBDIS	0x0	rw	DAC1 output buffer disable 0: DAC1 output buffer enabled 1: DAC1 output buffer disabled
Bit 0	D1EN	0x0	rw	DAC1 enable 0: DAC1 disabled 1: DAC1 enabled

DAC software trigger register (DAC_SWTRG)

Bit	Register	Reset value	Type	Description
Bit 31:2	Reserved	0x0000 0000	resd	Kept at its default value
Bit 1	D2SWTRG	0x0	rw	DAC2 software trigger 0: DAC2 software trigger disabled 1: DAC2 software trigger enabled Note: This bit is cleared by hardware (one APB1 clock cycle later) once the DAC_D2DTH data is loaded into the DAC_D2ODT register.
Bit 0	D1SWTRG	0x0	rw	DAC1 software trigger 0: DAC1 software trigger disabled 1: DAC1 software trigger enabled Note: This bit is cleared by hardware (one APB1 clock cycle later) once the DAC_D1DTH data is loaded into the DAC_D1ODT register.

DAC1 12-bit right-aligned data holding register (DAC_D1DTH12R)

Bit	Register	Reset value	Type	Description
Bit 31:12	Reserved	0x00000	resd	Kept at its default value
Bit 11:0	D1DT12R	0x000	rw	DAC1 12-bit right-aligned data

DAC1 12-bit left-aligned data holding register (DAC_D1DTH12L)

Bit	Register	Reset value	Type	Description
Bit 31:16	Reserved	0x0000	resd	Kept at its default value
Bit 15:4	D1DT12L	0x000	rw	DAC1 12-bit left-aligned data
Bit 3:0	Reserved	0x0	resd	Kept at its default value

DAC1 8-bit right-aligned data holding register (DAC_D1DTH8R)

Bit	Register	Reset value	Type	Description
Bit 31:8	Reserved	0x000000	resd	Kept at its default value
Bit 7:0	D1DT8R	0x00	rw	DAC1 8-bit right-aligned data

DAC2 12-bit right-aligned data holding register (DAC_D2DTH12R)

Bit	Register	Reset value	Type	Description
Bit 31:12	Reserved	0x00000	resd	Kept at its default value
Bit 11:0	D2DT12R	0x000	rw	DAC2 12-bit right-aligned data

DAC2 12-bit left-aligned data holding register (DAC_D2DTH12L)

Bit	Register	Reset value	Type	Description
Bit 31:16	Reserved	0x0000	resd	Kept at its default value
Bit 15:4	D2DT12L	0x000	rw	DAC2 12-bit left-aligned data
Bit 3:0	Reserved	0x0	resd	Kept at its default value

DAC2 8-bit right-aligned data holding register (DAC_D2DTH8R)

Bit	Register	Reset value	Type	Description
Bit 31:8	Reserved	0x000000	resd	Kept at its default value
Bit 7:0	D2DT8R	0x00	rw	DAC2 8-bit right-aligned data

Dual DAC 12-bit right-aligned data holding register (DAC_DDTH12R)

Bit	Register	Reset value	Type	Description
Bit 31:28	Reserved	0x0	resd	Kept at its default value
Bit 27:16	DD2DT12R	0x000	rw	DAC2 12-bit right-aligned data
Bit 15:12	Reserved	0x0	resd	Kept at its default value
Bit 11:0	DD1DT12R	0x000	rw	DAC1 12-bit right-aligned data

Dual DAC 12-bit left-aligned data holding register (DAC_DDTH12L)

Bit	Register	Reset value	Type	Description
Bit 31:20	DD2DT12L	0x000	rw	DAC2 12-bit left-aligned data
Bit 19:16	Reserved	0x0	resd	Kept at its default value
Bit 15:4	DD1DT12L	0x000	rw	DAC1 12-bit left-aligned data
Bit 3:0	Reserved	0x0	resd	Kept at its default value

Dual DAC 8-bit right-aligned data holding register (DAC_DDTH8R)

Bit	Register	Reset value	Type	Description
Bit 31:16	Reserved	0x0000	resd	Kept at its default value
Bit 15:8	DD2DT8R	0x00	rw	DAC2 8-bit right-aligned data
Bit 7:0	DD1DT8R	0x00	rw	DAC1 8-bit right-aligned data

DAC1 data output register (DAC_D1ODT)

Bit	Register	Reset value	Type	Description
Bit 31:12	Reserved	0x00000	resd	Kept at its default value
Bit 11:0	D1ODT	0x000	rw	DAC1 output data

DAC2 data output register (DAC_D2ODT)

Bit	Register	Reset value	Type	Description
Bit 31:12	Reserved	0x00000	resd	Kept at its default value
Bit 11:0	D2ODT	0x000	rw	DAC2 output data

21 CAN

CAN introduction

CAN (Controller Area Network) is a distributed serial communication protocol for real-time and reliable data communication among various nodes. It supports the CAN protocol version 2.0A and 2.0B.

Main features

- Baud rates up to 1M bit/s
- Supports the time triggered communication
- Interrupt enable and mask
- Configurable automatic retransmission mode

Transmission

- Three transmit mailboxes
- Configurable transmit priority
- Supports the time stamp on transmission

Reception

- Two FIFOs with three-level depth
- 14 filter banks
- Supports the identifier list mode
- Supports the identifier mask mode
- FIFO overrun management

Time triggered communication mode

- 16-bit timers
- Time stamp on transmission

Baud rate

The nominal bit time of the CAN bus consists of three parts as follows:

Synchronization segment (SYNC_SEG): This segment has one time unit, and its time duration is defined by the BRDIV[11:0] bit in the CAN_BTMG register.

Bit segment 1 (BIT SEGMENT 1): It is referred to as BSEG1 including the PROP_SEG and PHASE_SEG1 of the CAN standard. Its duration is between 1 and 16 time units, defined by the BTS1[3:0] bit.

Big segment 2 (BIT SEGMENT 2): It is referred to as BSEG2 including the PHASE_SEG2 of the CAN standard. Its duration is between 1 and 8 time units, defined by the BTS2[2:0] bit.

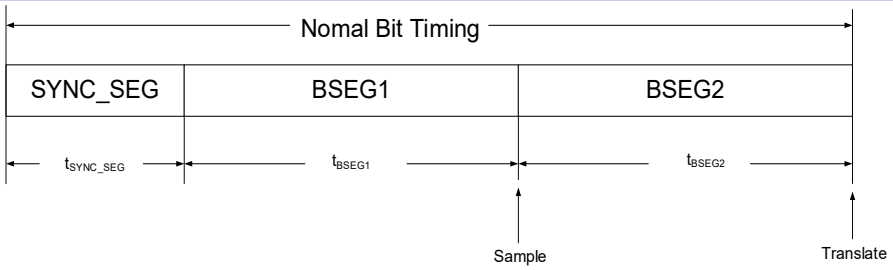


Figure 153. Bit timing

Baud rate formula:

$$\text{BaudRate} = \frac{1}{\text{Nomal Bit Timimg}}$$

$$\text{Nomal Bit Timing} = t_{SYNC_SEG} + t_{BSEG1} + t_{BSEG2}$$

where

$t_{SYNC_SEG} = 1 \times tq$

$t_{BSEG1} = (1+BTS1[3:0]) \times t_q$

$t_{BSEG2} = (1+BTS2[2:0]) \times t_q$

$t_q = (1+BRDIV[11:0]) \times t_{pclk}$

Hard synchronization and resynchronization

The start location of each bit in CAN nodes is always in synchronization segment by default, and the sampling is performed at the edge location of bit segment 1 and big segment 2 simulatenously.

During the actual transmission, each bit of the CAN nodes has certain phase error due to the oscillator drift, transmission delay among the network nodes and noise interference. To avoid the impact on the communication, the start-bit edge and its subsequent falling edge can be synchronized or resynchronized. The time length of the synchronization compensation can not be greater than the resynchronization width (1 to 4 time units, defined by the RSAW[1:0] bit).

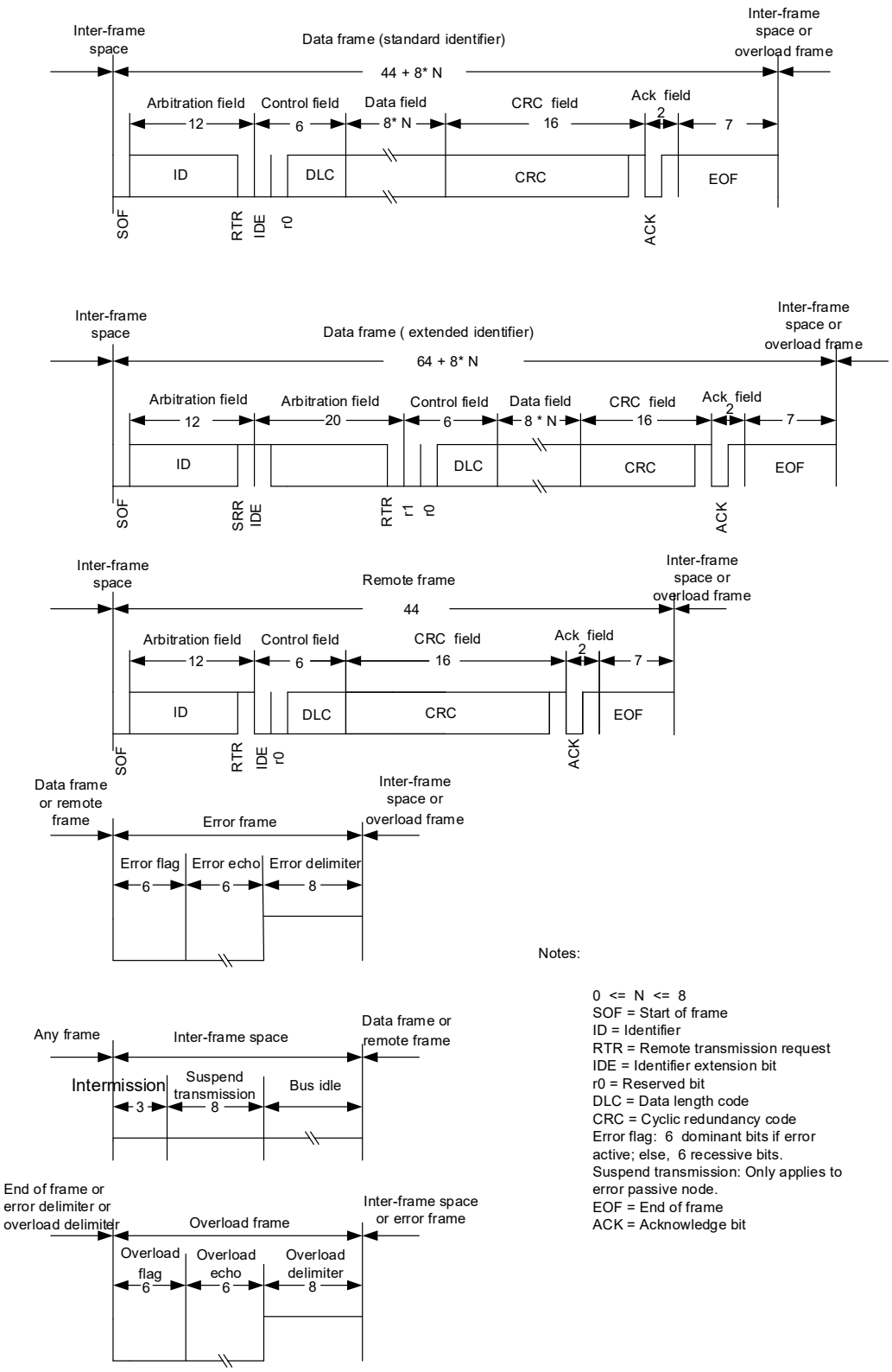


Figure 154. Transmit interrupt generation

Interrupt management

The CAN controller contains four interrupt vectors that can be used to enable or disable interrups by setting the CAN_INTEN register.

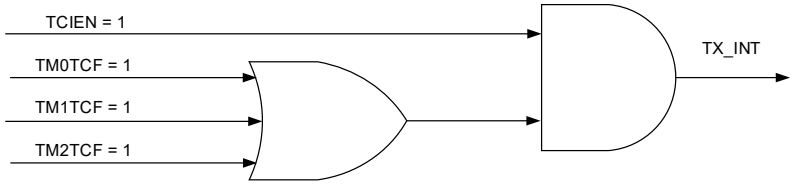


Figure 155. Transmit interrupt generation

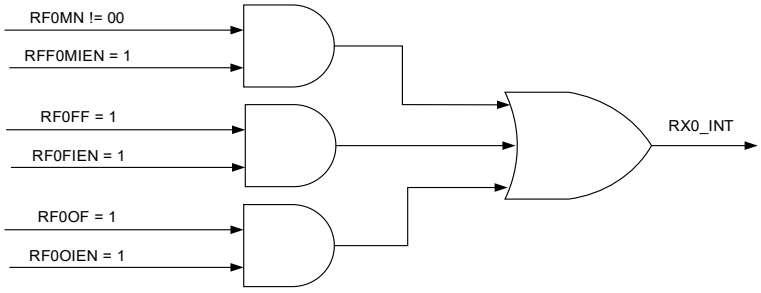


Figure 156. Receive interrupt 0 generation

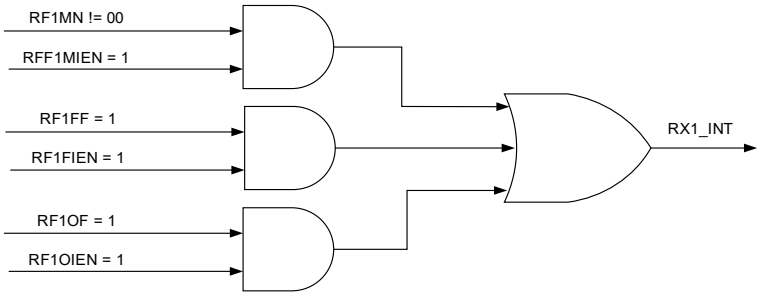


Figure 157. Receive interrupt 1 generation

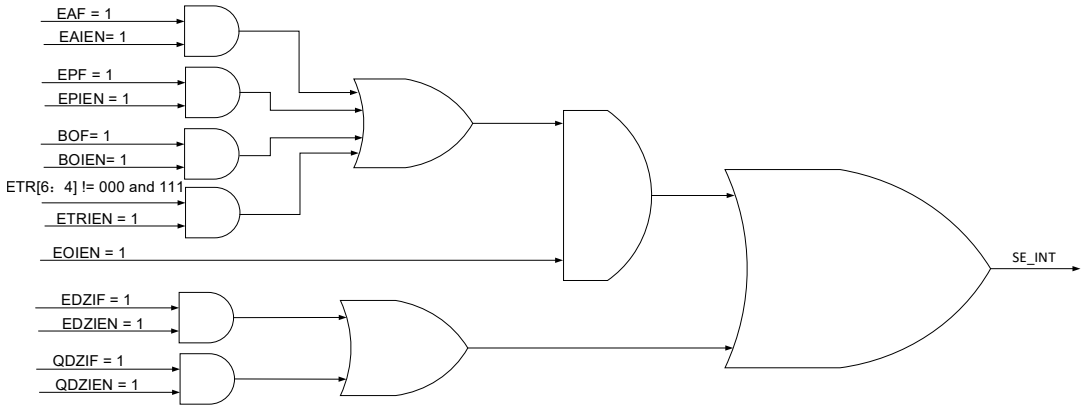


Figure 158. Status error interrupt generation

Interrupt management

The following information can be used as reference for CAN application development:

- Debug control
  When the system enters the debug mode, the CAN controller stops or continues to work normally, depending on the CANx_PAUSE bit in the DEBUG_CTRL register or the PTD bit in the CAN_MCTRL register.
- Time triggered communication
  The timer triggered communication is used to improve the real-time performance so as to avoid bus competition. It is activated by setting TTCEN=1 in the CAN_MCTRL register. The internal 16-bit timer is incremented each CAN bit time, and is sampled on the Start Of Frame bit to generate the time stamp value, which is stored in the CAN_RFCx and CAN_TMCx register.
- Register access protection
  The CAN_BTMG register can be modified only when the CAN is in frozen mode.

Although the transmission of incorrect data will not cause problems at the network level, it can have severe impact on the application. Thus a transmit mailbox can be modified only when it is in empty state.

The filter configuration in the CAN_FMCFG, CAN_FBWCFG and CAN_FRF registers can be modified only when FCS=1. The CAN_FiFBx register can be modified only when FCS=1 or FAENx=0.

Functional descriptions

General description

As the number of nodes in the CAN network and the number of messages grows, an enhanced filtering mechanism is required to handle all types of meassages in order to reduce the processing time of message reception. A FIFO scheme is used to ensure that the CPU can concentrate on application tasks for a long period of time without the loss of messages. In the meantime, the priority order of the messages to be transmitted is configured by hardware. Standard identifiers (11-bit) and extended identifiers (29-bit) are fully supported by hardware.

Based on the above mentioned conditions, the CAN controller provides 14 scalable/configurable identifier filter banks, 2 receive FIFOs with storing 3 complete messages each and being totally managed by hardware, and 3 transmit mailboxes with their transmit priority order defined by the transmit scheduler.

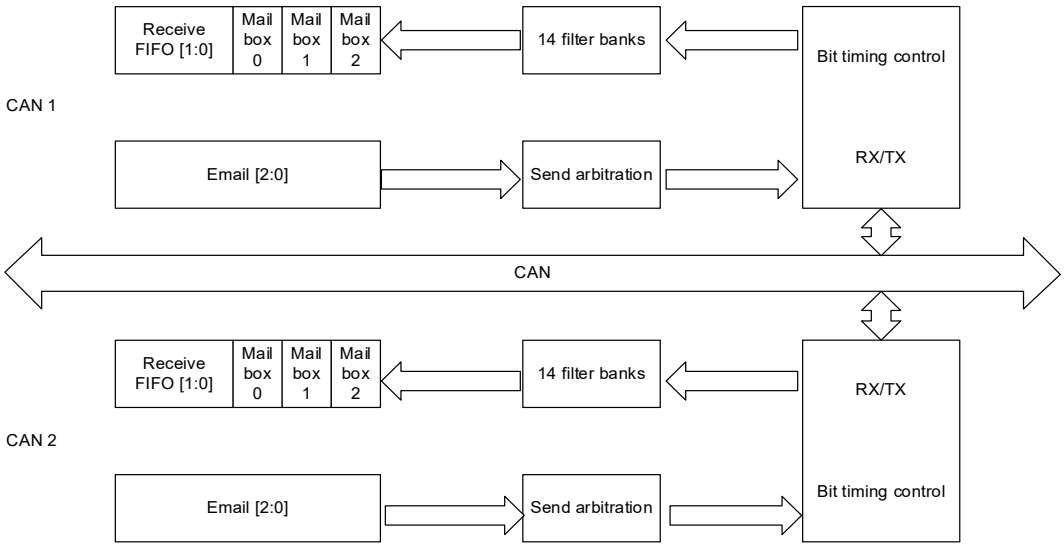


Figure 159. CAN block diagram

Operating modes

The CAN controller has three operating modes:

Sleep mode

After a system reset, the CAN controller is in Sleep mode. In this mode, the CAN clock is stopped to reduce power consumption and an internal pull-up resistance is disabled. However, the software can still access to the mailbox registers.

The software requests the CAN controller to enter Sleep mode by setting the DZEN bit in the CAN_MCTRL register. The hardware confirms the request by setting the DZC bit in the CAN_MSTS register.

Exit Sleep mode in two ways: The CAN controller can be woke up by hardware clearing the DZEN bit when the AEDEN bit in the CAN_MCTRL register and the CAN bus activity is detected. It can also be woke up by software clearing the DZEN bit.

Switch to Frozen mode: The CAN controller switches from Sleep mode to Frozen mode when the FZEN bit is set in the CAN_MCTRL register and the DZEN bit is cleared. Such switch operation is confirmed by hardware setting the FZC bit in the CAN_MSTS register.

Switch to Communication mode: The CAN controller enters Communication mode when the FZEN and DZEN bits are both cleared and the CAN controller has synchronized with the bus. In other words, it must wait for 11 consecutive recessive bits to be detected on the CANRX pin.

Frozen mode

The software initialization can be done only in Frozen mode, including the CAN_BTMG and CAN_MCTRL registers. But the initialization of the 14 CAN filter banks (mode, scale, FIFO association, activation and filter values) can be done in non-Frozen mode. When the CAN controller is in Frozen mode, message reception and transmission are both disabled.

Switch to Communication mode: The CAN controller leaves Frozen mode when the FZEN bit is cleared in the CAN_MCTRL register. This switch operation is confirmed by hardware clearing the FZC bit in the CAN_MSTS register. The CAN controller must be synchronized with the bus.

Switch to Sleep mode: The CAN controller enters Sleep mode if DZEN=1 and FZEN=0 in the CAN_MCTRL register. This switch operation is confirmed by hardware setting the DZC bit in the CAN_MSTS register.

Communication mode

After the CAN_BTMG and CAN_MCTRL registers are configured in Frozen mode, the CAN controller enters Communication mode and is ready for message reception and transmission.

Switch to Sleep mode: The CAN controller switches to Sleep mode when the DZEN bit is set in the CAN_MCTRL register and the current CAN bus transmission is complete.

Switch to Frozen mode: The CAN controller enters Frozen mode when the FZEN bit is set in the CAN_MCTRL register and the current CAN bus transmission is complete.

Test modes

The CAN controller defines three test modes, including Listen-only mode, Loop back mode and combined Listen-only and Loop back mode. Test mode can be selected by setting the LOEN and LBEN bits in the CAN_BTMG register.

- Listen-only mode is selected when the LOEN bit is set in the CAN_BTMG register. In this mode, the CAN is able to receive data, but only recessive bits are output on the CANTX pin. In the meantime, the dominant bits output on the CANTX can be monitored by the receive side but without affecting the CAN bus.
- Loop back mode is selected by setting the LBEN bit in the CAN_BTMG register. In this mode, The CAN only receives the level signal on its CANTX pin. Meanwhile, the CAN can also send data to the external bus. The Loop back mode is mainly used for self-test functions.
- It is possible to combine the Listen-only and Loop back mode by setting the LOEN and LBEN bits in the CAN_BTMG register. In this case, the CAN is disconnected from the bus network, the CANTX pin remains in recessive state, and the transmit side is connected to the receive side.

Message filtering

The received message has to go through filtering by its identifier. If passed,the message will be stored in the correspoinding FIFOs. If not, the message will be discarded. The whole operation is done by hardware without using CPU resources.

Filter bit width

The CAN controller provides 14 configurable and scalable filter banks (0~13). Each filter bank has two 32-bit registers, CAN_FiFB1 and CAN_FiFB. The filter bit width can be configured as two 16 bits or one 32 bits, depending on the corresponding bits in the CAN_FBWCFG register.

32-bit fliter register CAN_FiFBx includes the SID[10: 0], EID[17:0], IDT and RTR bits.

CAN_FiFB1[31:21]	CAN_FiFB1[20: 3]	CAN_FiFB1[2:0]		
CAN_FiFB2[31:21]	CAN_FiFB2[20: 3]	CAN_FiFB2[2:0]		
SID[10:0]/EID[28:18]	EID[17:0]	IDT	RTR	0

Two 16-bit filter register CAN_FiFBx includes SID[10: 0], IDT, RTR and EID[17:15] bits

CAN_FiFB1[31:21]	CAN_FiFB1 [20:19]		CAN_FiFB1 [18:16]	CAN_FiFB1 [15:5]	CAN_FiFB1 [4:3]		CAN_FiFB1 [2:0]
CAN_FiFB2[31:21]	CAN_FiFB2 [20:19]		CAN_FiFB2 [18:16]	CAN_FiFB2 [15:5]	AN_FiFB2 [4:3]		CAN_FiFB2 [2:0]
SID[10: 0]	IDT	RTR	EID[17:15]	SID[10:0]	IDT	RTR	EID[17:15]

Filtering mode

The filter can be configured in identifier mask mode or in identifier list mode by setting the FMSELx bit in the CAN_FMCFG register. The mask mode is used to specify which bits must match the pre-programmed identifiers, and which bits do not need. In identifier list mode, the identifier must match the pre-programmed identifier. The two modes can be used in conjunction with filter width to deliver four filtering modes below:

ID	CAN_FiFB1[31:21]	CAN_FiFB1[20:3]	CAN_FiFB1 [2:0]		
Mask	CAN_FiFB2[31:21]	CAN_FiFB2[20:3]	CAN_FiFB2 [2:0]		
Mapping	SID[10:0]	EID[17:0]	IDT	RTR	0

Figure 160. 32-bit identifier mask mode

ID	CAN_FiFB1[31:21]	CAN_FiFB1[20:3]	CAN_FiFB1 [2:0]		
ID	CAN_FiFB2[31:21]	CAN_FiFB2[20:3]	CAN_FiFB2 [2:0]		
Mapping	SID[10:0]	EID[17:0]	IDT	RTR	0

Figure 161. 32-bit identifier list mode

ID	CAN_FiFB1[15:5]	CAN_FiFB1[4:0]		
Mask	CAN_FiFB1[31:21]	CAN_FiFB1[20:16]		
ID	CAN_FiFB2[15:5]	CAN_FiFB2[4:0]		
Mask	CAN_FiFB2[31:21]	CAN_FiFB2[20:16]		
Mapping	SID[10:0]	RTR	IDT	EID[17:15]

Figure 162. 16-bit identifier mask mode

ID	CAN_FiFB1[15:8]	CAN_FiFB1[7:0]		
ID	CAN_FiFB1[31:24]	CAN_FiFB1[23:16]		
ID	CAN_FiFB2[15:8]	CAN_FiFB2[7:0]		
ID	CAN_FiFB2[31:24]	CAN_FiFB2[23:16]		
Mapping	SID[10:0]	RTR	IDT	EID[17:15]

Figure 163. 16-bit identifier list mode

Filter match number

14 filter banks have different filtering effects dependent on the bit width mode. For example, 32-bit identifier mask mode contains the filters numbered n while 16-bit identifier list mode contains the filters numbered n, n+1, n+2 and n+3. When a frame of message passes through the filter number N, the number N is stored in the RFFMN[7:0] bit in the CAN_RFCx register. The distribution of the filter number does not take into account the activation state of the filter banks.

The following are examples of filter numbering.

Table 71. Examples of filter numbering

Filter bank	FIFO0	Active	Filter number
0	CAN_F0FB1[31:0]-ID	Yes	0
	CAN_F0FB2[31:0]-ID		1
1	CAN_F1FB1[15:0]-ID	Yes	2
	CAN_F1FB1[31:16]-ID		3
	CAN_F1FB2[15:0]-ID		4
	CAN_F1FB2[31:16]-ID		5
2	CAN_F2FB1[31:0]-ID	Yes	6
	CAN_F2FB2[31:0]-Mask		
6	CAN_F6FB1[15:0]-ID	No	7
	CAN_F6FB1[31:16]-Mask		
	CAN_F6FB2[15:0]-ID		8
	CAN_F6FB2[31:16]-Mask		
9	CAN_F9FB1[31:0]-ID	No	9
	CAN_F9FB2[31:0]-ID		10
10	CAN_F10FB1[15:0]-ID	Yes	11
	CAN_F10FB1[31:16]-Mask		
	CAN_F10FB2[15:0]-ID		12
	CAN_F10FB2[31:16]-Mask		
12	CAN_F12FB1[15:0]-ID	No	13
	CAN_F12FB1[31:16]-ID		14
	CAN_F12FB2[15:0]-ID		15
	CAN_F12FB2[31:16]-ID		16

Filter bank	FIFO1	Active	Filter number
3	CAN_F3FB1[15:0]-ID	Yes	0
	CAN_F3FB1[31:16]-ID		1
	CAN_F3FB2[15:0]-ID		2
	CAN_F3FB2[31:16]-ID		3
4	CAN_F4FB1[31:0]-ID	Yes	4
	CAN_F4FB2[31:0]-Mask		
5	CAN_F5FB1[15:0]-ID	No	5
	CAN_F5FB1[31:16]-Mask		
	CAN_F5FB2[15:0]-ID		6
	CAN_F5FB2[31:16]-Mask		
7	CAN_F7FB1[15:0]-ID	No	7
	CAN_F7FB1[31:16]-ID		8
	CAN_F7FB2[15:0]-ID		9
	CAN_F7FB2[31:16]-ID		10
8	CAN_F8FB1[31:0]-ID	Yes	11
	CAN_F8FB2[31:0]-Mask		
11	CAN_F11FB1[31:0]-ID	Yes	12
	CAN_F11FB2[31:0]-ID		13
13	CAN_F13FB1[15:0]-ID	Yes	14
	CAN_F13FB1[31:16]-ID		15
	CAN_F13FB2[15:0]-ID		16
	CAN_F13FB2[31:16]-ID		17

Priority rules

When the CAN controller receives a frame of message, the message may pass through severl filters. In this case, the filter match number stored in the receive mailbox is determined according to the following priority rules:

- A 32-bit filter has priority over a 16-bit filter
- For filters with identical bit width, the identifier list mode has priority over the identifier mask mode
- For filter with identical bit width and identifier mode, the lower number has priority over the higher number.

Filter configuration

- The CAN filters ar configured by setting the FCS in the CAN_FCTRL register.
- Identifier mask mode or identifier list mode can be selected by setting the FMSELx bit in the CAN_FMCFG register.
- The filter bit width can be configured as two 16 bits or one 32 bits by setting the FBWSELx bit in the CAN_FBWCFG register.
- The filter x is associated with FIFO0 or FIFO1 by setting the FRFSELx bit in the CAN_FRF register.
- The filter banks x are activated by setting FAENx=1 in the CAN_FACFG register.
- Configure 0~13 filter banks by writing to the CAN_FiFBx register (i=0…13; x=1,2).
- Complete the CAN filter configuration by setting FCS=0 in the CAN_FCTRL register.

Message transmission

Register configuration

To transmit a message, it is necessary to select one transmit mailbox and configure it through the CAN_TMIx, CAN_TMCx, CAN_TMDTLx and CAN_TMDTHx registers. Once the mailbox configuration is complete, setting the TMSR bit in the CAN_TMIx register can initiate CAN transmission.

Message transmission

The mailbox enters pending state immediately after the mailbox is configured and the CAN controller receives the transmit request. At this point, the CAN controlle will confirm whether the mailbox is given the highest priority or not. If yes, it will enter SCHEDULED STATE, otherwise, it will wait to get the highest priority. The mailbox in SCHEDULED state will monitor the CAN bus state so that the messages in SCHEDULED mailbox can be transmitted as soon as the CAN bus becomes idle. The mailbox will enter EMPTY state at the end of the message transmission.

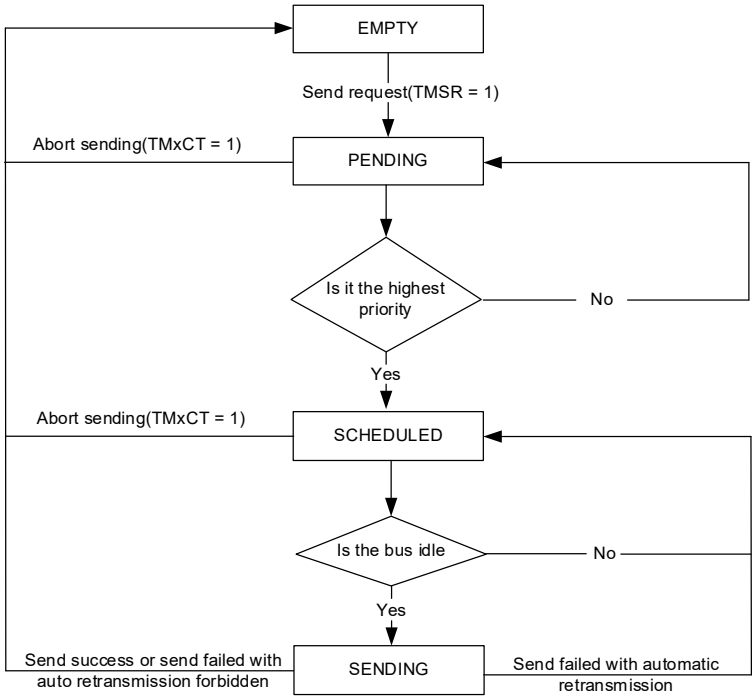


Figure 164. Transmit mailbox status

Transmit priority configuration

When two or more transmit boxes are in PENDING state, their transmit priority must be given.

By identifier: When MMSSR=0 in the CAN_MCTRL register, the transmit order is defined by the identifier of the message in the mailbox. The message with lower identifier value has the highest priority. If the identifier values are the same, the message with lower mailbox number will be transmitted first.

By transmit request order: When MMSSR=1 in the CAN_MCTRL register, the transmit priority is given by the transmit request order of mailboxes.

Transmit status and error status

The TMxTCF, TMxTSF, TMxALF, TMxTEF and TMxEF bits in the CAN_TSTS register are used to indicate transmit status and error status.

TMxTCF bit: Transmission complete flag, indicating that the data transmission is complete when TMxTCF=1.

TMxTSF bit: Transmission success flag, indicating that the data has been transmitted successfully when TMxTSF =1.

TMxALF bit: Transmission arbitration lost flag, indicating that the data transmission arbitration is lost when TMxALF=1.

TMxTEF bit: Transmission error flag, indicating that the data transmission failed due to bus error, and an error frame is sent when TMxTEF=1.

TMxEF bit: Mailbox empty flag, indicating that the data transmission is complete and the mailbox becomes empty when TMxEF=1.

Transmit abort

The TMxCT bit is set in the CAN_TSTS register to abort the transmission of the current mailbox, detailed as follows:

When the current transmission fails or arbitration is lost, if the automatic retransmission mode is disabled, the tranmist mailbox become EMPTY; if the automatic retransmission mode is enbled, the tranmist mailbox becomes SCHEDULED, the mailbox transmission then is aborted and becomes EMPTY.

When the current transmission is complete successfully, the mailbox becomes EMPTY.

Message reception

Register configuration

The CAN_RFIx, CAN_RFCx, CAN_RFDTLx and CAN_RFDTHx registers can be used by user applications to obtain valid messages.

Message reception

The CAN controller boasts two FIFO with three levels to receive messages. FIFO rule is adopted. When the message is received correctly and has passed the identifier filtering, it is regarded as a valid message and is stored in the corresponding FIFO. The number of the received messages RFxMN[1:0] will be incremented by one whenever the receive FIFO receives a valid message. If a valid message is received when RFxMN[1:0]=3, the controller will select either to overwrite the previous messages or discard the new incoming message through the MDRSEL bit in the CAN_MCTRL register.

In the meantime, when the user reads a frame of message and the RFxR is set in the CAN_RFx register, one FIFO mailbox is released, and RFxMN[1:0] bit is descremented by one in the CAN_RFx register.

Receive FIFO status

RFxMN[1:0], RFxFF and RFxOF bits in the RFx register are used to indicate receive FIFO status.

RFxMN[1:0]: indicates the number of valid messages stored in the FIFOx.

RFxFF: indicates that three valid messages are stroed in the FIFOx (i.e. the three mailboxes are full), as shown in (c) of the following diagram.

RFxOF: indicates that a new valid message has been received while the FIFOx is full, as shown in (d) of the following diagram.

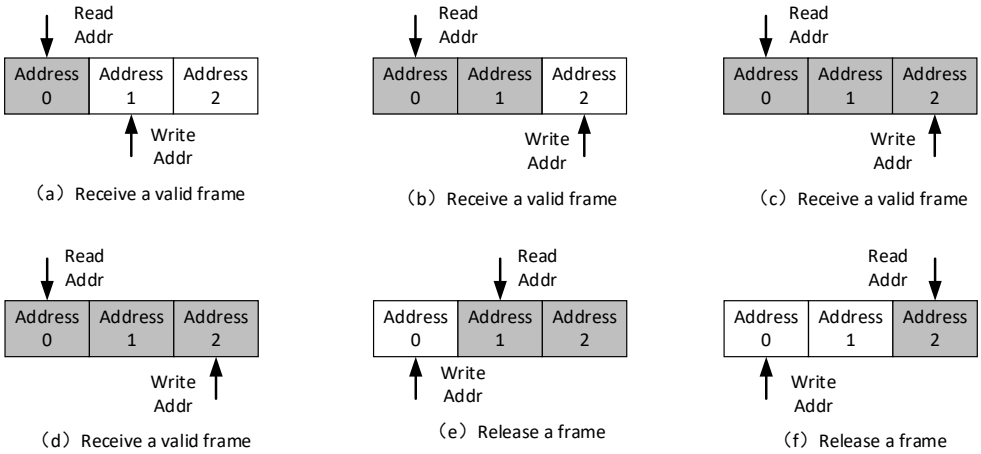


Figure 165. Receive FIFO status

Error management

The status of CAN nodes is indicated by the receive error counter (TEC) and transmit error counter (REC) bits in the CAN_ESTS register. In the meantime, the ETR[2:0] bit in the CAN_ESTS register is used to record the last error source, and the corresponding interrupts will be generated when the CAN_INTEN register is enabled.

- Error active flag: When both TEC and REC are lower than 128, the system is in the error active state. An error active flag is set when an error is detected.
- Error passive flag: When either TEC or REC is greater than 127, the system is in the error passive state. An error passive flag is set when an error is detected.
- Bus-off state: The bus-off state is entered when TEC is greater than 255. In this state, the CAN is no logner able to transmit and receive messages. There are two ways to resume CAN from bus-off state.
  - Option 1: When AEBOEN=0 in the CAN_MCTRL register, in communication mode, the software requests to enter Frozen mode and exit Frozen mode, and CAN will then resume from bus-off state after 128 occurences of 11 consecutive recessive bits have been detected on the CAN RX pin.
  - Option 2: When AEBOEN=1 in the CAN_MCTRL register, the CAN will resume from bus-off state automatically after 128 occurrences of 11 consecutive recessive bits have been detected on the CAN RX pin

CAN registers

These peripheral registers must be accessed by word (32 bits).

Table 72. CAN register map and reset values

Register	Offset	Reset value
MCTRL	000h	0x0001 0002
MSTS	004h	0x0000 0C02
TSTS	008h	0x1C00 0000
RF0	00Ch	0x0000 0000

Register	Offset	Reset value
FR1	010h	0x0000 0000
INTEN	014h	0x0000 0000
ESTS	018h	0x0000 0000
BTMG	01Ch	0x0123 0000
Reserved	020h~17Fh	xx
TMI0	180h	0xXXXX XXXX
TMC0	184h	0xXXXX XXXX
TMDTL0	188h	0xXXXX XXXX
TMDTH0	18Ch	0xXXXX XXXX
TMI1	190h	0xXXXX XXXX
TMC1	194h	0xXXXX XXXX
TMDTL1	198h	0xXXXX XXXX
TMDTH1	19Ch	0xXXXX XXXX
TMI2	1A0h	0xXXXX XXXX
TMC2	1A4h	0xXXXX XXXX
TMDTL2	1A8h	0xXXXX XXXX
TMDTH2	1ACh	0xXXXX XXXX
RFI0	1B0h	0xXXXX XXXX
RFC0	1B4h	0xXXXX XXXX
RFDTL0	1B8h	0xXXXX XXXX
RFDTH0	1BCh	0xXXXX XXXX
RFI1	1C0h	0xXXXX XXXX
RFC1	1C4h	0xXXXX XXXX
RFDTL1	1C8h	0xXXXX XXXX
RFDTH1	1CCh	0xXXXX XXXX
Reserved	1D0h~1FFh	xx
FCTRL	200h	0x2A1C 0E01
FMCFG	204h	0x0000 0000
Reserved	208h	xx
FSCFG	20Ch	0x0000 0000
Reserved	210h	xx
FRF	214h	0x0000 0000
Reserved	218h	xx
FACFG	21Ch	0x0000 0000
Reserved	220h~23Fh	xx
FB0F1	240h	0xXXXX XXXX
FB0F2	244h	0xXXXX XXXX
FB1F1	248h	0xXXXX XXXX
FB1F2	24Ch	0xXXXX XXXX
…	…	...
FB13F1	2A8h	0xXXXX XXXX
FB13F2	2ACh	0xXXXX XXXX

CAN control and status registers

CAN master control register (CAN_MCTRL)

Bit	Register	Reset value	Type	Description
Bit 31:17	Reserved	0x0000	resd	Kept at its default value.
Bit 16	PTD	0x1	rw	Prohibit trans when debug 0: Transmission works during debug 1: Transmission is prohibited during debug. Receive FIFO can be still accessible normally. Note: Transmission can be disabled only when PTD and CANx_PAUSE bits in the DEBUG_CTRL register are set simultaneously.
Bit 15	SPRST	0x0	rw1s	Software partial reset 0: Normal 1: Software partial reset Note: SPRST only reset receive FIFO and MCTRL register. The CAN enters Sleep mode after reset. Then this bit is automatically cleared by hardware.
Bit 14:8	Reserved	0x00	resd	Kept at its default value.
Bit 7	TTCEN	0x0	rw	Time triggered communication mode enable 0: Time triggered communication mode disabled 1: Time triggered communication mode enabled
Bit 6	AEBOEN	0x0	rw	Automatic exit bus-off enable 0: Automatic exit bus-off disabled 1: Automatic exit bus-off enabled Note: When Automatic exit bus-off mode is enabled, the hardware will automatically leave bus-off mode as soon as an exit timing is detected on the CAN bus. When Automatic exit bus-off mode is disabled, the software must enter/leave the freeze mode once more, and then the bus-off state is left only when an exit timing is detected on the CAN bus.
Bit 5	AEDEN	0x0	rw	Automatic exit doze mode enable 0: Automatic exit sleep mode disabled 1: Automatic exit sleep mode enabled Note: When Automatic exit sleep mode is disabled, the sleep mode is left by software clearing the sleep request command. When Automatic exit sleep mode is enabled, the sleep mode is left without the need of software intervention as soon as a message is monitored on the CAN bus.
Bit 4	PRSFEN	0x0	rw	Prohibit retransmission enable when sending fails enable 0: Retransmission is enabled. 1: Retransmission is disabled.
Bit 3	MDRSEL	0x0	rw	Message discard rule select when overflow 0: The previous message is discarded. 1: The new incoming message is discarded.

Bit	Register	Reset value	Type	Description
Bit 2	MMSSR	0x0	rw	Multiple message transmit sequence rule 0: The message with the smallest identifier is first transmitted. 1: The message with the first request order is first transmitted.
Bit 1	DZEN	0x1	rw	Doze mode enable 0: Sleep mode is disabled. 1: Sleep mode is enabled. Note: The hardware will automatically leave sleep mode when the AEDEN ib set and a message is monitored on the CAN bus. After CAN reset or partial software reset, this bit is forced to be set by hardware, that is, the CAN will keep in sleep mode, by default.
Bit 0	FZEN	0x0	rw	Freeze mode enable 0: Freeze mode disabled 1: Freeze mode enabled Note: The CAN leaves Freeze mode once 11 consecutive recessive bits have been detected on the RX pin. For this reason, the software acknowledges the entry of Freeze mode after the FZC bit is cleared by hardware. The Freeze mode is entered only when the current CAN activity (transmission or reception) is completed. Thus the sotware acknowledges the exit of Freeze mode after the FZC bit is cleared by hardware.

CAN master status register (CAN_MSTS)

Bit	Register	Reset value	Type	Description
Bit 31:12	Reserved	0x00000	resd	Kept at its default value.
Bit 11	REALRX	0x1	ro	Real time level on RX pin 0: Low 1: High
Bit 10	LSAMPRX	0x1	ro	Last sample level on RX pin 0: Low 1: High Note: This value keeps updating with the REALRX.
Bit 9	CURS	0x0	ro	Current receive status 0: No reception occurs 1: Reception is in progress Note: This bit is set by hardware when the CAN reception starts, and it is cleared by hardware at the end of reception.
Bit 8	CUSS	0x0	ro	Current transmit status 0: No transmit occurs 1: ransmit is in progress Note: This bit is set by hardware when the CAN transmission starts, and it is cleared by hardware at the end of transmission.
Bit 7:5	Reserved	0x0	resd	Kept at its default value.

Bit	Register	Reset value	Type	Description
Bit 4	EDZIF	0x0	rw1c	Enter doze mode interrupt flag 0: Sleep mode is not entered or no condition for flag set. 1: Sleep mode is entered. Note: This bit is set by hardware only when EDZIEN=1 and the CAN enters Sleep mode. Whe set, this bit will generate a status change interrupt. This bit is cleared by software (writing 1 to itself) or by hardware when DZC is cleared.
Bit 3	QDZIF	0x0	rw1c	Exit doze mode interrupt flag 0: Sleep mode is not left or no condition for exit. 1: Sleep mode has been left or exit condition has generated. Note: This bit is cleared by software (writing 1 to itself) Sleep mode is left when a SOF is detected on the bus. When QDZIEN=1, this bit will generate a status change interrupt.
Bit 2	EOIF	0x0	rw1c	Error occur interrupt flag 0: No error interrupt or no condition for error interrupt flag 1: Error interrupt is generated. Note: This bit is cleared by software (writing 1 to itself). This bit is set by hardware only when the corresponding bit is set in the CAN_ESTS register and the corresponding interrupt enable bit in the CAN_INTEN register is enabled. When EOIEN=1, this bit generates a status change interrupt.
Bit 1	DZC	0x1	ro	Doze mode acknowledge 0: The CAN is not in Sleep mode. 1: CAN is in Sleep mode. Note: This bit is used to decide whether the CAN is in Sleep mode or not. This bit acknowledges the Sleep mode request generated by software. The Sleep mode can be entered only when the current CAN activity (transmission or reception) is completed. For this reason, the software acknowledges the entry of Sleep mode after this bit is set by hardware. The Sleep mode is left only once 11 consecutive recessive bits have been detect on the CAN RX pin. For this reason, the software acknowledges the exit of Sleep mode after this bit is cleared by hardware.

Bit	Register	Reset value	Type	Description
Bit 0	FZC	0x0	ro	Freeze mode confirm 0: The CAN is not in Freeze mode 1: The CAN is in Freeze mode Note: This bit is used to decide whether the CAN is in Freeze mode or not. This bit acknowledges the Freeze mode request generated by software. The Freeze mode can be entered only when the current CAN activity (transmission or reception) is completed. For this reason, the software acknowledges the entry of Freeze mode after this bit is set by hardware. The Freeze mode is left only once 11 consecutive recessive bits have been detect on the CAN RX pin. For this reason, the software acknowledges the exit of Freeze mode after this bit is cleared by hardware.

CAN transmit status register (CAN_TSTS)

Bit	Register	Reset value	Type	Description
Bit 31	TM2LPF	0x0	ro	Transmit mailbox 2 lowest priority flag 0: Mailbox 2 is not given the lowest priority. 1: Lowest priority (This indicates that more than one mailboxes are pending for transmission, the mailbox 2 has the lowest priority.)
Bit 30	TM1LPF	0x0	ro	Transmit mailbox 1 lowest priority flag 0: Mailbox 1 is not given the lowest priority. 1: Lowest priority (This indicates that more than one mailboxes are pending for transmission, the mailbox 1 has the lowest priority.)
Bit 29	TM0LPF	0x0	ro	Transmit mailbox 0 lowest priority flag 0: Mailbox 0 is not given the lowest priority. 1: Lowest priority (This indicates that more than one mailboxes are pending for transmission, the mailbox 0 has the lowest priority.)
Bit 28	TM2EF	0x1	ro	Transmit mailbox 2 empty flag This bit is set by hardware when no transmission is pending in the mailbox 2.
Bit 27	TM1EF	0x1	ro	Transmit mailbox 1 empty flag This bit is set by hardware when no transmission is pending in the mailbox 1.
Bit 26	TM0EF	0x1	ro	Transmit mailbox 0 empty flag This bit is set by hardware when no transmission is pending in the mailbox 0.

Bit	Register	Reset value	Type	Description
Bit 25:24	TMNR	0x0	ro	Transmit Mailbox number record Note: If the transmit mailbox is free, these two bits refer to the number of the next transmit mailbox free. For example, in case of free CAN, the value of these two bit becomes 01 after a message transmit request is written. If the transmit box is full, these two bits refer to the number of the transmit mailbox with the lowest priority. For example, when there are three messages are pending for transmission, the identifiers of mailbox 0, mailbox 1 and mailbox 2 are 0x400, 0x433 and 0x411 respectively, and the value of these two bits becomes 01.
Bit 23	TM2CT	0x0	ro	Transmit mailbox 2 cancel transmit 0: No effect 1: Transmission is cancelled. Note: Software sets this bit to abort the transmission of mailbox 2. This bit is cleared by hardware when the transmit message in the mailbox 2 is cleared. Setting this bit has no effect if the mailbox 2 is free.
Bit 22:20	Reserved	0x0	resd	Kept at its default value.
Bit 19	TM2TEF	0x0	rw1c	Transmit mailbox 2 transmission error flag 0: No error 1: Mailbox 2 transmission error Note: This bit is set when the mailbox 2 transmission error occurred. It is cleared by software writing 1 or by hardware at the start of the next transmission
Bit 18	TM2ALF	0x0	rw1c	Transmit mailbox 2 arbitration lost flag 0: No arbitration lost 1: Transmit mailbox 2 arbitration lost Note: This bit is set when the mailbox 2 transmission failed due to an arbitration lost. It is cleared by software writing 1 or by hardware at the start of the next transmission
Bit 17	TM2TSF	0x0	rw1c	Transmit mailbox 2 transmission success flag 0: Transmission failed 1: Transmission was successful. Note: This bit indicates whether the mailbox 2 transmission is successful or not. It is cleared by software writing 1.

Bit	Register	Reset value	Type	Description
Bit 16	TM2TCF	0x0	rw1c	Transmit mailbox 2 transmission completed flag 0: Transmission is in progress 1: Transmission is completed Note: This bit is set by hardware when the transmission/abort request on mailbox 2 has been completed. It is cleared by software writing 1 or by hardware when a new transmission request is received. Clearing this bit will clear the TM2TSF, TM2ALF and TM2TEF bits of mailbox 2.
Bit 15	TM1CT	0x0	rw1s	Transmit mailbox 1 cancel transmit 0: No effect 1: Mailbox 1 cancel transmit Note: This bit is set by software to abort the transmission request on mailbox 1. Clearing the message transmission on mailbox 1 will clear this bit. Setting by this software has no effect when the mailbox 1 is free.
Bit 14:12	Reserved	0x0	resd	Kept at its default value.
Bit 11	TM1TEF	0x0	rw1c	Transmit mailbox 1 transmission error flag 0: No error 1: Mailbox 1 transmission error Note: This bit is set when the mailbox 1 transmission error occurred. It is cleared by software writing 1 or by hardware at the start of the next transmission
Bit 10	TM1ALF	0x0	rw1c	Transmit mailbox 1 arbitration lost flag 0: No arbitration lost 1: Transmit mailbox 1 arbitration lost Note: This bit is set when the mailbox 1 transmission failed due to an arbitration lost. It is cleared by software writing 1 or by hardware at the start of the next transmission
Bit 9	TM1TSF	0x0	rw1c	Transmit mailbox 1 transmission success flag 0: Transmission failed 1: Transmission was successful. Note: This bit indicates whether the mailbox 1 transmission is successful or not. It is cleared by software writing 1.
Bit 8	TM1TCF	0x0	rw1c	Transmit mailbox 1 transmission completed flag 0: Transmission is in progress 1: Transmission is completed Note: This bit is set by hardware when the transmission/abort request on mailbox 1 has been completed. It is cleared by software writing 1 or by hardware when a new transmission request is received. Clearing this bit will clear the TM1TSF, TM1ALF and TMF1TEF bits of mailbox 1.

Bit	Register	Reset value	Type	Description
Bit 7	TM0CT	0x0	rw1s	Transmit mailbox 0 cancel transmit 0: No effect 1: Mailbox 0 cancel transmit Note: This bit is set by software to abort the transmission request on mailbox 0. Clearing the message transmission on mailbox 0 will clear this bit. Setting by this software has no effect when the mailbox 0 is free.
Bit 6:4	Reserved	0x0	resd	Kept at its default value.
Bit 3	TM0TEF	0x0	rw1c	Transmit mailbox 0 transmission error flag 0: No error 1: Mailbox 0 transmission error Note: This bit is set when the mailbox 0 transmission error occurred. It is cleared by software writing 0 or by hardware at the start of the next transmission
Bit 2	TM0ALF	0x0	rw1c	Transmit mailbox 0 arbitration lost flag 0: No arbitration lost 1: Transmit mailbox 0 arbitration lost Note: This bit is set when the mailbox 0 transmission failed due to an arbitration lost. It is cleared by software writing 1 or by hardware at the start of the next transmission
Bit 1	TM0TSF	0x0	rw1c	Transmit mailbox 0 transmission success flag 0: Transmission failed 1: Transmission was successful. Note: This bit indicates whether the mailbox 0 transmission is successful or not. It is cleared by software writing 1.
Bit 0	TM0TCF	0x0	rw1c	Transmit mailbox 0 transmission completed flag 0: Transmission is in progress 1: Transmission is completed Note: This bit is set by hardware when the transmission/abort request on mailbox 0 has been completed. It is cleared by software writing 1 or by hardware when a new transmission request is received. Clearing this bit will clear the TM0TSF, TM0ALF and TM0TEF bits of mailbox 0.

CAN receive FIFO 0 register (CAN_RF0)

Bit	Register	Reset value	Type	Description
Bit 31:6	Reserved	0x0000000	resd	Kept at its default value.
Bit 5	RF0R	0x0	rw1s	Receive FIFO 0 release 0: No effect 1: Release FIFO Note: This bit is set by software to release FIFO 0. It is cleared by hardware when the FIFO 0 is released. Seting this bit by software has no effect when the FIFO 0 is empty. If there are more than two messages pending in the FIFO 0, the software has to release the FIFO 0 to access the second message.
Bit 4	RF0OF	0x0	rw1c	Receive FIFO 0 overflow flag 0: No overflow 1: Receive FIFO 0 overflow Note: This bit is set by hardware when a new message has been received and passed the filter while the FIFO 0 is full. It is cleared by software by writing 1.
Bit 3	RF0FF	0x0	rw1c	Receive FIFO 0 full flag 0: Receive FIFO 0 is not full 1: Receive FIFO 0 is full Note: This bit is set by hardware when three messages are pending in the FIFO 0. It is cleared by software by writing 1.
Bit 2	Reserved	0x0	resd	Kept at its default value.
Bit 1:0	RF0MN	0x0	ro	Receive FIFO 0 message num Note: These two bits indicate how many messages are pending in the FIFO 0. RF0ML bit is incremented by one each time a new message has been received and passed the fitler while the FIFO 0 is not full. RF0ML bit is decremented by one each time the software releases the receive FIFO 0 by writing 1 to the RF0R bit.

CAN receive FIFO 1 register (CAN_RF1)

Bit	Register	Reset value	Type	Description
Bit 31:6	Reserved	0x0000000	resd	Kept at its default value.
Bit 5	RF1R	0x0	rw1s	Receive FIFO 1 release 0: No effect 1: Release FIFO Note: This bit is set by software to release FIFO 1. It is cleared by hardware when the FIFO 1 is released. Seting this bit by software has no effect when the FIFO 1 is empty. If there are more than two messages pending in the FIFO 0, the software has to release the FIFO 1 to access the second message.
Bit 4	RF1OF	0x0	rw1c	Receive FIFO 1 overflow flag 0: No overflow 1: Receive FIFO 1 overflow Note: This bit is set by hardware when a new message has been received and passed the filter while the FIFO 1 is full. It is cleared by software by writing 1.
Bit 3	RF1FF	0x0	rw1c	Receive FIFO 1 full flag 0: Receive FIFO 1 is not full 1: Receive FIFO 1 is full Note: This bit is set by hardware when three messages are pending in the FIFO 1. It is cleared by software by writing 1.
Bit 2	Reserved	0x0	resd	Kept at its default value.
Bit 1:0	RF1MN	0x0	ro	Receive FIFO 1 message num Note: These two bits indicate how many messages are pending in the FIFO 1. RF1ML bit is incremented by one each time a new message has been received and passed the fitler while the FIFO 1 is not full. RF1ML bit is decremented by one each time the software releases the receive FIFO 1 by writing 1 to the RF1R bit.

CAN interrupt enable register (CAN_INTEN)

Bit	Register	Reset value	Type	Description
Bit 31:18	Reserved	0x0000	resd	Kept at its default value.
Bit 17	EDZIEN	0x0	rw	Enter doze mode interrupt enable 0: Enter sleep mode interrupt disabled 1: Enter sleep mode interrupt enabled Note: EDZIF flag bit corresponds to this interrupt. An interrupt is generated when both this bit and EDZIF bit are set.
Bit 16	QDZIEN	0x0	rw	Quit doze mode interrupt enable 0: Quit sleep mode interrupt disabled 1: Quit sleep mode interrupt enabled Note: The flag bit of this interrupt is the QDZIF bit. An interrupt is generated when both this bit and QDZIF bit are set.
Bit 15	EOIEN	0x0	rw	Error occur interrupt enable 0: Error interrupt disabled 1: Error interrupt enabled Note: The flag bit of this interrupt is the EOIF bit. An interrupt is generated when both this bit and EOIF bit are set.
Bit 14:12	Reserved	0x0	resd	Kept at its default value.
Bit 11	ETRIEN	0x0	rw	Error type record interrupt enable 0: Error type record interrupt disabled 1: Error type record interrupt enabled Note: EOIF is set only when this interrupt is enabled and the ETR[2:0] is set by hardware.
Bit 10	BOIEN	0x0	rw	Bus-off interrupt enable 0: Bus-off interrupt disabled 1: Bus-off interrupt enabled Note: EOIF is set only when this interrupt is enabled and the BOF is set by hardware.
Bit 9	EPIEN	0x0	rw	Error passive interrupt enable 0: Error passive interrupt disabled 1: Error passive interrupt enabled Note: EOIF is set only when this interrupt is enabled and the EPF is set by hardware.
Bit 8	EAIEN	0x0	rw	Error active interrupt enable 0: Error warning interrupt disabled 1: Error warning interrupt enabled Note: EOIF is set only when this interrupt is enabled and the EAF is set by hardware.
Bit 7	Reserved	0x0	resd	Kept at its default value.
Bit 6	RF1OIEN	0x0	rw	Receive FIFO 1 overflow interrupt enable 0: Receive FIFO 1 overflow interrupt disabled 1: Receive FIFO 1 overflow interrupt enabled Note: The flag bit of this interrupt is the RF1OF bit. An interrupt is generated when this bit and RF1OF bit are set.

Bit	Register	Reset value	Type	Description
Bit 5	RF1FIEN	0x0	rw	Receive FIFO 1 full interrupt enable 0: Receive FIFO 1 full interrupt disabled 1: Receive FIFO 1 full interrupt enabled Note: The flag bit of this interrupt is the RF1FF bit. An interrupt is generated when this bit and RF1FF bit are set.
Bit 4	RF1MIEN	0x0	rw	FIFO 1 receive message interrupt enable 0: FIFO 1 receive message interrupt disabled 1: FIFO 1 receive message interrupt enabled Note: The flag bit of this interrupt is RF1MN bit, so an interrupt is generated when this bit and RF1MN bit are set.
Bit 3	RF0OIEN	0x0	rw	Receive FIFO 0 overflow interrupt enable 0: Receive FIFO 0 overflow interrupt disabled 1: Receive FIFO 0 overflow interrupt enabled Note: The flag bit of this interrupt is RF0OF bit, so an interrupt is generated when this bit and RF0OF bit are set.
Bit 2	RF0FIEN	0x0	rw	Receive FIFO 0 full interrupt enable 0: Receive FIFO 0 full interrupt disabled 1: Receive FIFO 0 full interrupt enabled Note: The flag bit of this interrupt is the RF0FF bit. An interrupt is generated when this bit and RF0FF bit are set
Bit 1	RF0MIEN	0x0	rw	FIFO 0 receive message interrupt enable 0: FIFO 0 receive message interrupt disabled 1: FIFO 0 receive message interrupt enabled Note: The flag bit of this interrupt is the RF0MN bit. An interrupt is generated when this bit and RF0MN bit are set
Bit 0	TCIEN	0x0	rw	Transmit mailbox empty interrupt enable 0: Transmit mailbox empty interrupt disabled 1: Transmit mailbox empty interrupt enabled Note: The flag bit of this interrupt is the TMxTCF bit. An interrupt is generated when this bit and TMxTCF bit are set

CAN error status register (CAN_ESTS)

Bit	Register	Reset value	Type	Description
Bit 31:24	REC	0x00	ro	Receive error counter This counter is implemented in accordance with the receive part of the falut confinement mechanism of the CAN protocol.
Bit 23:16	TEC	0x00	ro	Transmit error counter This counter is implemented in accordance with the transmit part of the falut confinement mechanism of the CAN protocol.
Bit 15:7	Reserved	0x00	resd	Kept at its default value.
Bit 6:4	ETR	0x0	rw	Error type record 000: No error 001: Bit stuffing error 010: Format error 011: Acknowledgement error 100: Recessive bit error 101: Dominant bit error 110: CRC error 111: Set by software Note: This field is used to indicate the current error type. It is set by hardware according to the error condition detected on the CAN bus. It is cleared by hardware when a message has been transmitted or received successfully. If the error code 7 is not used by hardware, this field can be set by software to monitor the code update.
Bit 3	Reserved	0x0	resd	Kept at its default value.
Bit 2	BOF	0x0	ro	Bus-off flag 0: Bus-off state is not entered. 1: Bus-off state is entered. Note: When the TEC is greater than 255, the bus-off state is entered, and this bit is set by hardware.
Bit 1	EPF	0x0	ro	Error passive flag 0: Error passive state is not entered 1: Error passive state is entered Note: This bit is set by hardware when the current error times has reached the Error passive state limit (Receive Error Counter or Transmit Error Counter > 127)
Bit 0	EAF	0x0	ro	Error active flag 0: Error active state is not entered 1: Error active state is entered Note: This bit is set by hardware when the current error times has reached the Error active state limit (Receive Error Counter or Transmit Error Counter ≥ 96)

CAN bit timing register (CAN_BTMG)

Bit	Register	Reset value	Type	Description
Bit 31	LOEN	0x0	rw	Listen-Only mode 0: Listen-Only mode disabled 1: Listen-Only mode enabled
Bit 30	LBEN	0x0	rw	Loop back mode 0: Loop back mode disabled 1: Loop back mode enabled
Bit 29:26	Reserved	0x0	resd	Kept at its default value.
Bit 25:24	RSAW	0x1	rw	Resynchronization width TRSAW = TCAN × (RSAW[1:0] + 1) Note: This field defines the maximum of time unit that the CAN handware is allowed to lengthen or shorten in a bit.
Bit 23	Reserved	0x0	resd	Kept at its default value.
Bit 22:20	BTS2	0x2	rw	Bit time segment 2 TBTS2 = TCAN × (BTS2[2:0] + 1) Note: This field defines the number of time unit in Bit time segment 2.
Bit 19:16	BTS1	0x3	rw	Bit time segment 1 TBTS1 = TCAN × (BTS1[3:0] + 1) Note: This field defines the number of time unit in Bit time segment 1.
Bit 15:12	Reserved	0x0	resd	Kept at its default value.
Bit 11:0	BRDIV	0x000	rw	Baud rate division Tq = (BRDIV[11:0]+1) × TPCLK Note: This field defines the length of a time unit (Tq).

CAN mailbox registers

This section describes the registers of the transmit and receive mailboxes. Refer to Section 21.6.5 for more information on register map.

Transmit and receive mailboxes are the same except:

- RFFMN field in the CAN_RFCx register
- A receive mailbox is read only
- A transmit mailbox can be written only when empty. TMxEF=1 in the CAN_TSTS register indicates that the mailbox is empty.

There are three transmit mailboxes and two receive mailboxes. Each receive mailbox has 3-level depth of FIFO, and can only access to the first received message in the FIFO.

Each mailbox contains four registers.

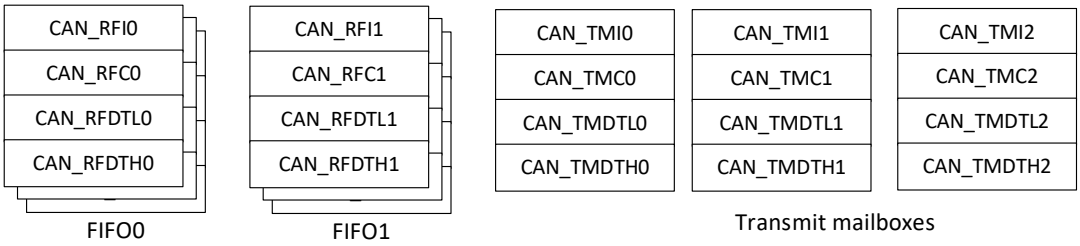


Figure 166. Transmit and receive mailboxes

Transmit mailbox identifier register (CAN_TMIx) (x=0..2)

Note: 1. This register is write protected when its mailboxes are pending for transmission.
2. This register implements the Transmit Request control (bit 0) – reset value 0.

Bit	Register	Reset value	Type	Description
Bit 31:21	TMSID/ TMEID	0xXXX	rw	Transmit mailbox standard identifier or extended identifier high bytes Note: This field defines the 11-bit high bytes of the standard identifier or extended identifier.
Bit 20: 3	TMEID	0xXXXXX	rw	Transmit mailbox extended identifier Note: This field defines the 18-bit low bytes of the extended identifier.
Bit 2	TMIDSEL	0xX	rw	Transmit mailbox identifier type select 0: Standard identifier 1: Extended identifier
Bit 1	TMFRSEL	0xX	rw	Transmit mailbox frame type select 0: Data frame 1: Remote frame
Bit 0	TMSR	0x0	rw	Transmit mailbox send request 0: No effect 1: Transmit request Note: This bit is cleared by hardware when the transmission has been completed (The mailbox becomes empty)

Transmit mailbox data length and time stamp register (CAN_TMCx) (x=0..2)

All the bits in the register are write protected when the mailbox is not in empty state.

Bit	Register	Reset value	Type	Description
Bit 31:16	TMTS	0xXXXX	rw	Transmit mailbox time stamp Note: This field contains the value of the CAN timer sampled at the SOF transmission.
Bit 15:9	Reserved	0xXX	resd	Kept at its default value
Bit 8	TMTSTEN	0xX	rw	Transmit mailbox time stamp transmit enable 0: Time stamp is not sent 1: Time stamp is sent Note: This bit is valid only when the time-triggered communication mode is enabled. In the time stamp MTS[15:0], the MTS[7:0] is stored in the TMDT7, and MTS[15:8] in the TMDT6. The data length must be programmed as 8 to send time stamp.

Bit	Register	Reset value	Type	Description
Bit 7:4	Reserved	0xX	resd	Kept at its default value
Bit 3:0	TMDTBL	0xX	rw	Transmit mailbox data byte length Note: This field defines the data length of a transmit message. A transmit message can contain from 0 to 8 data bytes.

Transmit mailbox data low register (CAN_TMDTLx) (x=0..2)

All the bits in the register are write protected when the mailbox is not in empty state.

Bit	Register	Reset value	Type	Description
Bit 31:24	TMDT3	0xXX	rw	Transmit mailbox data byte 3
Bit 23:16	TMDT2	0xXX	rw	Transmit mailbox data byte 2
Bit 15:8	TMDT1	0xXX	rw	Transmit mailbox data byte 1
Bit 7:0	TMDT0	0xXX	rw	Transmit mailbox data byte 0

Transmit mailbox data high register (CAN_TMDTHx) (x=0..2)

All the bits in the register are write protected when the mailbox is not in empty state.

Bit	Register	Reset value	Type	Description
Bit 31:24	TMDT7	0xXX	rw	Transmit mailbox data byte 7
Bit 23:16	TMDT6	0xXX	rw	Transmit mailbox data byte 6 Note: This field will be replaced with MTS[15:8] when the time-triggered communication mode is enabled and the corresponding time stamp transmit is enabled.
Bit 15:8	TMDT5	0xXX	rw	Transmit mailbox data byte 5
Bit 7:0	TMDT4	0xXX	rw	Transmit mailbox data byte 4

Receive FIFO mailbox identifier register (CAN_RFIx) (x=0..1)

Note: All the receive mailbox registers are read only.

Bit	Register	Reset value	Type	Description
Bit 31:21	RFSID / RFEID	0xXXX	ro	Receive FIFO standard identifier or receive FIFO extended identifier Note: This field defines the 11-bit high bytes of the standard identifier or extended identifier.
Bit 20: 3	RFEID	0xXXXXX	ro	Receive FIFO extended identifier Note: This field defines the 18-bit low bytes of the extended identifier.
Bit 2	RFIDI	0xX	ro	Receive FIFO identifier type indication 0: Standard identifier 1: Extended identifier
Bit 1	RFFRI	0xX	Ro	Receive FIFO frame type indication 0: Data frame 1: Remote frame
Bit 0	Reserved	0x0	resd	Kept at its default value

Receive FIFO mailbox data length and time stamp register (CAN_RFCx) (x=0..1)

Note: All the receive mailbox registers are read only.

Bit	Register	Reset value	Type	Description
Bit 31:16	RFTS	0xXXXX	ro	Receive FIFO time stamp Note: This field contains the value of the CAN timer sampled at the start of a receive frame.
Bit 15:8	RFFMN	0xXX	ro	Receive FIFO filter match number Note: This field contains the filter number that a message has passed through.
Bit 7:4	Reserved	0xX	resd	Kept at its default value
Bit 3:0	RFDTL	0xX	ro	Receive FIFO data length Note: This field defines the data length of a receive message. A transmit message can contain from 0 to 8 data bytes. For a remote frame, its data length RFDTl is fixed 0.

Receive FIFO mailbox data low register (CAN_RFDTLx) (x=0..1)

Note: All the receive mailbox registers are read only.

Bit	Register	Reset value	Type	Description
Bit 31:24	RFDT3	0xXX	ro	Receive FIFO data byte 3
Bit 23:16	RFDT2	0xXX	ro	Receive FIFO data byte 2
Bit 15:8	RFDT1	0xXX	ro	Receive FIFO data byte 1
Bit 7:0	RFDT0	0xXX	ro	Receive FIFO data byte 0

Receive FIFO mailbox data high register (CAN_RFDTHx) (x=0..1)

Note: All the receive mailbox registers are read only.

Bit	Register	Reset value	Type	Description
Bit 31:24	RFDT7	0xXX	ro	Receive FIFO data byte 7
Bit 23:16	RFDT6	0xXX	ro	Receive FIFO data byte 6
Bit 15:8	RFDT5	0xXX	ro	Receive FIFO data byte 5
Bit 7:0	RFDT4	0xXX	ro	Receive FIFO data byte 4

CAN filter registers

CAN filter control register (CAN_FCTRL)

Note: All the non-reserved bits of this register are controlled by software completely.

Bit	Register	Reset value	Type	Description
Bit 31:1	Reserved	0x160E0700	resd	Kept at its default value
Bit 0	FCS	0x1	rw	Filter configuration switch 0: Disabled (Filter bank is active) 1: Enabled (Filter bank is in configuration mode) Note: The initialization of the filter bank can be configured only when it is in configuration mode.

CAN filter mode configuration register (CAN_FMCFG)

Note: This register can be written only when FCS=1 in the CAN_FCTRL register (The filter is in configuration mode)

Bit	Register	Reset value	Type	Description
Bit 31:14	Reserved	0x00000	resd	Kept at its default value
Bit 13:0	FMSELx	0x0000	rw	Filter mode select Each bit corresponds to a filter bank. 0: Identifier mask mode 1: Identifier list mode

CAN filter bit width configuration register (CAN_ FBWCFG)

Note: This register can be written only when FCS=1 in the CAN_FCTRL register (The filter is in configuration mode)

Bit	Register	Reset value	Type	Description
Bit 31:14	Reserved	0x00000	resd	Kept at its default value
Bit 13:0	FBWSELx	0x0000	rw	Filter bit width select Each bit corresponds to a filter bank. 0: Dual 16-bit 1: Single 32-bit

CAN filter FIFO association register (CAN_ FRF)

Note: This register can be written only when FCS=1 in the CAN_FCTRL register (The filter is in configuration mode)

Bit	Register	Reset value	Type	Description
Bit 31:14	Reserved	0x00000	resd	Kept at its default value
Bit 13:0	FRFSELx	0x0000	rw	Filter relation FIFO select Each bit corresponds to a filter bank. 0: Associated with FIFO0 1: Associated with FIFO1

CAN filter activation control register (CAN_ FACFG)

Bit	Register	Reset value	Type	Description
Bit 31:14	Reserved	0x00000	resd	Kept at its default value
Bit 13:0	FAENx	0x0000	rw	Filter active enable Each bit corresponds to a filter bank. 0: Disabled 1: Enabled

CAN filter bank i filter bit register (CAN_ FiFBx) (i=0..13; x=1..2)

Note: There are 14 filter banks (i=0..13). Each filter bank consists of two 32-bit registers, CAN_ FiFB[2:1]. This register can be modified only when the FAENx bit of the CAN_FACFG register is cleared or the FCS bit of the CAN_FCTRL register is set.

Bit	Register	Reset value	Type	Description
Bit 31:0	FFDB	0xXXXX XXXX	rw	Filters filter data bit Identifier list mode: The configuration value of the register matches with the level of the corresponding bit of the data received on the bus (If it is a standard frame, the value of the corresponding bit of the extended frame is neglected.) Identifier mark mode: Only the bit with its register configuration value 1 can match with the level of the corresponding bit of the data received on the bus. It don't care when the register value is 0.

22 External memory controller

XMC introduction

XMC peripheral block can translate the AHB signals into the external memory signals and vice versa. It boasts two chip-select signals for interfacing up to external memories at a time. The supported external memories include a NAND Flash and a static memory device featuring multiplexed signals or additional address latch function. Such static memory device includes a static random memory (SRAM), NOR Flash and PSRAM.

Main features

NOR/PSRAM has the following features:

- Two chip-select signals with their own control registers
- Support access to static memory devices with multiplexed signals or additional address latch function, including:
  - Statis random access memory (SRAM)
  - NOR Flash
  - PSRAM
- 8-bit or 16-bit wide memory
- Various timing mode selection
  - Two modes with the same timings for read and write
  - Four modes with different timings for read and write
  - Multiplexed address/data mode
  - Synchronous mode
- Programmable timing control registers
- Translate the AHB data size into the appropriate external memory data size

NAND has the following features:

- A chip-select signal
- 8-bit or 16-bit wide NAND Flash
- Two storage sapce with programmable timing control registers
- Translate the AHB data size into the appropriate external memory data size
- Support ECC calculation

XMC architecture

Block diagram

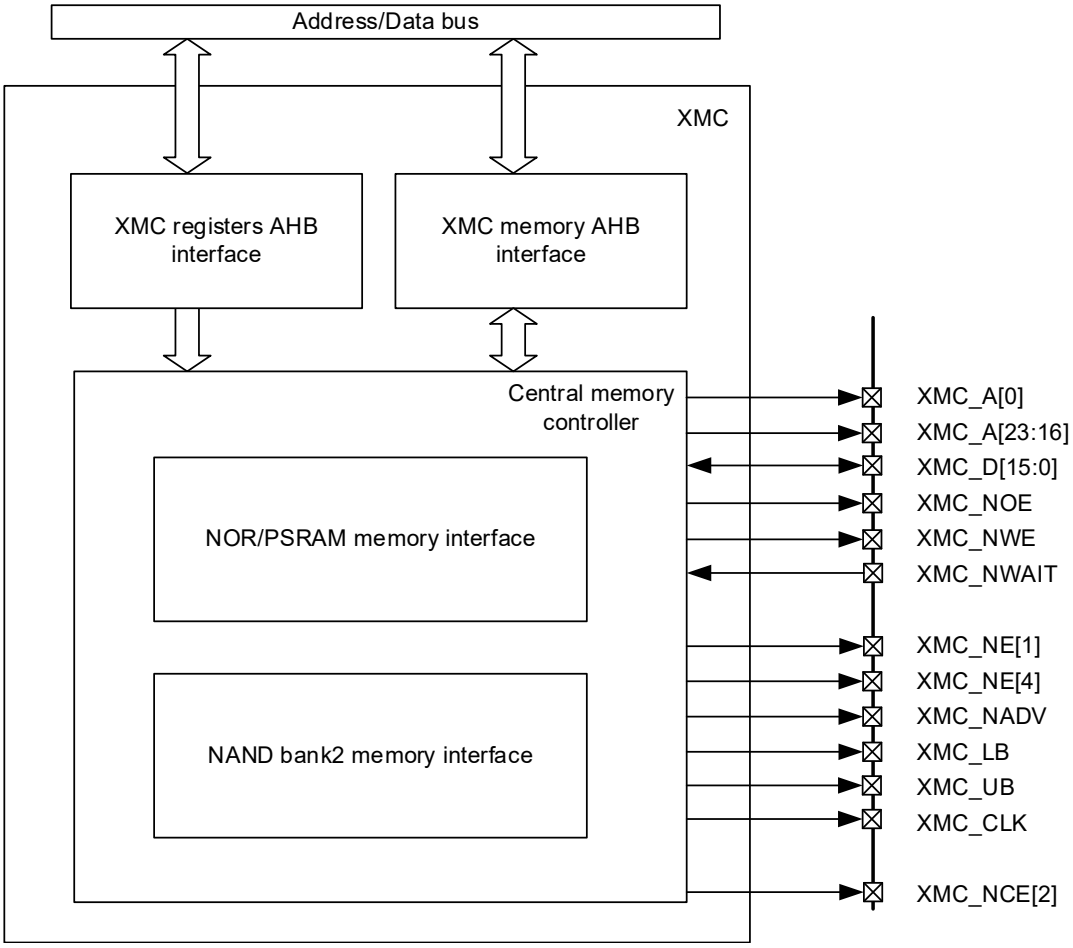


Figure 167. XMC block diagram

While interfacing to the external memory, NOR/PSRAM uses different pins from that of NAND, as shown in the following tables.

Table 73. NOR/PSRAM pins

Pin name	I/O	Description
XMC_CLK	Output	Clock
XMC_NE[x], x=1,4	Output	Chip select
XMC_NADV	Output	Address latch or address valid (NL) signal
XMC_A[x]	Output	Address but
XMC_NOE	Output	Output enable signal
XMC_NWE	Output	Write enable signal
XMC_LB, XMC_UB	Output	Byte select signal
XMC_D[15:0]	Read input/write output	Data bus/multiplexed address data
XMC_NWAIT	Input	Wait signal

Table 74. NAND pins

Pin name	I/O	Description
XMC_NCE[2]	Output	Chip select
XMC_A[17]	Output	Address latch (ALE) signal
XMC_A[16]	Output	Command latch (CLE) signal
XMC_NOE	Output	Output enable (NRE) signal
XMC_NWE	Output	Write enable signal
XMC_D[15:0]	Read input/write output	Data bus
XMC_NWAIT	Input	Ready/Busy

Address mapping

XMC address is divided into multiple memory banks, as shown below.

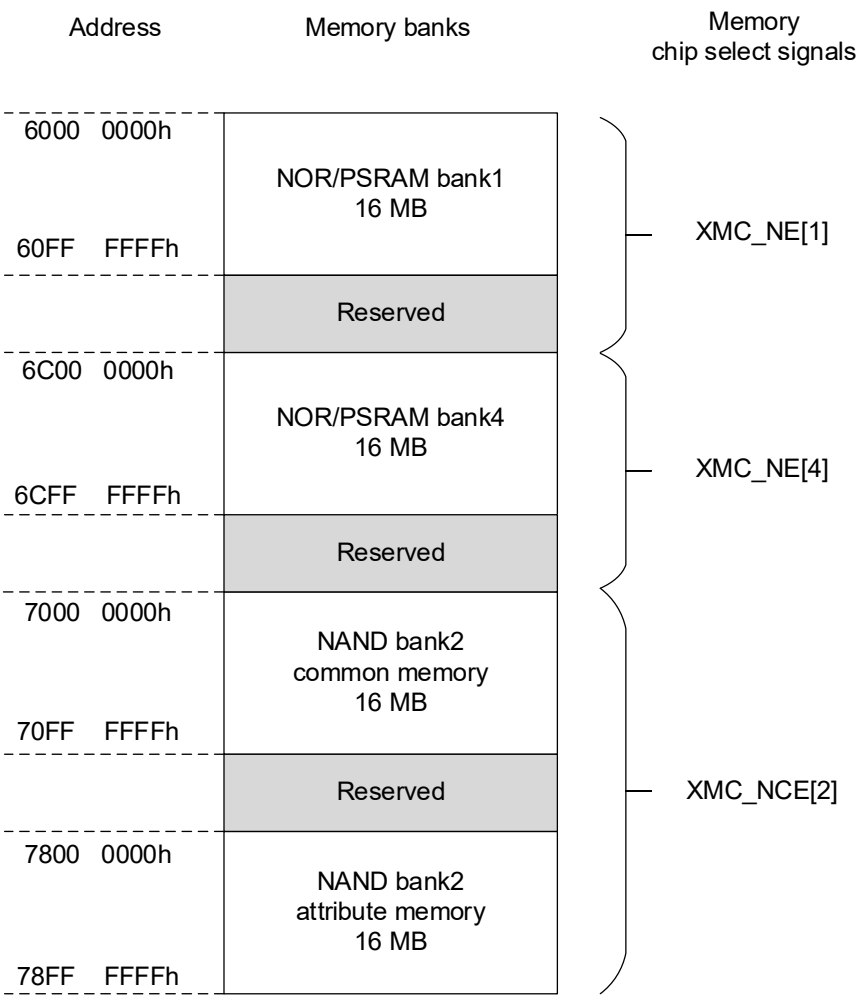


Figure 168. XMC memory banks

Some HADDR bits are used to select which bank to access, as shown in the following table.

Table 75. Memory bank selection

HADDR[31:28]	HADDR[27:26]	
0110: NOR/PSRAM	00: bank1	
	11: bank4	
HADDR[31:28]	HADDR[27]	HADDR[17:16]
0111: NAND bank2	0: Regular space	00: Data area
		01: Command area
		1x: Address area
	1:Special space	00: Data area
		01: Command area
		1x: Address area

NOR/PSRAM

NOR/PSRAM offers multiple access modes with different timings to drive multiple memories including NOR Flash, SRAM, PSRAM and Cellular RAM.

There are two banks, bank 1 and bank, with independent control registers. Such two banks can be accessed by means of different timings and different chip-select signals.

Operation mode

Pin function

Pin signals vary from external memory to external memory. The following table lists typical pin signals.

Table 76. Pin signals for NOR and PSRAM

XMC pin name	NOR Flash	PSRAM
XMC_CLK	Clock (synchronous mode)	Clock (synchronous mode)
XMC_NE[x]	Chip-select	Chip-select
XMC_NADV	Address latch or address valid	Address latch or address valid
XMC_A[23:16], XMC_A[0]	Address bus	Address bus
XMC_NOE	Output enable	Output enable
XMC_NWE	Write enable	Write enable
XMC_LB, XMC_UB	Without using XMC_LB, XMC_UB	XMC_LB: lower byte XMC_UB: upper byte
XMC_D[15:0]	Data bus multiplexed address data bus (multiplex and synchronous mode)	Data bus multiplexed address data bus (multiplex and synchronous mode)
XMC_NWAIT	NOR Flash wait request	PSRAM wait request

Note: If the memory data size is 8-bit, the typical data bus is XMC_D[7:0].

Access address

The upper bytes of the HADDR bit is used to select a memory bank while the lower bytes to data memory address. HADDR is a byte address whereas the XMC supports the memory addressed in words or half words. Address translation between them is shown in the following table. As long as read/write access to a specific address occurs, the XMC will enable chip-select signals and write/read operation to the external memories according to the HADDR bit.

Table 77. Address translation between HADDR and external memory

External memory data width	Address connection	Accessible maximm memory space (bits)
8-bit	HADDR[23:16] is linked to XMC_A[23:16]. HADDR[0] is connected to XMC_A[0]. In multiplexed and synchronous mode, HADDR[15:0] is connected to XMC_D[15:0] during address latch period.	16 Mbyte×8=128 Mbit
16-bit	HADDR[23:17] is connected to XMC_A[22:16]. HADDR[1] is connected to XMC_A[0]. In multiplexed and synchronous mode, HADDR[16:1] is connected to XMC_D[15:0] during address latch period	(16 Mbyte×16)/2=128 Mbit

Data access

In case that the AHB data width is not equal to that of the memories, the XMC will make appropriate arrangement according to the typical signals of the external memories. The following table lists the operation modes supported by XMC.

Table 78. Data access width vs. external memory data width

Memory	Mode	AHB data width	Memory width	Description
SRAM	Asynchronous read/write	8/16/32	8	One-time access, or split into two or four accesses
	Asynchronous read/write	8/16/32	16	XMC_LB and XMC_UB, One-time, or split into two access
NOR Flash	Asynchronous read	8	16	
	Asynchronous read/write	16	16	
	Asynchronous read/write	32	16	Split into two XMC accesses
	Synchronous read	16	16	
	Synchronous read	32	16	Split into two XMC accesses
PSRAM	Asynchronous read	8	16	
	Asynchronous write	8	16	Use XMC_LB and XMC_UB
	Asynchronous read/write	16	16	
	Asynchronous read/write	32	16	Split into two XMC accesses
	Synchronous write	8	16	Use XMC_LB and XMC_UB
	Synchronous read/write	16	16	
	Synchronous read/write	32	16	Split into two XMC accesses

Access mode

The XMC offers various access modes. Each access mode is operated based on the timing parameters, as shown in the following table. Users can perfrom programming operations according to the specifications of the external memory and application needs.

Access modes available in the XMC:

- Read/write operation with the same timings: Mode 1 and Mode 2
- Read/write operation with different timings: Mode A, B, C and D
- Multiplexed address data lines
- Clock-based synchronous mode

Table 79. NOR/PSRAM parameter registers

Parameter register	Function	Access mode	Unit
ADDRST	Address set-up time	1, 2, A, B, C, D and multiplexed	HCLK cycle
ADDRHT	Addess-hold time	D and multiplexed	HCLK cycle
DTST	Data set-up time	1, 2, A, B, C, D and multiplexed	HCLK cycle
DTLAT	Data latency time	Synchronous	XMC_CLK cycle
CLKPSC	Clock prescaler	Synchronous	HCLK cycle

In addition to timing parameter registers for timing control, if the wait enable bit (NWASEN or NWSEN) is enabled, the XMC will start to check whether the XMC_NWAIT signal is in wait request state during data set time. If so, the XMC will wait until the XMC_NWAIT returns to the ready state before data transfer.

Read/write operation with the same timings

The timing of read and write operation in mode 1 and mode 2 is based on the XMC_BK1TMG register configuration.

Mode 1

As configured in the following tables, the XMC uses mode 1 to access the external memory. The timing of read operation is shown in Figure 169. The timing of write operation is shown in Figure 170.

Table 80. Mode 1 – SRAM/NOR Flash chip select control register (XMC_BK1CTRL)

Bit	Description	Configuration
Bit 31:20	Reserved	0x0
Bit 19	MWMC: Memory write mode control	0x0
Bit 18:16	CRPGS: CRAM page size	0x0
Bit 15	NWASEN: NWAIT in asynchronous transfer enable	Configure according to memory specifications
Bit 14	RWTD: Read-write timing different	0x0
Bit 13	NWSEN: NWAIT in synchronous transfer enable	0x0
Bit 12	WEN: Write enable	Configure according to needs
Bit 11	NWTCFG: NWAIT timing configuration	0x0
Bit 10	WRAPEN: Wrapped enable	0x0
Bit 9	NWPOL: NWAIT polarity	Configure according to memory specifications
Bit 8	SYNCBEN: Synchronous burst enable	0x0
Bit 7	Reserved	0x1
Bit 6	NOREN: NOR flash access enable	0x0
Bit 5:4	EXTMDBW: External memory data bus width	Configure according to memory specifications

Bit	Description	Configuration
Bit 3:2	DEV: Memory device type	Configure according to memory specifications. It is valid except 0x2 (NOR Flash)
Bit 1	ADMUXEN: Address/data multiplexing enable	0x0
Bit 0	EN: Memory bank enable	0x1

Table 81. Mode 1 – SRAM/NOR Flash chip select timing register (XMC_ BK1TMG)

Bit	Description	Configuration
Bit 31:30	Reserved	0x0
Bit 29:28	ASYNCM: Asynchronous mode	0x0
Bit 27:24	DTLAT: Data latency	0x0
Bit 23:20	CLKPSC: Clock prescale	0x0
Bit 19:16	BUSLAT: Bus latency	Indicates the time the XMC_NE[x] from the rising edge to the falling edge. Configure according to needs and memory specifications
Bit 15:8	DTST: Data setup time	Refer to the following diagrams. Configure according to needs and memory specifications.
Bit 7:4	ADDRHT: Address-hold time	0x0
Bit 3:0	ADDRST: Address setup time	Refer to the following diagrams. Configure according to needs and memory specifications.

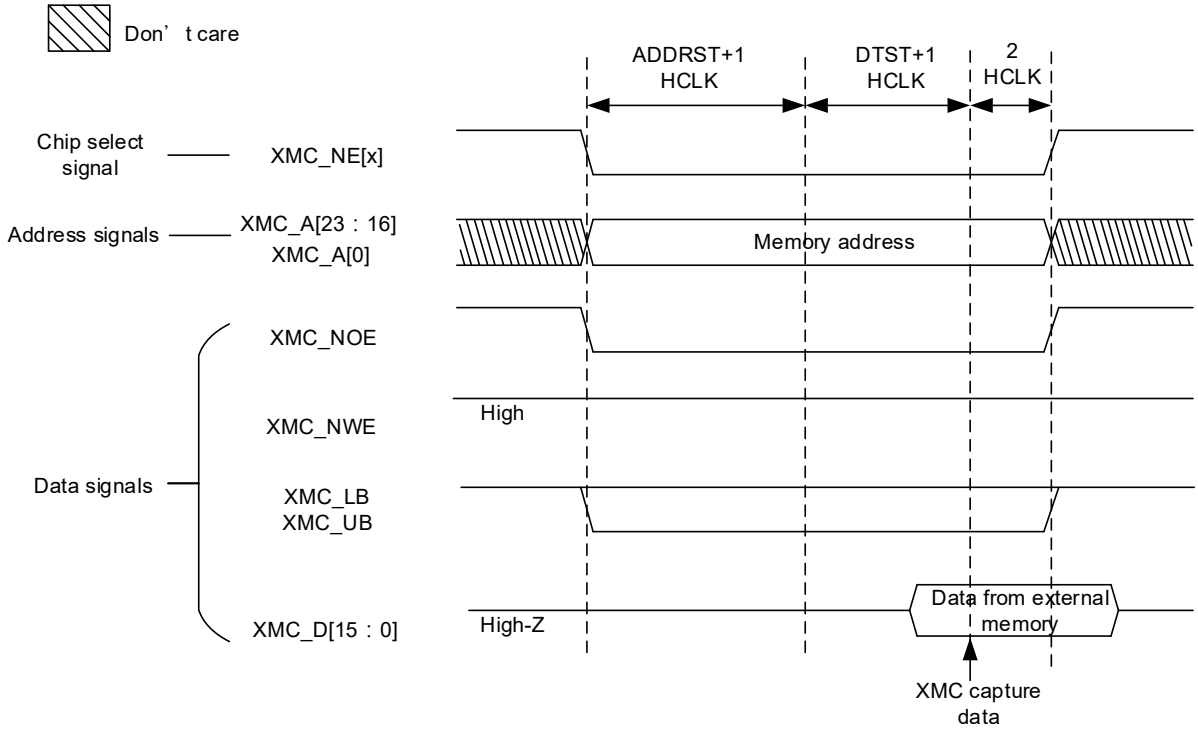


Figure 169. NOR/PSARM mode 1 read access

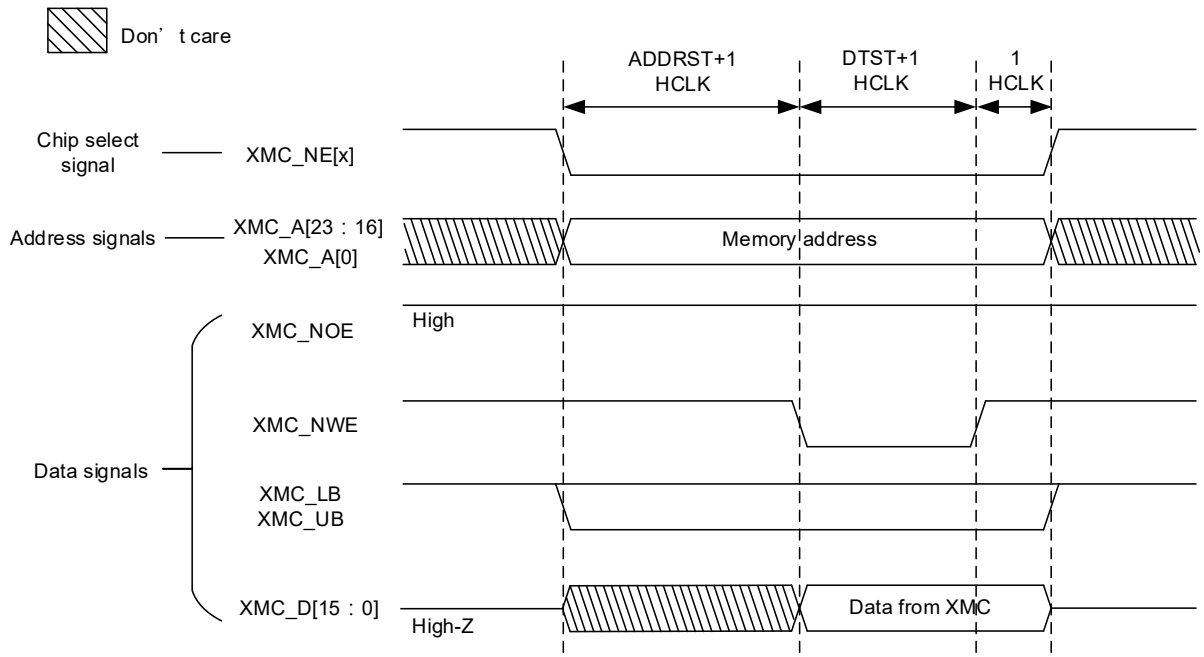


Figure 170. NOR/PSARM mode 1 write access

Mode 2

As configured in the following tables, the XMC uses mode 2 to access the external memory. The timing of read operation is shown in Figure 171. The timing of write operation is shown in Figure 172.

Table 82. Mode 2 – SRAM/NOR Flash chip select control register (XMC_BK1CTRL)

Bit	Description	Configuration
Bit 31:20	Reserved	0x0
Bit 19	MWMC: Memory write mode control	0x0
Bit 18:16	CRPGS:CRAM page size	0x0
Bit 15	NWASEN: NWAIT in asynchronous transfer enable	Configure according to memory specifications.
Bit 14	RWTD: Read-write timing different	0x0
Bit 13	NWSEN: NWAIT in synchronous transfer enable	0x0
Bit 12	WEN: Write enable	Configure according to needs.
Bit 11	NWTCFG: NWAIT timing configuration	0x0
Bit 10	WRAPEN: Wrapped enable	0x0
Bit 9	NWPOL: NWAIT polarity	Configure according to memory specifications.
Bit 8	SYNCBEN: Synchronous burst enable	0x0
Bit 7	Reserved	0x1
Bit 6	NOREN:NOR Flash access enable	0x1
Bit 5:4	EXTMDBW: External memory data bus width	Configure according to memory specifications.
Bit 3:2	DEV: Memory device type	0x2 (NOR Flash)

Bit	Description	Configuration
Bit 1	ADMUXEN: Address/data multiplexing enable	0x0
Bit 0	EN: Memory bank enable	0x1

Table 83. Mode 2 – SRAM/NOR Flash chip select timing register (XMC_BK1TMG)

Bit	Description	Configuration
Bit 31:30	Reserved	0x0
Bit 29:28	ASYNCM: Asynchronous mode	0x0
Bit 27:24	DTLAT: Data latency	0x0
Bit 23:20	CLKPSC: Clock prescale	0x0
Bit 19:16	BUSLAT: Bus latency	Indicates the time the XMC_NE[x] from the rising edge to the falling edge. Configure according to needs and memory specifications
Bit 15:8	DTST: Data setup time	Refer to the following diagrams. Configure according to needs and memory specifications.
Bit 7:4	ADDRHT: Address-hold time	0x0
Bit 3:0	ADDRST: Address setup time	Refer to the following diagrams. Configure according to needs and memory specifications.

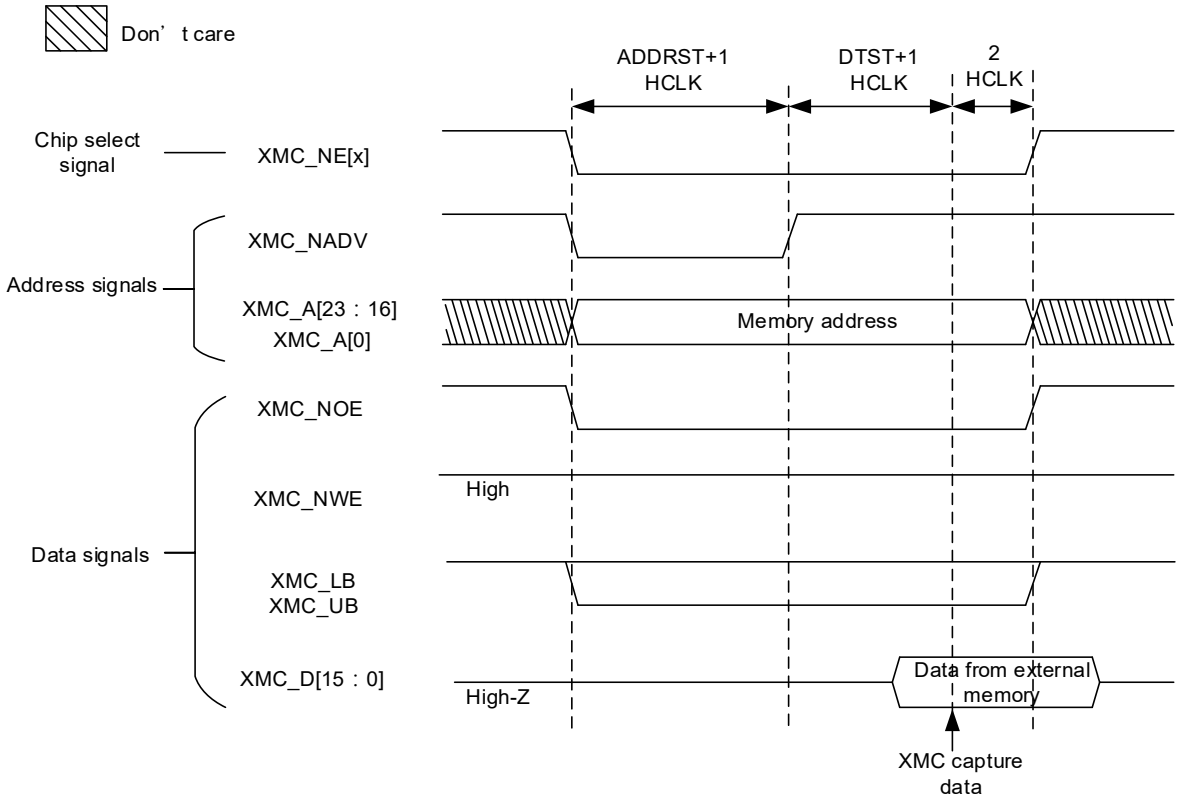


Figure 171. NOR/PSARM mode 2 read access

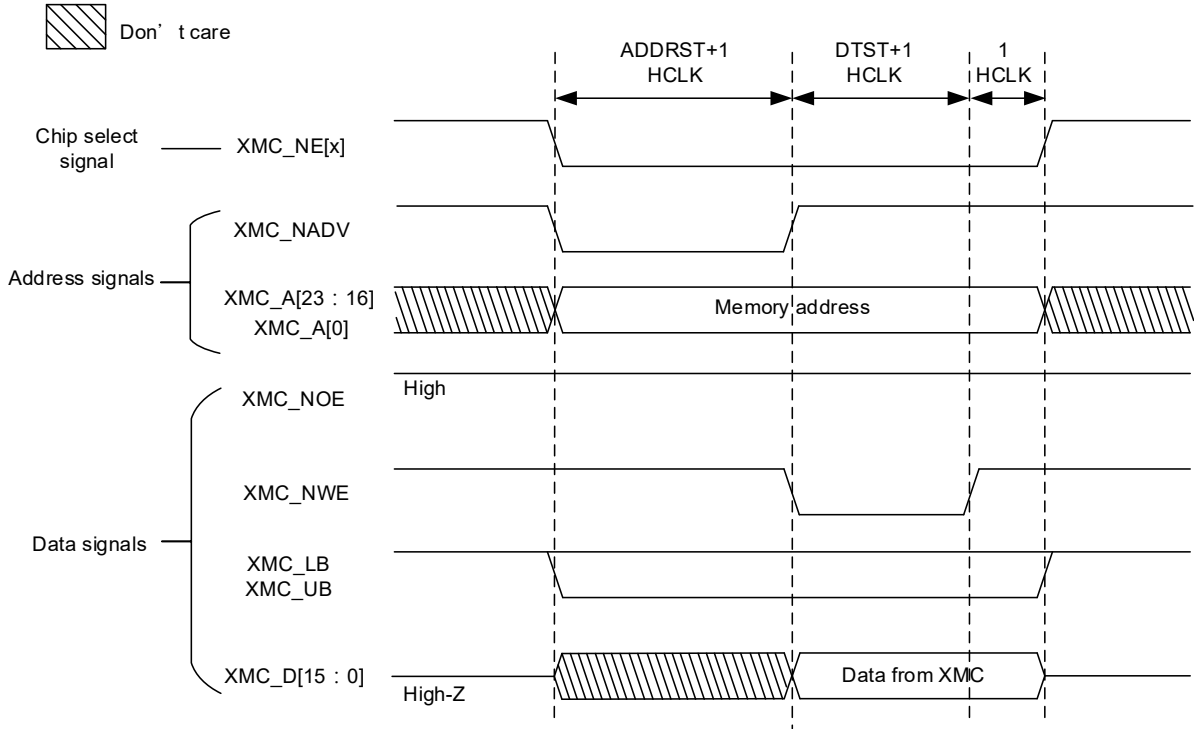


Figure 172. NOR/PSARM mode 2 write access

Read/write operation with different timings

The timing of read operation in mode A/B/C/D is based on the SRAM/NOR Flash chip select timing register (XMC_BK1TMG). The timing of write operation is based on SRAM/NOR Flash write timing register (XMC_BK1TMGWR). In addition to this, it is possible to mix A, B, C and D modes for read and write opeartions.

Mode A

As configured in the following tables, the XMC uses mode A to access the external memory. The timing of read operation is shown in Figure 173. The timing of write operation is shown in Figure 174.

Table 84. Mode A – SRAM/NOR Flash chip select control register (XMC_BK1CTRL)

Bit	Description	Configuration
Bit 31:20	Reserved	0x0
Bit 19	MWMC: Memory write mode control	0x0
Bit 18:16	CRPGS:CRAM page size	0x0
Bit 15	NWASEN: NWAIT in asynchronous transfer enable	Configure according to memory specifications.
Bit 14	RWTD: Read-write timing different	0x1
Bit 13	NWSEN: NWAIT in synchronous transfer enable	0x0
Bit 12	WEN: Write enable	Configure according to needs.
Bit 11	NWTCFG: NWAIT timing configuration	0x0

Bit	Description	Configuration
Bit 10	WRAPEN: Wrapped enable	0x0
Bit 9	NWPOL: NWAIT polarity	Configure according to memory specifications.
Bit 8	SYNCBEN: Synchronous burst enable	0x0
Bit 7	Reserved	0x1
Bit 6	NOREN:NOR Flash access enable	0x0
Bit 5:4	EXTMDBW: External memory data bus width	Configure according to memory specifications.
Bit 3:2	DEV: Memory device type	Configure according to memory specifications. It is valid except 0x2 (NOR Flash)
Bit 1	ADMUXEN: Address/data multiplexing enable	0x0
Bit 0	EN: Memory bank enable	0x1

Table 85. Mode A – SRAM/NOR Flash chip select timing register (XMC_BK1TMG)

Bit	Description	Configuration
Bit 31:30	Reserved	0x0
Bit 29:28	ASYNCM: Asynchronous mode	0x0 (Mode A)
Bit 27:24	DTLAT: Data latency	0x0
Bit 23:20	CLKPSC: Clock prescale	0x0
Bit 19:16	BUSLAT: Bus latency	Indicates the time the XMC_NE[x] from the rising edge to the falling edge. Configure according to needs and memory specifications
Bit 15:8	DTST: Data setup time	Refer to Figure 173. Configure according to needs and memory specifications.
Bit 7:4	ADDRHT: Address-hold time	0x0
Bit 3:0	ADDRST: Address setup time	Refer to Figure 173. Configure according to needs and memory specifications.

Table 86. Mode A – SRAM/NOR Flash write timing register (XMC_BK1TMGWR)

Bit	Description	Configuration
Bit 31:30	Reserved	0x0
Bit 29:28	ASYNCM: Asynchronous mode	0x0 (Mode A)
Bit 27:20	Reserved	0x0
Bit 19:16	BUSLAT: Bus latency	Indicates the time the XMC_NE[x] from the rising edge to the falling edge. Configure according to needs and memory specifications
Bit 15:8	DTST: Data setup time	Refer to Figure 174. Configure according to needs and memory specifications.
Bit 7:4	ADDRHT: Address-hold time	0x0
Bit 3:0	ADDRST: Address setup time	Refer to Figure 174. Configure according to needs and memory specifications.

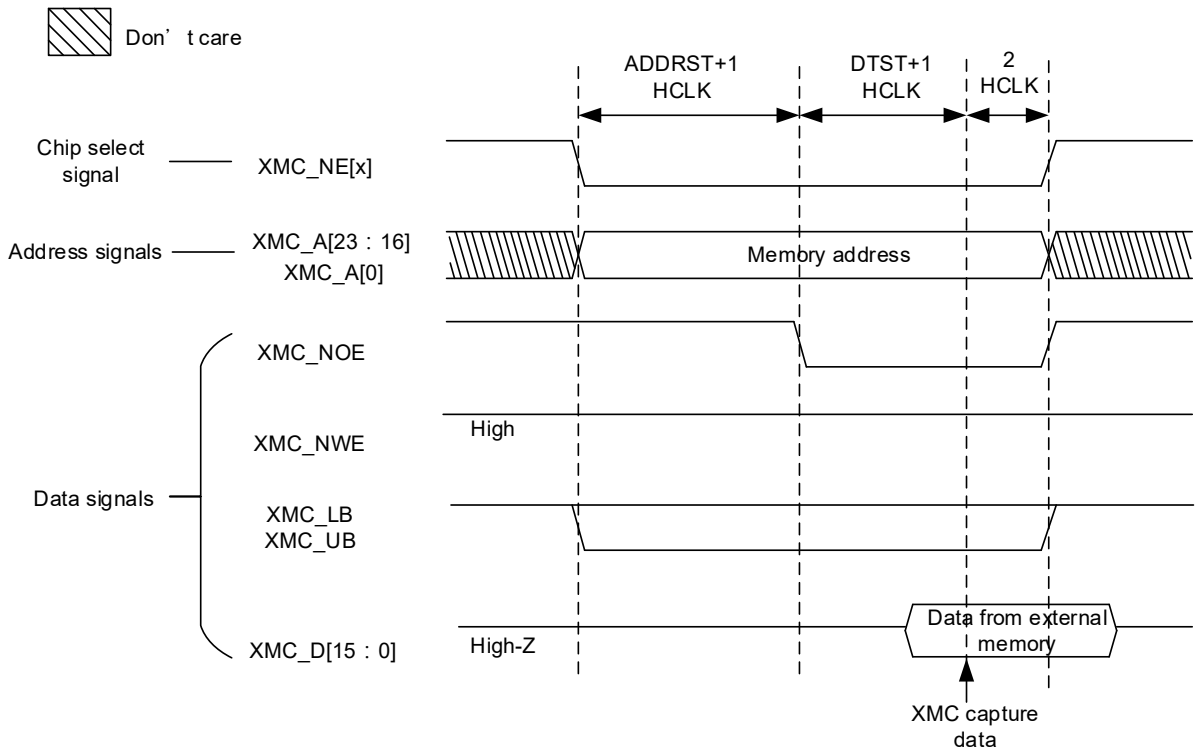


Figure 173. NOR/PSARM mode A read access

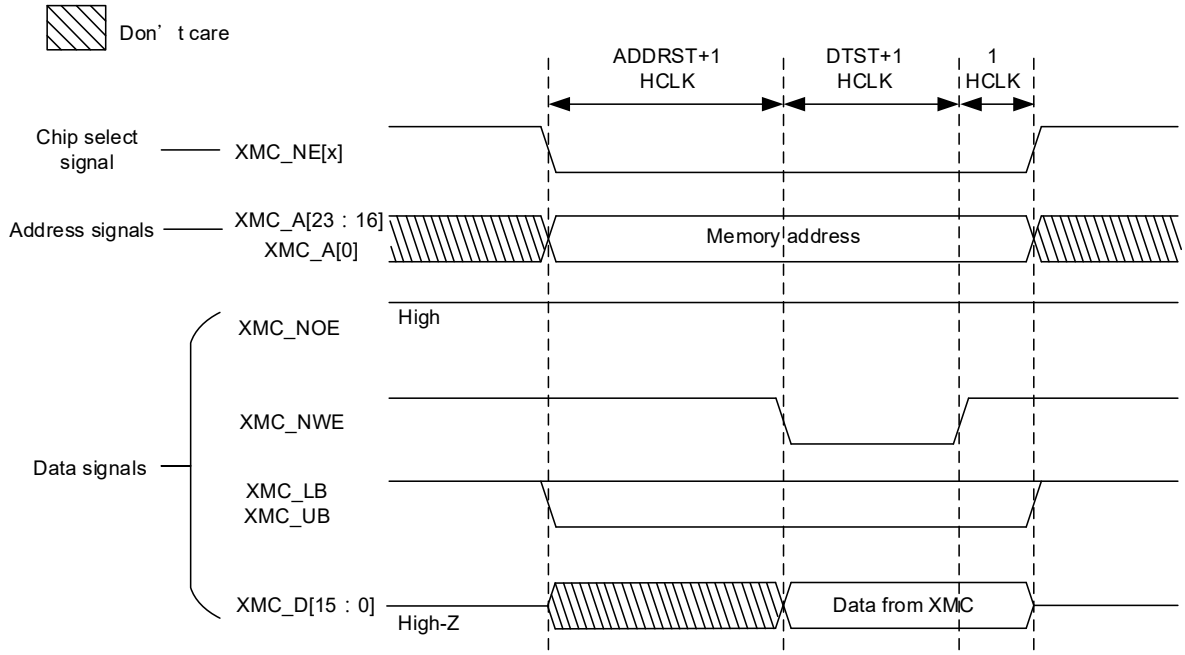


Figure 174. NOR/PSARM mode A write access

Mode B

As configured in the following tables, the XMC uses mode B to access the external memory. The timing of read operation is shown in Figure 175. The timing of write operation is shown in Figure 176.

Table 87. Mode B – SRAM/NOR Flash chip select register (XMC_BK1CTRL)

Bit	Description	Configuration
Bit 31:20	Reserved	0x0
Bit 19	MWMC: Memory write mode control	0x0
Bit 18:16	CRPGS:CRAM page size	0x0
Bit 15	NWASEN: NWAIT in asynchronous transfer enable	Configure according to memory specifications.
Bit 14	RWTD: Read-write timing different	0x1
Bit 13	NWSEN: NWAIT in synchronous transfer enable	0x0
Bit 12	WEN: Write enable	Configure according to needs.
Bit 11	NWTCFG: NWAIT timing configuration	0x0
Bit 10	WRAPEN: Wrapped enable	0x0
Bit 9	NWPOL: NWAIT polarity	Configure according to memory specifications.
Bit 8	SYNCBEN: Synchronous burst enable	0x0
Bit 7	Reserved	0x1
Bit 6	NOREN:NOR Flash access enable	0x1
Bit 5:4	EXTMDBW: External memory data bus width	Configure according to memory specifications.
Bit 3:2	DEV: Memory device type	0x2 (NOR Flash)
Bit 1	ADMUXEN: Address/data multiplexing enable	0x0
Bit 0	EN: Memory bank enable	0x1

Table 88. Mode B – SRAM/NOR Flash chip select timing register (XMC_BK1TMG)

Bit	Description	Configuration
Bit 31:30	Reserved	0x0
Bit 29:28	ASYNCM: Asynchronous mode	0x1 (Mode B)
Bit 27:24	DTLAT: Data latency	0x0
Bit 23:20	CLKPSC: Clock prescale	0x0
Bit 19:16	BUSLAT: Bus latency	Indicates the time the XMC_NE[x] from the rising edge to the falling edge. Configure according to needs and memory specifications
Bit 15:8	DTST: Data setup time	Refer to Figure 175. Configure according to needs and memory specifications.
Bit 7:4	ADDRHT: Address-hold time	0x0
Bit 3:0	ADDRST: Address setup time	Refer to Figure 175. Configure according to needs and memory specifications.

Table 89. Mode B – SRAM/NOR Flash write timing register (XMC_BK1TMGWR)

Bit	Description	Configuration
Bit 31:30	Reserved	0x0
Bit 29:28	ASYNCM: Asynchronous mode	0x1 (Mode B)
Bit 27:20	Reserved	0x0
Bit 19:16	BUSLAT: Bus latency	Indicates the time the XMC_NE[x] from the rising edge to the falling edge. Configure according to needs and memory specifications
Bit 15:8	DTST: Data setup time	Refer to Figure 176.Configure according to needs and memory specifications.
Bit 7:4	ADDRHT: Address-hold time	0x0
Bit 3:0	ADDRST: Address setup time	Refer to Figure 176. Configure according to needs and memory specifications.

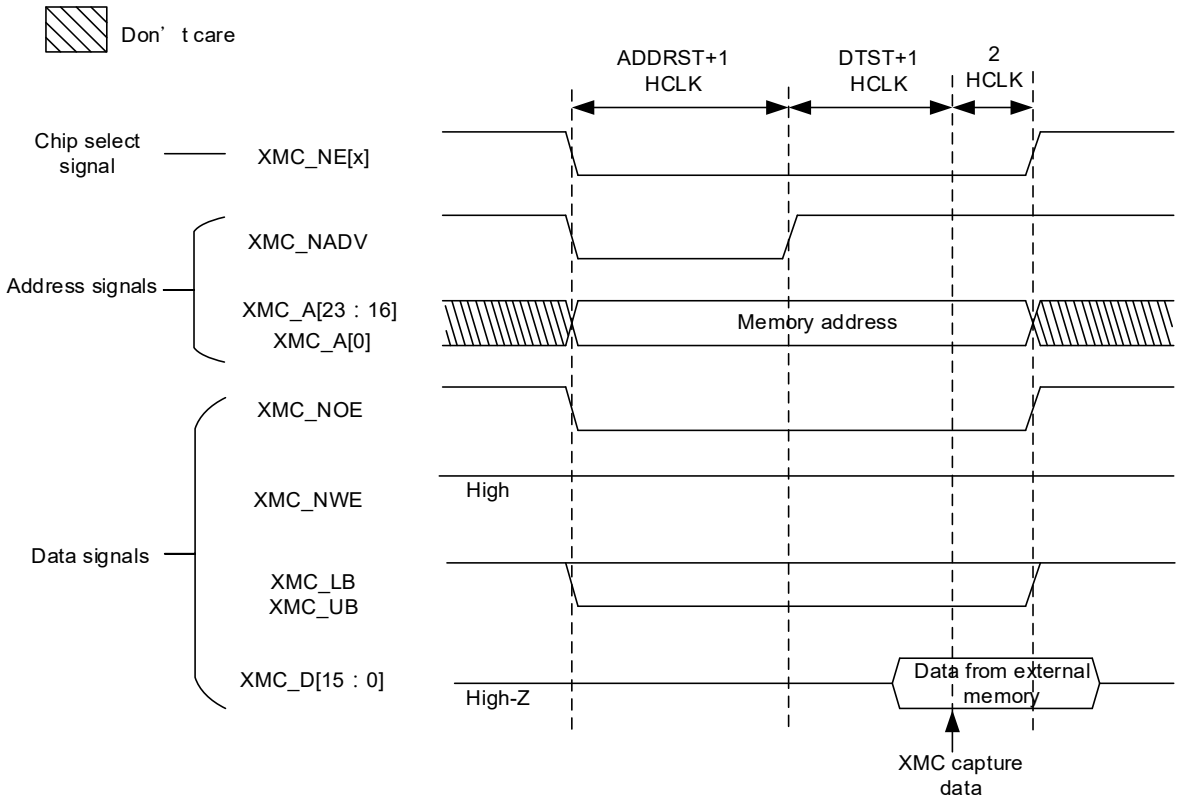


Figure 175. NOR/PSARM mode B read access

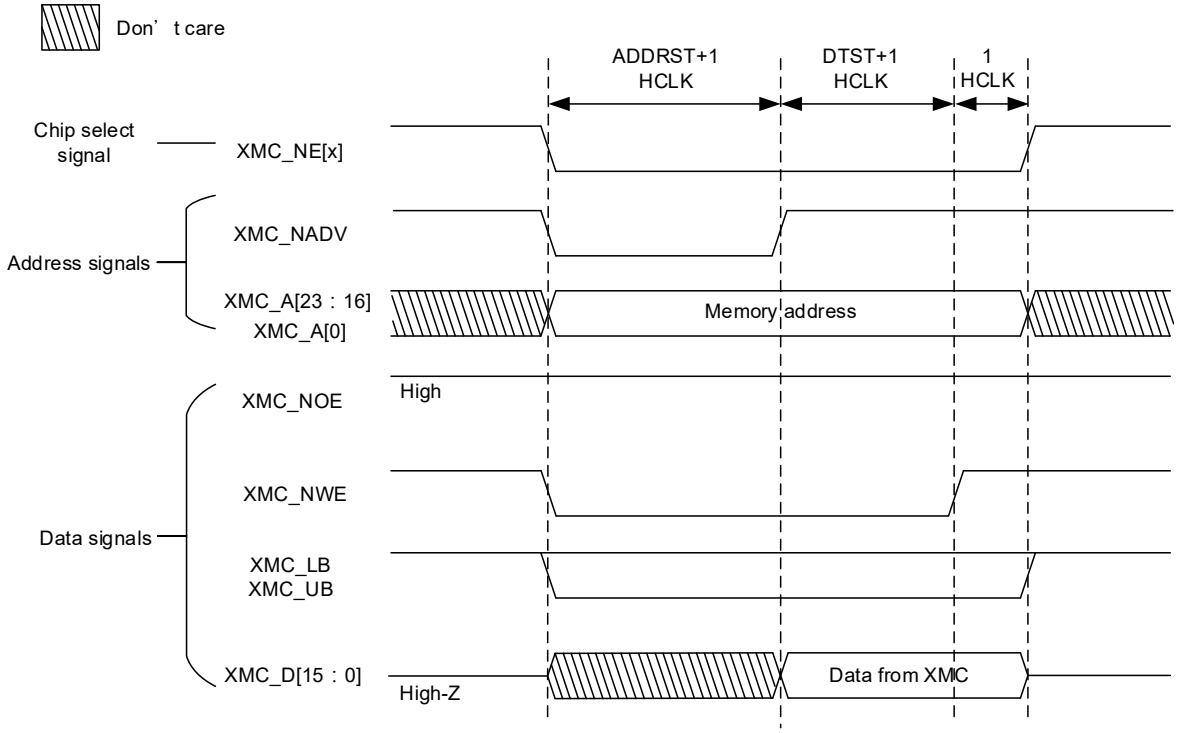


Figure 176. NOR/PSARM mode B write access

Mode C

As configured in the following tables, the XMC uses mode C to access the external memory. The timing of read operation is shown in Figure 177. The timing of write operation is shown in Figure 178.

Table 90. Mode C – SRAM/NOR Flash chip select register (XMC_BK1CTRL)

Bit	Description	Configuration
Bit 31:20	Reserved	0x0
Bit 19	MWMC: Memory write mode control	0x0
Bit 18:16	CRPGS:CRAM page size	0x0
Bit 15	NWASEN: NWAIT in asynchronous transfer enable	Configure according to memory specifications.
Bit 14	RWTD: Read-write timing different	0x1
Bit 13	NWSEN: NWAIT in synchronous transfer enable	0x0
Bit 12	WEN: Write enable	Configure according to needs.
Bit 11	NWTCFG: NWAIT timing configuration	0x0
Bit 10	WRAPEN: Wrapped enable	0x0
Bit 9	NWPOL: NWAIT polarity	Configure according to memory specifications.
Bit 8	SYNCBEN: Synchronous burst enable	0x0
Bit 7	Reserved	0x1
Bit 6	NOREN:NOR Flash access enable	0x1

Bit	Description	Configuration
Bit 5:4	EXTMDBW: External memory data bus width	Configure according to memory specifications.
Bit 3:2	DEV: Memory device type	0x2 (NOR Flash)
Bit 1	ADMUXEN: Address/data multiplexing enable	0x0
Bit 0	EN: Memory bank enable	0x1

Table 91. Mode C – SRAM/NOR Flash chip select timing register (XMC_BK1TMG)

Bit	Description	Configuration
Bit 31:30	Reserved	0x0
Bit 29:28	ASYNCM: Asynchronous mode	0x2 (Mode C)
Bit 27:24	DTLAT: Data latency	0x0
Bit 23:20	CLKPSC: Clock prescale	0x0
Bit 19:16	BUSLAT: Bus latency	Indicates the time the XMC_NE[x] from the rising edge to the falling edge. Configure according to needs and memory specifications
Bit 15:8	DTST: Data setup time	Refer to Figure 177. Configure according to needs and memory specifications.
Bit 7:4	ADDRHT: Address-hold time	0x0
Bit 3:0	ADDRST: Address setup time	Refer to Figure 177. Configure according to needs and memory specifications.

Table 92. Mode C – SRAM/NOR Flash write timing register (XMC_BK1TMGWR)

Bit	Description	Configuration
Bit 31:30	Reserved	0x0
Bit 29:28	ASYNCM: Asynchronous mode	0x1 (Mode C)
Bit 27:20	Reserved	0x0
Bit 19:16	BUSLAT: Bus latency	Indicates the time the XMC_NE[x] from the rising edge to the falling edge. Configure according to needs and memory specifications
Bit 15:8	DTST: Data setup time	Refer to Figure 178. Configure according to needs and memory specifications.
Bit 7:4	ADDRHT: Address-hold time	0x0
Bit 3:0	ADDRST: Address setup time	Refer to Figure 178. Configure according to needs and memory specifications.

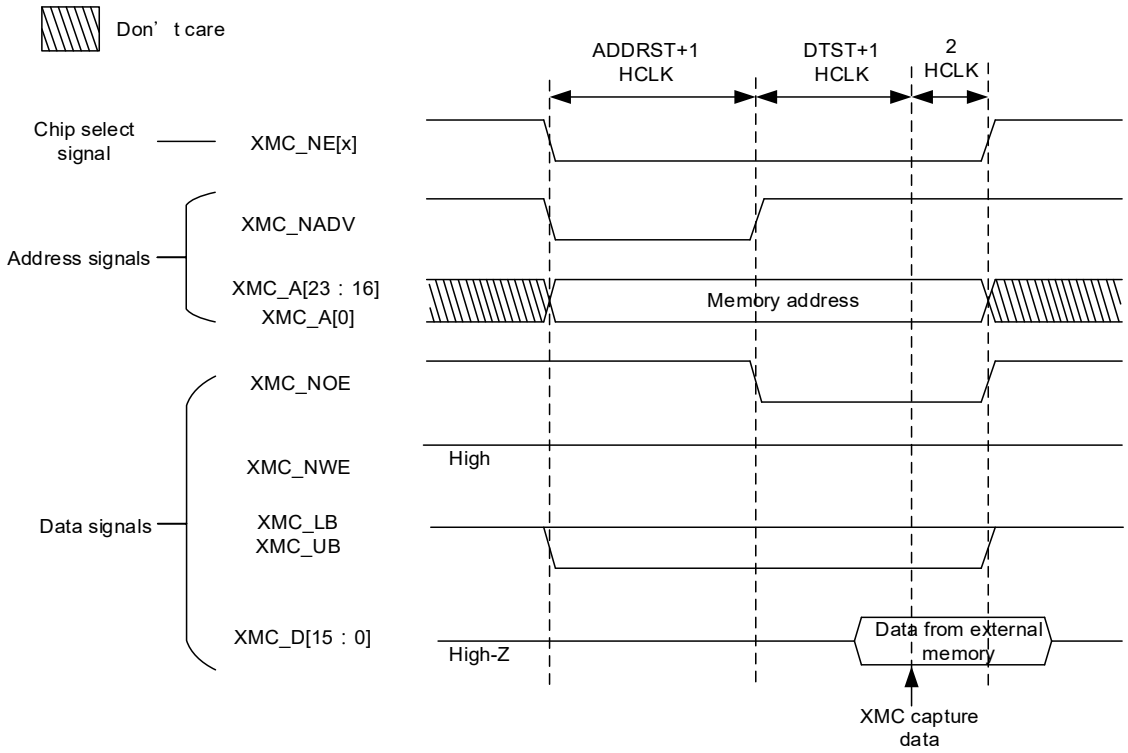


Figure 177. NOR/PSARM mode C read access

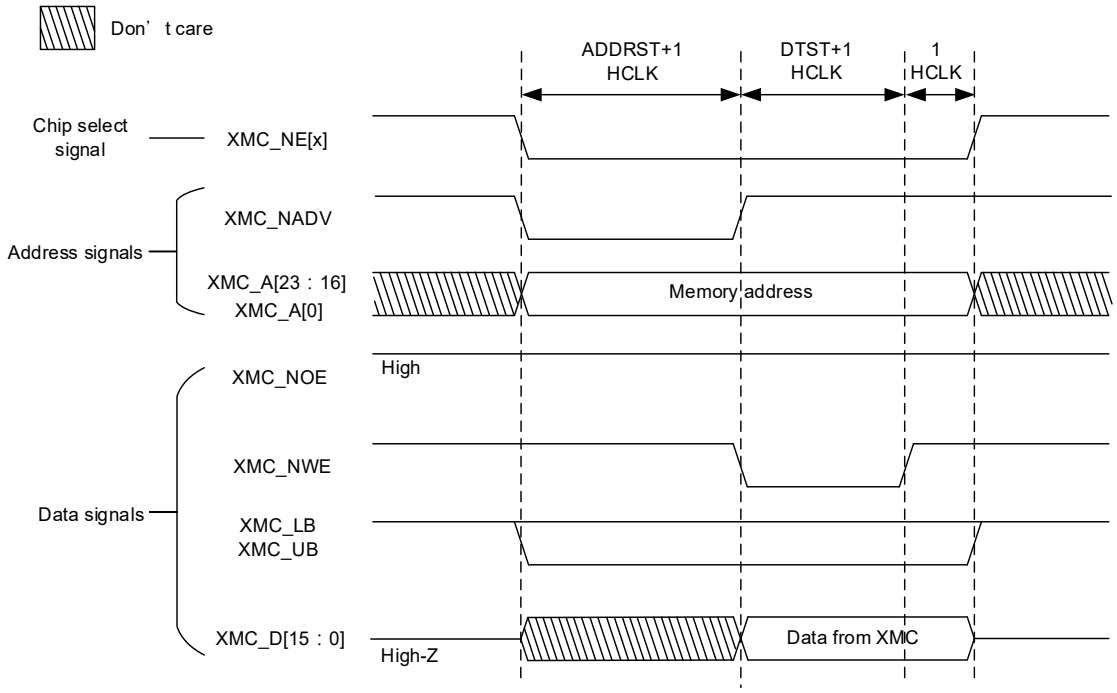


Figure 178. NOR/PSARM mode C write access

Mode D

As configured in the following tables, the XMC uses mode D to access the external memory. The timing of read operation is shown in Figure 179. The timing of write operation is shown in Figure 180.

Table 93. Mode D – SRAM/NOR Flash chip select register (XMC_BK1CTRL)

Bit	Description	Configuration
Bit 31:20	Reserved	0x0
Bit 19	MWMC: Memory write mode control	0x0
Bit 18:16	CRPGS:CRAM page size	0x0
Bit 15	NWASEN: NWAIT in asynchronous transfer enable	Configure according to memory specifications.
Bit 14	RWTD: Read-write timing different	0x1
Bit 13	NWSEN: NWAIT in synchronous transfer enable	0x0
Bit 12	WEN: Write enable	Configure according to needs.
Bit 11	NWTCFG: NWAIT timing configuration	0x0
Bit 10	WRAPEN: Wrapped enable	0x0
Bit 9	NWPOL: NWAIT polarity	Configure according to memory specifications.
Bit 8	SYNCBEN: Synchronous burst enable	0x0
Bit 7	Reserved	0x1
Bit 6	NOREN:NOR Flash access enable	Configure according to memory specifications.
Bit 5:4	EXTMDBW: External memory data bus width	Configure according to memory specifications.
Bit 3:2	DEV: Memory device type	Configure according to memory specifications.
Bit 1	ADMUXEN: Address/data multiplexing enable	0x0
Bit 0	EN: Memory bank enable	0x1

Table 94. Mode D – SRAM/NOR Flash chip select timing register (XMC_BK1TMG)

Bit	Description	Configuration
Bit 31:30	Reserved	0x0
Bit 29:28	ASYNCM: Asynchronous mode	0x3 (Mode D)
Bit 27:24	DTLAT: Data latency	0x0
Bit 23:20	CLKPSC: Clock prescale	0x0
Bit 19:16	BUSLAT: Bus latency	Indicates the time the XMC_NE[x] from the rising edge to the falling edge. Configure according to needs and memory specifications
Bit 15:8	DTST: Data setup time	Refer to Figure 179. Configure according to needs and memory specifications.
Bit 7:4	ADDRHT: Address-hold time	Refer to Figure 179. Configure according to needs and memory specifications.
Bit 3:0	ADDRST: Address setup time	Refer to Figure 179. Configure according to needs and memory specifications.

Table 95. Mode D – SRAM/NOR Flash write timing register (XMC_BK1TMGWR)

Bit	Description	Configuration
Bit 31:30	Reserved	0x0
Bit 29:28	ASYNCM: Asynchronous mode	0x3 (Mode D)
Bit 27:20	Reserved	0x0
Bit 19:16	BUSLAT: Bus latency	Indicates the time the XMC_NE[x] from the rising edge to the falling edge. Configure according to needs and memory specifications
Bit 15:8	DTST: Data setup time	Refer to Figure 180. Configure according to needs and memory specifications.
Bit 7:4	ADDRHT: Address-hold time	Refer to Figure 180. Configure according to needs and memory specifications.
Bit 3:0	ADDRST: Address setup time	Refer to Figure 180. Configure according to needs and memory specifications.

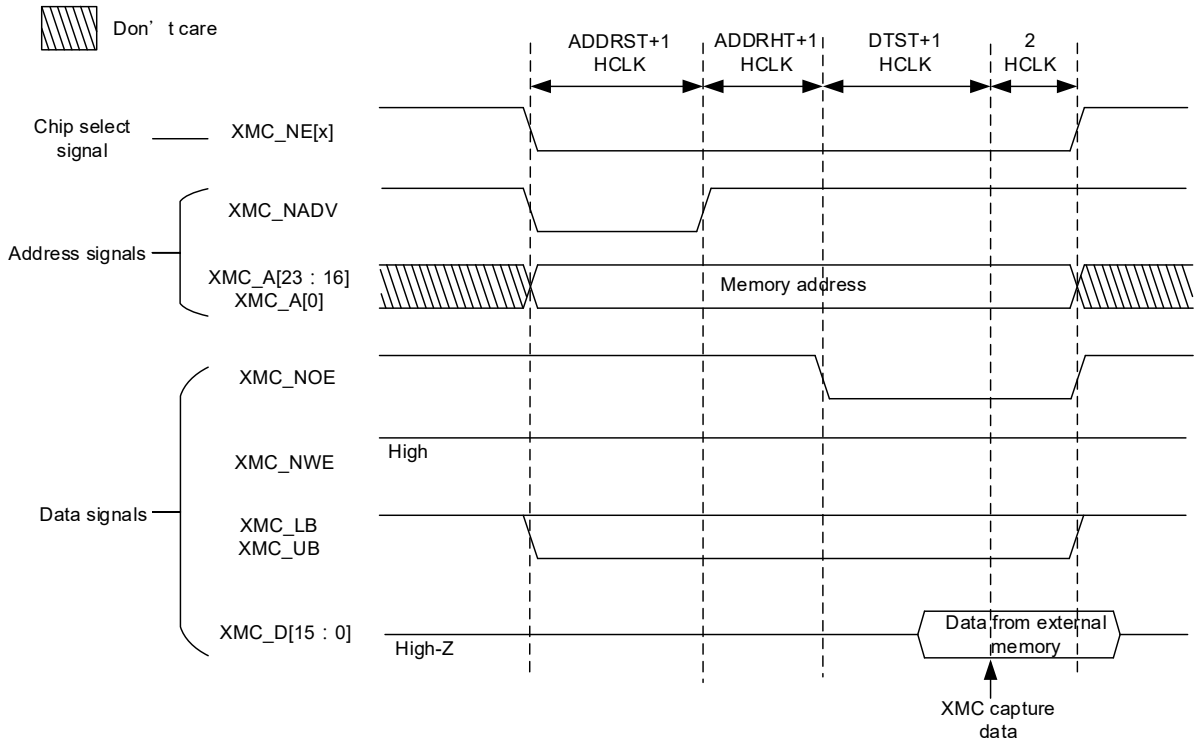


Figure 179. NOR/PSARM mode D read access

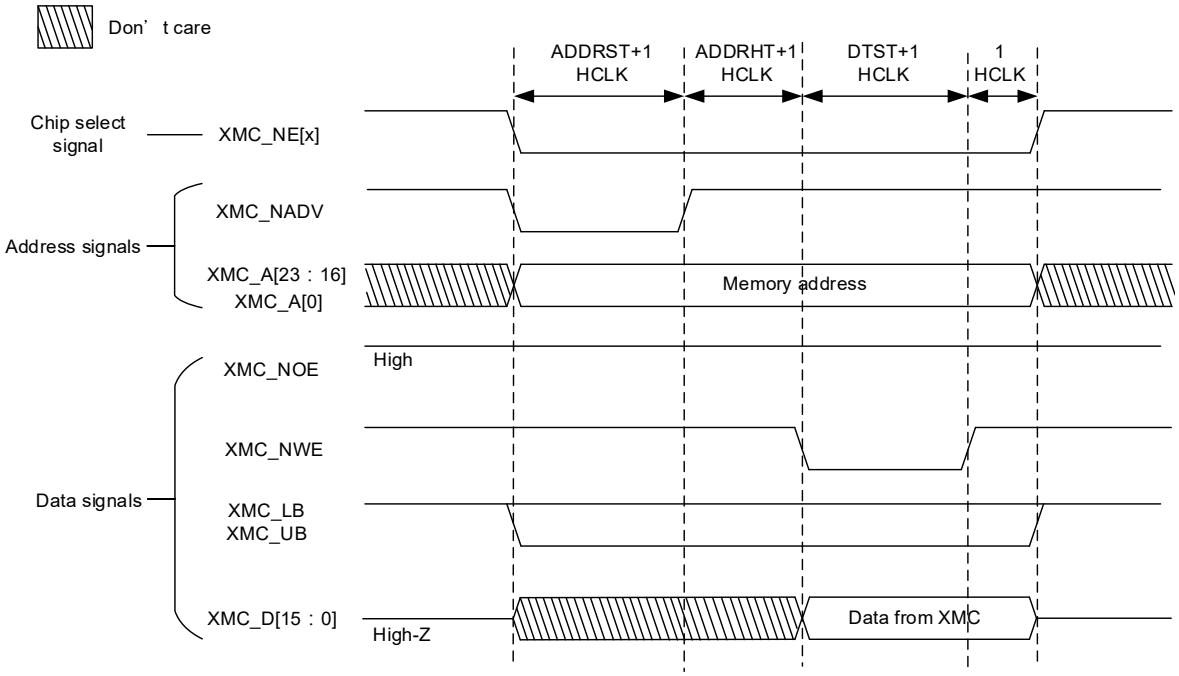


Figure 180. NOR/PSARM mode D write access

Multiplexed mode

As configured in the following tables, the XMC uses mode A to access the external memory. The timing of read operation is shown in Figure 181. The timing of write operation is shown in Figure 182.

Table 96. Multiplexed mode – SRAM/NOR Flash chip select control register (XMC_BK1CTRL)

Bit	Description	Configuration
Bit 31:20	Reserved	0x0
Bit 19	MWMC: Memory write mode control	0x0
Bit 18:16	CRPGS:CRAM page size	0x0
Bit 15	NWASEN: NWAIT in asynchronous transfer enable	Configure according to memory specifications.
Bit 14	RWTD: Read-write timing different	0x0
Bit 13	NWSEN: NWAIT in synchronous transfer enable	0x0
Bit 12	WEN: Write enable	Configure according to needs.
Bit 11	NWTCFG: NWAIT timing configuration	0x0
Bit 10	WRAPEN: Wrapped enable	0x0
Bit 9	NWPOL: NWAIT polarity	Configure according to memory specifications.
Bit 8	SYNCBEN: Synchronous burst enable	0x0
Bit 7	Reserved	0x1
Bit 6	NOREN:NOR Flash access enable	Configure according to memory specifications.
Bit 5:4	EXTMDBW: External memory data bus width	Configure according to memory specifications.

Bit	Description	Configuration
Bit 3:2	DEV: Memory device type	Configure according to memory specifications. It is valid except 0x0 (SRAM)
Bit 1	ADMUXEN: Address/data multiplexing enable	0x1
Bit 0	EN: Memory bank enable	0x1

Table 97. Multiplexed mode – SRAM/NOR Flash chip select timing register (XMC_BK1TMG)

Bit	Description	Configuration
Bit 31:30	Reserved	0x0
Bit 29:28	ASYNCM: Asynchronous mode	0x0
Bit 27:24	DTLAT: Data latency	0x0
Bit 23:20	CLKPSC: Clock prescale	0x0
Bit 19:16	BUSLAT: Bus latency	Indicates the time the XMC_NE[x] from the rising edge to the falling edge. Configure according to needs and memory specifications
Bit 15:8	DTST: Data setup time	Refer to the following diagrams. Configure according to needs and memory specifications.
Bit 7:4	ADDRHT: Address-hold time	Refer to the following diagrams. Configure according to needs and memory specifications.
Bit 3:0	ADDRST: Address setup time	Refer to the following diagrams. Configure according to needs and memory specifications.

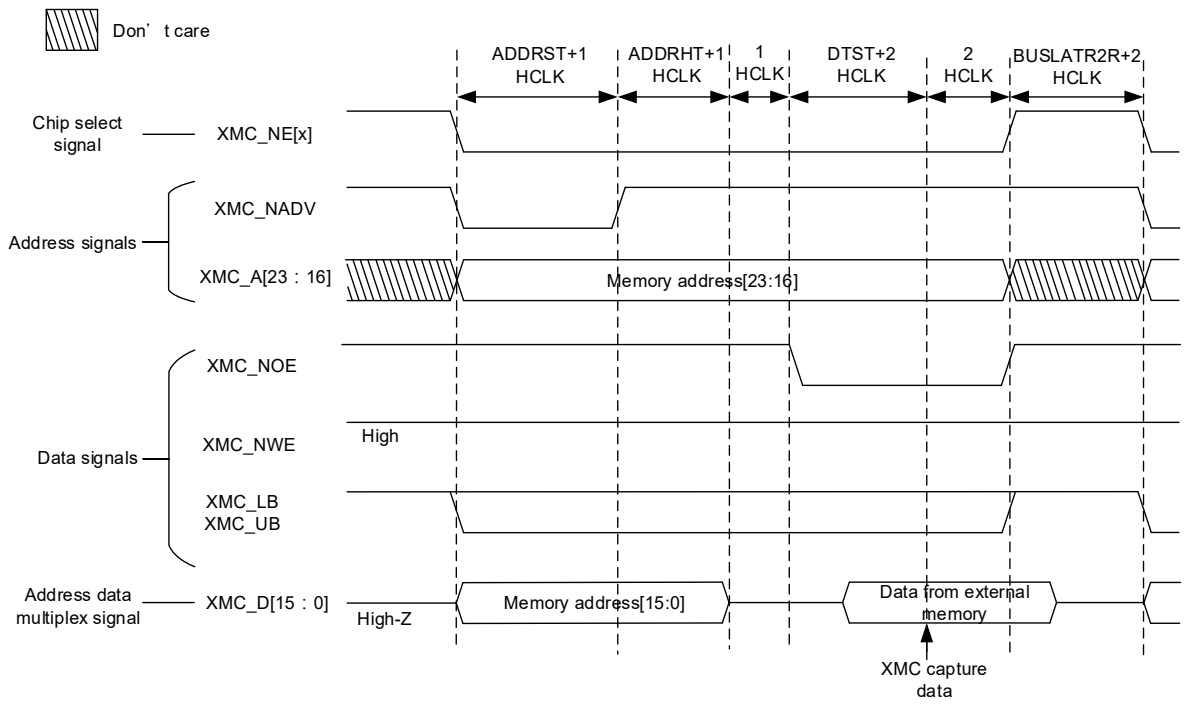


Figure 181. NOR/PSARM multiplexed mode read access

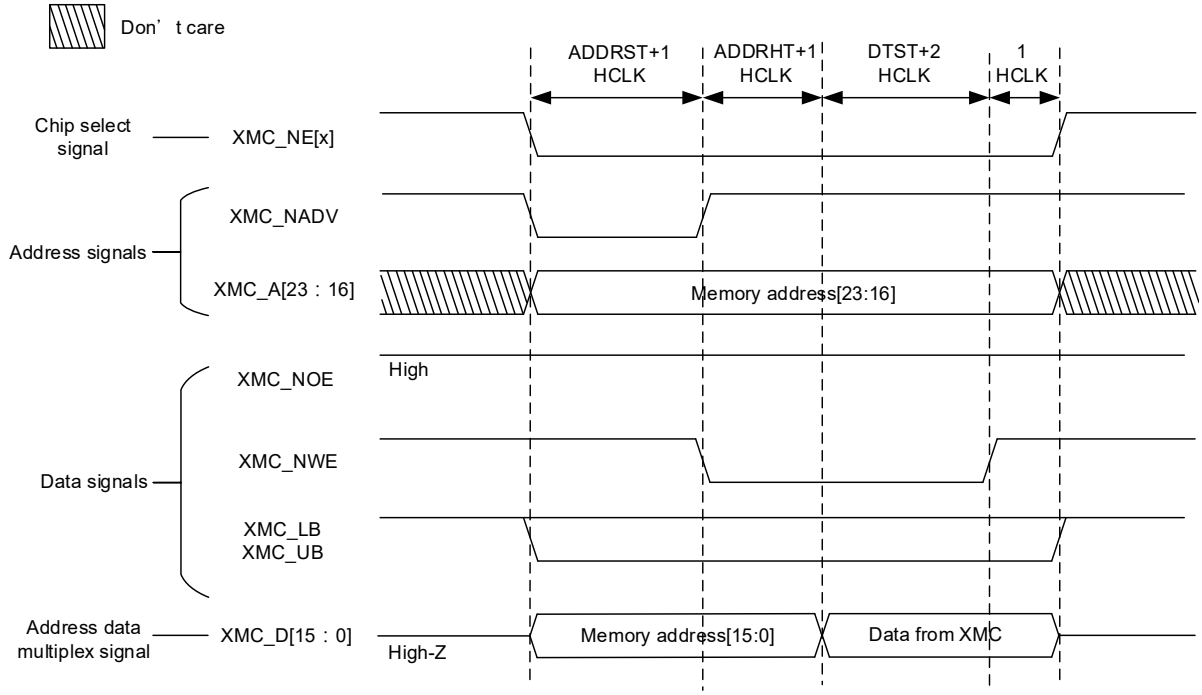


Figure 182. NOR/PSARM multiplexed mode write access

Synchronous mode

As configured in the following tables, the XMC uses synchronous mode to access the external memories.

If the memory insertes XMC_NWAIT signal between the address latch and data transfer, the XMC will not only wait (DTLAT+1) CLK clok cycles but also have to take into account the XMC_NWAIT signal. During data transmission, the XMC will, depending on the NWTCFG configuration, select to wait either one cycle after the XMC_NWAIT signal is active or when the XMC_NWAIT signal is active.

Figure 183 shows the timing for read access, while Figure 184 shows the timing for write access. The following diagrams are exmples of XMC waiting in the next cycle of XMC_NWAIT signal (NWTCFG=0)

Table 98. Synchronous mode – SRAM/NOR Flash chip select control register (XMC_BK1CTRL)

Bit	Description	Configuration
Bit 31:20	Reserved	0x0
Bit 19	MWMC: Memory write mode control	0x1
Bit 18:16	CRPGS:CRAM page size	Configure according to memory specifications.
Bit 15	NWASEN: NWAIT in asynchronous transfer enable	0x0
Bit 14	RWTD: Read-write timing different	0x0
Bit 13	NWSEN: NWAIT in synchronous transfer enable	Configure according to memory specifications.

Bit	Description	Configuration
Bit 12	WEN: Write enable	Configure according to needs.
Bit 11	NWTCFG: NWAIT timing configuration	Configure according to memory specifications.
Bit 10	WRAPEN: Wrapped enable	Configure according to needs.
Bit 9	NWPOL: NWAIT polarity	c
Bit 8	SYNCBEN: Synchronous burst enable	0x1
Bit 7	Reserved	0x1
Bit 6	NOREN:NOR Flash access enable	Write synchronization: 0x0 Read synchronization: Configure according to memory specifications.
Bit 5:4	EXTMDBW: External memory data bus width	Configure according to memory specifications.
Bit 3:2	DEV: Memory device type	Write synchronization: 0x1 Read synchronization: Configure according to memory specifications. It is valid except 0x0 (SRAM)
Bit 1	ADMUXEN: Address/data multiplexing enable	Configure according to needs.
Bit 0	EN: Memory bank enable	0x1

Table 99. Synchronous mode – SRAM/NOR Flash chip select timing register (XMC_BK1TMG)

Bit	Description	Configuration
Bit 31:30	Reserved	0x0
Bit 29:28	ASYNCM: Asynchronous mode	0x0
Bit 27:24	DTLAT: Data latency	Refer to the following diagrams
Bit 23:20	CLKPSC: Clock prescale	XMC_CLK cycle is HCLK cycle × (CLKPSC + 1). Refer to the following diagrams
Bit 19:16	BUSLAT: Bus latency	Indicates the time the XMC_NE[x] from the rising edge to the falling edge. Configure according to needs and memory specifications
Bit 15:8	DTST: Data setup time	0x0
Bit 7:4	ADDRHT: Address-hold time	0x0
Bit 3:0	ADDRST: Address setup time	0x0

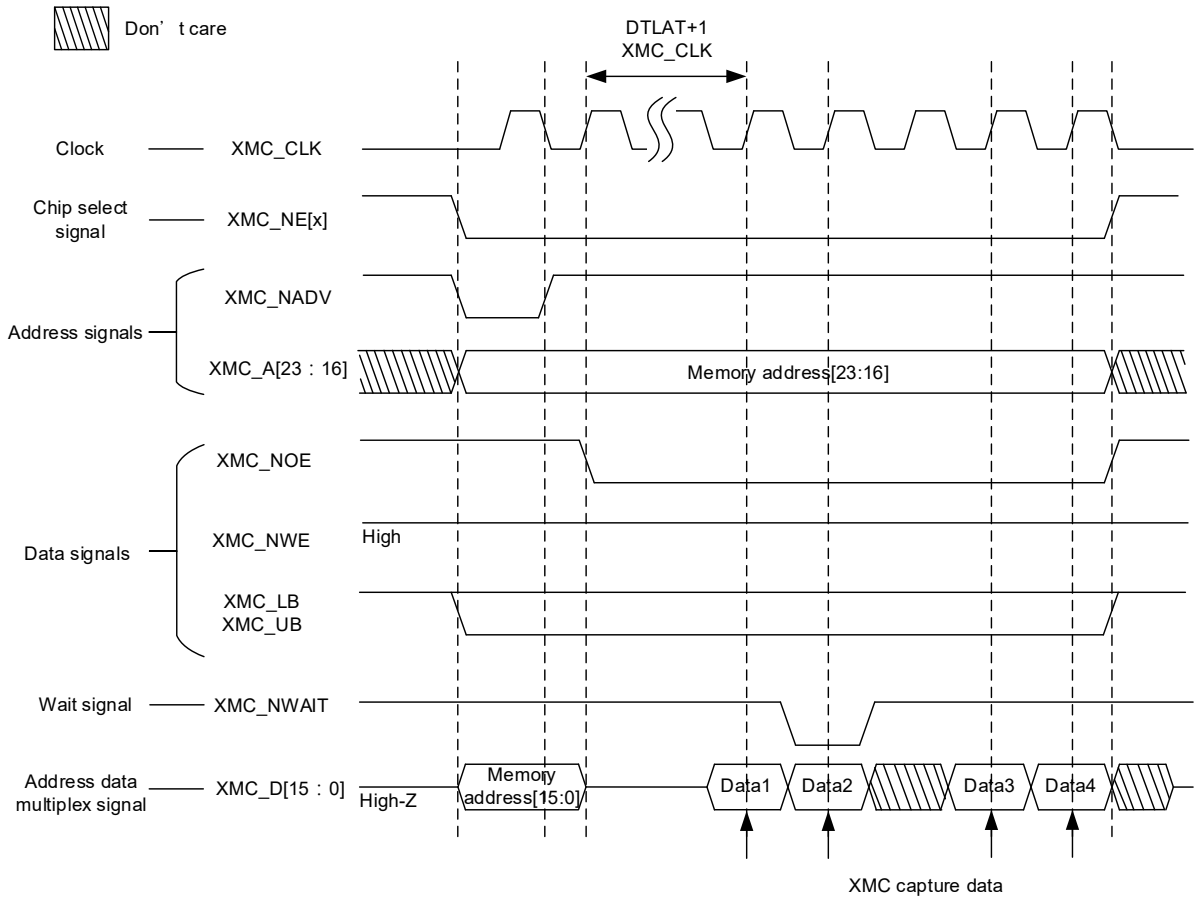


Figure 183. NOR/PSARM synchronous multiplexed mode read access

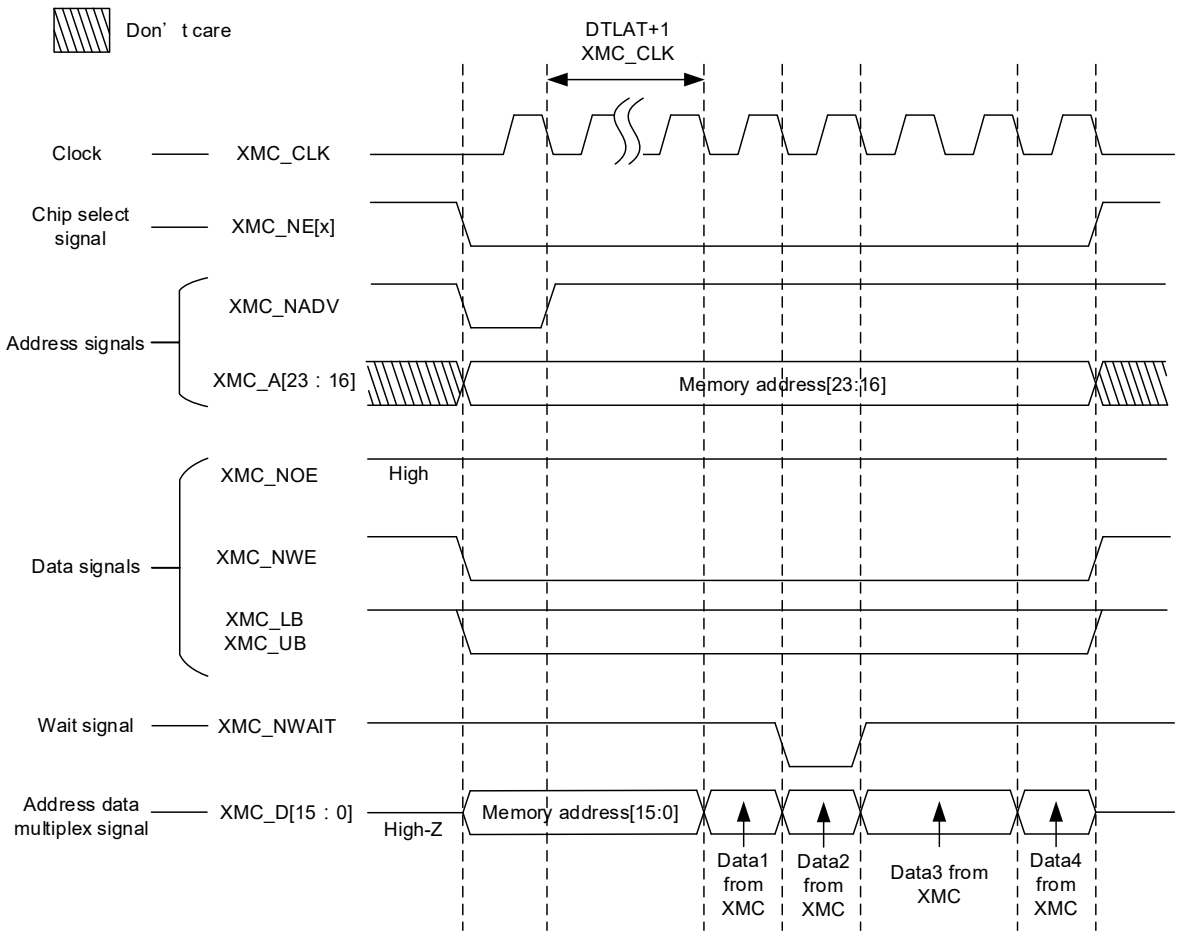


Figure 184. NOR/PSARM synchronous multiplexed mode write access

NAND

NAND interface can be used to drvie NAND Flash. It is divided into two storage banks: regular bank and special bank, each with its separate timing registers. Both banks can be accessible with different timings.

Operation mode

Pin function

Pin signals vary from external memory to external memory. the following table lists typical pin signals.

Table 100. Typical pin signals for NAND Flash

XMC pin name	8-bit NAND Flash	16-bit NAND Flash
XMC_NCE[2]	Chip-select	Chip-select
XMC_A[17]	Address latch enable (ALE)	Address latch enable (ALE)
XMC_A[16]	Command latch ebabke (CLE)	Command latch ebabke (CLE)
XMC_NOE	Output enable (NRE)	Output enable (NRE)
XMC_NWE	Write enable	Write enable
XMC_D[15:0]	Data bus	Do not use XMC_D[15:8] Use XMC_D[7:0] as data bus.
XMC_NWAIT	Ready/Busy (R/B)	Ready/Busy (R/B)

Access address

The upper bytes of the HADDR bit is used to select a memory bank while the lower bytes to data memory address. HADDR is a byte address whereas the XMC supports the memory addressed in words or half words. Address translation between them is shown in Table 72. As long as read/write access to a specific address occurs, the XMC will enable chip-select signals and write/read operation to the external memories according to the HADDR bit.

The HADDR is only used to select memory banks. Refer to Table 70 for more information.

The user writes the command value in the command section, the destination address in the address section, and reads or writes the data from or to the data section. As the access addresses are transmitted through data bus, the HADDR is actually not associated with NAND Flash size, so theoretically the XMC has no limitation on the NAND Flash capacity accessible.

Data access

In case that the AHB data width is not equal to that of the memories, the XMC will make appropriate arrangement according to the typical signals of the external memories. The following table lists the operation modes supported by XMC.

Table 101. Data access width vs. external memory data width

Memory	Mode	AHB data width	Memory width	Description
8-bit NAND	R/W	8	8	
	R/W	16	8	Split into two XMC accesses
	R/W	32	8	Split into four XMC accesses
16-bit NAND	R	8	16	
	R/W	16	16	
	R/W	32	16	Split into two XMC accesses

Access timings

The XMC access the NAND Flash according to the timing parameters, as shown in the following table and Table 86. Users can perform programming operations according to the specifications of the external memory and application needs.

Table 102. NAND parameter registers

Parameter register	Function	Access mode	Unit
RGDHIZT/SPDHIZT	Number of cycles during which the data bus is kept in high-Z state	W	HCLK cycle
RGST/SPST	Memory set up time	R/W	HCLK cycle
RGWT/SPWT	Memory set up time	R/W	HCLK cycle
RGHT/SPHT	Memory set up time	R/W	HCLK cycle

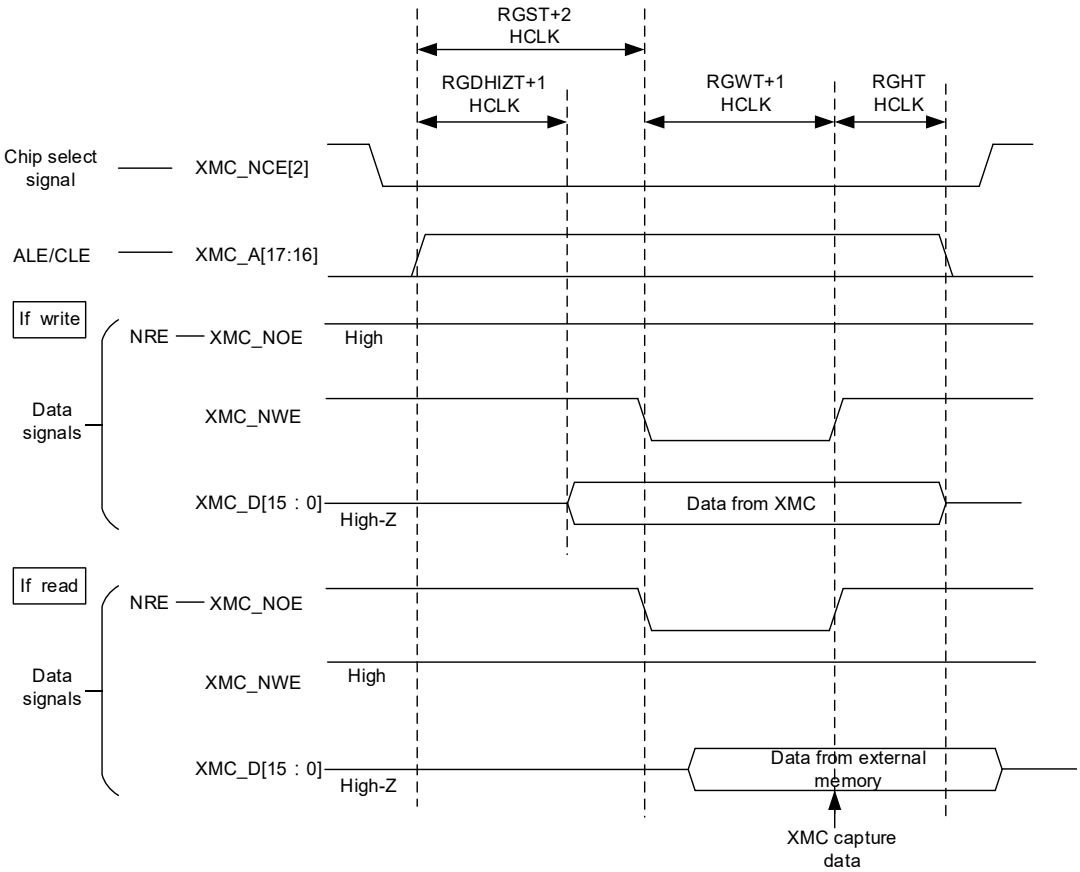


Figure 185. NAND read access

When the NWEN bit is enabled, the XMC will monitor whether the XMC_NWAIT signal is pulled low or not at the end of memory hold-up period, if so, it wil keep the XMC_NCE[2] low until the XMC_NWAIT goes high.

Some NAND Flash devices require that, after receiveing the last address byte, that the XMC_NCE[2] remains low until it enters ready state. This can be done by means of a special timing register and NWEN bits:

The user has to configure the time duration during which the NAND Flash shifts from the rising edge of the XMC_NWE to the falling edge of the XMC_NWAIT into a SPHT register, and write the last address byte into a special memory address section so that the XMC can perform write operations based on the timings of the special memory timing register, as shown in Address 3 in the following diagram.

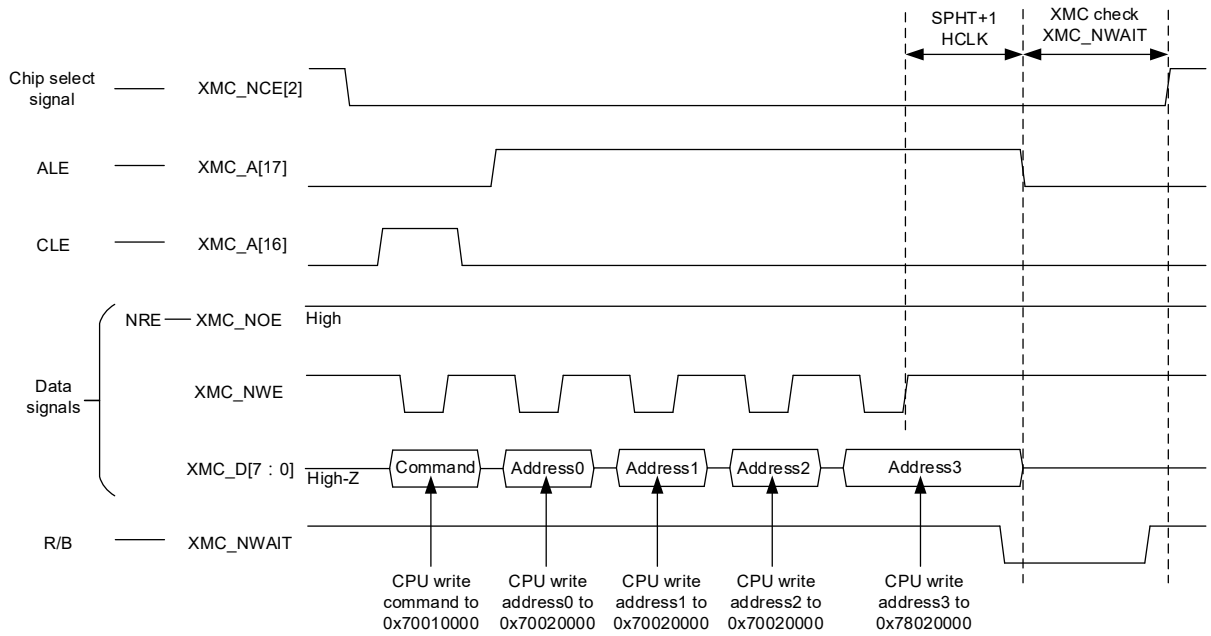


Figure 186. NAND wait functionality

ECC computation

The NAND interface contains ECC computation module so that the data will be used for ECC computation when the NAND interface access the NAND Flash. The computed value is stored into the XMC_BK2ECC register.

To perfrom an ECC computation:

1. Configure the ECCPGS bit to select the number of bytes to be computed by ECC module: 256, 512, 1024, 2048, 4096 or 8192 bytes.
2. Enable the ECCEN bit.
3. Read/write from and to the data section.
4. After receiving/sending the same number of bytes as the value programed in the ECCPGS, the XMC will store the ECC computed value into the XMC_BK2ECC registers.
5. Software reads/write the last byte and waits until the FIFO flag is set.
6. Software reads the XMC_BK2ECC register and performs the corresponding error correction routine.
7. Clear the ECCEN bit by software. Repeat from 2 to 6.

Table 103. lists the ECC result bits corresponding to the number of bytes

ECCPGS	000	001	010	011	100	101
Number of bytes	256	512	1024	2048	4096	8192
ECC result bits	ECC[21:0]	ECC[23:0]	ECC[25:0]	ECC[27:0]	ECC[29:0]	ECC[31:0]

XMC registers

These peripheral registers must be accessed by words (32 bits).

Table 104. XMC register address mapping

Register	Offset	Reset value
XMC_BK1CTRL1	0x000	0x0000 30DB
XMC_BK1TMG1	0x004	0x0FFF FFFF
XMC_BK1CTRL4	0x018	0x0000 30D2
XMC_BK1TMG4	0x01C	0x0FFF FFFF
XMC_BK2CTRL	0x060	0x0000 0018
XMC_BK2IS	0x064	0x0000 0040
XMC_BK2TMGRG	0x068	0xFCFC FCFC
XMC_BK2TMGSP	0x06C	0xFCFC FCFC
XMC_BK2ECC	0x074	0x0000 0000
XMC_BK1TMGWR1	0x104	0x0FFF FFFF
XMC_BK1TMGWR4	0x11C	0x0FFF FFFF
XMC_EXT1	0x220	0x0000 0808
XMC_EXT4	0x22C	0x0000 0808

NOR Flash and PSRAM control registers

SRAM/NOR Flash chip select control register 1 (XMC_BK1CTRL1)

Bit	Register	Reset value	Type	Description
Bit 31:20	Reserved	0x000	resd	Kept at its default value.
Bit 19	MWMC	0x0	rw	Memory write mode control 0: Write operations are performed in asynchronous mode 1: Write operations are performed in synchronous mode
Bit 18:16	CRPGS	0x0	rw	CRAM page size Cellular RAM 1.5 does not allow synchronous access to cross the address boundaries between pages. When these bits are configured in synchronous mode, the XMC will automatically split the access when the page size is reached. 000: No split access when crossing address boundary (default value) 001: 128 bytes 010: 256 bytes 011: 512 bytes 100: 1024 bytes Others: Reserved.
Bit 15	NWASEN	0x0	rw	NWAIT in asynchronous transfer enable 0: NWAIT signal is disabled 1: NWAIT signal is enable

Bit	Register	Reset value	Type	Description
Bit 14	RWTD	0x0	rw	Read-write timing different Different timings are used for read and write operations, that is, the XMC_BK1TMGWR register is enabled. 0: Same timings for read and write operations 1: Different timings for read and write operations
Bit 13	NWSEN	0x1	rw	NWAIT enable during synchronous transfer 0: NWAIT signal is disabled 1: NWAIT signal is enabled
Bit 12	WEN	0x1	rw	Write enable 0: Disabled 1: Enabled
Bit 11	NWTCFG	0x0	rw	NWAIT timing configuration It is valid only in synchronous mode. 0: NWAIT signal is active one data cycle before the wait state 1: NWAIT signal is active one data cycle during the wait state
Bit 10	WRAPEN	0x0	rw	Wrapped enable This bit defines whether the XMC will split a wrapped AHB access into two accesses. 0: Direct wrapped access is not allowed 1: Direct wrapped access is allowed
Bit 9	NWPOL	0x0	rw	NWAIT polarity This bit defines the polarity of the NWAIT signal in synchronous mode. 0: NWAIT active low 1: NWAIT active high
Bit 8	SYNCBEN	0x0	rw	Synchronous burst enable This bit allowes synchronous access to Flash memories. 0: Synchronous burst disabled 1: Synchronous burst enabled
Bit 7	Reserved	0x1	resd	Kept at its default value.
Bit 6	NOREN	0x1	rw	Nor flash access enable 0: Nor flash access is disabled 1: Nor flash access is enabled
Bit 5:4	EXTMDBW	0x1	rw	External memory data bus width This field defines the external memory data bus width. 00: 8 bits 01: 16 bits 10: Reserved 11: Reserved
Bit 3:2	DEV	0x2	rw	Memory device type 00: SRAM/ROM 01: PSRAM (Cellular RAM or CRAM) 10: NOR Flash 11: Reserved
Bit 1	ADMUXEN	0x1	rw	Address/data multiplexing enable 0: Address/data not multiplexed 1: Address/data multiplexed
Bit 0	EN	0x1	rw	Memory bank enable 0: Memory bank disabled 1: Memory bank enabled

SRAM/NOR Flash chip select control register 4 (XMC_BK1CTRL4)

Bit	Register	Reset value	Type	Description
Bit 31:20	Reserved	0x000	resd	Kept at its default value.
Bit 19	MWMC	0x0	rw	Memory write mode control 0: Write operations are performed in asynchronous mode 1: Write operations are performed in synchronous mode
Bit 18:16	CRPGS	0x0	rw	CRAM page size Cellular RAM 1.5 does not allow synchronous access to cross the address boundaries between pages. When these bits are configured in synchronous mode, the XMC will automatically split the access when the page size is reached. 000: No split access when crossing address boundary (default value) 001: 128 bytes 010: 256 bytes 011: 512 bytes 100: 1024 bytes Others: Reserved
Bit 15	NWASEN	0x0	rw	NWAIT enable during asynchronous transfer 0: NWAIT signal is disabled 1: NWAIT signal is enable
Bit 14	RWTD	0x0	rw	Read-write timing different Different timings are used for read and write operations, that is, the XMC_BK1TMGWR register is enabled. 0: Same timings for read and write operations 1: Different timings for read and write operations
Bit 13	NWSEN	0x1	rw	NWAIT enable during synchronous transfer 0: NWAIT signal is disabled 1: NWAIT signal is enabled
Bit 12	WEN	0x1	rw	Write enable 0: Disabled 1: Enabled
Bit 11	NWTCFG	0x0	rw	NWAIT timing configuration It is valid only in synchronous mode. 0: NWAIT signal is active one data cycle before the wait state 1: NWAIT signal is active one data cycle during the wait state
Bit 10	WRAPEN	0x0	rw	Wrapped enable This bit defines whether the XMC will split a wrapped AHB access into two accesses. 0: Direct wrapped access is not allowed 1: Direct wrapped access is allowed
Bit 9	NWPOL	0x0	rw	NWAIT polarity This bit defines the polarity of the NWAIT signal in synchronous mode. 0: NWAIT active low 1: NWAIT active high
Bit 8	SYNCBEN	0x0	rw	Synchronous burst enable This bit allowes synchronous access to Flash memories. 0: Synchronous burst disabled 1: Synchronous burst enabled

Bit	Register	Reset value	Type	Description
Bit 7	Reserved	0x1	resd	Kept at its default value.
Bit 6	NOREN	0x1	rw	Nor flash access enable 0: Nor flash access is disabled 1: Nor flash access is enabled
Bit 5:4	EXTMDBW	0x1	rw	External memory data bus width This field defines the external memory data bus width. 00: 8 bits 01: 16 bits 10: Reserved 11: Reserved
Bit 3:2	DEV	0x0	rw	Memory device type 00: SRAM/ROM 01: PSRAM (Cellular RAM or CRAM) 10: NOR Flash 11: Reserved
Bit 1	ADMUXEN	0x1	rw	Address/data multiplexing enable 0: Address/data not multiplexed 1: Address/data multiplexed
Bit 0	EN	0x0	rw	Memory bank enable 0: Memory bank disabled 1: Memory bank enabled

SRAM/NOR Flash chip select timing register 1, 4 (XMC_BK1CTRL1, 4)

Bit	Register	Reset value	Type	Description
Bit 31:30	Reserved	0x0	resd	Kept at its default value.
Bit 29:28	ASYNCM	0x0	rw	Asynchronous mode This field is valid only when the RWTD bit is enabled. 00: Mode A 01: Mode B 10: Mode C 11: Mode D
Bit 27:24	DTLAT	0xF	rw	Data latency This field is valid only in synchronous mode. 0000: 0 XMC_CLK cycle is inserted 0001: 1 additional XMC_CLK cycle is inserted …… 1111: 15 additional XMC_CLK cycles are inserted
Bit 23:20	CLKPSC	0xF	rw	Clock prescaler This field is valid only in synchronous mode. It defines the frequency of the XMC_CLK clock. 0000: Reserved 0001: XMC_CLK cycle= 2 × HCLK clock cycles 0010: XMC_CLK cycle =3 × HCLK clock cycles …… 1111: XMC_CLK cycle = 6 × HCLK cycles

Bit	Register	Reset value	Type	Description
Bit 19:16	BUSLAT	0xF	rw	Bus latency To avoid data bus conflict, a latency is inserted on the data bus if one read operation is followed by writing XMC in multiplexed or synchronous mode. 0000: 1 HCLK cycle is inserted 0001: 2 HCLK cycles are inserted …… 1111: 16 HCLK cycles are inserted
Bit 15:8	DTST	0xFF	rw	Data setup time 0000: 0 HCLK cycle is inserted 0001: 1 additional HCLK cycle is inserted …… 1111: 15 additional HCLK cycles are inserted
Bit 7:4	ADDRHT	0xF	rw	Address-hold time 0000: 0 HCLK cycle is inserted 0001: 1 additional HCLK cycle is inserted …… 1111: 15 additional HCLK cycles are inserted
Bit 3:0	ADDRST	0xF	rw	Address setup time 0000: 0 HCLK cycle is inserted 0001: 1 additional HCLK cycle is inserted …… 1111: 15 additional HCLK cycles are inserted

SRAM/NOR Flash write timing register 1, 4 (XMC_BK1 TMGWR1, 4)

Bit	Register	Reset value	Type	Description
Bit 31:30	Reserved	0x0	resd	Kept at its default value.
Bit 29:28	ASYNCM	0x0	rw	Asynchronous mode This field is valid only when the RWTD bit is enabled. 00: Mode A 01: Mode B 10: Mode C 11: Mode D
Bit 27:20	Reserved	0xFF	resd	Kept at its default value.
Bit 19:16	BUSLAT	0xF	rw	Bus latency To avoid data bus conflict, a latency is inserted on the data bus if write operatin is followed by reading XMC in multiplexed or synchronous mode. 0000: 1 HCLK cycle is inserted 0001: 2 HCLK cycles are inserted …… 1111: 16 HCLK cycles are inserted
Bit 15:8	DTST	0xFF	rw	Data setup time 0000: 0 HCLK cycle is inserted 0001: 1 additional HCLK cycle is inserted …… 1111: 15 additional HCLK cycles are inserted

Bit	Register	Reset value	Type	Description
Bit 7:4	ADDRHT	0xF	rw	Address-hold time 0000: 0 HCLK cycle is inserted 0001: 1 additional HCLK cycle is inserted …… 1111: 15 additional HCLK cycles are inserted
Bit 3:0	ADDRST	0xF	rw	Address setup time 0000: 0 HCLK cycle is inserted 0001: 1 additional HCLK cycle is inserted …… 1111: 15 additional HCLK cycles are inserted

SRAM/NOR Flash extra timing register 1, 4 (XMC_EXT1, 4)

Bit	Register	Reset value	Type	Description
Bit 31:16	Reserved	0x0000	resd	Kept at its default value.
Bit 15:8	BUSLATR2R	0x08	rw	Bus turnaround phase for consecutive read duration This field is used to define the bus turnaround phase duration for consecutive read operations. A delay is inserted bwteen two consecutive operations in order to avoid bus conflicts. 00000000: 1 HCLK cycle is inserted for consecutive read operations 00000001: 2 HCLK cycles are inserted for consecutive read operations …… 00001000: 9 HCLK cycles are inserted for consecutive read operations (default value) …… 11111111: 256 HCLK cycles are inserted for consecutive read operations
Bit 7:0	BUSLATW2W	0x08	rw	Bus turnaround phase for consecutive write duration This field is used to define the bus turnaround phase duration for consecutive write operations. A delay is inserted between two consecutive write operations in order to avoid bus conflicts. 00000000: 1 HCLK cycle is inserted for consecutive write operations 00000001: 2 HCLK cycles are inserted for consecutive write operations …… 00001000: 9 HCLK cycles are inserted for consecutive write operations (default value) …… 11111111: 256 HCLK cycles are inserted for consecutive write operations

NAND Flash control registers

NAND Flash control register 2 (XMC_BK2CTRL)

Bit	Register	Reset value	Type	Description
Bit 31:20	Reserved	0x000	resd	Kept at its default value.
Bit 19:17	ECCPGS	0x0	rw	ECC page size 000: 256 bytes 001:512 bytes 010: 1024 bytes 011: 2048 bytes 100: 4096 bytes 101: 8192 bytes
Bit 16:13	TAR	0x0	rw	ALE to RE delay This field specifies the delay from the falling edge of the ALE to that of the RE. 0000:1 HCLK cycle …… 1111: 16 HCLK cycles
Bit 12:9	TCR	0x0	rw	CLE to RE delay This field specifies the delay from the falling edge of the CLE to that of the RE. 0000: 1 HCLK cycle …… 1111: 16 HCLK cycles
Bit 8:7	Reserved	0x0	resd	Kept at its default value.
Bit 6	ECCEN	0x0	rw	ECC enable 0: ECC disabled 1: ECC enabled
Bit 5:4	EXTMDBW	0x1	rw	External memory data bus width This field specifies the external NAND Flash width. 00: 8 bits 01: 16 bits 10: Reserved 11: Reserved
Bit 3	DEV	0x1	rw	Memory device type 0: Reserved 1: NAND Flash
Bit 2	EN	0x0	rw	Memory bank enable 0: Memory bank disabled 1: Memory bank enabled
Bit 1	NWEN	0x0	rw	Wait feature enable his bit is used to enable NAND Flash wait function. 0: Disabled 1: Enabled
Bit 0	Reserved	0x0	resd	Kept at its default value.

Interrupt enable and FIFO status register 2 (XMC_BK2IS)

Bit	Register	Reset value	Type	Description
Bit 31:7	Reserved	0x000000	resd	Kept at its default value.
Bit 6	FIFOE	0x1	rw	FIFO empty This bit is set by hardware when the FIFO is empty. 0: FIFO is not empty 1: FIFO is empty XMC FIFO size is 16 words. It is used to store the data from AHB.
Bit 5	FEIEN	0x0	rw	Falling edge interrupt enable 0: Falling edge interrupt disabled 1: Falling edge interrupt enabled
Bit 4	HLIEN	0x0	rw	High-level interrupt enable 0: High-level interrupt disabled 1: High-level interrupt enabled
Bit 3	REIEN	0x0	rw	Rising edge interrupt enable 0: Rising edge interrupt disabled 1: Rising edge interrupt enabled
Bit 2	FES	0x0	rw	Falling edge status This bit is set by hardware and cleared by software. 0: No falling edge interrupt generated 1: Falling edge interrupt generated
Bit 1	HLS	0x0	rw	High-level status This bit is set by hardware and cleared by software. 0: No high level interrupt generated 1: High level interrupt generated
Bit 0	RES	0x0	rw	Rising edge status This bit is set by hardware and cleared by software. 0: No rising edge interrupt generated 1: Rising edge interrupt generated

Regular memory timing register 2 (XMC_ BK2TMGRG)

Bit	Register	Reset value	Type	Description
Bit 31:24	RGDHIZT	0xFC	rw	Regular memory databus High resistance time This field defines the databus high resistance duration when write access to NAND Flash is started in a regular space. 00000000: 0 HCLK cycle is inserted 00000001: 1 additional HCLK cycle is inserted …… 11111111: 255 additional HCLK cycles are inserted
Bit 23:16	RGHT	0xFC	rw	Regular memory hold time This field defines the databus hold time when access to NAND Flash in a regular memory. 00000000: Reserved 00000001: 1 HCLK cycle is inserted …… 11111111: 255 HCLK cycles are inserted
Bit 15:8	RGWT	0xFC	rw	Regular memory wait time Specifies the regular memory wait time when the XMC_NWE and XMC_NOE is low. 00000000: 0 HCLK cycle is inserted 00000001: 1 additonal HCLK cycle is inserted …… 11111111: 255 additiona HCLK cycles are inserted
Bit 7:0	RGST	0xFC	rw	Regular memory setup time This field defines the address seti[time when access to NAND Flash in a regular memory. 00000000: 0 HCLK cycle is inserted 00000001: 1 additonal HCLK cycle is inserted …… 11111111: 255 additiona HCLK cycles are inserted

Special memory timing register 2 (XMC_ BK2TMGSP)

Bit	Register	Reset value	Type	Description
Bit 31:24	SPDHIZT	0xFC	rw	Special memory databus High resistance time This field defines the databus high resistance duration when write access to NAND Flash is started in a special space. 00000000: 0 HCLK cycle is inserted 00000001: 1 additional HCLK cycle is inserted …… 11111111: 255 additional HCLK cycles are inserted
Bit 23:16	SPHT	0xFC	rw	Special memory hold time This field defines the databus hold time when access to NAND Flash in a special memory. 00000000: Reserved 00000001: 1 HCLK cycle is inserted …… 11111111: 255 HCLK cycles are inserted
Bit 15:8	SPWT	0xFC	rw	Special memory wait time Specifies the special memory wait time when the XMC_NWE and XMC_NOE is low. 00000000: 0 HCLK cycle is inserted 00000001: 1 additonal HCLK cycle is inserted …… 11111111: 255 additiona HCLK cycles are inserted
Bit 7:0	SPST	0xFC	rw	Special memory setup time This field defines the address seti[time when access to NAND Flash in a special memory. 00000000: 0 HCLK cycle is inserted 00000001: 1 additonal HCLK cycle is inserted …… 11111111: 255 additiona HCLK cycles are inserted

ECC value register 2 (XMC_ BK2ECC)

Bit	Register	Reset value	Type	Description
Bit 31:0	ECC	0x0000 0000	ro	EECC value This field contains the computed ECC value.

23 SDIO interface

SDIO introduction

The SD/SDIO MMC card host interface (SDIO) provides an interface between the AHB peripheral bus and MultiMediaCards (MMC), SD memory cards and SDIO cards.

SD memory card and SDI/O card system specifications are available through the SD card association website www.sdcard.org.

The MultiMediaCard system specifications published by the MMCA technical committee are available through the MultiMediaCard association website www.mmca.org.

Main features

- Full compatibility with SD memory card specifications version 2.0
- Full compatibility with SDI/O card specification version 2.0 and support 1-bit and 4-bit databus modes.
- Full compatibility with MultiMedia card specification version 4.2 and support 1-bit, 4-bit and 8-bit databus modes.
- Full compatibility with previous versions of MultiMedia card specifications
- DMA transfer
- Data transfer up to 50 MHz in 8-bit bus mode
- Interrupt requests

Note: The SDIO is not compatible with SPI communication mode. It supports only one SD/SDIO/MMC 4.2 card at any one time.

Communication on the bus is based on command and data transfers.

- Command: A command is a token that starts an operation. Commands are sent from the host either to a single card (addressed command) or to all conncected cards (broadcast command). Commands are transferred serially on the CMD line.
- Response: A response is a token that is sent from a card to the host as an answer to a previously received command. Responses are 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 SDIO_D data line.

The basic operation on the MMC card/SD/SDI I/O bus is the command/response structure. These types of bus operation transfer their information through the command or bus mechanism. In addition, some operations have a data token.

Data transfers to/from SD/SDIO memory cards are done in data blocks. Data blocks are always followed by CRC bit, defining single and multiple block operations. Data transfers to/from MMC are done in data blocks or streams, as shown in figure below.

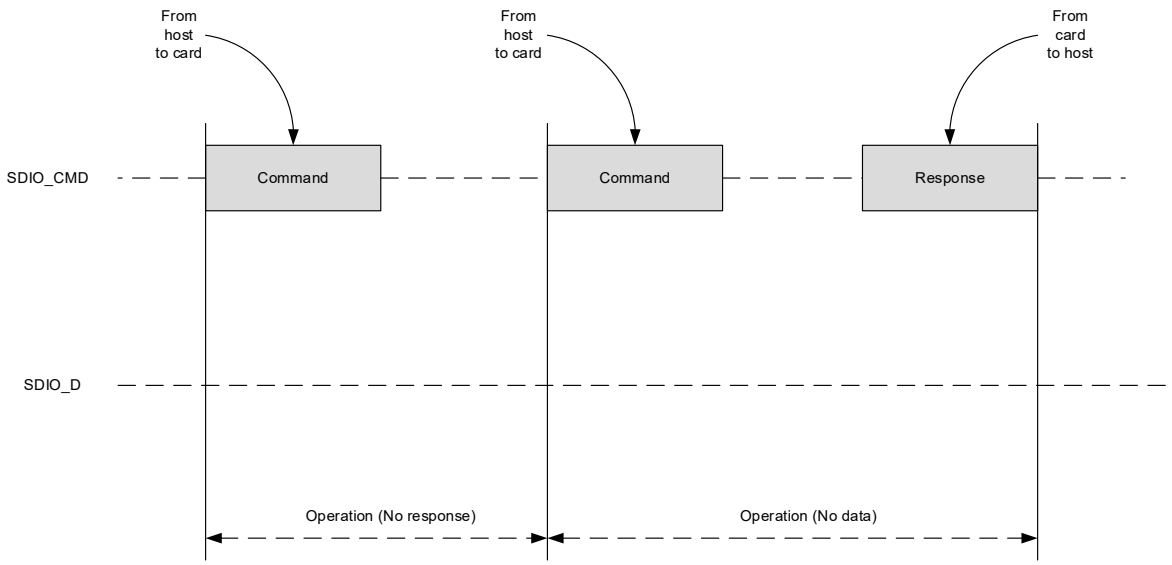


Figure 187. SDIO "no response" and "no data" operations

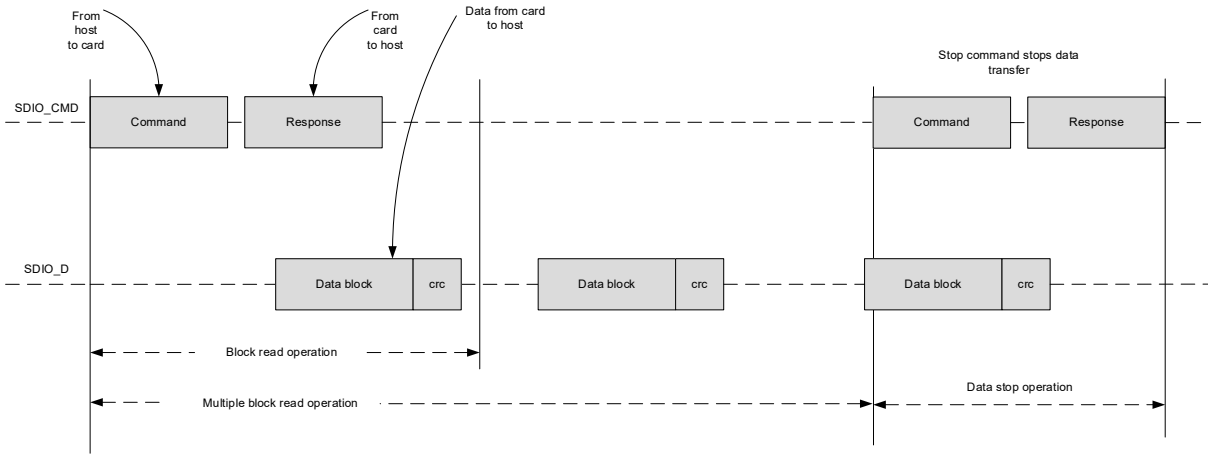


Figure 188. SDIO multiple block read operation

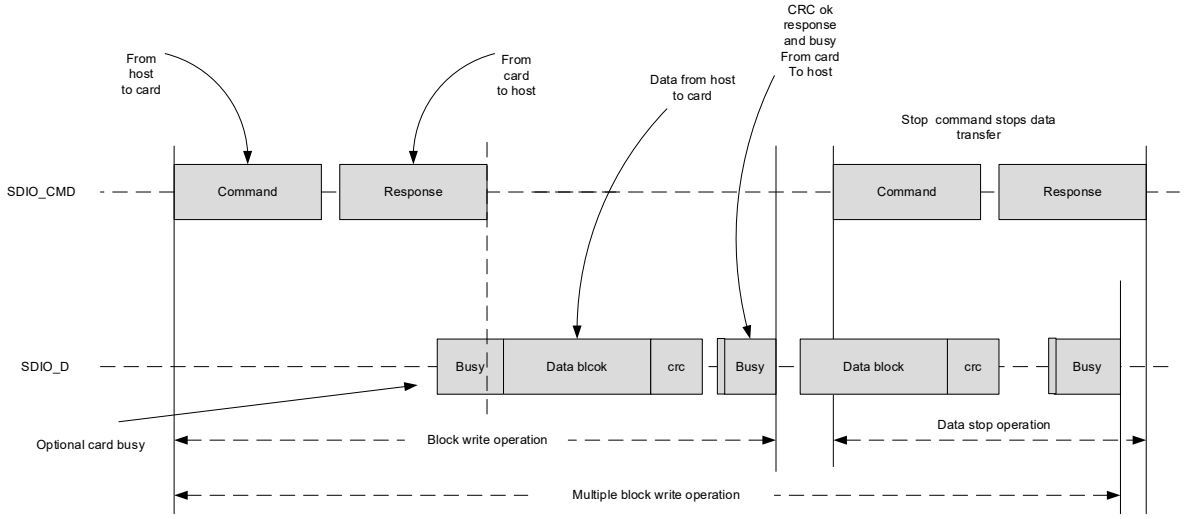


Figure 189. SDIO multiple block write operation

Note: The SDIO will not send any data as long as the Busy signal is set (SDIO_D0 pulled low).

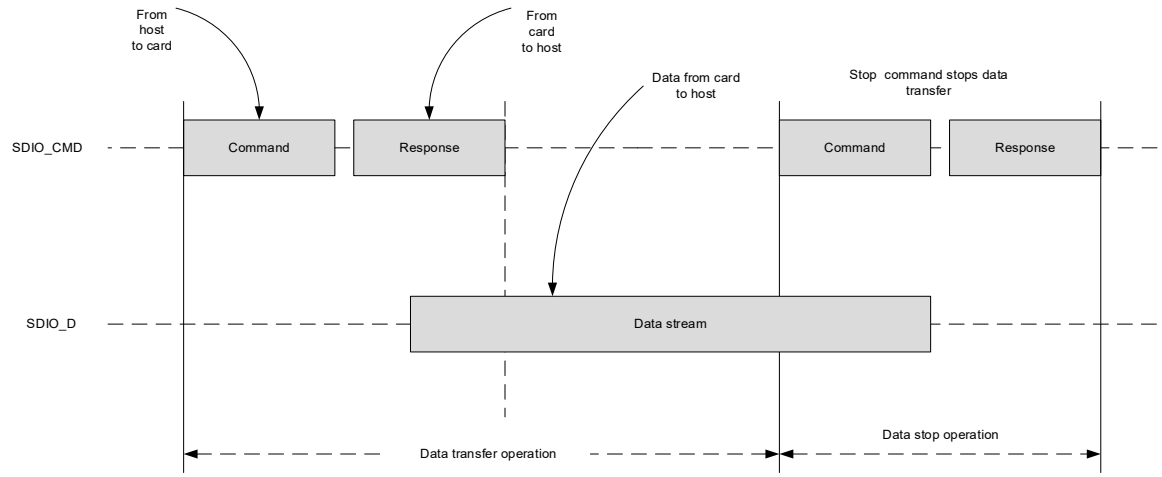


Figure 190. SDIO sequential read operation

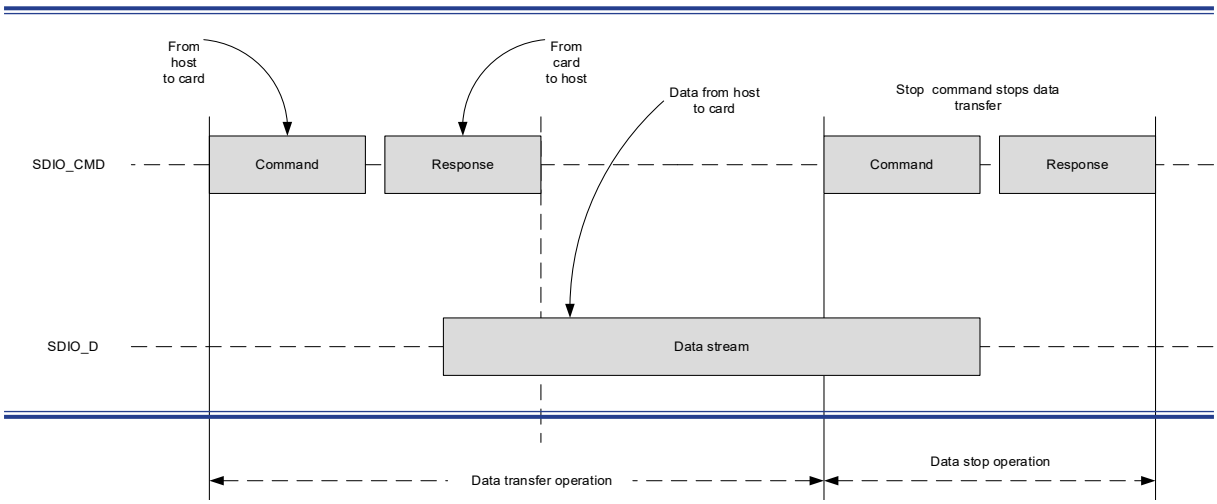


Figure 191. SDIO sequential write operation

Functional descriptions

Card functional description

All the communications between the host and the cards are controlled by the card. The host sends two different types of commands: broadcast command and addressed (point-to-point) command.

- Broadcast command: applicable to all cards, some need responses
- Addressed command: sent to the addressed card, and responses received from the card

Memory card defines two types of operational modes:

- Card identification mode
- Data transfer mode

Card identification mode

In card indentification mode, the host resets all cards, validates the operation voltage range, identifies cards and sets a relative card address (RCA) for each card on the CMD. All communications in the card identification mode use the command line (CMD).

Card identification process

The card identification process varies from card to card, and the host sends different commands. There are SD, SDI/O and MMC cards. It is possible to send a CMD5 command to identify the type of a card. If the host receives a response, it is a SDI/O card. If no response is received, the host will continue to send ACMD41 command, if a response is received, it is a SD card, otherwise a MMC card.

The card identification process is described as follows:

1. The bus is activated to confirm whether the card is connected or not. The clock frequency is at 0-400kHz during the card identification process.
2. The SDIO host sends a SD card, SDI/O card or MMC card.

3. Card Initialization
   - SD card: The SDIO host sends CMD2 (ALL_SEND_CID) to obtain its unique CID number. After receiving a response (CID number) from the card, the host will send CMD3 (SEND_RELATIVE_ADDR), instructing the card to issue the relative card address (RCA), which is shorter than the CID and is used to address the card in the data transfer mode.
   - SDI/O card: The SDIO host sends CMD3 (SEND_RELATIVE_ADDR) to instruct the card to release the relative card address (RCA), which is shorter than the CID and is used to address the card in the data transfer mode.
   - MMC card: The SDIO host sends CMD1 (SEND_OP_COND), followed by CMD2 and CMD3.
4. If the host wants to assign another RCA number, it can instruct the card to issue a new number by sending another CMD3 command. The last RCA is the actual RCA number of the card. The host repeats the card indetification process (CMD2 and CMD3 cycle of each card)

Data transfer mode

The host will enter data transfer mode after identifying all cards on the bus. In data transfer mode, the host can operate cards within the range of 0 - 50MHz. It can send CMD9 (SEND_CSD) to get data specific to a card (CSD register) such as block length and car memory size. All communications between the host and the selected card are point-to-point transferred, and the CMD bus will confirm all addressed commands as a response. Data transfer read/write can be done in data block mode or stream mode, configured by the TFRMODE bit in the SDIO_DTCTRL register. In the data stream mode, data is transferred in bytes and without CRC appended to the end of each data block.

Wide bus selection/deselection

Wide bus (4-bit bus width) operation mode is selected or deselected by using ACMD6 (SET_BUS_WIDTH). The default bus width after power-up or CMD0 (GO_IDLE_STATE) is 1 bit. The ACMD6 is only valid in a transfer state, indicating that the bus width can be changed only after a card is selected by CMD7.

Stream read/write (MultiMedia card only)

Read:

1. The host sends CMD11 (READ_DAT_UNTIL_STOP) for stream read.
2. Until the host sends CMD12 (STOP_TRANSMISSION). The stop command has an execution delay due to the serial command transmission and the data transfers stops after the end bit of the the stop command.

Write:

1. The host sends CMD20 (WRITE_DAT_UNTIL_STOP) for stream write.
2. Until the host sends CMD12 (STOP_TRANSMISSION). As the amount of data to be transferred is not determined in advance, the CRC cannot be used. When the memory range is reached, the command will be discarded by the card and remain in a transfer state, and a respons is issued by setting the ADDRESS_OUT_OF_RANGE bit.

Data block read

In block read mode, the basic unit of data transfer is a block whose maximum size (fixed length 512 bytes) is defined in the CSD (READ_BL_LEN). If the READ_BL_PARTIAL is set, smaller blocks

whose start and end addresses are entirely contained within 512 bytes may also be transmitted. A CRC is appended to the end of each block to ensuring data transfer integrity. Several commands related to data block read are as follows:

- CMD17 (READ_SINGLE_BLOCK): initiates a data block read and returns to the transfer state after the completion of the transfer.
- CMD18 (READ_MULTIPLE_BLOCK): starts a transfer of several consecutive data blocks.

Data blocks will keep transferring until the host sends CMD12(STOP_TRANSMISSION). The stop command has an execution delay due to the serial command transmission and the data transfers stops after the end bit of the the stop command.

Data block write

During block write (CMD24-27), one or more blocks of data are transferred from the host to the card with a CRC appended to the end of each block. If the CRC failed, the card indicates a failure on the SDIO_D signal line and the transferred data are discarded and not written, and all transmitted data blocks are ignored.

If the host uses partial blocks with accumulated length is not block aligned, and 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 sets the ADDRESS_ERROR bit in the SDIO_STS register and waits the stop command in a receive state while ignoring all further data transfer. If the host attempts to perform write operation on a write-protected area, the write operation will be aborted. In this case, however, the card should set the WP_VIOLATION bit.

Programming of the CID and CSD registers does not require a block length setting in advance. The transferred data is also CRC protected. If a part of the CSD or CID register is stored in the ROM, then the unchangeable part must match the corresponding part of the receive buffer. If this match failed, then the card will report an error and does not change any register contents. Some cards may require long and unpredictable times to write a block of data. After receiving a block of data and completing the CRC check, the card begins writing and holds the SDIO_D signal line low if its write buffer is full and unable to accect new data from a new WRITE_BLOCK command. The host can check the status of the card with SEND_STATUS command (CMD13) at any time, and the card will respond with its status.

The READY_FOR_DATA status bit indicates whether the car can accept new data or whether the write process in still in progress. The host can deselect the card by issuing CMD7 (select another card), which will place the card in the disconnect state and release the SDIO_D line without interrupting the write operation. when reselecting the card, it will reactivate busy indication by pulling SDIO_D line to low if the programming is still in progress and the write buffer is unavailable.

Erase

The erasable unit of the MultiMedia card and SD card is the erase group. The erase group is calculated in wirte blocks, which are the basic writble units of the card. The size of the erase group is a parameter specific to the card and defined in the CSD.

The host can erase a contiguous range of erase groups. There are three steps to start the erase process, but the commands sent by the MultiMedia card and SD card are different.

1. The host defines the start address of the range using the following command:
   - SD card: issue CMD32 (ERASE_WR_BLK_START)
   - MMC car: issue CMD35 (ERASE_GROUP_START)
2. The host defines the end address of the rang using the following command:
   - SD card: issue CMD33 (ERASE_WR_BLK_END)
   - MMD car: issue CMD36 (ERASE_GROUP_END)
3. The host starts the erase process by sending CMD38 (ERASE)

Protection management

Three write protection methods are supported in the SDIO card host module to ensure that the protected data is not erased or changed.

Mechanical write protect switch

There is a mechanical sliding switch on the side of the card to allow the user to set/clear the write protection on the card. When the sliding switch is positioned with the window open, the card is write-protected, and when the window is closed, the card is not write-protected.

Internal card write protection

Card data can be protected against write and erase. The entire card can be permanently write-protected by the manufacturer or the content provider by setting the permanent or temporary write-protect bits in the CSD. For cards that support write protection of groups of sectors by setting the WP_GRP_ENABLE bit in the CSD, part of the data can be protected, and the write protection can be changed by the application. The SET_WRITE_PROT commands set the write protection of the addressed group. The CLR_WRITE_PROT commands clear the write protection of the addressed group. The SEND_WRITE_PROT command is similar to a single block read command. The card sends a data block containing 32 write protection bits (representing 32 write protect groups starting at the specified address) followed by 16 CRC bits. The address filed in the write protect commands is a group address in byte units.

Password protect

The password protection function enables the SDIO card host to lock and unlock a card with a password. The password is stored in the 128-bit PWD register and its size is set in the 8-bit PWD_LEN register. These registers are nonvolatile so that the content is not erased after power-off.

Locked cards can support certain commands, indicating that the host is allowed to reset, initialize and query for status, but not allowed to access data on the card. When the password is set (PWD_LEN is nonzero value), the card is locked automatically after power-up.

Like the CSD and CID register write commands, the lock/unlock commands are valid only in a transfer state. The command does not include an address parameter and thus the card must be selected before using it.

The card lock/unlock commands have the structure and bus transaction types of a regular single-block write command. The transferred data block include all the information required for the command (the password setting mode, PWD content and card lock/unlock indication). The command data block size is defined by the SDIO card host module before it sents the card lock/unlock command. The lock/unlock command structure is shown below:

Table 105. Lock/unlock command structure

Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	Reserved(set to 0)				ERASE	LOCK_ UNLOCK	CLR_ PWD	SET_ PWD
1	PWDS_LEN							
2	password data							
…								
PWDS_LEN+1								

- ERASE: Setting it will force an erase operation. All other bits must be zero, and only the command byte is sent.
- LOCK_UNLOCK: Setting it will lock the card. Clearing it will unlock the card. LOCK_UNLOCK can be set simultaneously with SET_PWD, but not with the CLR_PWD.
- CLR_PWD: Setting it will clear the password data.
- SET_PWD: Setting this bit will save the password data to memory.
- PWD_LEN: This bit defines the length of the password in bytes. When the password is changed, the length is the combination of the old and new passwords.
- PWD: password (new or currently used, depending on the command)

The data block size should be defined by the host before it sends the card lock/unlock command. The block length should be equal to or greater than the data structure required for the lock/unlock command.

The following sections list the command sequence to set/clear a password, lock/unlock a card, and force an erase operation.

Setting the password

1. Select a card using CMD7 (SELECT/DESELECT_CARD), if none is selected previously
2. Define the block length with CMD16(SET_BLOCKLEN) to send in the 8-bit card lock/unlock mode, 8-bit PWD_LEN, and the number of bytes of the new password. When a password is replaced, the block size must take into account the length of both the old and the new passwords sent with the command.
3. Send CMD42(LOCK/UNLOCK) with the appropriate data block size on the data line including the 16-bit CRC. The data block indicates the operation mode (SET_PWD=1), the length (PWD_LEN) and the password (PWD). When a password replacement is done, the length value includes the length of the both passwords, the old and the new one, and the PWD field includes the old password (currently used) followed by the new password.
4. When the old password is matched, the new password and its size are saved into the PWD and PWD_LEN fields, respectively. When the old password sent is not correct (in size and/or content), the LOCK_UNLOCK_FAILED error bit is set in the SDIO_STS register, and the old password is not changed. When the old password sent is correct (in size and content), the given new password and its size will be saved in the PWD and the PWD_LEN registers, respectively.

The password length field (PWD_LEN) indicates whether a password is currently set or not. When the field is a zero value, it means that a password is not set currently. When the field is nonzero, it means that a password is set and the card locks itself after power-up. It is possible to lock the card immediately in the current power session by setting the LOCK_UNLOCK bit or sending an additional card lock command.

Clearing the password

1. Select a card using CMD7 (SELECT/DESELECT_CARD), if none is selected previously.
2. Define the block length with CMD16(SET_BLOCKLEN) to send in the 8-bit card lock/unlock mode, 8-bit PWD_LEN, and the number of bytes of the new password.
3. Send CMD42(LOCK/UNLOCK) with the appropriate data block size on the data line including the 16-bit CRC. The data block indicates the operation mode (SET_PWD=1), the length (PWD_LEN) and the password (PWD). When a password is matched, the PWD field is cleared and PWD_LEN is set to 0. If the password sent does not correspond to the expected password (in size and/or content), the LOCK_UNLOCK_FAILED error bit is set in the SDIO_STS register, and the password is not changed.

Locking a card

1. Select a card using CMD7 (SELECT/DESELECT_CARD), if none is selected previously.
2. Define the block length with CMD16(SET_BLOCKLEN) to send in the 8-bit card lock/unlock mode, 8-bit PWD_LEN, and the number of bytes of the new password.
3. Send CMD42(LOCK/UNLOCK) with the appropriate data block size on the data line including the 16-bit CRC. The data block indicates the operation mode (SET_PWD=1), the length (PWD_LEN) and the password (PWD).
4. When a password is matched, the card is locked and CARD_IS_LOCKED is set in the SDIO_STS register. If the password sent does not correspond to the expected password (in size and/or content), the LOCK_UNLOCK_FAILED error bit is set in the SDIO_STS register, and the lock fails.

If the password is previously set (PWD_LEN is not 0), the card is locked automatically after power-on reset. An attempt to lock a locked card or to lock a card that does not have a password fails and the LOCK_UNLOCK_FAILED error bit is set in the SDIO_STS register.

Unlocking a card

1. Select a card using CMD7 (SELECT/DESELECT_CARD), if none is selected previously
2. Define the block length with CMD16(SET_BLOCKLEN) to send in the 8-bit card lock/unlock mode, 8-bit PWD_LEN, and the number of bytes of the new password.
3. Send CMD42(LOCK/UNLOCK) with the appropriate data block size on the data line including the 16-bit CRC. The data block indicates the operation mode (SET_PWD=1), the length (PWD_LEN) and the password (PWD).
4. When a password is matched, the card is unlocked and CARD_IS_LOCKED is cleared in the SDIO_STS register. If the password sent does not correspond to the expected password (in size and/or content), the LOCK_UNLOCK_FAILED error bit is set in the SDIO_STS register, and the lock remains locked.

The unlocking function is valid only for the current power session. The card is locked automatically on the next power-up as long as the PWD field is not cleared.

An attempt to unlock an unlocked card fails and the LOCK_UNLOCK_FAILED error bit is set in the SDIO_STS register.

Forcing erase

If the user forgot the password (PWD content), it is possible to access the card after clearing all the data on the card. This forced erase operation will erase all card data and all password data.

1. Select a card using CMD7 (SELECT/DESELECT_CARD), if none is selected previously
2. Define the block length with CMD16(SET_BLOCKLEN) to send in the 8-bit card lock/unlock mode, 8-bit PWD_LEN, and the number of bytes of the new password.
3. Send CMD42(LOCK/UNLOCK) with the appropriate data block size on the data line including the 16-bit CRC. The data block indicates the operation mode (ERASE =1). All other bits must be zero.
4. When the ERASE bit is the only bit in the data field, all card content will be erased, including the PWD and the PWD_LEN field, and the card is no longer locked. When any other bits are not zero, the LOCK_UNLOCK_FAILED error bit is set in the SDIO_STS register, and the card data are retained, and the card remains locked.

An attempt to force erase an unlocked card fails and the LOCK_UNLOCK_FAILED error bit is set in the SDIO_STS register.

Commands and responses

Commands

Command types

Four commands are available to control the SD memory card:

1. Broadcast command: sent to all cards, no responses returned
2. Broadcast command with response: sent to all cards, responses received from all cards simultaneously
3. Addressed command: sent to the selected card, and no data transfer on the SDIO_D line
4. Addressed data transfer command: sent to the selected card, and data transfer is present on the SDIO_D line

Command description

The SDIO host module system is designed to provide a standard interface for a variety of application types. In the meantime, specific customer/application features must also be taken into account. Because of this, two types of general commands are defined in the standard: general commands (GEN_CMD) and application-specific commands (ACMD).

To use the application-specific commands, the SDIO host must send CMD55(APP_CMD) first, and waits the response from the card, which indicates that the APP_CMD bit is set and an ACMD is expected, before sending ACMD.

Table 106. Commands

CMD index	Type	Parameter	Response format	Abbreviation	Description
CMD0	bc	[31:0]=stuff bits	—	GO_IDLE_STATE	Reset all cards to idle state
CMD1	bc	[31:0]=OCR	R3	SEND_OP_COND	In idle state, request a card to send OCR register content through the CMD bus
CMD2	bcr	[31:0]=stuff bits	R2	ALL_Send_CID	Request all cards to send CID data through the CMD bus
CMD3	bcr	[31:0]=stuff bits	R6	SEND_RELATIVE_ADDR	Request a card to issue a new relative card address
CMD4	bc	[31:16]=DSR [15:0]= stuff bits	—	SET_DSR	Set the DSR register of all cards
CMD5	bcr	[31:24]Reserved [23:0] I/O OCR	R4	IO_SEND_OP_COND	Used only for the SDI/O card to query the voltage range of the required I/O card
CMD6	ac	[31:26] set to 0 [25:24] access [23:6] index [15:8] value [7:3] set to 0 [2:0] command set	R1b	SWITCH	Used only for the MMC card to switch the operation modes or modify the EXT_CSD register
CMD7	ac	[31:16]=RCA [15:0]= stuff bits	R1b	SELECT/DESELECT_CARD	This command is used to switch a card between the standby state and the send state, or between the programmed and the disconnected state. The relative card address is used to select a card. Address 0 is used to deselect the card.
CMD8 (SD)	bcr	[31:12] Reserved [11:8]operating voltage (VHS) [7:0]check mode	R7	SEND_IF_COND	Send the host power supply voltage to the SD card and check whether the card supports the voltage or not.
CMD8 (MMC)	adtc	[31:0]= stuff bits	R1	SEND_EXT_CSD	Used only for the MMC card to send its own EXT_CSD register as a data block
CMD9	ac	[31:16]=RCA [15:0]= stuff bits	R2	SEND_CSD	The selected card sends CSD (card-specific data) through the CMD bus
CMD10	ac	[31:16]=RCA [15:0]= stuff bits	R2	SEND_CID	The selected card sends CID (card flag) through the CMD bus
CMD12	ac	[31:0]= stuff bits	R1b	STOP_TRANSMISSION	Force the card to stop transmission
CMD13	ac	[31:16]=RCA [15:0]= stuff bits	R1	SEND_STATUS	Selected card sendstatus register
CMD15	ac	[31:16]=RCA [15:0]= stuff bits	—	GO_INACTIVE_STATE	Selected card switch to inactive state

Table 107. Data block read commands

CMD index	Type	Parameter	Response format	Abbreviation	Description
CMD16	ac	[31:0]=data block length	R1	SET_ BLOCKLEN	This command is used to set the length of data blocks (in bytes) for all block commands. The default value is 512 bytes.
CMD17	adtc	[31:0]=data address	R1	READ_ SINGLE_ BLOCK	Read a data block of the size set by CMD16
CMD18	adtc	[31:0]=data address	R1	READ_ MULTIPLE_ BLOCK	Continously read data from the card to the host until the STOP_ TRANSMISSION is received

Table 108. Data stream read/write commands

CMD index	Type	Parameter	Response format	Abbreviation	Description
CMD11	adtc	[31:0]=data address	R1	READ_DAT_ UNTIL_STOP	Read data stream form the card starting from a given address until the STOP_ TRANSMISSION is received.
CMD20	adtc	[31:0]=data address	R1	WRITE_DAT_ UNTIL_STOP	Read data stream form the host starting from a given address until the STOP_ TRANSMISSION is received.

Table 109. Data block write commands

CMD index	Type	Parameter	Response format	Abbreviation	Description
CMD16	ac	[31:0]=data block length	R1	SET_ BLOCKLEN	This command is used to set the length of data blocks (in bytes) for all block commands. The default value is 512 bytes.
CMD23	ac	[31:16]=set to 0 [15:0]=data block size	R1	SET_BLOCK_ COUNT	Define the number of blocks to be transferred in the data block read/write that follows
CMD24	adtc	[31:0]=data address	R1	WRITE_ BLOCK	Write a data block of the size set by the CMD16
CMD25	adtc	[31:0]=data address	R1	WRITE_ MULTIPLE BLOCK	Continuously write data blocks until the STOP_TRANSMISSION is received
CMD26	adtc	[31:0]=stuff bits	R1	PROGRAM_ CID	Program the card identification register
CMD27	adtc	[31:0]=stuff bits	R1	PROGRAM_ CSD	Program the programmable bits of the CSD

Table 110. Block-based write protect commands

CMD index	Type	Parameter	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 specified group. The properties of write protection are placed in the card-specific area (WP_GRP_SIZE).
CMD29	ac	[31:0]=data address	R1b	CLR_WRITE_PROT	If the card has write protection features, this command clears the write protection bit of the specified group.
CMD30	adtc	[31:0]=write protect data address	R1	SEND_WRITE_PROT	If the card has write protection features, this command asks the card to send the status of the write protection bits.

Table 111. Erase commands

CMD index	Type	Parameter	Response format	Abbreviation	Description
CMD32 … CMD34	Reserved. These command indexes canot be used in order to maintain backward compatibility with older versions of the MultiMedia card.				
CMD35	ac	[31:0]=data address	R1	ERASE_GROUP_START	Sets the address of the first erase group within a range to be selecte for erase
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
CMD37	Reserved. These command indexes canot be used in order to maintain backward compatibility with older versions of the MultiMedia card.				
CMD38	ac	[31:0]=stuff bits	R1b	ERASE	Erase all previously selected data blocks

Table 112. I/O mode commands

CMD index	Type	Parameter	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 specifies 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 specified register. This command accesses application-specific registers that are not defined in the MultiMedia card standard.
CMD40	bcr	[31:0]=stuff bits	R5	GO_IRQ_STATE	Place the system in the interrupt mode

Table 113. Card lock commands

CMD index	Type	Parameter	Response format	Abbreviation	Description
CMD42	adtc	[31:0]=stuff bits	R1	LOCK_UNLOCK	Sets/clears the password or locks/unlocks the card. The size of the data block is set by CMD16.

Table 114. Application-specific commands

CMD index	Type	Parameter	Response format	Abbreviation	Description
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 read a data block from the card for general-purpose/application-specific commands. The size of the data block is defined by the SET_BLOCK_LEN command.
CMD57 … CMD59	Reserved				
CMD60 … CMD63	Reserved for manufacturer				

Response formats

All responses are sent via the CMD bus. The response transmission always starts with the left bit of the bit string corresponding to the response code word. The length depends on the response type.

A response always starts with a start bit (always 0), folled by the transmission bit indicating the direction of transmission (card =0). A value denoted by – in the tables below stands for a variable entry. All responses, except the R3 response type, are protected by a CRC. Every command code word is terminated with the end bit (always 1).

R1 (normal response command)

Code length = 48 bits. The 45:40 bits indicate the index of the command to be responded to. This value is interpreted as a binary-coded number (between 0 and 63). The status of the card is coded in 32 bits. Note that if it involves a data transfer to a card, a busy signal may appear on the data line after each data block is transmitted. The host should check the busy signal after data block transfer.

Table 115. R1 response

Bit	47	46	[45:40]	[39:8]	[7:1]	0
Field width	1	1	6	32	7	1
Value	0	0	—	—	—	1
Description	Start bit	Transmission bit	Command index	Card status	CRC7	End bit

R1b

It is the same as R1 with an optional busy signal transmitted on the data line. The card may become busy after receiving these commands based on its state prior to the command reception. The host should check the busy signal.

R2 (CID & CSD registers)

Code length = 136 bits. The contents of the CID register are sent as a response to the CMD2 and CMD10 commands. The contents of the CSD register are sent as a response t the CMD9. Only the bits [127 ... 1] of the CID and CSD are transmitted, and the reserved bit [0] of these registers is replaced with the end bit of the response.

Table 116. R2 response

Bit	135	134	[133:128]	[127:1]	0
Field width	1	1	6	127	1
Value	1	0	111111	—	1
Description	Start bit	Transmission bit	Reserved	CID or CSD register	End bit

R3

Code length = 48 bits. The contents of the OCR register are sent as a response to ACMD41.

Table 117. R3 response

Bit	47	46	[45:40]	[39:8]	[7:1]	0
Field width	1	1	6	32	7	1
Value	1	0	111111	—	111111	1
Description	Start bit	Transmission bit	Reserved	OCR register	Reserved	End bit

R4 (Fast I/O)

Code length = 48 bits. The parameter field contains the RCA of the specified card, the register address to be read out or written to, and its contents.

Table 118. R4 response

Bit	47	46	[45:40]	[39:8]			[7:1]	0
Field width	1	1	6	16	8	8	7	1
Value	1	0	100111	—	—	—	—	1
Description	Start bit	Transmission bit	CMD39	RCA	Register address	Read register contents	CRC7	End bit

R4b

For SD I/O only, an SDIO card will respond with a unique SDIO response R4 after receiving the CMD5.

Table 119. R4b response

Bit	47	46	[45:40]	[39:8]					[7:1]	0
Field width	1	1	6	1	3	1	3	24	7	1
Value	1	0	—	—	—	—	—	—	—	1
Description	Start bit	Tx bit	Res.	Card ready	Numberof I/O functions	Current memory	Stuff bit	I/O OCR	Res.	End bit

R5 (interrupt request)

For MultiMedia card only. Code length = 48 bits. If the response is generated by the host, the RCA field in the parameter will be 0x0.

Table 120. R5 response

Bit	47	46	[45:40]	[39:8]		[7:1]	0
Field width	1	1	6	16	16	7	1
Value	1	0	101000	—	—	—	1
Description	Start bit	Start bit	Tx bit	RCA[31:16] of a successful card or of the host	Not defined. Maybe used for interrupt data	CRC7	End bit

R6 (interrupt request)

For SD I/O card only. this is a normal response to CMD3 by a memery device.

Table 121. R6 response

Bit	47	46	[45:40]	[39:8]		[7:1]	0
Field width	1	1	6	16	16	7	1
Value	1	0	000011	—	—	—	1
Description	Start bit	Tx bit	CMD3	RCA[31:16] of a successful card or of the host	Card status	CRC7	End bit

The card status bit [23:8] will be changed when the CMD3 is sent to an I/O-only card. In this case, the 16 bits of response are the SD I/O-only values.

- Bit 15=COM_CRC_ERROR
- Bit 14=ILLEGAL_COMMAND
- Bit 13=ERROR
- Bit [12:0]=Reserved

SDIO functional description

SDIO consists of four parts:

- SDIO adapter block: contains a control unit, command path and data path that provides all functions specific to the MMC/SD/SD I/O card such as the clock generation, command and data transfer
  - Control unit: manages and generates clock signals
  - Command path: manages command transfer
  - Data path: manages data transfer
- AHB interface: generates interrupt and DMA request signals
- Adapter register: system register
- BUF: used for data transfer

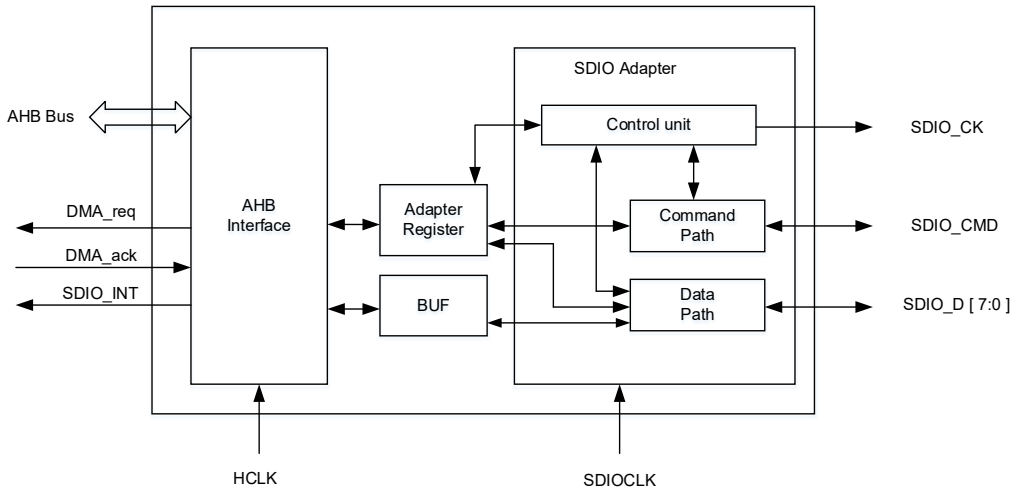


Figure 192. SDIO block diagram

SDIO adapter

SDIO_CK is a clock to the MultiMedia/SD/SDIO car provided by the host. One bit of command or data is transferred on both command and data lines with each clock cycle. The clock frequency can very between different cards and different protocols.

- MultiMedia card
  - V3.31 protocol 0 – 20 MHz
  - V4.0/4.2 protocol 0 – 50 MHz
- SD card
  - 0 – 50 MHz
- SD I/O card
  - 0 – 50 MHz

SDIO_CMD is a bidirectional command channel and used for the initialization of a card and command transfer. When the host sends a command to a card, the card will issue a response to the host. The SDIO_CMD has two operational modes:

- Open-drain mode for initialization (only for MMCV3.31 or previous)
- Push-pull mode for command transfer (SD/SD I/O card and MMC V4.2 also use push-pull drivers for initialization)

SDIO_D[7:0] is a bidirectional data channel. After initialization, the host can change the width of the data bus. After reset, the SDIO_D0 is used for data transfer by default. MMCV3.31 or previous supports only one bit of data line, so only SDIO_DO can be used.

The table below is used for the MultiMedia card/SD/SD I/O card bus:

Table 122. SDIO pin definitions

Pin	Direction	Description
SDIO_CK	Output	MultiMedia card/SD/SDIO card clock. This pin is the clock from the host to a card.
SDIO_CMD	Bidirectional	MultiMedia card/SD/SDIO card command. This pin is the bidirectional command/response signal .
SDIO_D[7:0]	Bidirectional	MultiMedia card/SD/SDIO card data. This pin is the bidirectional databus.

Control unit

The control unit consists of a power management sub-unit and a clock management sub-unit. The power management subunit is controlled by the SDIO_PWRCTRL register. The PS bit is used to define power-up/power-off state. During the power-off and power-up phases, the power management subunit will disable the card bus output signals. The clock management subunit is controlled by the SDIO_CLKCTRL register where the CLKDIV bit is used to define the divider factor between the SDIOCLK and the SDIO output clock. If BYPSEN = 0, the SDIO_CK output signal is driven by the SDIOCLK divided according to the CLKDIV bit; if BYPSEN = 1, the SDIO_CK output signal is directly driven by the SDIOCLK. The HFCEN is set to enable hardwar flow control feature in order to avoid the occurrence of an error at transmission underflow or reception overflow. The PWRSVEN bit can be set by software to enable power save mode, and the SDIO_CK can be output only when the bus is active.

Command path

The command path unit sends commands to and receives responses from the cards. When the CCSMEN bit is set in the SDIO_CMDCTRL register, a command transfer starts. First sends a command to a card by the SDIO_CMD, the command length is 48 bits. The data on the SDIO_CMD is synchronized with the rising edge of the SDIO_CK. A block of data is transferred with each SDIO_CK, including start bit, transfer bit, command index defined by the SDIO_CMDCTRL_CMDIDX bit, parameters defined by the SDIO_ARG, 7-bit CRC and end bit. Then receives responses from the card. There are two response types: 48-bit short response and 136-bit long response. Both use CRC erro check. The received responses are saved in the area from SDIO_RSP1 to SDIO_RSP4. The command path can generate command flag, which can be defined by the SDIO_STS register.

Table 123. Command formats

Bit	47	46	[45:40]	[39:8]	[7:1]	0
Width	1	1	6	32	7	1
Value	0	1	—	—	—	1
Description	Start bit	Tx bit	Command index	Parameter	CRC7	End bit

Response: A response is sent from a specified card to the host (or synchronously from all cards for MMCV3.31 or previous), as an answer to a previously received command. Responses are transferred serially on the CMD line.

Table 124. Short response format

Bit	47	46	[45:40]	[39:8]	[7:1]	0
Width	1	1	6	32	7	1
Value	0	0	—	—	—	1
Description	Start bit	Tx bit	Command index	Parameter	CRC7 (or 1111111)	End bit

Table 125. Long response format

Bit	135	134	[133:128]	[127:1]	0
Width	1	1	6	127	1
Value	0	0	111111	—	1
Description	Start bit	Tx	Reserved	CID or CSD (including internal CRC7)	End bit

Table 126. Command path status flags

Flag	Description
CMDRSPCMPL	A response is already received (CRC OK)
CMDFAIL	A command response is already received (CRC fails)
CMDCMPL	A command is sent (does not require a response)
CMDTIMEOUT	Command response timeout (64 SDIO_CK cycles)
DOCMD	Command transfer is in progress

Command channel state machine (CCSM)

When the CCSMEN bit is set in the SDIO_CMDCTR register, command transfer starts. When the command has been sent, the command channel state machine (CCSM) will set the status flags and enters the idle state if a response is not required. When a response is received, the received CRC code and the internally generated CRC code are compared, and the appropriate status flags are set.

- The CCSM remains in the idle state for at least 8 SDIO_CK cycles to meet the Ncc (the minimum delay beteen two host commands) and NRC (the minimum delay between the host command and the card response).
- When the wait state is entered, the command timer is enabled. if the NCR timeout (resonse time to a command), that is, 64 SDIO_CK periods, is reached before the CCSM moves to the receive state, the timeout flag is set (CMDTIMEOUT) and the idle state is entered.

If the interrupt bit is set in the command register, the timer is disabled and the CCSM waits for an interrupt request from one card. If the pending bit is set in the command register, the CCSM will enter pend state and wait for a CmdPend signal from the data path subunit. When detecting the CmdPend, the CCSM goes to the send state, which triggers the data counter to send the stop command.

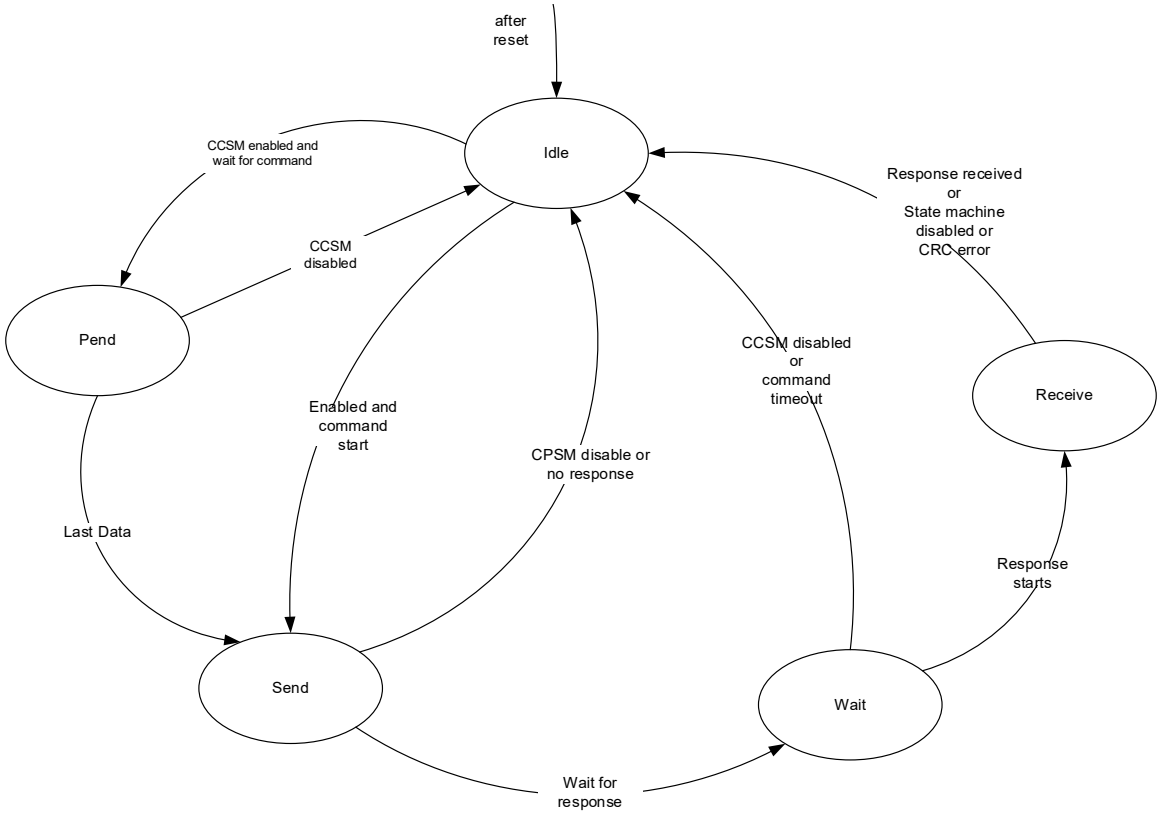


Figure 193. Command channel state machine (CCSM)

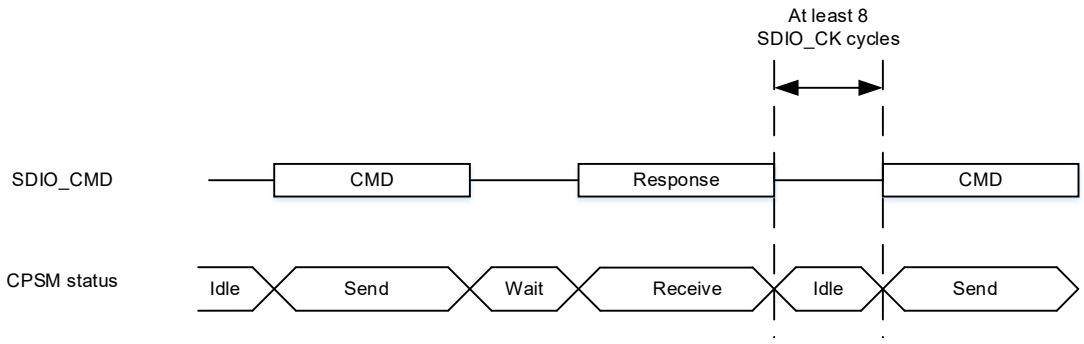


Figure 194. SDIO command transfer

Data path

The data path subunit transfers data between the host and the cards. The databus width can be configured using the BUSWS bit in the SDIO_CLKCTRL register. By default, only the SDIO_DO signal line is used for transfer. Only one bit of data is transferred with each clock cycle. If the 4-bit wide bus mode is selected, four bits are transferred per clock cycle over the SDIO_D[3:0] signal line. If the 8-bit wide bus mode is selected, eight bits are transferred per clock cycle over the SDIO_D[7:0] signal line. The TFRDIR bit is set in the SDIO_DTCTR register to define the transfer direction. If TFRDIR=0, it indicates that the data is transferred from the controller to the card; if TFRDIR=1, it indicates that the data is transferred from the card to the controller. The TFRMODE bit can be used to select block data transfer or stream transfer for the MultiMedia card. If the TFREN bit is set, data transfer starts. Depending on the TFRDIR bit, the DCSM enters Wait_S or Wait_R state.

Data channel state machine (DCSM)

The DCSM has seven states, in send and receive mode, as shown in the figure below:

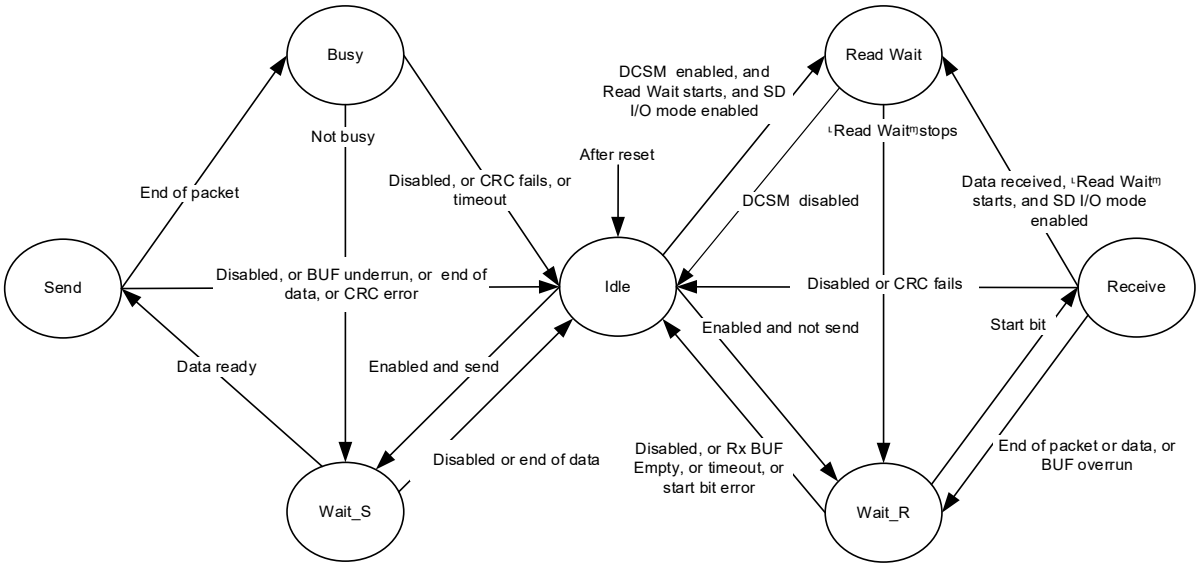


Figure 195. Data channel state machine (DCSM)

Send mode

- Idle: The data channel is inactive, either in the Wait_S or the Wait_R state.
- Wait_S: Waits until the BUF flag becomes empty or the data transmission is completed. The DCSM must remain in the Wait_S state for at leat two clock periods to meet the N_{WR} timing requirements where the N_{WR} is the interval between the reception of the card response and the start of the data transfer from the host.
- Send: The DCSM sends data to a card, and the data transfer mode can be either block or stream, depending on the SDIO_DTCTRL_TFRMODE bit. If an overflow error occurred, the DCSM then moves to the idle state
- Busy: The DCSM waits for the CRC flag. If the DCSM receives a correct CRC status and is not busy, it will enter the Wait_S state. If it does not receive a correct CRC status or a timeout occurs while the DCSM is in the busy state, a CRC fail flag or timeout flag is generated.
- Wait_R: The start bit of the Wait_R. If a timeout occurs before it detects a start bit, the DCSM moves to the idle state and generates a timeout flag.
- Receive: Data is received from a card and writtento the BUF. The data transfer mode can be either block or stream, depending on the SDIO_DTCTRL _TFRMODE bit. If an overflow error occurs, it then returns to Wait_R state.

Table 127. Data token formats

Description	Start bit	Data	CRC16	End bit
Block data	0	—	Y	1
Stream data	0	—	N	1

Data BUF

The data BUF contains a transmit and receive unit. It is a 32-bit wide and 32-word deep data buffer. Becase the data BUF operates in the AHB clock domain (HCLK), all signals connected to the SDIO clock domain (SDIOCLK) are resynchronized.

- Transmit BUF: Data can be written to the transmit BUF via the AHB interface when the SDIO transmission feature is enabled.
  The transmit BUF has 32 sequential addresses. It contains a data output register that holds the data word pointed by the read pointer. When the data path has loaded its shift register, it moves its read pointer to the next data and outputs data.
  If the transmit BUF is disabled, all status flags are inactive. The data path sets the DOTX when it transmits data.
- Receive BUF: When the data path receives a data word, it will write the data to the BUF. The write pointer is incremented automatically after the end of the wrie operation. On the other side, a read pointer always points to the current data in the BUF. If the receive BUF is disabled, all status flags are cleared, and the read and write pointers are reset as well. The data path sets the DORX when it receives data.

SDIO AHB interface

The AHB interface generates the interrupt and DMA requests, and access the SDIO interface registers and the data BUF.

SDIO interrupts

The interrupt logic generats an interrupt request when one of the selected status flags is hight. The SDIO_INTEN register is used to select the conditions that will generat an interrupt.

SDIO/DMA interface: data transfer pocess between the SDIO and memory

In the following examples, data is transferred from the host to the card. the SDIO BUF is filled with data stored in a memory through the DMA controller.

1. Card identification process
2. Increase the SDIO_CK frequency
3. Select a card by sending CMD7
4. Enable the DMA2 controller and clear all interrupt flag bits, configure the DMA2 channel4 source address register as the memory buffer's base address, and the DMA2 channel4 destination address register as the SDIO_BUF register address. Then configure the DMA2 channel4 control register (memory increment, non-peripheral increment, and peripheral and source data width is word width). Finally enable DMA2 channel4.
5. Send CMD24 (WRITE_BLOCK) as follows:

   Program the SDIO data length register (SDIO_DTLEN), the BLKSIZE bit in the SDIO data control register (SDIO_DTCTRL), and the SDIO parameter register (SDIO_ARG) with the address of the card where data is to be transferred, and program the SDIO command register (SDIO_CMD), enable the CCSMEN bit, wait for SDIO_STS[6]=CMDRSPCMPL interrupt, and then program the SDIO data control register (SDIO_DTCTRL): TFREN=1 (enable the SDIO card host to send data), TFRDIR=0 (from the controller to the card), TFRMODE=0 (block data transfer), DMAEN=1 (enable DMA), BLKSIZE=9 (512 bytes), and wait from SDIO_STS[10]=DTBLKCMPL.
6. Check that no channesl are still enabled by confirming the DMA channel enable SDIO status register (SDIO_STS)

Hardware flow control

The HFCEN bit is set in the SDIO_CLKCTRL register to enable hardware flow control, which is sued to avoid BUF underflow and overflow errors. Read/write access to the BUF is still active even if flow control is enabled.

SDIO I/O card-specific operations

The SDIO can support the following operations (except read suspend, for it does not require specific hardware operation) when the SDIO_DTCTRL[11] is set.

SDIO read wait operation by SDIO_D2 siganl lines

The optional read wait operation is used only for SD card 1-bit or 4-bit mode. The read wait operation can instruct the host to stop data transfer temporarily while the host is reading from multiple registers (IO_RW_EXTENDED, CMD53), and also allows the host to send commands to other functions in the SD I/O device in order to start a read wait process after the reception of the first data block. The detailed process as follows:

- Enable data path (SDIO_DTCTRL[0] = 1)
- Enalbe SDIO-specific operation (SDIO_DTCTRL[11] = 1)
- Start read wait (SDIO_DTCTRL[10]=0 and SDIO_DTCTRL[8]=1)
- Data direction is from a card to the SDIO host (SDIO_DTCTRL[1]=1)

- The data unit in the SDIO adapter will enter read wait state, and drive the SDIO_D2 to 0 after 2 SDIO_CK cycles
- The data unit starts waiting to receive data from a card. the DCSM will not enter read wait even if read wait start is set. The read wait process will start after the CRC is received. The RDWTSTOP has to be cleared to start a new read wait operation.

Durint the read wait period, the SDIO host can detect the SDIO interrupts over the SDIO_D1.

SDIO read wait operation by stopping clock

If the SDIO card does not support the mentioned above read wait operation, the SDIO can enter a read wait by stopping SDIO_CK, described as follows:

- Enable data path (SDIO_DTCTRL[0] = 1)
- Enalbe SDIO-specific operation (SDIO_DTCTRL[11] = 1)
- Start read wait (SDIO_DTCTRL[10]=0 and SDIO_DTCTRL[8]=1)

The DCSM stos the clock two SDIO_CK cycles after the end bit of the current received data block and starts the clock again after the read wait end bit is set.

Note that as the SDIO_CK is stopped, the SDIO host cannot send any command to the card. the SDIO host can detect the SDIO interrupt over the SDIO_D1.

SDIO suspend/resume operation (write and read operation suspend)

To free the bus to provide higher-priority transfers for other functions or memories, the host can suspend data transfer to certain functions or memories. As soon as the higher-priority transfer is completed, the previous transfer operation will restart at the suspended location.

While sending data to a card, the SDIO can suspend the write operation. the SDIO_CMD[11] bit is set and indicates to the CCSM that the current command is a suspend command. The CCSM analyzes the response and when a ACK signal is received from the card (suspend accepted), it acknowledges the DCSM that enters the idle state after receiving the CRC of the current data block.

SDIO interrupts

There is a pin with interrupt feature on the SD interface in oder to enable the SD I/O card to interrupt the MultiMedia card/SD module. In 4-bit SD mode, this pin is SDIO_D1. The SD I/O interrupts are detected when the level is active. In other words, the interrupt signal line must be active (low) before it is recognized and responded by the MultiMedia card/SD module, and will remain inactive (high) at the end of the interrupt routine.

When the SDIO_DTCTRL[11] bit is set, the SDIO interrupts are detected on the SDIO_D1 signal line.

SDIO registers

The device communicates with the system through 32-bit control registers accessible via AHB.

The peripheral registers must be accessed by words (32-bit).

Table 128. A summary of the SDIO registers.

Register	Offset	Reset value
SDIO_PWRCTRL	0x00	0x0000 0000
SDIO_CLKCTRL	0x04	0x0000 0000
SDIO_ARG	0x08	0x0000 0000
SDIO_CMD	0x0C	0x0000 0000
SDIO_RSPCMD	0x10	0x0000 0000
SDIO_RSP1	0x14	0x0000 0000
SDIO_RSP2	0x18	0x0000 0000
SDIO_RSP3	0x1C	0x0000 0000
SDIO_RSP4	0x20	0x0000 0000
SDIO_DTTMR	0x24	0x0000 0000
SDIO_DTLEN	0x28	0x0000 0000
SDIO_DTCTRL	0x2C	0x0000 0000
SDIO_DTCNTR	0x30	0x0000 0000
SDIO_STS	0x34	0x0000 0000
SDIO_INTCLR	0x38	0x0000 0000
SDIO_INTEN	0x3C	0x0000 0000
SDIO_BUFCNTR	0x48	0x0000 0000
SDIO_BUF	0x80	0x0000 0000

SDIO power control register (SDIO_ PWRCTRL)

Bit	Register	Reset value	Type	Description
Bit 31:2	Reserved	0x0000 0000	resd	Kept at its default value.
Bit 1:0	PS	0x0	rw	Power switch These bits are set or cleared by software. They are used to define the current status of the card clock. 00: Power-off, the card clock is stopped 01: Reserved 10: Reserved 11: Power-on, the card clock is started

Note: Write access to this register is not allowed within seven HCLK clock periods after data is written.

SDIO clock control register (SDIO_ CLKCTRL)

The SDIO_CLKCTRL register controls the SDIO_CK output clock.

Bit	Register	Reset value	Type	Description
Bit 31:17	Reserved	0x0000	resd	Kept at its default value.
Bit 16:15	CLKDIV	0x0	rw	Clock division This field is set or cleared by software. It defines the clock division relations between the SDIOCLK and the SDIO_CK: SDIO_CK frequency=SDIOCLK / [CLKDIV[9:0] + 2].
Bit 14	HFCEN	0x0	rw	Hardware flow control enable This bit is set or cleared by software. 0: Hardware flow control disabled 1: Hardware flow control enabled Note: When hardware flow control is enabled, refer to the SDIO_STS register for the meaning of the TXBUF_E and RXBUF_F interrupt signals.
Bit 13	CLKEGS	0x0	rw	SDIO_CK edge selection This bit is set or cleared by software. 0: SDIO_CK generated on the rising edge of the master clock SDIOCLK 1: SDIO_CK generated on the falling edge of the master clock SDIOCLK
Bit 12:11	BUSWS	0x0	rw	Bus width selection This bit is set or cleared by software. 00: Default bus mode, SDIO_D0 used 01: 4-bit bus mode,SDIO_D[3:0] used 10: 8-bit bus mode, SDIO_D[7:0] used
Bit 10	BYPSEN	0x0	rw	Clock divider bypass enable bit This bit is set or cleared by software. When disabled, the SDIO_CK output signal is driven by the SDIOCLK that is divided according to the CLKDIV value. When enabled, the SDIO_CK output signal is directly driven by the SDIOCLK. 0: Clock divider bypass disabled 1: Clock divider bypass enabled
Bit 9	PWRSVEN	0x0	rw	Power saving mode enable This bit is set or cleared by software. When disabled, the SDIO_CK is always output; when enabled, the SDIO_CK is only output when the bus is active. 0: Power saving mode disabled 1: Power saving mode enabled
Bit 8	CLKOEN	0x0	rw	Clock output enable This bit is set or cleared by software. 0: Clock output disabled 1: Clock output enabled
Bit 7:0	CLKDIV	0x00	rw	Clock division This field is set or cleared by software. It defines the clock division relations between the SDIOCLK and the SDIO_CK: SDIO_CK frequency=SDIOCLK / [CLKDIV[9:0] + 2].

Note: 1. While the SD/SDIO card or MultiMedia car is in identification mode, the SDIO_CK freqeuency must be less than 400kHz.
2. When all card are assigned with relative card addresses, the clock frequency can be changed to the maxmimum card frequency.

3. This register cannot be written within seven HCLK clock periods after data is written. The SDIO_CK can be stopped during the read wait period for SD I/O cards. In this case, the SDIO_ CLKCTRL register does not control the SDIO_CK.

SDIO argument register (SDIO_ARG)

The SDIO_ARG register contains 32-bit command argument, which is sent to a card as pat of a command.

Bit	Register	Reset value	Type	Description
Bit 31:0	ARGU	0x0000 0000	rw	Command argument Command argument is sent to a card as part of a command. If a ommand contains an argument, it must be loaded into this register before writing a command to the command register.

SDIO command register (SDIO_CMD)

The SDIO_CMD register contains the command index and command type bits. The command index is sent to a card as part of a command. The command type bits control the command channel state machine (CCSM).

Bit	Register	Reset value	Type	Description
Bit 31:12	Reserved	0x00000	resd	Kept at its default value.
Bit 11	IOSUSP	0x0	rw	SD I/O suspend command This bit is set or cleared by software. If this bit is set, the command to be sent is a suspend command (used for SDIO only). 0: SD I/O suspend command disabled 1: SD I/O suspend command enabled
Bit 10	CCSMEN	0x0	rw	Command channel state machine (CCSM) enable bit This bit is set or cleared by software. 0: Command channel state machine disabled 1: Command channel state machine enabled
Bit 9	PNDWT	0x0	rw	CCSM Waits for ends of data transfer (CmdPend internal signal) This bit is set or cleared by software. If this bit is set, the CCSM waits for the end of data transfer before it starts sending a command. 0: Disabled 1: Enabled
Bit 8	INTWT	0x0	rw	CCSM waits for interrupt request This bit is set or cleared by software. If this bit is set, thE CCSM disables command timeout and waits for an interrupt request. 0: Disabled 1: Enabled

Bit	Register	Reset value	Type	Description
Bit 7:6	RSPWT	0x0	rw	Wait for response bits This bit is set or cleared by software. This bit indicates whether the CCSM is to wait for a response, and if yes, it will indicates the response type. 00: No response 01: Short response 10: No response 11: Long response
Bit 5:0	CMDIDX	0x00	rw	Command index The command indes is sent to a card as part of a command.

Note: 1. This register cannot be written withing sevel HCLK clock periods after data is written.
2. MultiMedia card can sent two types of responses: 48-bit short response or 136-bit short response. The SD card and SD I/O card can send only short responses, and the argument can vary according to the type of response. The software will distinguish the tpe fo response accorind to the command sent.

SDIO command response register (SDIO_RSPCMD)

The SDIO_RSPCMD register contains the command index of the last command response received. If the command response transmiision does not contain the command index (long or OCR response), the SDIO_RSPCMD field is unknon, alghout it should have contained 111111b (the value of the reserved field from a response)

Bit	Register	Reset value	Type	Description
Bit 31:6	Reserved	0x0000000	resd	Kept at its default value.
Bit 5:0	RSPCMD	0x00	ro	Response command index This field contains the command indes of the command response received.

SDIO response 1..4 register (SDIO_RSPx)

The SDIO_RSPx (x=1..4) register contains the status of a card, which is parf ot the response received.

Bit	Register	Reset value	Type	Description
Bit 31:0	CARDSTSx	0x0000 0000	ro	See Table 92

The card status size is 32 or 127 bits, dependion on the response type.

Table 129. Response type and SDIO_RSPx register

Register	Short response	Long response
SDIO_RSP1	Card status [31:0]	Card status [127:96]
SDIO_RSP2	Unused	Card status [95:64]
SDIO_RSP3	Unused	Card status [63:32]
SDIO_RSP4	Unused	Card status [31:1]

The most significant bit of the card status is always received first. The least significant bit of the SDIO_RSP4 register is always 0.

SDIO data timer register (SDIO_DTTMR)

The SDIO_DTTMR register contains the data timeout period in the unit of card bus clock periods. A counter loads the value from the SDIO_DTTMR register and starts decrementing when the DCSM enters the Wait_R or busy state. If the counter reaches 0 while the DCSM is in either of these states, a timeout status flag will be set.

Bit	Register	Reset value	Type	Description
Bit 31:0	TIMEOUT	0x0000 0000	rw	Data timeout period Data timeout period in card bus clock cycles.

Note: A data transfer must be written to the SDIO_DTCNTR and the SDIO_DTLEN register before being written to the SDIO data control register (SDIO_DTCTRL).

SDIO data length register (SDIO_DTLEN)

The SDIO_DTLEN register contains the number of data bytes to be transferred. the value is loaded into the data counter when data transfer starts.

Bit	Register	Reset value	Type	Description
Bit 31:25	Reserved	0x00	resd	Kept at its default value.
Bit 24:0	DTLEN	0x0000000	rw	Data length value Number of data bytes to be transferred.

Note: For a block data transfer, the value in the SDIO_DTLEN msut be a multiple of the block data size.

A data transfer must be written to the SDIO_DTCNTR and the SDIO_DTLEN register before being written to the SDIO data control register (SDIO_DTCTRL).

SDIO data control register (SDIO_DTCTRL)

The SDIO_DTCTRL register controls the data channel statue machine (DCSM).

Bit	Register	Reset value	Type	Description
Bit 31:12	Reserved	0x00000	resd	Kept at its default value.
Bit 11	IOEN	0x0	rw	SD I/O enable functions This bit is set or cleared by software. If the bit is set, the DCSM performs an SD IO card-specific operation. 0: Disabled 1: Enabled
Bit 10	RDWTMODE	0x0	rw	Read wait mode This bit is set or cleared by software. If disabled, the SDIO_D2 contols the read wait; if enabled, the SDIO_CK controls the read wait. 0: Disabled 1: Enabled
Bit 9	RDWTSTOP	0x0	rw	Read wait stop This bit is set or cleared by software.While the the RDWTSTART is set, If this bit is set, it indicates that read wait is stopped; if this bit cleared, it indicates that the read wait is in progress. 0: Read wait is in progress if the RDWTSTART is set 1: Read wait is stopped if the RDWTSTART is set

Bit	Register	Reset value	Type	Description
Bit 8	RDWTSTART	0x0	rw	Read wait start This bit is set or cleared by software. When this bit is set, read wait starts; when this bit is cleared, no ations occurs. 0: Read wait disabled 1: Read wait enabled
Bit 7:4	BLKSIZE	0x0	rw	Data block size This bit is set or cleared by software. This field defines the length of data block when the block data transfer is selected. 0000: block length = 2^0 = 1 byte 0001: block length = 2^1 = 2 bytes 0010: block length = 2^2 = 4 bytes 0011: block length = 2^3 = 8 bytes 0100: block length = 2^4 = 16 bytes 0101: block length = 2^5 = 32 bytes 0110: block length = 2^6 = 64 bytes 0111: block length = 2^7 = 128 bytes 1000: block length = 2^8 = 256 bytes 1001: block length = 2^9 = 512 bytes 1010: block length = 2^{10} = 1024 bytes 1011: block length = 2^{11} = 2048 bytes 1100: block length = 2^{12} = 4096 bytes 1101: block length = 2^{13} = 8192 bytes 1110: block length = 2^{14} = 16384 bytes 1111: Reserved
Bit 3	DMAEN	0x0	rw	DMA enable bit This bit is set or cleared by software. 0: Disabled 1: Enabled
Bit 2	TFRMODE	0x0	rw	Data transfer mode selection This bit is set or cleared by software. If this bit is set, it indicates stread data transfer; if this bit cleared, it indicates block data transfer. 0: Disabled 1: Enabled
Bit 1	TFRDIR	0x0	rw	Data transfer direction selection This bit is set or cleared by software. If this bit is set, data transfer is from a card to a controller; if this bit is cleared, data transfer is from a controlle to a card. 0: Disabled 1: Enabled
Bit 0	TFREN	0x0	rw	Data transfer enabled bit This bit is set or cleared by software. If this bit is set, data transfer starts. The DCSM enters the Wait_S or Wait_R state, depending on the direction bit TFRDIR. The DCSM goes to the read wait state if the RDWTSTART bit is set from the beginning of the transfer. It is not necessary to clear the enable bit after the end of data transfer but the SDIO_DTCTRL must be updated to enable a new data transfer. 0: Disabled 1: Enabled

Note: This register cannot be written within seven HCLK clock periods after data is written.

SDIO data counter register (SDIO_DTCNTR)

The SDIO_DTCNTR register loads the value from the SDIO_DTLEN register when the DCSM moves from the idle state to the Wait_R or Wait_S state. During the data transfer, the counter value decrements to 0, and then the DCSM enters the idle state and sets the data status end flag bit DTCMPL.

Bit	Register	Reset value	Type	Description
Bit 31:25	Reserved	0x00	resd	Kept at its default value.
Bit 24:0	CNT	0x0000000	ro	Data count value When this register is read, the number of data bytes to be transferred is returned. Write access has no effect.

Note: This register can be read only when the data transfer is complete.

SDIO status register (SDIO_STS)

The SDIO_STS is a read-only register, containing two types of flags:

- Static flags (bits [23:22, 10: 0]): These bits can be cleared by writing to the SDIO_INTCLR register.
- Dynamic flags (bit [21:11]): These bit status changes with the state of the corresponding logic (for example, BUT full or empty flag is set or cleared as data written to the BUF)

Bit	Register	Reset value	Type	Description
Bit 31:23	Reserved	0x000	resd	Kept at its default value.
Bit 22	IOIF	0x0	ro	SD I/O interrupt received
Bit 21	RXBUF	0x0	ro	Data available in receive BUF
Bit 20	TXBUF	0x0	ro	Data available in transmit BUF
Bit 19	RXBUFE	0x0	ro	Receive BUF empty
Bit 18	TXBUFE	0x0	ro	Transmit BUF empty If hardware flow control is enabled, the TXBUF_E signal becomes valid when the BUF contains two words.
Bit 17	RXBUFF	0x0	ro	Receive BUF full If hardware flow control is enabled, the RXBUF_F becomes valid two words before the BUF is full.
Bit 16	TXBUFF	0x0	ro	Transmit BUF full
Bit 15	RXBUFH	0x0	ro	Receive BUF half full There are at least 8 words in the BUF. This flag bit can be used as DMA request.
Bit 14	TXBUFH	0x0	ro	Transmit BUF half empty: At least 8 words can be written to the BUF. This flag bit can be used as DMA request.
Bit 13	DORX	0x0	ro	Data receive in progress
Bit 12	DOTX	0x0	ro	Data transmit in progress
Bit 11	DOCMD	0x0	ro	Command transfer in progress
Bit 10	DTBLKCMPL	0x0	ro	Data block sent/received CRC check passed
Bit 9	SBITERR	0x0	ro	Start bit not detected on all data signals in wide bus mode
Bit 8	DTCMPL	0x0	ro	Data end (data counter, SDIO CNT, is zero)
Bit 7	CMDCMPL	0x0	ro	Command sent (no response required

Bit	Register	Reset value	Type	Description
Bit 6	CMDRSPCMPL	0x0	ro	Command response (CRC check passed)
Bit 5	RXERRO	0x0	ro	Received BUF overrun error
Bit 4	TXERRU	0x0	ro	Transmit BUF underrun error
Bit 3	DTTIMEOUT	0x0	ro	Data timeout
Bit 2	CMDTIMEOUT	0x0	ro	Command response timeout The command timeout is a fixed value of 64 SDIO_CK clock periods.
Bit 1	DTFAIL	0x0	ro	Data block sent/received (CRC check failed)
Bit 0	CMDFAIL	0x0	ro	Command response received (CRC check failed)

SDIO clear interrupt register (SDIO_INTCLR)

The SDIO_INTCLR is a read-only register. Writing 1 to the corresponding register bit will clear the correspond bit in the SDIO_STS register.

Bit	Register	Reset value	Type	Description
Bit 31:23	Reserved	0x000	resd	Kept at its default value.
Bit 22	IOIF	0x0	rw	SD I/O interface flag clear bit This bit is set by software to clear the IOIF flag.
Bit 21:11	Reserved	0x000	resd	Kept at its default value.
Bit 10	DTBLKCMPL	0x0	rw	DTBLKCMPL flag clear bit This bit is set by software to clear the DTBLKCMPL flag.
Bit 9	SBITERR	0x0	rw	SBITERR flag clear bit This bit is set to clear the SBITERR flag.
Bit 8	DTCMPL	0x0	rw	DTCMPL flag clear bit This bit is set by software to clear the DTCMPL flag.
Bit 7	CMDCMPL	0x0	rw	CMDCMPL flag clear bit This bit is set by software to clear the CMDCMPL flag.
Bit 6	CMDRSPCMPL	0x0	rw	MDRSPCMPL flag clear bit This bit is set by software to clear the CMDRSPCMPL flag.
Bit 5	RXERRO	0x0	rw	RXERRO flag clear bit This bit is set by software to clear the RXERRO flag.
Bit 4	TXERRU	0x0	rw	TXERRU flag clear bit This bit is set by software to clear the TXERRU flag.
Bit 3	DTTIMEOUT	0x0	rw	DTTIMEOUT flag clear bit This bit is set by software to clear the DTTIMEOUT flag.
Bit 2	CMDTIMEOUT	0x0	rw	CMDTIMEOUT flag clear bit This bit is set by software to clear the CMDTIMEOUT flag.
Bit 1	DTFAIL	0x0	rw	DTFAIL flag clear bit This bit is set by software to clear the DTFAIL flag.
Bit 0	CMDFAIL	0x0	rw	CMDFAIL flag clear bit This bit is set by software to clear the CMDFAIL flag.

SDIO interrupt mask register (SDIO_INTEN)

The SDIO_INTEN register determines which status bit generates an interrupt by setting the corresponding bit.

Bit	Register	Reset value	Type	Description
Bit 31:23	Reserved	0x000	resd	Kept at its default value.
Bit 22	IOIFIEN	0x0	rw	SD I/O mode received interrupt enable This bit is set or cleared by software to enable/disable the SD I/O mode received interrupt function. 0: Disabled 1: Enabled
Bit 21	RXBUFIEN	0x0	rw	Data available in RxBUF interrupt enable This bit is set or cleared by software to enable/disable the Data Available in RxBUF Interrupt. 0: Disabled 1: Enabled
Bit 20	TXBUFIEN	0x0	rw	Data available in TxBUF interrupt enable This bit is set or cleared by software to enable/disable the Data Available in TxBUF Interrupt. 0: Disabled 1: Enabled
Bit 19	RXBUFEIEN	0x0	rw	RxBUF empty interrupt enable This bit is set or cleared by software to enable/disable the RxBUF empty interrupt. 0: Disabled 1: Enabled
Bit 18	TXBUFEIEN	0x0	rw	TxBUF empty interrupt enable This bit is set or cleared by software to enable/disable the TxBUF empty interrupt. 0: Disabled 1: Enabled
Bit 17	RXBUFFIEN	0x0	rw	RxBUF full interrupt enable This bit is set or cleared by software to enable/disable the RxBUF full interrupt. 0: Disabled 1: Enabled
Bit 16	TXBUFFIEN	0x0	rw	TxBUF full interrupt enable This bit is set or cleared by software to enable/disable the TxBUF full interrupt. 0: Disabled 1: Enabled
Bit 15	RXBUFHIEN	0x0	rw	RxBUF half full interrupt enable This bit is set or cleared by software to enable/disable the RxBUF half full interrupt. 0: Disabled 1: Enabled
Bit 14	TXBUFHIEN	0x0	rw	TxBUF half empty interrupt enable This bit is set or cleared by software to enable/disable the TxBUF half empty interrupt. 0: Disabled 1: Enabled

Bit	Register	Reset value	Type	Description
Bit 13	DORXIEN	0x0	rw	Data receive acting interrupt enable This bit is set or cleared by software to enable/disable the Data receive acting interrupt. 0: Disabled 1: Enabled
Bit 12	DOTXIEN	0x0	rw	Data transmit acting interrupt enable This bit is set or cleared by software to enable/disable the Data transmit acting interrupt. 0: Disabled 1: Enabled
Bit 11	DOCMDIEN	0x0	rw	Command acting interrupt enable This bit is set or cleared by software to enable/disable the Command acting interrupt. 0: Disabled 1: Enabled
Bit 10	DTBLKCMPLIEN	0x0	rw	Data block end interrupt enable This bit is set or cleared by software to enable/disable the Data block end interrupt. 0: Disabled 1: Enabled
Bit 9	SBITERRIEN	0x0	rw	Start bit error interrupt enable This bit is set or cleared by software to enable/disable the Start bit error interrupt. 0: Disabled 1: Enabled
Bit 8	DTCMPLIEN	0x0	rw	Data end interrupt enable This bit is set or cleared by software to enable/disable the Data end interrupt. 0: Disabled 1: Enabled
Bit 7	CMDCMPLIEN	0x0	rw	Command sent interrupt enable This bit is set or cleared by software to enable/disable the Command sent interrupt. 0: Disabled 1: Enabled
Bit 6	CMDRSPCMPLIEN	0x0	rw	Command response received interrupt enable This bit is set or cleared by software to enable/disable the Command response received interrupt. 0: Disabled 1: Enabled
Bit 5	RXERROIEN	0x0	rw	RxBUF overrun error interrupt enable This bit is set or cleared by software to enable/disable the RxBUF overrun error interrupt. 0: Disabled 1: Enabled
Bit 4	TXERRUIEN	0x0	rw	TxBUF underrun error interrupt enable This bit is set or cleared by software to enable/disable the TxBUF underrun error interrupt. 0: Disabled 1: Enabled

Bit	Register	Reset value	Type	Description
Bit 3	DTTIMEOUTIEN	0x0	rw	Data timeout interrupt enable This bit is set or cleared by software to enable/disable the Data timeout interrupt. 0: Disabled 1: Enabled
Bit 2	CMDTIMEOUTIEN	0x0	rw	Command timeout interrupt enable This bit is set or cleared by software to enable/disable the Command timeout interrupt. 0: Disabled 1: Enabled
Bit 1	DTFAILIEN	0x0	rw	Data CRC fail interrupt enable This bit is set or cleared by software to enable/disable the Data CRC fail interrupt. 0: Disabled 1: Enabled
Bit 0	CMDFAILIEN	0x0	rw	Command CRC fail interrupt enable This bit is set or cleared by software to enable/disable the Command CRC fail interrupt. 0: Disabled 1: Enabled

SDIOBUF counter register (SDIO_BUFCNTR)

The SDIO_BUFCNTR register contains the number of words to be written to or read from the BUF. The BUF counter loads the value from the SDIO_DTLEN register when the data transfter bit TFREN is set in the SDIO_DTCTRL register. If the data length is not word-aligned, the remaining 1 to 3 bytes are regarded as a word.

Bit	Register	Reset value	Type	Description
Bit 31:24	Reserved	0x00	resd	Kept at its default value.
Bit 23:0	CNT	0x000000	ro	Number of words to be written to or read from the BUF.

SDIO data BUF register (SDIO_BUF)

The receive and data BUF is group of a 32-bit wide registers that can be written or read. The BUF contains 32 registers on 32 sequential addresses. The CPU can use BUF for read/write multiple operations.

Bit	Register	Reset value	Type	Description
Bit 31:0	DT	0x0000 0000	rw	Receive and transmit BUF data The BUF data occupies 32 × 32-bit words, the address: SDIO base + 0x80 to SDIO base + 0xFC

24 Universal serial bus full-seed device interface (USBFS)

USBFS introduction

The USBFS implements the USB2.0 full-speed protocols. At the bus speed of 1212Mb/s, it supports control transfer, bulk transfer, synchronous transfer and interrupt transfer, as well as USB suspend/resume.

The USBFS has eight programmable bidirectional endpoints that can be configured as different types of transfers according to specific requirements. It contains a SRAM with dual endpoints for data transfer between the endpoint and user application. In the meantime, it has achieved a dual-buffer mechanism for bulk/synchronous endpoints in order to enhance the transfer efficiency. There is an internal DP pull-up resistance in the USBFS PHY to satisfy the needs of the devices.

USBFS clock and pin configuration

USB clock configuration

The USB full-speed device module interface has two clocks: USB control clock and APB1 bus clock. The USB full-speed device bus speed standard is 12Mb/s±0.25%, so it is necessary to supply 48MHz±0.25% for the USBFS to implement USB bus sampling.

USBFS 48M clock has two sources:

- HICK 48M
  When the HICK 48M clock is used as a USB control clock, it is recommended to enable ACC feature.
- Divided by PLL
  The PLL output frequency must ensure that the USBDIV (see the CRM_CFG register) can be divided to 48MHz.

Note: The APB1 clock frequency must be greater than 12MHz when the USBFS is enabled.

USB pin configuration

When the USB module is enabled in the CRM, PA11 and PA12 can be multiplexed as DP/DM; PA8 can be multiplexed as SOF ouput and configured as a push-pull multiplexed output feature.

Table 130. USB pin configuration

Pin	GPIO	Condition
USB_DM	PA11	USB module enabled in the CRM
USB_DP	PA12	USB module enabled in the CRM
USB_SOF	PA8	Optional. If the USB module is enabled in the CRM, the SOF output feature is enabled, and the PA8 can be configured as a push-pull multiplexed output.

Functional descriptions

USB initialization

After the USB module is enabled (enalbe USBFS clock in the CRM), it is necessary to initialize the USBFS prior to the host enumeration as follows:

1. Clear software set by setting CSRST = 0
2. Clear all status flags by setting INTSTS=0
3. Enable USB Core by setting DEVADDR.CEN=1
4. Configure interrupt enable bits
5. Enable USB PHY by setting CTRL.DISUSB=0

Endpoint configuration

The USBFS supports up to 8 bidirectional and 16 unidirectional endpoints (8 IN and 8 OUT). Each point has its corresponding USBFS endpoint n register (USBFS_EPTn) that is used to store the endpoint status. The endpoint configuration includes:

- Endpoint number (by configuring the EPTADDR, the endpoint number of each endpoint register is programmable)
- Transfer type (control transfer, bulk transfer, isochronous transfer and interrupt transfer)
- Buffer for IN/OUT endpoint (buffer allocation is described in the next section)
- IN/OUT Toggle status (correspond to DATA0/DATA1)
- IN/OUT status (VALID, NAK, STALL, DISABLE)

Note: Endpoint 0 acts as a control point by default. It is usually configuread after receiving a reset signal sent by the host.

USB buffer

USB has a dual-port SRAM buffer for data transfter between endpoint and user application. Both the user application and the USBFS module can access to the buffer at the same time. The buffer size can be automatically adjusted according to the CAN status. The following table lists its mapping address and size.

Table 131. Buffer size configuration table

Working conditions	USBBUFS	0	1			
	CAN1 status	Enable/Disable	Disable	Disable	Enable	Enable
	CAN2 status	Enable/Disable	Disable	Enable	Disable	Enable
Buffer size		512 Byte	1280 Byte	1024 Byte	1024 Byte	768 Byte
Address range		0x4000 6000~ 0x4000 63FF	0x4000 7800~ 0x4000 81FF	0x4000 7800~ 0x4000 7FFF	0x4000 7800~ 0x4000 7FFF	0x4000 7800~ 0x4000 7DFF

The buffer is composed of the endpoint register buffer description field and the endpoint buffer. The endpoint register description table describes the offset address of endpoint receive/transmit. The figure below gives an example of a buffer structure (512 bytes).

Start address: 0x40006000

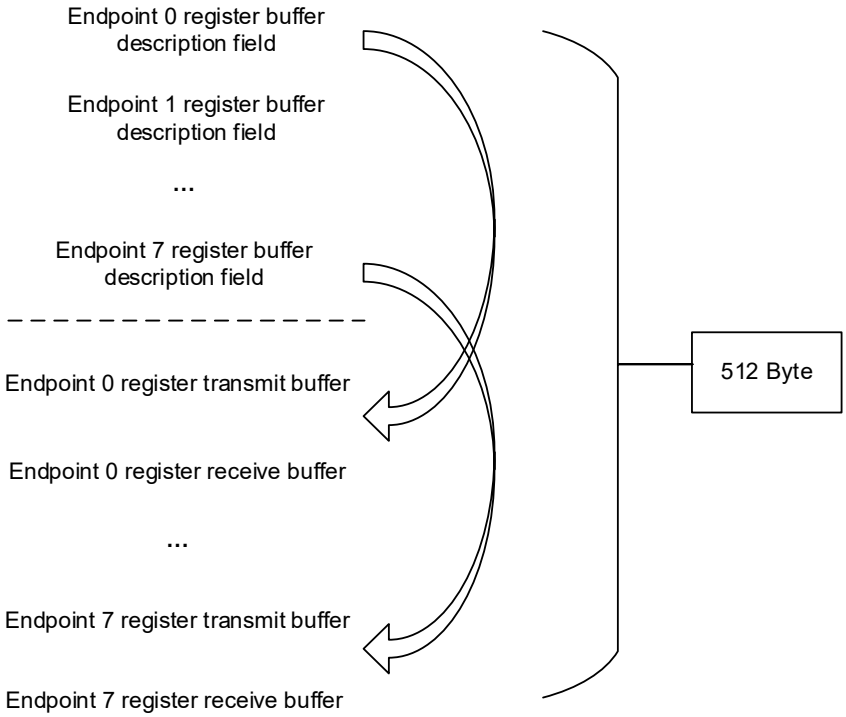


Figure 196. Structure of 512 bytes endpoint register buffer

In the USBFS module, each endpoint register corresponds to a buffer description field, which is used to describe the buffer and data length of the endpoint receive/transmit. The structure of a regular endpoint register buffer description field is shown as follows (dual buffer description table is detailed in the next section):

Endpoint n:

Table 132. The structure of a regular endpoint register buffer description field

0	2	4	6	8	10	12	14
TnADDR	Reserved	TnLEN	Reserved	RnADDR	Reserved	RnLEN	Reserved
Transmit buffer description				Buffer receive description			

The start address of a buffer description field=buffer address + BTADDR × 2. The user application puts the endpoint register buffer description in different locations according to specific needs. The BTADDR is by default 0.

The USBFS module contains 8 endpoint registers. Each endpoint register description table acutally occupies 8 bytes. When programming, the user should reserve enough space for buffer description field according to the endpoint register used. While allocating transmit/receive buffer to the endpoint, the offset address is not allowed to occupy the buffer description and transmit/receive buffer of other endpoints.

Note: The APB1 bus width is 32 bits, while the buffer is 16-bit wide memory, meaning that the APB1 bus can write only two-byte data to a packet buffer each time. For this reason, 8 consecutive write accesses are required when a 16-byte data is to be written.

Double-buffered endpoints

Double-buffered mode is designed in the USBFS module in order to increase the transaction rate of bulk transfer and isochronous transfer.

One IN endpoint (or OUT endpoint) corresponds to two buffers. The following diagram shows the differences of buffer description table between regular endpoint and double-buffered endpoint.

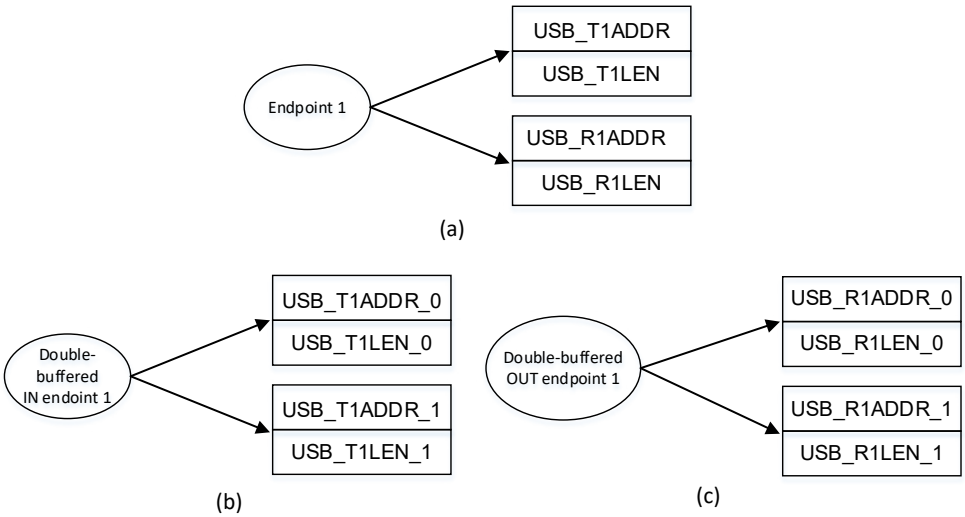


Figure 197. Buffer description table of regular endpoint vs. double-buffered endpoint

Two endpoint transfer modes support double-buffered function:

- Bulk transfer endpoints
  Enable double-buffered feature by setting TRANS_TPYE=00 and EXF=1
- Isochronous transfer endpoints
  When TRANS_TPYE=10, double-buffered feature is enabled.
  Two buffers are used to increase the transfer rate. The USBFS and the user application access to a different buffer at the same time so as to process data simultaneously. The USBFS and the user application must determine which one of the existing buffers is accessible, which can be done by the SBUF flag in the USBFS.
- Double-buffered OUT endpoints: SBUF corresponds to bit 6 in the USB_EPTn
  SBUF=1, USBFS uses RnADDR_0 and RnLEN_0, while the user application uses RnADDR_1 and RnLEN_1
  SBUF=0, USBFS uses RnADDR_1 and RnLEN_1, while the user application uses RnADDR_0 and RnLEN_0
- Double-buffered IN endpoints: SBUF corresponds to bit 14 in the USB_EPTn
  SBUF=1, USBFS uses TnADDR_0 and TnLEN_0, while the user application uses TnADDR_1 and TnLEN_1
  SBUF=0, USBFS uses TnADDR_1 and TnLEN_1, while the user application uses TnADDR_0 and TnLEN_0

Note: Endpoint 0 cannot be used as a double-buffered endpoint.

SOF output

A SOF flag is generated when the USBFS receives a SOF sent by the host. Meanwhile, the current frame number can be read from the USBFS_SOFRNUM register. By setting the SOFOUTEN bit in the USBFS_CFG register, a SOF pulse sigal with a width of 24 APB1 clock cycles will be output onto the pin.

Suspend/Resume

The USB2.0 full-speed standard defines a low-power consumption state, called Suspend. In suspend mode, a suspend state is entered when the idle state is detected on the bus for more than 3ms. The USBFS can enter suspend state in two ways:

1. Detect the lack of three consecutive SOF packets

2. Control register SSP is set by the application

When the device detects the lack of three consecutive SOF packets, the SSP and LPM registers must be set by the application in order to disable SOF detection and the statistic power consumtion of the USB physical transceiver.

Resume refers to a process when the USB device returns from a suspend state to a normal state. When the USB device is in suspend mode, any non-idle state (such as packet start signal SOF) of its upstream port can resume it. In addition, the USB device can also apply to activate resume operation by setting the GRESUME register, which is called a remote wakeup.

USBFS interrupts

Low-priority interrupts: interrupt vector number 20 and 74. When a high-priority interrupt is disabled, all the USBFS interrupts can be handled by low-priority interrupts.

High-priority interrupts: interrupt vector number 19 and 73. Only isochronous transfer and double-buffered bulk endpoints can trigger high-priority interrupts.

USB wakeup interrupts: interrupt vector number 42. While the USB enters a suspend state, the chip can be waken up by this interrupt after moving to Deep Sleep mode.

By default, the USBFS shares the interrupt vector number (19 and 20) with the CAN1, causing that the USBFS and CAN1 cannot be used at the same time. For this reason, new interrupt vector numbers (73 and 74) are created. The user application can map the USBFS interrupt vector numbers to 73 and 74 by setting the USBINTMAP bit in the CRM_INTMAP register.

USBFS registers

These peripheral registers must be accessed by words (32 bits).

Table 133. USBFS register map and reset values

Register	Offset	Reset value
USBFS_EPT0	0x00	0x0000
USBFS_EPT1	0x04	0x0000
USBFS_EPT2	0x008	0x0000
USBFS_EPT3	0x0C	0x0000
USBFS_EPT4	0x10	0x0000

Register	Offset	Reset value
USBFS_EPT5	0x14	0x0000
USBFS_EPT6	0x18	0x0000
USBFS_EPT7	0x1C	0x0000
USBFS_CTRL	0x40	0x0003
USBFS_INTSTS	0x44	0x0000
USBFS_SOFRNUM	0x48	0x0XXX
USBFS_DEVADDR	0x4C	0x0000
USBFS_BUFTBL	0x50	0x0000
USBFS_CFG	0x60	0x0000
USBFS_TnADDR	[USB_BUFTBL] × 2 + n × 16	0xXXXX
USBFS_TnLEN	[USB_BUFTBL] × 2 + n × 16 + 4	0xXXXX
USBFS_RnADDR	[USB_BUFTBL] × 2 + n × 16 + 8	0xXXXX
USBFS_RnLEN	[USB_BTABLE] × 2 + n × 16 + 12	0xXXXX

USBFS endpoint n register (USBFS_EPTn), n=[0..7]

Bit	Register	Reset value	Type	Description
Bit 15	RXTC	0x0	rw0c	Rx transaction completed This bit is set when an OUT/SETUP transaction is completed, indicating that Rx transaction is complete. 0: Software clears this bit 1: OUT/SETUP transaction is completed.
Bit 14	RXDTS	0x0	tog	Rx Data Toggle (DAT0/DATA1) Synchronization This is not ISO transfer. This bit indicates that the current transaction is DATA0/DATA1. 0: DATA0 1: DATA1
Bit 13:12	RXSTS	0x0	tog	Rx Status This field indicates the endpoint status in response to the OUT transaction of the host. There are four states: DISABLE, NAK, STALL and ACK. 00: DISABLED, endpoint ignores all reception requests. 01: STALL, endpoint responds to all reception requests with STALL packets 10: NAK, endpoint responds to all reception requests with NAK packets 11: VALID, endpoint can be used for reception
Bit 11	SETUPTC	0x0	rog	Setup transaction completed When the RXTC is set, this bit is used to determine whether OUT/SETUP transaction is completed. 0: OUT transaction is completed 1: SETUP transaction is completed
Bit 10: 9	TRANS_ TYPE	0x0	rw	Transfer type This field defines four types of USB transfers: Control, Bulk, Interrupt and ISO. 00: BULK endpoint, can work with EXF bit register 01: CTRL endpoint, can work iwth EXF bit register 10: ISO endpoint 11: INT endpoint

Bit	Register	Reset value	Type	Description
Bit 8	EXF	0x0	rw	Endpoint Extend function USB endpoint extend function is used for Bulk and Control transfers. For Bulk transfer, this bit is set to indicate that double-buffered is enabled. For Control transfer, if this bit is set, it detects whether the data length in the SETUP transaction is 0 or not, a STALL is returned if the value is not 0.
Bit 7	TXTC	0x0	rw0c	Tx transaction completed This bit is set when IN transaction is completed, indicating that Tx transaction is completed. 0: Software clears Tx transaction complete flag 1: IN transaction reception is completed
Bit 6	TXDTS	0x0	tog	Tx Data Toggle (DAT0/DATA1) Synchronization This is non-ISO endpoint, indicating that the current IN transaction is DATA0/DATA1. 0: DATA0 1: DATA1
Bit 5:4	TXSTS	0x0	tog	Tx Status This field indicates the endpoint status in response to the IN transaction of the host. There are four states: DISABLE, NAK, STALL, ACK. 00: DISABLED, endpoint ignores all transmission requests. 01: STALL, endpoint responds to all transmission requests with STALL packets 10: NAK, endpoint responds to all transmission requests with NAK packets 11: VALID, endpoint can be used for transmission
Bit 3:0	EPTADDR	0x0	rw	Endpoint address

USBFS control register (USBFS_CTRL)

Bit	Register	Reset value	Type	Description
Bit 15	TCIEN	0x0	rw	Transmission complete interrupt enable 0: Disabled 1: Enabled
Bit 14	UCFORIEN	0x0	rw	USB Core fifo overrun interrupt enable 0: Disabled 1: Enabled
Bit 13	BEIEN	0x0	rw	Bus error interrupt enable 0: Disabled 1: Enabled
Bit 12	WKIEN	0x0	rw	Wakeup/Remote wakeup interrupt enable 0: Disabled 1: Enabled
Bit 11	SPIEN	0x0	rw	Bus suspend interrupt enable 0: Disabled 1: Enabled
Bit 10	RSTIEN	0x0	rw	Bus reset interrupt enable 0: Disabled 1: Enabled

Bit	Register	Reset value	Type	Description
Bit 9	SOFIEN	0x0	rw	Start of frame interrupt enable 0: Disabled 1: Enabled
Bit 8	LSOFIEN	0x0	rw	Lost start of frame interrupt enable 0: Disabled 1: Enabled
Bit 7:5	Reserved	0x0	resd	Kept at its default value.
Bit 4	GRESUME	0x0	rw	Generate Resume request In suspend mode, the software can set this bit to send a resume signal to the host in order to wake up it. It must be cleared between 10 ms and 15 ms.
Bit 3	SSP	0x0	rw	Software suspend config This bit is set by software when a suspend flag is detected. When exiting suspend state, this bit must be cleared by software. 0: Software leaves suspend mode 1: Software enters suspend mode
Bit 2	LPM	0x0	rw	Low power mode While entering suspend mode, this bit can be set to reduce power consumption. It is cleared automatically when the USB Core is waken up. 0: No low-power mode 1: Low-power mode
Bit 1	DISUSB	0x1	rw	Disble USB PHY 0: USB PHY enabled 1: USB PHY disabled
Bit 0	CSRST	0x1	rw	Core soft Reset 0: Software clears reset 1: Software resets usb core, and a reset interrupt is generate.

USBFS interrupt status register (USBFS_INTSTS)

Bit	Register	Reset value	Type	Description
Bit 15	TC	0x0	ro	Transaction completed 0: Reset value 1: This bit is set to indicate that the USB has successfully completed one IN/OUT transaction
Bit 14	UCFOR	0x0	rw0c	USB Core fifo overrun 0: Reset value 1: USB Core fifo overrun
Bit 13	BE	0x0	rw0c	Bus error 0: Reset value 1: Bus error detected, such as, CRC check error, bit-stuffing error, Answer timeout error and frame format error.
Bit 12	WK	0x0	rw0c	Wakeup 0: Reset value 1: A wakeup signal is received when the USB is in suspend state.

Bit	Register	Reset value	Type	Description
Bit 11	SP	0x0	rw0c	Bus Suspend 0: Reset value 1: No data transfer has been received for 3ms, and the bus enters suspend state.
Bit 10	RST	0x0	rw0c	Bus reset 0: Reset value 1: A USB reset signal was detected on the bus.
Bit 9	SOF	0x0	rw0c	Start of frame 0: Reset value 1: This bit is set to indicate that a SOF transaction arrives through the bus.
Bit 8	LSOF	0x0	rw0c	Lost start of frame 0: Reset value 1: No SOF has been received for more than 1ms.
Bit 7:5	Reserved	0x0	resd	Kept at its default value.
Bit 4	INOUT	0x0	ro	IN/Out transaction When TC complete interruput is generated, this bit is used to indicate whether IN/OUT transcation has been completed. 0: IN transaction 1: OUT transaction
Bit 3:0	EPT_NUM	0x0	ro	Endpoint number When TC complete interruput is generated, this bit is used to indicate which endpoint transfer has been completed successfully.

USBFS SOF frame number register (USBFS_SOFRNUM)

Bit	Register	Reset value	Type	Description
Bit 15	DPSTS	0x0	ro	D+ status Indicates D+ status
Bit 14	DMSTS	0x0	ro	D- status Indicates D- status
Bit 13	CLCK	0x0	ro	Connect Locked This bit is set when two consecutive SOF packets have been received.
Bit 12:11	LSOFNUM	0x0	ro	Lost SOF number After LSOF, this bit indicates the number of SOF packet lost. It is cleared by hardware at the reception of an SOF transaction.
Bit 10: 0	SOFNUM	0xXXX	ro	Start of Frame number Indicates the current SOF frame number.

USBFS device address register (USBFS_DEVADDR)

Bit	Register	Reset value	Type	Description
Bit 15:8	Reserved	0x00	resd	Kept at its default value
Bit 7	CEN	0x0	rw	USB Core Enable 0: USB Core disabled 1: USB Core enabled
Bit 6:0	ADDR	0x00	rw	Host assign Device address These bits contain the device address assigned by the host during the enumeration process.

USBFS buffer table address register (USBFS_BUFTBL)

Bit	Register	Reset value	Type	Description
Bit 15:3	BTADDR	0x0000	rw	Endpoint buffer table start address This field indicates the start address of the buffer description table. It is 0 by defauly.
Bit 2:0	Reserved	0x0	resd	Forced to 0 by hardware.

USBFS CFG control register (USBFS_CFG)

Bit	Register	Reset value	Type	Description
Bit 15:2	Reserved	0x0000	resd	Kept at its default value.
Bit 1	PUO	0x0	rw	DP pull-up off 0: DP pull-up resistance enabled 1: DP pull-up resistance disabled
Bit 0	SOFOUTEN	0x0	rw	SOF output enable 0: No SOF pulse output 1: SOF pulse output to the pin

USBFS transmission buffer first address register (USBFS_TnADDR)

Bit	Register	Reset value	Type	Description
Bit 15:1	TnADDR	0xXXXX	rw	Transmission buffer first address This field indicates the start address of the buffer where data is to be transmitted at the reception of the next IN transaction request.
Bit 0	Reserved	0x0	resd	This bit must be 0 since the address of the packet buffer must be word-aligned.

USBFS transmission data length register (USBFS_TnLEN)

Bit	Register	Reset value	Type	Description
Bit 15:10	Reserved	0xXX	resd	Kept at its default value.
Bit 9:0	TnLEN	0xXXX	rw	Transmission length This field contains the number of data bytes to be transferred at the reception of the next IN transaction request.

USBFS reception buffer first address register (USBFS_RnADDR)

Bit	Register	Reset value	Type	Description
Bit 15:1	RnADDR	0xXXXX	rw	Reception buffer first address This field indicates the start address of the buffer where data is to be received at the reception of the next OUT/SETUP transaction request.
Bit 0	Reserved	0x0	resd	This bit must be 0 since the address of the packet buffer must be word-aligned.

USBFS reception data length register (USBFS_RnLEN)

Bit	Register	Reset value	Type	Description
Bit 15	BSIZE	0xX	rw	Block size This bit indicates the size of the reception buffer ot he current endpoint. If BSIZE=0, the block size is 2 byte large, and the the size of the packet buffer ranges from 2 to 62 bytes. If BSIZE=1, the block size is 32 byte large, and the size of the packet buffer ranges from 32 to 1024 bytes.
Bit 14:10	NBLK	0xXX	rw	Number of blocks This field defines the number of blocks allocated to the current endpoint reception buffer.
Bit 9:0	RnLEN	0xXXX	rw	Reception Length This field defines the data length received by the endpoint.

25 HICK auto clock calibration (ACC)

ACC introduction

HICK auto clock calibration (HICK ACC), which uses the SOF signal (1 ms of period) generated as a reference signal, implements the sampling and calibration for the HICK clocks.

The main pupose of this module is to provide a clock of 48MHz±0.25% for the USB device.

It is able to make the calibrated frequency as close to the target frequency as possible by means of "cross and return" algorithm.

Main features

- Programmable center frequency
- Programmable boundary frequency that triggers calibration function
- Center frequency precision ±0.25%
- Status detection flags
  - Calibration ready flag
- Error detection flags
  - Reference signal lost error flag
- Two interrupt source flag
  - Calibration ready flag
  - Reference signal lost error flag
- Two calibration modes: coarse calibration and fine calibration

Interrupt requests

Table 134. ACC interrupt requests

Interrupt event	Event flag	Enable bit
Calibration ready	CALRDY	CALRDYIEN
Reference signal lost	RSLOST	EIEN

ACC interrupt events are linked to the same interrupt vector (see the following diagram). Interrupt events include:

- During calibration process: When the calibration gets ready or reference signal lost occurs, the corresponding interrupt will be generated if the corresponding enable bit is enabled.

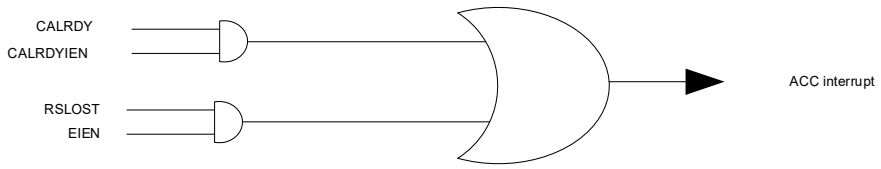


Figure 198. ACC interrupt mapping diagram

Functional descriptions

Auto clock calibration (HICK ACC), which uses the SOF signal (1 ms of period) generated as a reference signal, implements the sampling and calibration for the HICK clocks. In particular, the HICK clock frequency can be calibrated to a precision of ±0.25% so as to meet the needs of the high-precision clock applications such as USB.

The signals of the module are connected to the CRM and HICK inside the microcontroller instead of being connected to the pins externally.

- CRM_HICKCAL: the HICKCAL bit in the CRM module. This signal is used to calibrate the HICK in bypass mode. The value is defined by the HICKCAL[7:0] in the CRM_CTRL register.
- CRM_HICKTRIM: the HICKTRIM bit in the CRM module. This signal is used to calibrate the HICK in bypass mode. The value is defined by the HICKTRIM[5:0] in the CRM_CTRL register. The default value of the HICk is 32, which can be calibrated to 8MHz±0.25%. The HICK frequency can be adjusted by 20kHz (design value) each time when the CRM_HICKTRIM value changes. In other words, the HICK output frequency will increase by 20kHz eac time the CRM_HICKTRIM value is decremented by one; the HICK output frequency will reduce by 20kHz each time the CRM_HICKTRIM value is decremented by one.
- USB_SOF: USB Start-of-Frame signal given by the USB device. Its high-level width is 12 system clock cycles, a pulse signal of 1 ms.
- HICKCLK: HICK clock. The original HICK output frequency is 48MHz, but the sampling clock used by the HICK calibration module is frequency divider (1/6) clock, about 8MHz.
- HICKCAL: HICK module calibration signal. For the HICK clock after frequency divion (1/6), the HICK clock frequency will change by 40KHz (design value) each time the HICKCAL changes, which is positively correlated. In other words, the HICK clock frequency will incease by 40KHz (design value) eac time the HICKCAL is incremented by one; the HICK clock frequency will reduce by 40KHz each time the HICKCAL is decremented by one.
- HICKTRIM: HICK module calibration signal. For the HICK clock after frequency divion (1/6), the HICK clock frequency will change by 20KHz (design value) each time the HICKCAL changes, which is positively correlated.

Refer to Section 25.6 for more information about the bit definition in the registers.

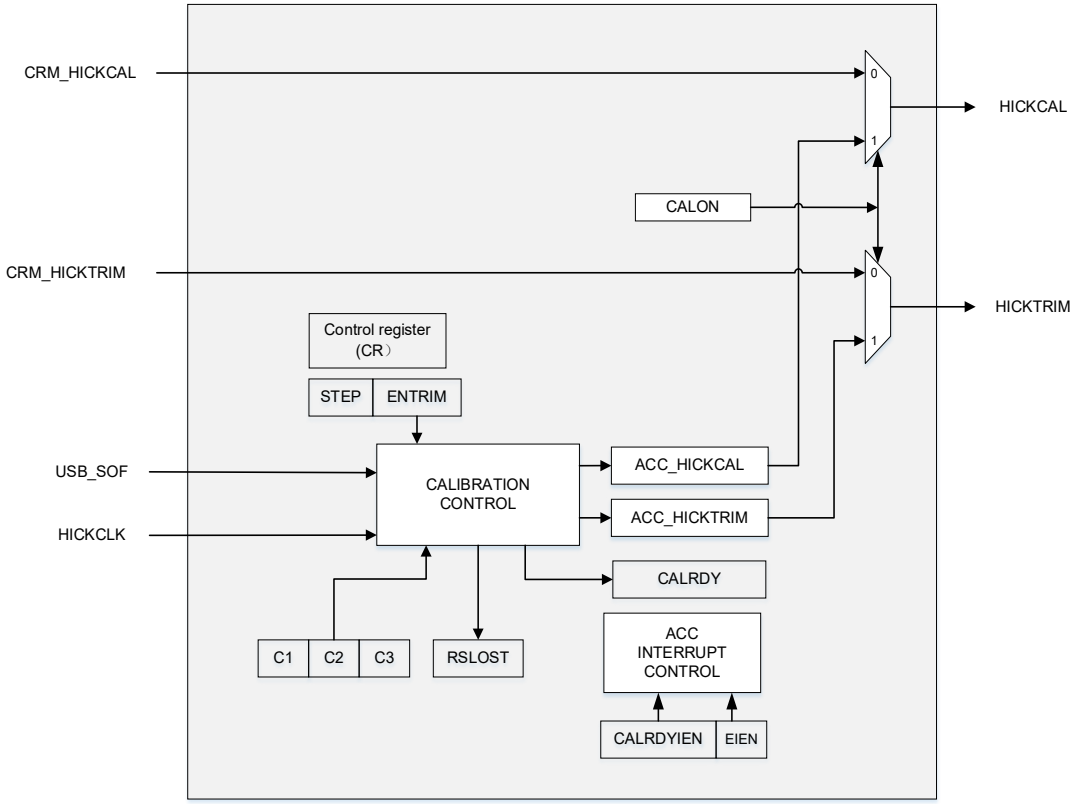


Figure 199. ACC block diagram

Principle

USB_SOF period signal: 1ms of period must be accurate, which is a prerequisite of the normal operaion of an auto calibration module.

cross-return algorithm: This is used to calculate a calibration value closest to the theoretic value. In theory, the actual frequency after calibration can be adjusted to be within an accuracy range of about 0.5 steps from the target frequency (8MHz)

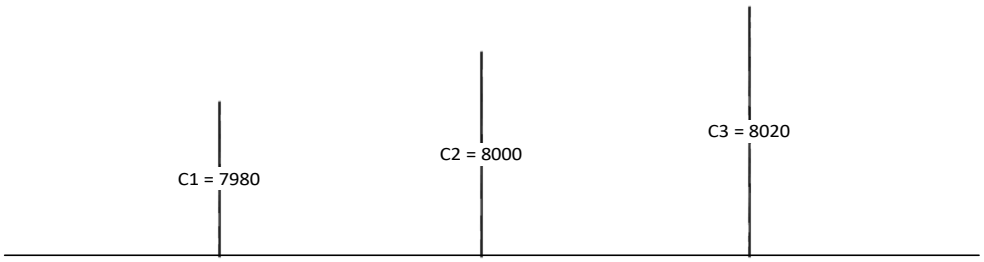


Figure 200. Cross-return algorithm

From the above figure, auto calibration function will adjust the HICKCAL or HICKTRIM according to the specified step as soon as the condition for trigerring auo calibration is reached.

Cross:

If the auto calibration condition is met, the actual sampling data in the first 1ms period will be either less than C2, or greater than C2.

When this value is less than C2, the auto calibration module will start increasing either the HICKCAL or HICKTRIM according to the step definition until the actual sampling value is greater than C2. In this way, the actual value will cross over C2 from small to large.

When this value is greater than C2, the auto calibration module will start decrease either the HICKCAL or HICKTRIM according to the step definition until the actual sampling value become less than C1. In this way, the actual value will cross over C2 from large to small.

Return:

After cross operation is completed, the actual value closest to C2 can be obtained by comparing the difference (calculated as absolute value) between the actual sampling value and C2 before and after crossing C2 so as to get the best calibration value HICKCAL or HICKTRIM.

If the difference after crossing is less than the one before crossing C2, the calibration value after crossing prevails, and stops the calibration process until the next condition for auto calibration appears.

If the difference after crossing is greater than the one before crossing C2, the calibration value before crossing prevails, and it will return by one step to the one before crossing, and stops the calibration process until the next condition for auto calibration appears.

According to the cross-return strategy, in theory, it is possible to get the frequency accuracy that is 0.5 steps away from the center frequency.

Four conditions for enabling auto calibration function are as follows:

1. The rising edge of the CANLON (from 0 to 1)

2. When CALON=1, reference signal is lost and restored

3. When the sample counter is less than C1

4. When the sample counter is greater than C3

Even though the sampling counter is between C1 and C3, at the rising edge the CANLON, the auto calibration module can also be activated so that the HICK frequency can be adjusted to be within a range of 0.5 steps of the center frequency as soon as the CANLON is enabled.

Under one of the above-mentioned circumstances, the HICK frequency cal be calibrated to be within 0.5 steps of the center frequency. To achieve the best calibration accuracy, it is recommended to remain step as 1 (defual value). If the steop is set to 0, either HICKCAL or HICKTRIM will not be able to be calibrated.

Register description

Refer to the list of abbreviations used in register descriptions.

These peripheral registers must be accessed by words (32 bits).

Table 135. ACC register map and reset values

Register	Offset	Reset value
ACC_STS	0x00	0x0000 000
ACC_CTRL1	0x0C	0x0000 0100
ACC_CTRL2	0x0C	0x0000 2080
ACC_C1	0x0C	0x0000 1F2C
ACC_C2	0x10	0x0000 1F40
ACC_C3	0x14	0x00000 1F54

Status register (ACC_STS)

Bit	Register	Reset value	Type	Description
Bit 31:2	Reserved	0x0000000	resd	Kept at its default value.
Bit 1	RSLOST	0x0	ro	Reference Signal Lost 0: Reference Signal is not lost 1: Reference Signal is lost Note: During the calibration, when the sample counter of the calibration module is twice that of C2, if a SOF reference signal is not detected, it means that the reference signal is lost. The internal statue machine will move to the idle state unless another SOF signal is detected, otherwise, the interal clock sample counter remains 0. The RSLOST bit is immediately cleared after the CALON bit is cleared or when the RSLOST is written with 0. Reference signal detection occurs only when CALON=1.
Bit 0	CALRDY	0x0	ro	Internal high-speed clock calibration ready 0: Interal 8MHz oscillator calibration is not ready 1: Interal 8MHz oscillator calibration is ready Note: This bit is set by hardware to indicate that internal 8MHz oscillator has been calibrated to the frequency closest to 8MHz. The CALRDY is immediately cleared after the CALON bit is cleared or when the CALRDY is written with 0.

Control register 1 (ACC_CTRL1)

Bit	Register	Reset value	Type	Description
Bit 31:12	Reserved	0x00000	resd	Forced by hardware to 0
Bit 11:8	STEP	0x1	rw	Calibrated step This field defines the value after each calibration. Note: It is recommended to set the step bit in order to get a more accurate calibration result. While ENTRIM=0, only the HICKCAL is calibrated. If the step is incremented or decremented by one, the HICKCAL will be incremented or decremented by one accordingly, and the HICK frequency will increase or decrease by 40 kHz (design value). This is a positive relationship. While ENTRIM=1, only the HICKTRIM is calibrated. If the step is incremented or decremented by one, the HICKTRIM will be incremented or decremented by one accordingly, and the HICK frequency will increase or decrease by 20 kHz (design value). This is a positive relationship.
Bit 7:6	Reserved	0x0	rw	Forced by hardware to 0
Bit 5	CALRDYIEN	0x0	rw	CALRDY interrupt enable This bit is set or cleared by software. 0: Interrupt generation disabled 1: ACC interrupt is generated when CALRDY=1 in the ACC_STS register
Bit 4	EIEN	0x0	rw	RSLOST error interrupt enable This bit is set or cleared by software. 0: Interrupt generation disabled 1: ACC interrupt is generated when RSLOST=1 in the ACC_STS register
Bit 3:2	Reserved	0x0	rw	Forced by hardware to 0
Bit 1	ENTRIM	0x0	rw	Enable trim This bit is set or cleared by software. 0: HICKCAL is calibrated. 1: HICKTRIM is calibrated. Note: It is recommended to set ENTRIM=1 in order to get higher calibration accuracy.
Bit 0	CALON	0x0	rw	Calibration on This bit is set or cleared by software. 0: Calibration disabled 1: Calibration enabled, and starts searching for a pulse on the USB_SOF. Note: This module cannot be used without the USB_SOF reference signal. If there are no requirements on the accuracy of the HICK clock, it is unnecessary to enable this module.

Control register 2 (ACC_CTRL2)

Bit	Register	Reset value	Type	Description
Bit 31:14	Reserved	0x00000	resd	Forced to 0 by hardware
Bit 13:8	HICKTRIM	0x00	ro	Internal high-speed auto clock trimming This field is read only, but not written. Internal high-speed clock is adjusted by ACC module, which is added to the ACC_HICKCAL[7:0] bit. These bits allow the users to input a trimming value to adjust the frequency of the HICKRC oscillator according to the variations in voltage and temperature. The default value is 32, which can trim the HICK to 8 MHz ± 0.25. The trimming value is 20 kHz (design value) between two consecutive ACC_HICKTRIM steps.
Bit 7:0	HICKCAL	0x00	ro	Internal high-speed auto clock calibration This field is read only, but not written. Internal high-speed clock is adjusted by ACC module. These bits allow the users to input a trimming value to adjust the frequency of the HICKPC oscillator according to the variations in voltage and temperature. The default value is 128, which can trim the HICK to 8 MHz ± 0.25. The trimming value is 40 kHz (design value) between two consecutive ACC_HICKCAL steps.

Compare value 1 (ACC_C1)

Bit	Register	Reset value	Type	Description
Bit 31:16	Reserved	0x0000	resd	Forced to 0 by hardware
Bit 15:0	C1	0x1F2C	rw	Compare 1 This value is the lower boundary for triggering calibration, and its default value is 7980. When the number of clocks sampled by ACC in 1 ms period is less than or equal to C1, auto calibration is triggered automatically. When the actual sampling value (number of clocks in 1 ms) is greater than C1 but less than C3, auto calibration is not enabled.

Compare value 2 (ACC_C2)

Bit	Register	Reset value	Type	Description
Bit 31:16	Reserved	0x0000	resd	Forced to 0 by hardware
Bit 15:0	C2	0x1F40	rw	Compare 2 This value defines the number of clocks sampled for 8MHz (ideal frequency) clock in 1ms period, and its default value is 8000 (theoretical value) As a center point of cross-return strategy, this value is used to calculate the calibration value closest to the theoretical value. In theory, the actual frequency after calibration can be trimmed to be within an accuracy of 0.5 steps from the targe frequency (8 MHz)

Compare value 3 (ACC_C3)

Bit	Register	Reset value	Type	Description
Bit 31:16	Reserved	0x0000	resd	Forced to 0 by hardware
Bit 15:0	C3	0x1F54	rw	Compare 3 This value is the upper boundary for triggering calibration. When the number of clock sampled by ACC in 1ms period is greater than or equal to C3, auto calibration is triggered automatically. When the actual sampling value (number of clocks in 1ms period) is greater than C1 but less than C3, auto calibration is not enabled.

26 Debug (DEBUG)

Debug introduction

Cortex™-M4F core provides poweful debugging features including halt and single step support, as well as trace function that is used for checking the details of the program execution. The debug features are implemented with two interfaces: serial wire debug (SWD) and JTAG debug port. Trace information is collected by a single-wire serial wire view interface, or by TRACE interface when a larger trace bandwidth is needed. Trace and debugging interfaces can be combined into one interface.

ARM Cortex™-M4F reference documentation:

- Cortex™-M4 Technical Reference Manual (TRM)
- ARM Debug Interface V5
- ARM CoreSight Design Kit revision r1p0 Technical Reference Manual

Debug and Trace

It is possible to support debugging for different peripherals, and configure the working status of peripherals during debugging. For timers and watchdogs, the user can select whether or not to stop or continue counting during debugging; For CAN, the user can select whether or not to stop or continue updating receive registers during debugging; For I^2C, the user can select whether or not to stop or continue SMBUS timeout counting.

In addition, code debugging is supported in Low-Power mode. In Sleep mode, the clock programmed by code remains active for HCLK and FCLK to continue to work. In DeepSleep mode, HICK oscillator is enabled to feed FCLK and HCLK.

There are several ID codes inside the MCU, which is accessible by the debugger using the DEBUG_IDCODE at address 0xE0042000. It is part of the DEBUG and is mapped on the external PPB bus. These codes are accessible using the JTAG debug port or the SWD debug port or by the user software. They are even accessible while the MCU is under system reset.

Two trace interface modes supported: single-pin mode for serial wire view and multi-pin trace interface.

I/O pin control

SWJ-DP is supported in different packages of HT32F49365/HT32F49395. It uses 5 general-purpose I/O ports. After reset, the SWJ-DP can be immediately used by the debugger as a default function. To ensure that JTAG input pins are not floating (especially SWCLK/JTCK), the JTAG input pins are embedded with internal pull-up or pull-down feature, NJTRST, JTDI and JTMS/SWDIO with internal pull-up feature, and JTCK/SWCLK with internal pull-down feature.

When the user wants to switch to a different debug port or disable debug feature, either IOMUX_MAPR or IOMUX_MAPR7 register can be configured to release these dedicated I/O pins. Once a corresponding debug I/O is released by the user, the GPIO controller takes control, and then these I/Os can be used as general-purpose I/Os.

For trace feature, it is possible to set the TRACE_IOEN and TRACE_MODE bits in the DEBUG_CTRL register to enable trace function and slect trace modes.

Table 136. Trace function enable

TRACE_IOEN	Description
0	No Trace (default state)
1	Trace enabled

Table 137. Trace function mode

<table>
<tr><th colspan="2">TRACE_MODE[1:0]</th><th>PB3/JTDO/ TRACESWO</th><th>PE2/ TRACECK</th><th>PE3/ TRACED[0]</th><th>PE4/ TRACED[1]</th><th>PE5/ TRACED[2]</th><th>PE6/ TRACED[3]</th></tr>
<tr><td>00</td><td>Asynchronous trace</td><td>TRACES WO</td><td colspan="5">Released (can be used as general-puspose I/Os)</td></tr>
<tr><td>01</td><td>Synchronous trace</td><td rowspan="3">Released (can be used as general-puspose I/Os)</td><td>TRAC ECK</td><td>TRAC ED[0]</td><td colspan="3">Released (can be used as general-puspose I/Os)</td></tr>
<tr><td>10</td><td>Synchronous trace</td><td>TRAC ECK</td><td>TRAC ED[0]</td><td>TRAC ED[1]</td><td colspan="2">Released (can be used as general-puspose I/Os)</td></tr>
<tr><td>11</td><td>Synchronous trace</td><td>TRACE CK</td><td>TRACE D[0]</td><td>TRACE D[1]</td><td>TRACE D[2]</td><td>TRACE D[3]</td></tr>
</table>

DEBUG registers

The following table shows debug register map and its reset values.

The peripheral registers can be accessed by words (32-bit).

Table 138. DEBUG register address and reset value

Register	Offset	Reset value
DEBUG_IDCODE	0xE004 2000	0xXXXX XXXX
DEBUG_CTRL	0xE004 2004	0x0000 0000

DEBUG device ID (DEBUG_IDCODE)

MCU integrates an ID code that is used to identify MCU's revision. The DEBUG_IDCODE register is mapped on the external PPB bus at address 0xE0042000. This code is accessible by the JTAG debug port or SW debug port or by the user code.

Bit	Register	Reset value	Type	Description
Bit 31:0	PID	0xXXXX XXXX	ro	PID information

Table 139. Description of ID code

PID[31:0]	HT32 part number	FLASH size	Packages
0x7005_0244	HT32F49365	256KB	LQFP100
0x7005_0245	HT32F49365	256KB	LQFP64
0x7005_0246	HT32F49365	256KB	LQFP48
0x7005_0247	HT32F49365	256KB	QFN48
0x7005_0340	HT32F49395	1024KB	LQFP100
0x7005_0341	HT32F49395	1024KB	LQFP64
0x7005_0342	HT32F49395	1024KB	LQFP48
0x7005_0343	HT32F49395	1024KB	QFN48

DEBUG control register (DEBUG_CTRL)

This register is asynchronously reset by POR Reset (not reset by system reset). It can be written by the debugger under reset.

Bit	Register	Reset value	Type	Description
Bit 31	I2C3_SMBUS_TIMEOUT	0x0	rw	I²C3 pause control bit 0: Work normally 1: I²C3 SMBUS timeout control is disabled
Bit 30	TMR11_PAUSE	0x0	rw	TMR11 pause control bit 0: Work normally 1: Timer is disabled
Bit 29	TMR10_PAUSE	0x0	rw	TMR10 pause control bit 0: Work normally 1: Timer is disabled
Bit 28	TMR9_PAUSE	0x0	rw	TMR9 pause control bit 0: Work normally 1: Timer is disabled
Bit 27	TMR14_PAUSE	0x0	rw	TMR14 pause control bit 0: Work normally 1: Timer is halted
Bit 26	TMR13_PAUSE	0x0	rw	TMR13 pause control bit 0: Work normally 1: Timer is disabled
Bit 25	TMR12_PAUSE	0x0	rw	TMR12 pause control bit 0: Work normally 1: Timer is disabled
Bit 24:22	Reserved	0x0	resd	Kept at 0
Bit 21	CAN2_PAUSE	0x0	rw	CAN2 pause control bit 0: CAN2 works normally 1: CAN2 receive registers do not continue to receive data
Bit 20	TMR7_PAUSE	0x0	rw	TMR7 pause control bit 0: Work normally 1: Timer is disabled
Bit 19	TMR6_PAUSE	0x0	rw	TMR6 pause control bit 0: Work normally 1: Timer is disabled
Bit 18	TMR5_PAUSE	0x0	rw	TMR5 pause control bit 0: Work normally 1: Timer is disabled
Bit 17	TMR8_PAUSE	0x0	rw	TMR8 pause control bit 0: Work normally 1: Timer is disabled
Bit 16	I2C2_SMBUS_TIMEOUT	0x0	rw	I²C2 pause control bit 0: Work normally 1: I²C2 SMBUS timeout control is disabled
Bit 15	I2C1_SMBUS_TIMEOUT	0x0	rw	I²C1 pause control bit 0: Work normally 1: I²C1 SMBUS timeout control is disabled

Bit	Register	Reset value	Type	Description
Bit 14	CAN1_PAUSE	0x0	rw	CAN1 pause control bit 0: CAN1 works normally 1: CAN1 receive registers do not continue to receive data
Bit 13	TMR4_PAUSE	0x0	rw	TMR4 pause control bit 0: Work normally 1: Timer is disabled
Bit 12	TMR3_PAUSE	0x0	rw	TMR3 pause control bit 0: Work normally 1: Timer is disabled
Bit 11	TMR2_PAUSE	0x0	rw	TMR2 pause control bit 0: Work normally 1: Timer is disabled
Bit 10	TMR1_PAUSE	0x0	rw	TMR1 pause control bit 0: Work normally 1: Timer is disabled
Bit 9	WWDT_PAUSE	0x0	rw	Window watchdog pause control bit 0: Window watchdog works normally 1: Window watchdog is stopped
Bit 8	WDT_PAUSE	0x0	rw	watchdog pause control bit 0: Watchdog works normally 1: Watchdog is stopped
Bit 7:6	TRACE_MODE	0x0	rw	Trace pin assignment control 00: Asynchronous mode 01: Snychronous mode with a data length of 1 10: Snychronous mode with a data length of 2 11: Snychronous mode with a data length of 4
Bit 5	TRACE_IOEN	0x0	rw	Trace pin assignment enable 0: No trace (default state) 1: Trace is enabled
Bit 4:3	Reserved	0x0	resd	Kept at 0
Bit 2	STANDBY_DEBUG	0x0	rw	Debug Standby mode control bit 0: The whole 1.2V digital circuit is unpowered in Standby mode 1: The whole 1.2V digital circuit is not unpowered in Standby mode, and the system clock is provided by the internal RC oscillator (HICK)
Bit 1	DEEPSLEEP_DEBUG	0x0	rw	Debug Deepsleep mode control bit 0: In Deepsleep mode, all clcoks in the 1.2V domain are disabled. When exiting from Deepsleep mode, the internal RC oscillator (HICK) is enabled, and HICK is used as the system clock source, and the software must reprogram the system clock according to application requirements. 1: In Deepsleep mode, system clock is provided by the internal RC oscillator (HICK). When exiting from Deepsleep mode, HICK is used as the system clock source, and the software must reprogram the system clock. according to application requirements.

Bit	Register	Reset value	Type	Description
Bit 0	SLEEP_DEBUG	0x0	rw	Debug Sleep mode control bit 0: When entering Sleep mode, CPU HCLK clock is disabled, but other clocks remain active. When exiting from Sleep mode, it is not necessary to reprogram the clock system. 1: When entering Sleep mode, all clocks keep running.

Copyright© 2023 by HOLTEK SEMICONDUCTOR INC. All Rights Reserved.

The information provided in this document has been produced with reasonable care and attention before publication, however, HOLTEK does not guarantee that the information is completely accurate. The information contained in this publication is provided for reference only and may be superseded by updates. HOLTEK disclaims any expressed, implied or statutory warranties, including but not limited to suitability for commercialization, satisfactory quality, specifications, characteristics, functions, fitness for a particular purpose, and non-infringement of any third-party's rights. HOLTEK disclaims all liability arising from the information and its application. In addition, HOLTEK does not recommend the use of HOLTEK's products where there is a risk of personal hazard due to malfunction or other reasons. HOLTEK hereby declares that it does not authorise the use of these products in life-saving, life-sustaining or safety critical components. Any use of HOLTEK's products in life-saving/sustaining or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold HOLTEK harmless from any damages, claims, suits, or expenses resulting from such use. The information provided in this document, including but not limited to the content, data, examples, materials, graphs, and trademarks, is the intellectual property of HOLTEK (and its licensors, where applicable) and is protected by copyright law and other intellectual property laws. No license, express or implied, to any intellectual property right, is granted by HOLTEK herein. HOLTEK reserves the right to revise the information described in the document at any time without prior notice. For the latest information, please contact us.