XIAOHUA SEMICONDUCTOR

# HC32L12x Series

## 32-bit ARM® Cortex®-M0+ Microcontroller

# Reference Manual

Rev1.20

August 2026

# Statement


## Copyright ©2026 Xiaohua Semiconductor Co., Ltd. All Rights Reserved.

All intellectual property rights related to the information contained in this document and its annexes belongs to Xiaohua Semiconductor Co., Ltd. (hereinafter referred to as "XHSC"). The customer only has the right to use the information contained in this document and its annexes. Without the written permission of XHSC, it is forbidden to extract, copy, modify, use or spread any part or all of this document in any form.

## Trademark Statement

XHSC XIAOHUA SEMICONDUCTOR, XHSC, and other trademarks are the trademarks of XHSC. All other products or services shown on XHSC products are the property of their respective owners.


## Precautions

- XHSC reserves the right to change, correct, enhance and modify the product and/or this document at any time without notice. Users can get the latest information before placing orders. XHSC products are sold in accordance with the sales terms and conditions stated in the basic purchase and sale contract.
- It is the customer's responsibility to select the appropriate XHSC product for your application and to design, validate and test your application to ensure that your application meets the appropriate standards and any safety, security or other requirements. The customer shall be solely responsible for this.
- XHSC hereby acknowledges that no intellectual property license has been granted by express or implied means.
- The resale of XHSC Products shall be invalidated by any warranty commitment of XHSC to such Products if its terms are different from those set forth herein.
- **Important Notice**: Before using this product, be sure to consult the latest official Errata Sheet to fully understand the product's known corrections and avoid usage risks.
- Information in this notification replaces information in previous versions.


## Xiaohua Semiconductor Co., Ltd.

| | |
|---|---|
| Address: | F4, Block A, No.1867 Zhongke Road, Pudong New Area, Shanghai, China |
| URL: | https://www.xhsc.com.cn |
| E-mail: | XHSC_MCU@xhsc.com.cn (Business Consulting)<br>XHSC_FAE@xhsc.com.cn (Technical Consulting) |

# Preface

## Bit/Bit Field

A register contains a bit (one bit wide) or a bit field (multiple bit width), and the bit field is usually described as [y:x], starting from the x bit and ending at the y bit. Unless otherwise specified, decimal numbers are used in the bit field.

It is not recommended to use unspecified fields; otherwise, unpredictable consequences may be caused.

## Register Address

The address of each register can be obtained by the base address of the system address division and the offset address of the register (see the "Register List" of each module for details).

## Access Type

The following abbreviations are used for register description:

| Access Type | Description |
|---|---|
| RW | Readable and writable. |
| RO | Read only. Writes are ignored. |
| WO | Write only. Read value undefined. |
| RWA | Readable and writable. Writable only when unlocked or in safe mode. |
| RWC | Readable and writable. Writing any value clears this bit. |
| RC | Read only. Writes are ignored. This bit is cleared automatically after reading. |
| RS | Read only. Writes are ignored. This bit is set automatically after reading. |

## Number Format

- 0x prefix is used for hexadecimal numbers
- 0b prefix is used for binary numbers
- Numbers without prefix are in decimal representation

## Reserved (Res.)

Registers and bit fields marked with *Reserved* are reserved for future use. These should be written to their reset value unless otherwise stated in the Register Description. The read value of Reserved bit may be different in different devices or features.

## Reset Value (Reset)

The reset value indicates the value after the register is reset. Registers need to be initialized and reset before use; otherwise, unpredictable consequences may be caused.

## Peripheral Availability Description

For HC32L12x Series 32-bit ARM® Cortex®-M0+ Microcontroller supported peripherals and their numbers, refer to the corresponding "Datasheet".

## Security Statement

There may be potential security problems due to the use of a certain function or protocol, which need to be declared to remind users and avoid security risks.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

# Contents

**Preface** ..... iii
**1 Product Introduction** ..... 25
**2 Storage and Architectures** ..... 26
2.1 Overview ..... 26
2.2 System Address Division ..... 27
2.3 Memory and Module Address Assignment ..... 28
**3 Operating Mode** ..... 29
3.1 Overview ..... 29
3.2 Working Mode Switching ..... 29
3.3 Active Mode ..... 30
3.4 Sleep Mode ..... 30
3.5 DeepSleep Mode ..... 31
3.6 Functional/Module Working Status ..... 32
3.7 Register Description ..... 34
3.7.1 System Control Register (SCB_SCR) (Cortex-M0+ Core System Control Register) ..... 34
**4 SYSCTRL** ..... 35
4.1 System Clock Introduction ..... 35
4.1.1 Clock Architecture Diagram ..... 35
4.1.2 Clock Types ..... 36
4.1.3 Clock Start Process ..... 37
4.2 System Clock Switching ..... 38
4.2.1 System Clock Switching Introduction ..... 38
4.2.2 Switching Process ..... 38
4.2.3 Examples of Switching ..... 39
4.2.4 Clock Switching Diagram ..... 41
4.3 Clock Calibration Module ..... 41
4.4 Interrupt Wakeup Control ..... 41
4.4.1 Executing Interrupt Service after Waking from DeepSleep mode ..... 42
4.4.2 Interrupt Service Not Executed after Waking from DeepSleep mode ..... 42
4.4.3 Using Sleep-on-exit Feature ..... 43
4.5 Reset Control ..... 43
4.5.1 Reset Controller Overview ..... 43
4.5.2 Reset Function ..... 44
4.6 Register Description ..... 45
4.6.1 Register List ..... 45
4.6.2 System Control Register 0 (SYSCTRL0) ..... 46
4.6.3 System Control Register 1 (SYSCTRL1) ..... 48
4.6.4 System Control Register 2 (SYSCTRL2) ..... 49
4.6.5 RCH Control Register (RCH_CR) ..... 50
4.6.6 XTH Control Register (XTH_CR) ..... 51
4.6.7 RCL Control Register (RCL_CR) ..... 53
4.6.8 XTL Control Register (XTL_CR) ..... 54

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd. 

4.6.9 Reset Flag Register (RESET_FLAG)....56
4.6.10 Peripheral Module Clock Control Register 0 (PERI_CLKEN0)....58
4.6.11 Peripheral Module Clock Control Register 1 (PERI_CLKEN1)....61
4.6.12 Peripheral Module Reset Control Register 0 (PERI_RESET0)....62
4.6.13 Peripheral Module Reset Control Register 1 (PERI_RESET1)....64
4.6.14 Peripheral Debug Function Enable Register (DEBUG_ACTIVE)....65
**5 NVIC....67**
5.1 Introduction....67
5.2 Functional Description....67
5.2.1 Interrupt Priority....67
5.2.2 Interrupt Digraph....68
5.2.3 Interrupt Input and Pending Behavior....69
5.2.4 Interrupt Waiting....71
5.2.5 Interrupt Source....71
5.2.6 Interrupt Structure Diagram....73
5.3 Operation Example....75
5.3.1 External Interrupt Enable....75
5.3.2 NVIC Interrupt Enable and Clear Enable....75
5.3.3 NVIC Interrupt Pending and Clear Pending....75
5.3.4 NVIC Interrupt Priority....75
5.3.5 NVIC Interrupt Mask....75
5.4 Register Description....77
5.4.1 Register List....77
5.4.2 Interrupt Enable Set Register (NVIC_ISER)....78
5.4.3 Interrupt Enable Clear Register (NVIC_ICER)....78
5.4.4 Interrupt Pending State Set Register (NVIC_ISPR)....79
5.4.5 Interrupt Pending State Clear Register (NVIC_ICPR)....79
5.4.6 Interrupt Priority Register (NVIC_IPR0)....80
5.4.7 Interrupt Priority Register (NVIC_IPR1)....80
5.4.8 Interrupt Priority Register (NVIC_IPR2)....81
5.4.9 Interrupt Priority Register (NVIC_IPR3)....81
5.4.10 Interrupt Priority Register (NVIC_IPR4)....82
5.4.11 Interrupt Priority Register (NVIC_IPR5)....82
5.4.12 Interrupt Priority Register (NVIC_IPR6)....83
5.4.13 Interrupt Priority Register (NVIC_IPR7)....83
5.4.14 Interrupt Mask Special Register (PRIMASK)....84
**6 FLASH....85**
6.1 Introduction....85
6.2 Functional Description....85
6.2.1 Capacity Division....85
6.2.2 Read Wait Period....85
6.2.3 Flash Security Protection....85
6.3 Operation Example....87
6.3.1 Sector Erase....87
6.3.2 Chip Erase....87

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd. 

6.3.3 Program....88
6.3.4 Cont-Program....89
6.3.5 Read....92
6.4 Register Description....93
6.4.1 Register List....93
6.4.2 CR Register (FLASH_CR)....94
6.4.3 IFR Register (FLASH_IFR)....95
6.4.4 ICLR Register (FLASH_ICLR)....96
6.4.5 BYPASS Register (FLASH_BYPASS)....97
6.4.6 SLOCK0 Register (FLASH_SLOCK0)....97
6.4.7 SLOCK1 Register (FLASH_SLOCK1)....98
6.4.8 Protection Status Register (FLASH_LOCK_ST)....99
**7 RAM....100**
7.1 Introduction....100
7.2 Functional Description....100
7.2.1 RAM Address Range....100
7.2.2 Read and Write Bit Width....100
7.2.3 Parity Check....100
7.3 Register Description....101
7.3.1 Register List....101
7.3.2 Control Register (RAM_CR)....102
7.3.3 Parity Check Error Address Register (RAM_ERRADDR)....102
7.3.4 Error Interrupt Flag Register (RAM_IFR)....103
7.3.5 Error Interrupt Flag Clear Register (RAM_ICLR)....103
**8 GPIO....104**
8.1 Introduction....104
8.2 Features....104
8.3 Functional Description....105
8.3.1 Port Configuration Function....105
8.3.2 Port Write....108
8.3.3 Port Read....108
8.3.4 Port Digital Multiplexing Function....110
8.3.5 Port Interrupt Function....113
8.3.6 Port Input Filter Function....113
8.3.7 Port Reset Status....113
8.3.8 HSI2C Communication Level Selection Function....113
8.4 Operation Example....114
8.4.1 Port Multiplexing Configured as Analog Port Operation Flow....114
8.4.2 Port Multiplexing Configured as Digital Universal Port Operation Flow....114
8.4.3 Port Multiplexing Configured as Digital Function Port Operation Flow....114
8.4.4 Port Multiplexing Configured as Debug Test Port Operation Flow....114
8.4.5 Port Multiplexing Configured as Infrared Output Signal Operation Flow....114
8.4.6 Port High Level Interrupt Operation Flow....115
8.4.7 Port Low Level Interrupt Operation Flow....115
8.4.8 Port Rising Edge Interrupt Operation Flow....115

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd. 

8.4.9 Port Falling Edge Interrupt Operation Flow....115
8.4.10 Port Pull-Up Enable Configuration Operation Flow....115
8.4.11 Port Pull-Down Enable Configuration Operation Flow....115
8.4.12 Port Enhancement Driver Configuration Operation Flow....116
8.4.13 Port Open-Drain Output Configuration Operation Flow....116
8.4.14 Port Bit Setting Operation Flow....116
8.4.15 Port Bit Clearing Operation Flow....116
8.4.16 Port Bit Setting and Clearing Operation Flow....116
8.4.17 Port Input Filtering Operation Flow....116
8.5 Register Description....117
8.5.1 Register List....117
8.5.2 Port General Register....121
8.5.3 Port Auxiliary Register....135
8.5.4 Port Function Selection Register....142
**9 DMAC....143**
9.1 Introduction....143
9.2 Features....143
9.3 Functional Description....144
9.3.1 Function Block Diagram....144
9.3.2 DMAC Transfer Mode....145
9.3.3 DMA Transfer Pause....148
9.3.4 Other Configurations of DMA....149
9.3.5 DMA Interrupt....149
9.4 Register Description....151
9.4.1 Register List....151
9.4.2 All Channel Configuration Register (DMAC_CONF)....152
9.4.3 Channel x Configuration A Register (DMAC_CONFAx) (x=0-1)....153
9.4.4 Channel x Configuration B Register (DMAC_CONFBx) (x=0-1)....155
9.4.5 Channel x Transfer Source Address Register (DMAC_SRCADRx) (x=0-1)....158
9.4.6 Channel x Transfer Destination Address Register (DMAC_DSTADRx) (x=0-1)....158
**10 CTRIM....159**
10.1 Introduction....159
10.2 Features....159
10.3 Functional Description....159
10.3.1 RCH Clock Calibration Mode....159
10.3.2 RCL Clock Calibration Mode....160
10.3.3 Low Power Timer Mode....161
10.4 Operation Example....161
10.4.1 RCH Clock Automatic Calibration Example....161
10.4.2 RCL Clock Automatic Calibration Example....162
10.5 Register Description....163
10.5.1 Register List....163
10.5.2 Autoload Register (CTRIM_ARR)....164
10.5.3 Error Counter Register (CTRIM_CNT)....164
10.5.4 Control Register 0 (CTRIM_CR0)....165


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

10.5.5 Control Register 1 (CTRIM_CR1)..... 167
10.5.6 Interrupt Enable Register (CTRIM_IER)..... 169
10.5.7 Interrupt and Status Register (CTRIM_ISR)..... 170
10.5.8 Interrupt Status Flag Register (CTRIM_ICR)..... 171
10.5.9 Error Result Register (CTRIM_FCAP)..... 172
10.5.10 TRIM Result Register (CTRIM_TVAL)..... 173
10.5.11 Error Upper Limit Register (CTRIM_FLIM)..... 173
**11 SYSTICK..... 174**
11.1 Introduction..... 174
11.2 Operation Example..... 174
11.2.1 Set SysTick..... 174
11.3 Register Description..... 175
11.3.1 Register List..... 175
11.3.2 SysTick Control and Status Register (CTRL)..... 176
11.3.3 SysTick Reload Register (LOAD)..... 177
11.3.4 SysTick Current Value Register (VAL)..... 178
11.3.5 SysTick Calibration Value Register (CALIB)..... 178
**12 LPTIM..... 179**
12.1 Introduction..... 179
12.2 Features..... 179
12.3 Functional Description..... 179
12.3.1 Function Block Diagram..... 179
12.3.2 Working Mode..... 179
12.3.3 Counting Function..... 181
12.3.4 Timing Function..... 181
12.3.5 PWM Pulse Output..... 181
12.3.6 GATE Interconnection..... 182
12.3.7 ETR Interconnection..... 182
12.4 Register Description..... 183
12.4.1 Register List..... 183
12.4.2 Counter Count Value Register (LPTIM_CNT)..... 184
12.4.3 Reload Register (LPTIM_ARR)..... 184
12.4.4 Compare Register (LPTIM_CMP)..... 185
12.4.5 Control Register (LPTIM_CR)..... 186
12.4.6 Interrupt Flag Register (LPTIM_IFR)..... 188
12.4.7 Interrupt Flag Clear Register (LPTIM_ICLR)..... 188
**13 TIM0/1/2/3..... 189**
13.1 Introduction..... 189
13.2 Features..... 189
13.2.1 TIM0/1/2 Features..... 189
13.2.2 TIM3 Features..... 190
13.3 Functional Description..... 191
13.3.1 Timer Clock..... 191
13.3.2 Timer Counter..... 191
13.3.3 Timer Prescaler..... 191

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd. 

13.3.4 Mode 0 Count Timer Function....192
13.3.5 Mode 1 Pulse Width Measurement PWC....195
13.3.6 Mode2/3 Compare Capture Mode....198
13.3.7 Mode2/3 Slave Mode....216
13.3.8 Quadrature Encoder Counting Function....218
13.3.9 Timer Triggers ADC....219
13.3.10 Cycle-By-Cycle Current Limiting and Brake Control....220
13.3.11 Interrupt....221
13.3.12 Timer Interconnection....222
13.3.13 Input Interconnection....222
13.3.14 DMA....222
13.4 Operation Example....223
13.4.1 Mode0 Counting Timer Function Configuration Example....223
13.4.2 Mode1 Pulse Width Measurement PWC Configuration Example....224
13.4.3 Mode2/3 Compare Capture Mode Configuration Example....224
13.4.4 Complementary PWM Output B Compare as ADC Trigger Function....226
13.4.5 Simultaneous PWM Output Startup for Timer TIM0/1/2 Configuration (Using TIM0 CTEN to Trigger TIM1/2) Example....226
13.4.6 Simultaneous PWM Output Startup for Timer TIM0/1/2 Configuration (Using TIM1 UG to Trigger TIM0/2) Example....227
13.4.7 DMA Configuration Example....227
13.5 Register Description....228
13.5.1 Register List....228
13.5.2 Mode0 Count Timer Register....230
13.5.3 Mode1 Pulse Width Measurement PWC Register....236
13.5.4 Mode2/3 Compare Capture Mode Register....244
**14 RTC....269**
14.1 Introduction....269
14.2 Features....269
14.3 Functional Description....270
14.3.1 Function Block Diagram....271
14.3.2 Power-on Setting....271
14.3.3 Clock Error Compensation....271
14.3.4 Timestamp Function....273
14.3.5 Wake-up Timer....273
14.3.6 RTC Interrupt....274
14.4 Operation Example....274
14.4.1 RTC Count Start Setting....274
14.4.2 Read Count Register....275
14.4.3 Write Count Register....275
14.4.4 Alarm Clock Setting....276
14.4.5 1Hz Output....276
14.5 Register Description....278
14.5.1 Register List....278
14.5.2 Control Register 0 (RTC_CR0)....279


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

14.5.3 Control Register 1 (RTC_CR1)..... 281
14.5.4 Second Count Register (RTC_SEC)..... 282
14.5.5 Minute Count Register (RTC_MIN)..... 282
14.5.6 Hour Count Register (RTC_HOUR)..... 283
14.5.7 Day Count Register (RTC_DAY)..... 285
14.5.8 Week Count Register (RTC_WEEK)..... 286
14.5.9 Month Count Register (RTC_MON)..... 287
14.5.10 Year Count Register (RTC_YEAR)..... 288
14.5.11 Clock Error Compensation Register (RTC_COMPEN)..... 289
14.5.12 High-Speed Compensation Accuracy Parameter Register (RTC_HCOMP)..... 291
14.5.13 Control Register 2 (RTC_CR2)..... 292
14.5.14 Alarm A Register (RTC_ALMA)..... 293
14.5.15 Alarm B Register (RTC_ALMB)..... 295
14.5.16 Timestamp Time Register (RTC_TSTR)..... 297
14.5.17 Timestamp Date Register (RTC_TSDR)..... 298
14.5.18 Wake-Up Timer Register (RTC_WUTR)..... 299
14.5.19 Interrupt Enable Register (RTC_IER)..... 299
14.5.20 Interrupt Flag Register (RTC_IFR)..... 301
14.5.21 Interrupt Clear Register (RTC_ICR)..... 302
14.5.22 Protection Register (RTC_WPR)..... 303
**15 IWDT..... 304**
15.1 Introduction..... 304
15.2 Features..... 304
15.3 Functional Description..... 304
15.3.1 Function Block Diagram..... 304
15.3.2 Window Watchdog Mode..... 304
15.3.3 Independent Watchdog Mode..... 305
15.3.4 Watchdog Overflow Handling..... 305
15.3.5 Watchdog Pause..... 305
15.3.6 Watchdog Stop and Run Again..... 305
15.3.7 Register Write Protection..... 305
15.4 Operation Example..... 305
15.4.1 Configure IWDT as Window Watchdog..... 305
15.4.2 Configure IWDT as Independent Watchdog..... 306
15.4.3 Watchdog Timeout Time (10kHz Independent Clock)..... 306
15.5 Register Description..... 307
15.5.1 Register List..... 307
15.5.2 Key Value Register (IWDT_KR)..... 308
15.5.3 Control Register (IWDT_CR)..... 309
15.5.4 Reload Register (IWDT_ARR)..... 310
15.5.5 Status Register (IWDT_SR)..... 311
15.5.6 Window Register (IWDT_WINR)..... 312
**16 WWDT..... 313**
16.1 Introduction..... 313
16.2 Features..... 313


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

16.3 Functional Description....313
16.3.1 Function Block Diagram....313
16.3.2 Enable Watchdog....314
16.3.3 Configure Watchdog Timeout....314
16.3.4 Update Down Counter....314
16.3.5 Pre-overflow Interrupt....314
16.4 Operation Example....314
16.4.1 Example of Configuring Watchdog Window Overflow Period....314
16.5 Register Description....315
16.5.1 Register List....315
16.5.2 Control Register 0 (WWDT_CR0)....316
16.5.3 Control Register 1 (WWDT_CR1)....317
16.5.4 Status Register (WWDT_SR)....318
**17 LPUART....319**
17.1 Introduction....319
17.2 Features....319
17.3 Functional Description....320
17.3.1 Function Block Diagram....320
17.3.2 Clock Description....320
17.3.3 Working Mode....320
17.3.4 Baud Rate Generation....326
17.3.5 Frame Error Detection....329
17.3.6 Multi-Machine Communication....329
17.3.7 Hardware Flow Control....330
17.3.8 DMA Transmission....331
17.3.9 Receive/Transmit Buffer....331
17.4 Operation Example....333
17.4.1 Transmit Data Example....333
17.4.2 Receive Data Example....333
17.5 Register Description....334
17.5.1 Register List....334
17.5.2 Data Register (LPUART_SBUF)....335
17.5.3 Control Register (LPUART_SCON)....336
17.5.4 Address Register (LPUART_SADDR)....339
17.5.5 Address Mask Register (LPUART_SADEN)....339
17.5.6 Interrupt Flag Bit Register (LPUART_ISR)....340
17.5.7 Interrupt Flag Bit Clear Register (LPUART_ICR)....341
17.5.8 Baud Rate Register (LPUART_SCNT)....341
17.5.9 Baud Rate Generate Register (LPUART_BSEL)....342
17.5.10 9600/4800 Baud Rate Modulator Register (LPUART_MODU)....343
**18 USART....345**
18.1 Introduction....345
18.2 Features....345
18.2.1 Asynchronous Serial Port Features....345
18.2.2 Synchronous Serial Port Features....345


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

18.2.3 Smart Card Interface Features..... 345
18.2.4 Infrared Communication Features..... 345
18.2.5 LIN Features..... 346
18.3 Functional Description..... 346
18.3.1 USART General Function..... 346
18.3.2 Asynchronous Serial Port Mode..... 355
18.3.3 Synchronous Serial Port Mode..... 359
18.3.4 Smart Card (7816) Mode..... 363
18.3.5 Infrared Mode..... 367
18.3.6 LIN Mode..... 368
18.4 Operation Example..... 373
18.4.1 Programming Example of Sending Data in Asynchronous Serial Port Mode..... 373
18.4.2 Programming Example of Receiving Data in Asynchronous Serial Port Mode..... 373
18.4.3 Infrared Mode 1: Transmission Data Setting..... 374
18.4.4 Infrared Mode 1: Reception Data Setting..... 374
18.5 Register Description..... 376
18.5.1 Register List..... 376
18.5.2 Control Register 1 (USART_CR1)..... 377
18.5.3 Control Register 2 (USART_CR2)..... 381
18.5.4 Control Register 3 (USART_CR3)..... 384
18.5.5 Baud Rate Register (USART_BRR)..... 387
18.5.6 Timer Counter Register (USART_TOR)..... 388
18.5.7 Status Register (USART_SR)..... 389
18.5.8 Transmit Data Register (USART_TDR)..... 392
18.5.9 Interrupt Enable Register (USART_IER)..... 393
18.5.10 Interrupt Flag Clear Register (USART_ICR)..... 395
18.5.11 Receive Data Register (USART_RDR)..... 396
18.5.12 Address Register (USART_ADDR)..... 396
18.5.13 Address Mask Register (USART_ADDMR)..... 397
18.5.14 Baud Rate Detection Value Register (USART_BDR)..... 397
18.5.15 Infrared Modulation Register (USART_IR)..... 398
**19 I2C..... 399**
19.1 Introduction..... 399
19.2 Features..... 399
19.3 Protocol Description..... 399
19.3.1 Data Transmission on the I2C Bus..... 399
19.3.2 Acknowledgment Signal on the I2C Bus..... 400
19.3.3 Arbitration on the I2C Bus..... 401
19.4 Functional Description..... 402
19.4.1 Function Block Diagram..... 403
19.4.2 Serial Clock Generator..... 403
19.4.3 Input Filter..... 404
19.4.4 Address Comparator..... 404
19.4.5 Acknowledgement Flag Bit..... 404
19.4.6 Interrupt Generator..... 404


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

19.4.7 Operating Mode....404
19.4.8 Status Code Expression....409
19.5 Operation Example....411
19.5.1 Master Sending Example....411
19.5.2 Master Receiving Example....411
19.5.3 Slave Sending Example....412
19.5.4 Slave Receiving Example....412
19.6 Register Description....414
19.6.1 Register List....414
19.6.2 Baud Rate Generator Enable Register (I2C_TMRUN)....415
19.6.3 Baud Rate Counter Configuration Register (I2C_TM)....415
19.6.4 Configuration Register (I2C_CR)....416
19.6.5 Data Register (I2C_DATA)....418
19.6.6 Status Register (I2C_STAT)....418
19.6.7 Slave Address Register 0 (I2C_ADDR0)....419
19.6.8 Slave Address Register 1 (I2C_ADDR1)....419
19.6.9 Slave Address Register 2 (I2C_ADDR2)....420
19.6.10 Slave Address Match Register (I2C_MATCH)....421
**20 HSI2C....422**
20.1 Introduction....422
20.2 Features....422
20.3 Functional Description....423
20.3.1 System Block Diagram....423
20.3.2 Protocol Description....423
20.3.3 Master Mode....424
20.3.4 Slave Mode....429
20.3.5 Communication Level Selection Function....431
20.4 Operation Example....431
20.4.1 Master Transmission Mode....431
20.4.2 Master Receiving Mode....432
20.4.3 Slave Transmission Mode....432
20.4.4 Slave Receiving Mode....433
20.4.5 Master High-Speed Mode Switching....433
20.5 Register Description....435
20.5.1 Register List....435
20.5.2 Master Control Register (I2C_MCR)....436
20.5.3 Master Status Register (I2C_MSR)....437
20.5.4 Master Interrupt Enable Register (I2C_MIER)....439
20.5.5 Master Status Clear Register (I2C_MSCR)....440
20.5.6 Master Configuration 1 Register (I2C_MCFGR1)....441
20.5.7 Master Configuration 2 Register (I2C_MCFGR2)....443
20.5.8 Master Configuration 3 Register (I2C_MCFGR3)....445
20.5.9 Master Clock Configuration Register 0 (I2C_MCCR0)....446
20.5.10 Master Clock Configuration Register 1 (I2C_MCCR1)....448
20.5.11 Master Transmit Data Register (I2C_MTDR)....450


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

20.5.12 Master Receive Data Register (I2C_MRDR)....451
20.5.13 Master Data Configuration Register (I2C_MDMR)....452
20.5.14 Master DMA Enable Register (I2C_MDER)....453
20.5.15 Slave Control Register 0 (I2C_SCR0)....454
20.5.16 Slave Status Register (I2C_SSR)....455
20.5.17 Slave Interrupt Enable Register (I2C_SIER)....458
20.5.18 Slave Status Clear Register (I2C_SSCR)....460
20.5.19 Slave Control 1 Register (I2C_SCR1)....461
20.5.20 Slave Control 2 Register (I2C_SCR2)....464
20.5.21 Slave Transmit ACK Register (I2C_STAR)....466
20.5.22 Slave Transmit Data Register (I2C_STDR)....467
20.5.23 Slave Receive Data Register (I2C_SRDR)....468
20.5.24 Slave DMA Enable Register (I2C_SDER)....469
20.5.25 Slave Address Register (I2C_SADDR)....470
20.5.26 Slave Address Status Register (I2C_SASR)....471
**21 SPI....472**
21.1 Introduction....472
21.2 Features....472
21.3 Functional Description....472
21.3.1 Function Block Diagram....472
21.3.2 Communication Format and Timing....473
21.3.3 Slave Select Pin Configuration....474
21.3.4 Full-duplex Communication....475
21.3.5 Single-wire Half-duplex Communication....476
21.3.6 Simplex Communication....476
21.3.7 Multi-machine Communication....477
21.3.8 Master Mode Delayed Sampling....478
21.3.9 Status Flags and Error Flags....479
21.3.10 Interrupt Generation and Clearing....482
21.4 Operation Example....482
21.4.1 Example of Sending and Receiving Data in Full-duplex by the Master....482
21.4.2 Example of Sending and Receiving Data in Full-duplex by the Slave....483
21.4.3 Single-wire Half-duplex Sending and Receiving Data Example....483
21.5 Register Description....485
21.5.1 Register List....485
21.5.2 SPI Control Register 0 (SPI_CR0)....486
21.5.3 SPI Control Register 1 (SPI_CR1)....488
21.5.4 SPI Single-wire Half-duplex Output Enable Register (SPI_HDOE)....490
21.5.5 SPI Internal Slave Select Register (SPI_SSI)....490
21.5.6 SPI Status Register (SPI_SR)....491
21.5.7 SPI Interrupt Clear Register (SPI_ICR)....493
21.5.8 SPI Data Register (SPI_DR)....494
**22 ADC....495**
22.1 Introduction....495
22.2 Features....495


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

22.3 Functional Description....496
22.3.1 ADC Block Diagram....496
22.3.2 Conversion Timing and Conversion Speed....496
22.3.3 Single Conversion Mode....497
22.3.4 Scan Conversion Mode....497
22.3.5 ADC Conversion Result Accumulation....499
22.3.6 ADC Conversion External Trigger Source....500
22.3.7 ADC Conversion Result Comparison....500
22.3.8 ADC Interrupt....501
22.3.9 Measure Ambient Temperature Using a Temperature Sensor....501
22.4 Operation Example....502
22.4.1 Start the ADC Single Conversion Operation Process through the ADC_SGLSTART.START Bit....502
22.4.2 ADC Single Conversion Operation Process through an External Trigger....503
22.4.3 Start the ADC SQR Scan Conversion Operation Process through the ADC_SQRSTART.START Bit....504
22.4.4 External Trigger Start ADC SQR Scan Conversion Operation Flow....504
22.4.5 Start ADC JQR Scan Conversion Operation Flow via ADC_JQRSTART.START Bit....505
22.4.6 External Trigger Start ADC JQR Scan Conversion Operation Flow....506
22.4.7 Start the ADC Continuous Conversion Accumulation Operation Process through the ADC_SQRSTART.START Bit....506
22.4.8 Start ADC Continuous Conversion Accumulation Operation Process through External Trigger....507
22.4.9 Operation Process of Measuring Ambient Temperature through ADC....508
22.5 Register Description....509
22.5.1 Register List....509
22.5.2 BGR Control Register (ADC_BGR)....511
22.5.3 ADC Basic Configuration Register 0 (ADC_CR0)....512
22.5.4 ADC Basic Configuration Register 1 (ADC_CR1)....515
22.5.5 ADC SQR Scan Conversion Channel Configuration Register 0 (ADC_SQR0)....516
22.5.6 ADC SQR Scan Conversion Channel Configuration Register 1 (ADC_SQR1)....517
22.5.7 ADC JQR Scan Conversion Channel Configuration Register (ADC_JQR)....518
22.5.8 ADC SQR Scan Conversion Channel x Result Register (ADC_SQRRESULTx) (x=0-5)....519
22.5.9 ADC JQR Scan Conversion Channel x Result Register (ADC_JQRRESULTx) (x=0-3)....519
22.5.10 ADC Conversion Result Register (ADC_RESULT)....520
22.5.11 ADC Conversion Result Accumulation Value Register (ADC_RESULTACC)....520
22.5.12 ADC Compare High Threshold Register (ADC_HT)....521
22.5.13 ADC Compare Low Threshold Register (ADC_LT)....521
22.5.14 ADC Interrupt Flag Register (ADC_IFR)....522
22.5.15 ADC Interrupt Clear Register (ADC_ICR)....523
22.5.16 ADC Single or SQR Scan Conversion External Interrupt Trigger Source Configuration Register (ADC_EXTTRIGGER0)....524
22.5.17 ADC JQR Scan Conversion External Interrupt Trigger Source Configuration Register (ADC_EXTTRIGGER1)....525
22.5.18 ADC Single Conversion Start Control Register (ADC_SGLSTART)....526
22.5.19 ADC SQR Scan Conversion Start Control Register (ADC_SQRSTART)....527
22.5.20 ADC JQR Scan Conversion Start Control Register (ADC_JQRSTART)....528
22.5.21 ADC Continuous Conversion Start/Stop Control Register (ADC_ALLSTART)....529
**23 LCD....530**
23.1 Introduction....530


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

23.2 Features....530
23.3 Functional Description....531
23.3.1 Function Block Diagram....531
23.3.2 Frame Rate....531
23.3.3 LCD Driving Waveform....531
23.3.4 Dead Zone Insertion....534
23.3.5 LCD Bias Generation Voltage....535
23.3.6 DMA....538
23.3.7 Interrupt....538
23.3.8 LCD Display Mode....538
23.4 Operation Example....544
23.4.1 Internal Resistance Mode Operating Process....544
23.4.2 External Capacitor Mode Operating Process....544
23.4.3 External Resistor Mode Operating Process....544
23.5 Register Description....546
23.5.1 Register List....546
23.5.2 Configuration Register 0 (LCD_CR0)....547
23.5.3 Configuration Register 1 (LCD_CR1)....549
23.5.4 Status Register (LCD_SR)....551
23.5.5 Interrupt Clear Register (LCD_INTCLR)....552
23.5.6 Output Configuration Register 0 (LCD_POEN0)....552
23.5.7 Output Configuration Register 1 (LCD_POEN1)....553
23.5.8 LCD_RAMx (x=0-7)....555
23.5.9 LCD_RAMx (x=8-F)....555
**24 VC....556**
24.1 Introduction....556
24.2 Features....556
24.3 Functional Description....557
24.3.1 Function Block Diagram....557
24.3.2 Setup/Response Time....557
24.3.3 Digital Filtering....558
24.3.4 Hysteresis Function....558
24.3.5 Configuration of Comparison Result Polarity....559
24.3.6 Window Comparison Function....559
24.3.7 Blank Window....559
24.4 Register Description....560
24.4.1 Register List....560
24.4.2 Control Register 0 (VCx_CR0) (x=0-1)....561
24.4.3 Control Register 1 (VCx_CR1) (x=0-1)....563
24.4.4 Control Register 2 (VCx_CR2) (x=0-1)....565
24.4.5 Status Register (VCx_SR) (x=0-1)....566
24.4.6 Offset Trim Register (VC_TRIM)....567
**25 LVD....568**
25.1 Introduction....568
25.2 Features....568


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

25.3 Functional Description....568
25.3.1 LVD Block Diagram....568
25.3.2 Digital Filtering....568
25.3.3 Hysteresis Function....569
25.4 Operation Example....570
25.4.1 LVD Configured as Low Voltage Reset....570
25.4.2 LVD Configured as Voltage Change Interrupt....570
25.5 Register Description....571
25.5.1 Register List....571
25.5.2 LVD Configuration Register (LVD_CR)....572
25.5.3 LVD Interrupt Flag Register (LVD_IFR)....575
**26 CRC....576**
26.1 Introduction....576
26.2 Functional Description....576
26.2.1 Function Block Diagram....576
26.2.2 CRC Parameter Model....576
26.2.3 CRC16 Encoding Mode....577
26.2.4 CRC16 Check Mode....577
26.2.5 CRC32 Encoding Mode....578
26.2.6 CRC32 Check Mode....578
26.3 Register Description....579
26.3.1 Register List....579
26.3.2 Control Register (CRC_CR)....580
26.3.3 Result Register (CRC_RESULT)....581
26.3.4 Data Register (CRC_DATA)....582
**27 TRNG....583**
27.1 Introduction....583
27.2 Functional Description....583
27.2.1 True Random Number Generation Principle....583
27.3 Operation Example....583
27.3.1 Generating 64bits True Random Number (The First Time After Power-On)....583
27.3.2 Generating 64bits True Random Number (Not Generated for the First Time After Power-on)....584
27.4 Register Description....585
27.4.1 Register List....585
27.4.2 Control Register (TRNG_CR)....586
27.4.3 Mode Register (TRNG_MODE)....587
27.4.4 Data Register 0 (TRNG_DATA0)....588
27.4.5 Data Register 1 (TRNG_DATA1)....588
**28 AES....589**
28.1 Overview....589
28.2 Features....590
28.3 Functional Description....590
28.3.1 Exception Mechanism....590
28.3.2 Runtime Instructions....590
28.4 Operation Example....591


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

28.4.1 Common Points of Encryption and Decryption Operations....591
28.4.2 Encryption Operation Procedures....591
28.4.3 Decryption Operation Procedures....591
28.4.4 Data Example....591
28.5 Register Description....595
28.5.1 Register List....595
28.5.2 Control Register (AES_CR)....596
28.5.3 Data Register (AES_DATAn) (n=0-3)....597
28.5.4 Key Register (AES_KEYn) (n=0-7)....598
28.5.5 Control Register Mapping Address (AES_CRDM)....599
**29 SWD Debug Interface....600**
29.1 Introduction....600
29.2 Functional Description....600
29.2.1 Debug System Diagram....600
29.2.2 ARM® Reference Documentation....601
29.2.3 Debug Port Pins....601
29.2.4 SWD Port....602
29.2.5 Core Debugging....605
29.2.6 BPU (Breakpoint Unit)....606
29.2.7 DWT (Data Watchpoint)....606
29.2.8 MCU Debug Component (DBG)....606
**30 Device Electronic Signature....608**
30.1 Introduction....608
30.2 Functional Description....608
30.2.1 Product Unique Identifier (UID)....608
30.3 Register Description....609
30.3.1 Product Unique Identifier (UID) Register (80 bits)....609
30.3.2 Product Model Register....609
30.3.3 FLASH Capacity Register....610
30.3.4 RAM Capacity Register....610
30.3.5 Pin Count/Flash Sector Size Configuration Register....611
**A Glossary....612**
**Revision History....613**

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd. 

# List of Figures

Figure 3-1 Working Mode State Transition Diagram....29
Figure 4-1 Clock Architecture Diagram....36
Figure 4-2 Crystal Oscillator Clock Startup Diagram....38
Figure 4-3 Clock Switching Diagram....41
Figure 4-4 Reset Source Diagram....44
Figure 5-1 Interrupt Vector Table....69
Figure 5-2 Interrupt Active and Pending States....70
Figure 5-3 Interrupt Pending State is Cleared and Then Reasserted....70
Figure 5-4 When the Interrupt Exits, If the Interrupt Request Remains High, It Will Cause the Interrupt Processing to Be Executed Again....70
Figure 5-5 Interrupt Pending Interrupts Generated During Interrupt Processing Can Also Be Acknowledged....71
Figure 5-6 Interrupt Structure Diagram....74
Figure 8-1 Port Circuit Diagram....105
Figure 8-2 AHB/FASTIO Bus Port Changes with System Clock....108
Figure 8-3 Port Pin Data Read Synchronization Diagram....108
Figure 8-4 GPIO Port Input Filter Function Timing Diagram....113
Figure 9-1 DMA Controller Functional Block Diagram....144
Figure 9-2 DMA Software Block Transfer Mode....145
Figure 9-3 DMA Software Burst Transfer Mode....146
Figure 9-4 DMA Hardware Block Transfer Mode....147
Figure 9-5 DMA Hardware Burst Transfer Mode....147
Figure 10-1 RCH Clock Calibration Operational Block Diagram....159
Figure 10-2 RCL Clock Calibration Operational Block Diagram....160
Figure 10-3 General Low Power Timer Scenario Operational Block Diagram....161
Figure 12-1 LPTimer Block Diagram....179
Figure 12-2 LPTimer Mode 1....180
Figure 12-3 LPTimer Mode 2....180
Figure 12-4 PWM Pulse Output Diagram....182
Figure 13-1 TIM0/1/2 Block Diagram....190
Figure 13-2 TIM3 Block Diagram....191
Figure 13-3 Mode0 Count Timer Free Counting Block Diagram....192
Figure 13-4 Mode0 Count Timer Reload Counting Block Diagram....193
Figure 13-5 Mode0 Count Timer 16-Bit Reload Counting Waveform....193
Figure 13-6 Mode0 Count Timer 32-Bit Free Counting Waveform....193
Figure 13-7 Mode0 Count Timer Free Counting Timing Diagram....194
Figure 13-8 Mode0 Count Timer Reload Counting Timing Diagram (Prescaler=2)....194
Figure 13-9 Mode1 Pulse Width Measurement PWC Function Block Diagram....195
Figure 13-10 Mode1: Pulse Width Measurement PWC High-Level Pulse Width Measurement....196
Figure 13-11 Mode1: Pulse Width Measurement PWC Falling Edge to Falling Edge Period Measurement....197
Figure 13-12 Mode1: Pulse Width Measurement PWC Rising Edge to Rising Edge Period Measurement....197
Figure 13-13 Mode1: Pulse Width Measurement PWC Rising Edge to Rising Edge Period Measurement Single Mode....197
Figure 13-14 Mode 2/3 Compare Capture Mode Counter Block Diagram....198
Figure 13-15 Mode 2 Compare Capture Mode Up-Counting Without Prescaler....199
Figure 13-16 Mode 2 Compare Capture Mode Up-Counting With Prescaler....199
Figure 13-17 Mode 2 Compare Capture Mode Down-Counting Without Prescaler....199
Figure 13-18 Mode 2 Compare Capture Mode Down-Counting With Prescaler....200
Figure 13-19 Mode2 Edge-Aligned Timer Waveform (M23CR.DIR=0)....200
Figure 13-20 Mode2 Edge-Aligned Timer Waveform (M23CR.DIR=1)....200
Figure 13-21 Mode3 Compare Capture Mode Up/Down-Counting with Prescaler....201
Figure 13-22 Mode3 Center-Aligned Counter Waveform....201

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd. 

Figure 13-23 Repetition Counter Generates Update Timing (RCR.OV=0, RCR.UND=0)....202
Figure 13-24 Update Timing Generated by Repetition Counter when Overflow Is Masked (RCR.OV=1, RCR.UND=0)....202
Figure 13-25 Update Timing Generated by Repetition Counter when Underflow Is Masked (RCR.OV=0, RCR.UND=1)....202
Figure 13-26 Mode 3 Triangular Wave Mode with Cache Enable....203
Figure 13-27 Mode 3 Triangular Wave Mode with Cache Invalid....203
Figure 13-28 Mode 2 Sawtooth Wave Mode Up-Counting with Cache Enable....204
Figure 13-29 Mode 2 Sawtooth Wave Mode Up-Counting with Cache Invalid....204
Figure 13-30 Mode 2 Sawtooth Wave Mode Down-Counting with Cache Enable....204
Figure 13-31 Mode 2 Sawtooth Wave Mode Down-Counting with Cache Invalid....205
Figure 13-32 Mode2 Sawtooth Wave Down-Counting Mode with Compare Cache Enable....205
Figure 13-33 OCREF Output Block Diagram....206
Figure 13-34 Sawtooth Counting Single Point Comparison OCREF Output Waveform (OCMyx=0b111)....207
Figure 13-35 Triangular Counting Single Point Comparison OCREF Output Waveform (OCMyx=0b111)....207
Figure 13-36 Sawtooth Counting Dual Point Comparison OCREF Output Waveform (OCMyx=0b111)....207
Figure 13-37 Triangular Counting Dual Point Comparison OCREF Output Waveform (OCMyx=0b111)....208
Figure 13-38 Independent PWM Output Block Diagram....209
Figure 13-39 PWM Output Waveform When CCPy=0....209
Figure 13-40 PWM Output Waveform When CCPy=1....209
Figure 13-41 Complementary PWM Output Block Diagram....210
Figure 13-42 Complementary PWM Output Waveform....210
Figure 13-43 Complementary PWM Output Waveform (with Dead Time)....211
Figure 13-44 Dead Time....211
Figure 13-45 Triangular Wave Mode Single Pulse Counting....212
Figure 13-46 Sawtooth Up-Counting Single Pulse Mode....212
Figure 13-47 Sawtooth Down-Counting Single Pulse Mode....213
Figure 13-48 Compare Interrupt Diagram....214
Figure 13-49 Capture Function Block Diagram....214
Figure 13-50 Capture Timing Diagram....214
Figure 13-51 TIM3 CH0A Capture Input Port Selection....215
Figure 13-52 CH0B Capture Input Port Selection....215
Figure 13-53 Mode2/3 Slave Mode Diagram....216
Figure 13-54 Mode2/3 Slave Mode Gating Function....217
Figure 13-55 Mode2/3 Slave Mode Trigger and Reset Function....218
Figure 13-57 Quadrature Encoder Counting Mode3 Counting Diagram....219
Figure 13-58 Timer Triggers ADC....220
Figure 13-59 DMA Request Trigger Sources....223
Figure 14-1 RTC Block Diagram....271
Figure 15-1 IWDT Function Block Diagram....304
Figure 16-1 WWDT Function Block Diagram....313
Figure 17-1 LPUART Function Block Diagram....320
Figure 17-2 Mode0 Data Transmitting....323
Figure 17-3 Mode0 Data Receiving....323
Figure 17-4 Mode1 Data Transmitting....324
Figure 17-5 Mode1 Data Receiving....324
Figure 17-6 Mode2/Mode3 Data Transmitting....324
Figure 17-7 Mode2/Mode3 Data Receiving....325
Figure 17-8 LPUART Hardware Flow Control....330
Figure 17-9 RTS Hardware Flow Control Signal....330
Figure 17-10 CTS Hardware Flow Control Signal....331
Figure 17-11 Receive Buffer....331
Figure 17-12 Transmit Buffer....332
Figure 18-1 Asynchronous Serial Port Mode Hardware Flow Control Example....351

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd. 

Figure 18-2 CTS Hardware Flow Control Signal....352
Figure 18-3 RTS Hardware Flow Control Signal....352
Figure 18-4 Asynchronous Serial Port Function Block Diagram....355
Figure 18-5 Asynchronous Serial Port Mode Data Format....356
Figure 18-6 Transmit Data Example....356
Figure 18-7 Receive Data Example....357
Figure 18-8 Multiprocessor Communication Data Format....358
Figure 18-9 Synchronize Serial Port Mode Data Format....359
Figure 18-10 Synchronous Serial Port Mode Sending Data Legend 1 (LSB/No CTS Function)(Use Internal Clock Source)....360
Figure 18-11 Synchronous Serial Port Mode Sending Data Legend 1 (LSB/No CTS Function) (Use External Clock Source)....361
Figure 18-12 Synchronous Serial Port Mode to Send Data Legend 2 (LSB/CTS Function)....361
Figure 18-13 Synchronous Serial Port Mode Receiving Data Legend 1 (LSB/No RTS Function)....362
Figure 18-14 Synchronous Serial Port Mode Receiving Data Legend 2 (LSB/RTS Function)....362
Figure 18-15 Precautions for Synchronous Serial Port Mode....363
Figure 18-16 Smart Card Connection Schematic Diagram....364
Figure 18-17 Smart Card Mode Internal Synchronization Timing and Sample Timing Diagram....364
Figure 18-18 Smart Card Mode Data Format....365
Figure 18-19 Smart Card Mode Data Format Example....365
Figure 18-20 Smart Card Mode Transmit Data Legend....366
Figure 18-21 Smart Card Mode Receive Data Legend....367
Figure 18-22 Infrared Data Modulation....367
Figure 18-23 38K Infrared Modulation....368
Figure 18-24 LIN Bus Data and Frame Structure....369
Figure 19-1 I2C Transmission Protocol....399
Figure 19-2 START, Repeated START and STOP....400
Figure 19-3 Bit Transfer on I2C Bus....400
Figure 19-4 Acknowledgment Signal on I2C Bus....401
Figure 19-5 Arbitration on the I2C Bus....402
Figure 19-6 I2C Function Block Diagram....403
Figure 19-7 Data Synchronization Diagram of Master Sending Mode....405
Figure 19-8 I2C Master Sending State Diagram....405
Figure 19-9 Data Synchronization Diagram of Master Receiving Mode....406
Figure 19-10 I2C Master Receiving State Diagram....406
Figure 19-11 Slave Receiving Mode Data Synchronization Diagram....407
Figure 19-12 Slave Receiving State Diagram....407
Figure 19-13 Slave Sending Mode Data Synchronization Diagram....408
Figure 19-14 I2C Slave Sending State Diagram....408
Figure 19-15 I2C Broadcast Call State Diagram....409
Figure 20-1 I2C System Block Diagram....423
Figure 20-2 I2C Bus Timing Diagram....424
Figure 20-3 I2C Bus Timing Parameters Diagram....425
Figure 21-1 SPI Block Diagram....472
Figure 21-2 SPI Communication Format Timing Diagram....474
Figure 21-3 Slave Select NSS Pin Configuration....475
Figure 21-4 SPI Full-duplex Communication....476
Figure 21-5 SPI Single-wire Half-duplex Communication....476
Figure 21-6 SPI Simplex Transmit Communication Diagram....477
Figure 21-7 SPI Simplex Receive Communication Diagram....477
Figure 21-8 SPI Multi-machine System....478
Figure 21-9 Explanation Diagram of General Flags in SPI Master Mode....481
Figure 21-10 SPI Slave Mode General Flag Bit Description Diagram....481
Figure 22-1 ADC Block Diagram....496

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd. 

Figure 22-2 ADC Conversion Timing Diagram....496
Figure 22-3 Examples of ADC SQR Scan Conversion Process....497
Figure 22-4 Examples of ADC JQR Scan Conversion Process....498
Figure 22-5 Examples of JQR Scan During ADC SQR Scan Process....499
Figure 22-6 Examples of Continuous ADC Conversion Accumulation Process....500
Figure 22-7 ADC Conversion Result Comparison Diagram....501
Figure 23-1 LCD Function Block Diagram....531
Figure 24-1 VC Function Block Diagram....557
Figure 24-2 VC Filter Response Time....558
Figure 24-3 VC Hysteresis Function....559
Figure 24-4 VC Blank Window....559
Figure 25-1 LVD Block Diagram....568
Figure 25-2 LVD Filter Output....569
Figure 25-3 LVD Hysteresis Response....570
Figure 26-1 CRC Application Diagram....576
Figure 27-1 True Random Number Generator Function Block Diagram....583
Figure 28-1 AES Encryption and Decryption Schematic Diagram....589
Figure 28-2 AES Encryption Flowchart....589
Figure 29-1 Debug Support Block Diagram....601

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

# List of Tables

Table 2-1 Address Division Table....28
Table 3-1 Reset Type Response Capabilities in Different Operational Mode....29
Table 3-2 Peripheral Working Status Table in Each Mode....32
Table 4-2 System Controller (SYSCTRL) Register Offset....45
Table 5-1 Cortex-M0+ Processor Interrupt List....67
Table 5-4 Interrupt Controller (NVIC) Register Offset Address....77
Table 6-1 Flash Capacity Division Table....85
Table 6-3 Flash Controller (FLASH) Register Offset....93
Table 7-1 RAM Address Mapping....100
Table 7-2 RAM Controller (RAM) Register Offset....101
Table 8-1 Port Status Truth Table....107
Table 8-2 Port Multiplexing Table....110
Table 8-3 General-Purpose I/O (GPIO) Register Offset....117
Table 9-1 DMAC Transfer Mode Comparison Table....145
Table 9-2 DMA Controller (DMA) Register Offset....151
Table 10-1 Clock Trim Module (CTRIM) Register Offset....163
Table 11-2 SysTick Timer Register Offset....175
Table 12-2 Low Power Timer (LPTIM) Register Offset....183
Table 13-8 CH0A Capture Input Signal....215
Table 13-13 TIM0/1/2/3 Interrupt List....221
Table 13-14 Timer Interconnection Signal List....222
Table 13-15 Mode0 Timer Register Offset....228
Table 13-16 Mode1 Pulse Width Measurement (PWM) Register Offset....228
Table 13-17 Mode2/3 Compare-Capture Mode Register Offset....228
Table 14-1 Basic Features of RTC....269
Table 14-2 Real Time Counter (RTC) Register Offset....278
Table 15-1 IWDT Timeout Time (10kHz Independent Clock)....306
Table 15-2 Independent Watchdog (IWDT) Register Offset....307
Table 16-1 Window Watchdog (WWDT) Register Offset....315
Table 17-1 LPUART Mode0/1/2/3 Data Structure....321
Table 17-2 B8 Data Meaning....321
Table 17-3 SCLK = 4MHz Baud Rate Calculation Table....326
Table 17-4 SCLK = 8MHz Baud Rate Calculation Table....327
Table 17-5 SCLK = 16MHz Baud Rate Calculation Table....327
Table 17-6 SCLK = 24MHz Baud Rate Calculation Table....327
Table 17-7 SCLK = 32MHz Baud Rate Calculation Table....328
Table 17-8 SCLK = 48MHz Baud Rate Calculation Table....328
Table 17-9 Low Power Universal Asynchronous Receiver/Transmitter (LPUART) Register Offset....334
Table 18-1 Each Mode Sample Count and Maximum Baud Rate....346
Table 18-2 Baud Rate Programming Example with PCLK at 4MHz....347
Table 18-3 Baud Rate Programming Example with PCLK at 8MHz....348
Table 18-4 Baud Rate Programming Example with PCLK at 16MHz....348
Table 18-5 Baud Rate Programming Example with PCLK at 24MHz....348
Table 18-6 Baud Rate Programming Example with PCLK at 32MHz....349
Table 18-7 Baud Rate Programming Example with PCLK at 48MHz....349
Table 18-8 Multi-Function Timing Counter Function....353
Table 18-9 Each Mode Interrupt Validity....354
Table 18-10 Universal Synchronous Asynchronous Receiver/Transmitter (USART) Register Offset....376
Table 19-1 I2C Clock Signal Baud Rate....403
Table 19-2 I2C Status Code Expression....409

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd. 

Table 19-3 I2C Bus (I2C) Register Offset..... 414
Table 20-1 Host Timing Parameters..... 426
Table 20-2 Synchronization Delay..... 426
Table 20-3 I2C Timing Parameter Limitations..... 426
Table 20-4 I2C Example Timing Configuration..... 427
Table 20-7 High Speed Integrated Circuit Bus (HSI2C) Register Offset..... 435
Table 21-1 SPI Interrupt..... 482
Table 21-2 Serial Peripheral Interface (SPI) Register Offset..... 485
Table 22-1 Preemptive Behavior State Table..... 499
Table 22-2 ADC Interrupt..... 501
Table 22-3 Temperature Sensor Calibration Value Information List..... 502
Table 22-4 Analog-to-Digital Converter (ADC) Register Offset..... 509
Table 23-12 Liquid Crystal Display Controller (LCD) Register Offset..... 546
Table 23-15 Configuration of COMy/SEGx PAD Functions Under Different Duty Ratios..... 548
Table 24-1 Analog Voltage Comparator (VC) Register Offset..... 560
Table 25-1 Low Voltage Detector (LVD) Register Offset..... 571
Table 26-1 CRC Parameter Model..... 576
Table 26-2 CRC Operation (CRC) Register Offset..... 579
Table 27-1 True Random Number Generator (TRNG) Register Offset..... 585
Table 28-1 AES Encryption and Decryption Running Time..... 590
Table 28-2 128-Bit Operation Register Example..... 592
Table 28-3 192-Bit Operation Register Example..... 593
Table 28-4 256-Bit Operation Register Example..... 594
Table 28-5 Advanced Encryption Standard Module (AES) Register Offset..... 595
Table 28-11 Fill-in data for Register..... 597
Table 28-14 Fill-in data for Register - 128-bit key..... 598
Table 28-15 Fill-in data for Register - 192-bit key..... 598
Table 28-16 Fill-in data for Register - 256-bit key..... 598
Table 29-3 Host Transmitted Data Packet Request..... 603
Table 29-4 Target Transmitted Acknowledgment Response..... 603
Table 29-5 Data Transfers Transmitted by Host or Target..... 603

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd. 

# 1 Product Introduction

HC32L12x Series is an ultra-low power consumption MCU with a wide operating voltage range designed to extend the battery life of portable measurement systems. Integrates a 12-bit 2Msps high-precision SARADC, comparator, built-in PWM timer, LCD display, USART, SPI, I2C, and other communication peripherals. Includes AES and TRNG security modules, delivering high integration, high noise immunity, high reliability, and ultra-low power consumption. This product adopts the Cortex-M0+ core with a maximum frequency of 48MHz, compatible with mature mainstream development and debugging software such as Keil, IAR, and Eclipse, and supports development using C language and assembly language.

## Typical Applications of Ultra-low Power MCU

Main application scenarios include:

- Sensor applications, IoT applications
- Smart meters, wireless modules, thermostats, electronic shelf labels
- Intelligent transportation, alarm systems
- Smart home, medical devices

## About This Manual

This manual describes the chip's functionalities, operational guidelines, and usage methods. For detailed chip specifications, refer to the corresponding "Datasheet".

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

# 2 Storage and Architectures

## 2.1 Overview

System consists of the following parts:

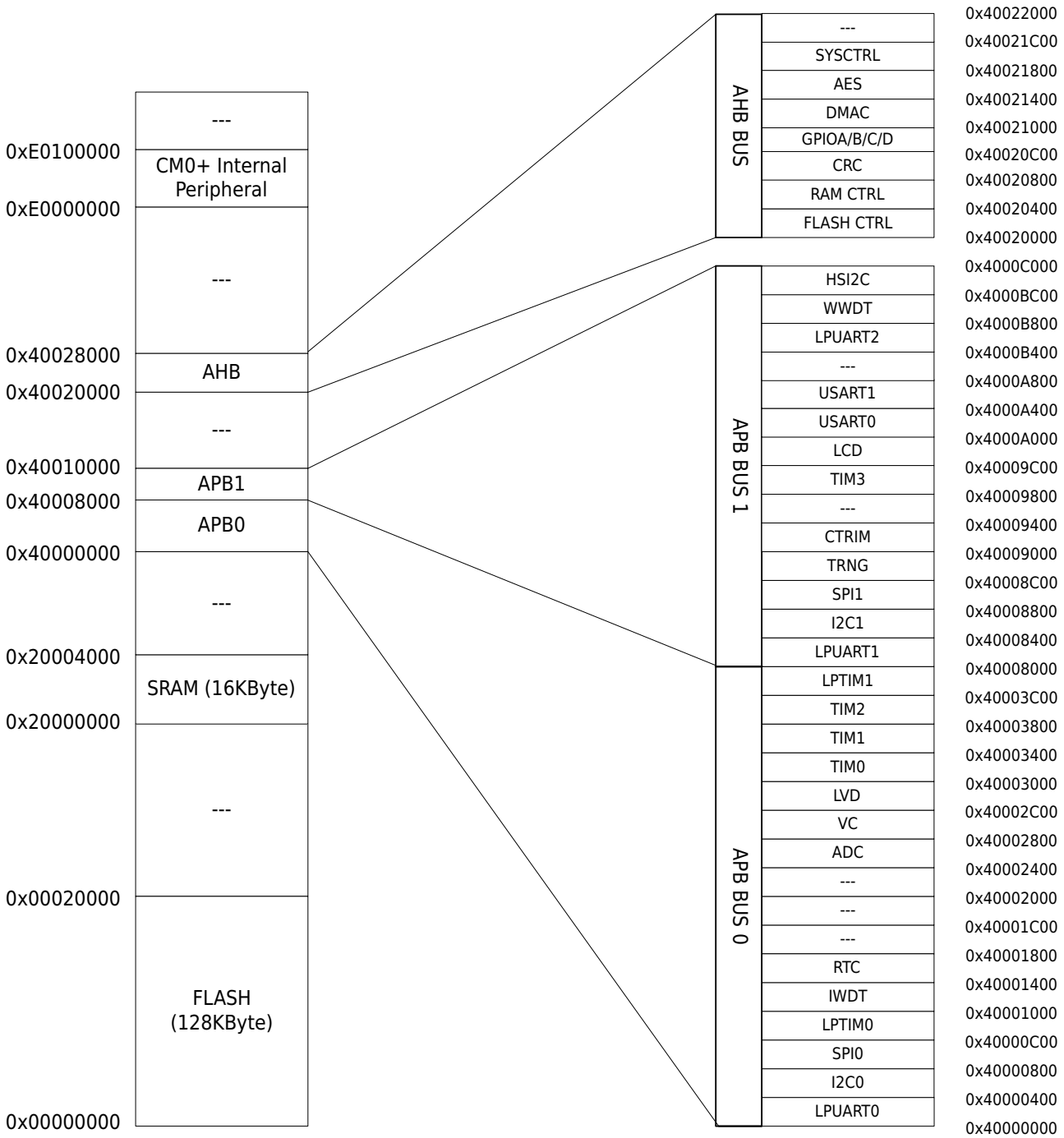
- 2 AHB bus Master:
  - Cortex-M0+
  - DMA controller
- Multiple AHB bus Slaves:
  - Flash Memory
  - SRAM Memory
  - AHB to APB Bridge 0 includes all APB0 interface peripherals
  - AHB to APB Bridge 1 includes all APB1 interface peripherals
  - AHB peripherals

The entire system bus structure is realized by multi-level AHB-lite bus interconnection.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

## 2.2 System Address Division

System of address region division:





All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

## 2.3 Memory and Module Address Assignment

Table 2-1 Address Division Table

| Categories | Boundary Address | Size | Memory Area |
|---|---|---|---|
| Memory | 0x00000000-0x0001FFFF | 128KByte | FLASH |
| | 0x20000000-0x20003FFF | 16KByte | SRAM |
| APB0 peripheral | 0x40000000-0x40007FFF | 32KByte | - |
| APB1 peripherals | 0x40008000-0x4000FFFF | 32KByte | - |
| AHB peripherals | 0x40020000-0x40027FFF | 32KByte | - |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

# 3 Operating Mode

## 3.1 Overview

This product can realize 3 working modes under the control of the power management module. The operable modules and their power consumption differ across working modes.

This product has the following working modes:

1. Active Mode: CPU running, peripheral functional modules running.
2. Sleep mode: CPU halted, peripheral functional modules running.
3. DeepSleep mode: CPU halted, high-speed clock modules disabled.

## 3.2 Working Mode Switching

The product can be freely switched among 3 working modes when it is working. In Active mode, executing specific instructions can transition the device into Sleep mode or DeepSleep mode. Wake-up from Sleep or DeepSleep mode via an interrupt and return to Active mode.

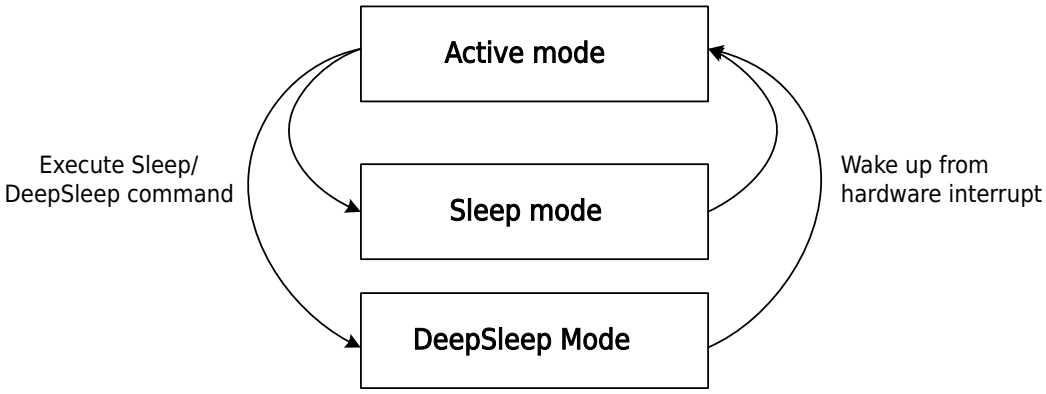



Figure 3-1 Working Mode State Transition Diagram

The interrupts that the CPU can respond in various working modes, please refer to Interrupt Source in "Interrupt Controller (NVIC)" Chapter.

In different working mode, the reset types system can respond following to:

Table 3-1 Reset Type Response Capabilities in Different Operational Mode

| Number | Reset Source | Active Mode | Sleep mode | DeepSleep mode |
|---|---|---|---|---|
| [0] | Power-on Reset (POR)/Power-down Reset (PDR) | ✓ | ✓ | ✓ |
| [1] | External Reset Pin Reset | ✓ | ✓ | ✓ |
| [2] | LVD Reset | ✓ | ✓ | ✓ |
| [3] | IWDT Reset | ✓ | ✓ | ✓ |
| [4] | Cortex-M0+ LOCKUP Hardware Reset | ✓ | × | × |
| [5] | Cortex-M0+ SYSRESETREQ Software Reset | ✓ | × | × |
| [6] | WWDT Reset | ✓ | ✓ | × |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

## 3.3 Active Mode

The MCU is in the Active mode after system is reset at power-on, or after waking up from various low power consumptions. Various low-power modes are available to conserve power when the CPU is not required to continue running, such as while waiting for an external event. Users need to select an optimal low-power mode based on the lowest power consumption, fastest startup time, and available wake-up sources.

In Active mode, several ways to reduce chip power consumption in run mode:

1. In Active mode, the speed of any system clock (HCLK, PCLK) can be reduced by programming the prescaler registers (SYSCTRL0.HCLK_PRS, SYSCTRL0.PCLK_PRS). The prescaler can also be used to slow down the clocks to peripherals before entering Sleep mode.
2. In Active mode, disable the clocks of unused peripherals (PERI_CLKENx) to reduce power consumption.
3. Turn off the clocks of unused peripheral clock (PERI_CLKENx) before executing WFI instructions.
4. In certain scenarios, low-power mode can be used instead of Active mode, as this product features extremely short wake-up time that can still meet system real-time response requirements.

## 3.4 Sleep Mode

Executing the WFI instruction enters Sleep mode. In Sleep mode, the CPU halted, but the clock module, system clock, NVIC interrupt processing and peripheral functional modules can still work.

When the system enters the sleep state, the port state will not be changed. Before entering the sleep state, the IO state can be changed as needed.

### Enter Sleep Mode

Enter sleep state by executing WFI instruction. There are two options available for selecting the Sleep mode entry mechanism based on value of the SLEEPONEXIT bit in the system control register of Cortex-M0+:

- SLEEP-NOW: If the SLEEPONEXIT bit is cleared, the MCU enters Sleep mode immediately when WFE is executed.
- SLEEP-ON-EXIT: If the SLEEPONEXIT bit is set, the MCU enters Sleep mode immediately when the system exits from the lowest priority interrupt handler.

### Exit Sleep Mode

If the WFI instruction is executed to enter the Sleep mode, any peripheral interrupt responded by the as the high-priority nested vector interrupt controller can wake up the system from the Sleep mode.

### Precautions

- The position of SLEEPONEXIT is 1, and the interrupt will automatically enter Sleep after execution, and the program does not need to write __wfi();
- SLEEPONEXIT bit is cleared to 0, main() executes __wfi() and enters Sleep, interrupt triggers and executes the interrupt program and returns to main(), then executes WFI instruction and enters Sleep. Wait for a subsequent interrupt to fire.
- The SLEEPONEXIT bit does not affect the execution of the __wfi() instruction. SLEEPONEXIT = 0: main() executes __wfi() and enters Sleep, interrupt triggers and executes the interrupt program and returns to main(), and then continues to execute;


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

- If Sleep is entered in an interrupt, only interrupts with a priority higher than this interrupt can wake up, and the high priority is executed first, and then the low priority is executed; interrupts with a priority lower than or equal to this interrupt cannot wake up.
- For details on entering and waking up from Sleep mode via WFE (Wait For Event) instruction, refer to the Cortex-M0+ relevant technical manual.

## 3.5 DeepSleep Mode

Using SLEEPDEEP with the WFI instruction enables entry into DeepSleep mode. In DeepSleep mode, the CPU halted, the high-speed clock is disabled, the low-speed clock can be configured to remain active or not, certain low-power peripheral modules may be selectively enabled, while the NVIC interrupt handling remains functional.

- When the system (Using high-speed clock) enters DeepSleep mode, the high-speed clock is automatically disabled, and the low-speed clock remains in the state before entering DeepSleep.
- The system (Using low-speed clock) enters the DeepSleep mode, the low-speed clock is not automatically disabled and maintains its operation. Only ARM Cortex-M0+ is not running, other modules are running.
- When the system clock is switched, all clocks will not be automatically disabled, and the corresponding clocks need to be disabled or enabled by software according to power consumption and system requirements.
- When the system enters the DeepSleep mode, it will not change the port state. Before entering the DeepSleep mode, change the IO state as needed.

### Enter DeepSleep Mode

First set the SLEEPDEEP bit of Cortex-M0+ in the system control register, and enter the DeepSleep mode by executing the WFI instruction. Depending on the value of the SLEEPONEXIT bit in the system control register, there are two options for selecting the DeepSleep mode entry mechanism:

- SLEEP-NOW: If the SLEEPONEXIT bit is cleared, the MCU enters DeepSleep mode immediately when WFI is executed.
- SLEEP-ON-EXIT: If the SLEEPONEXIT bit is set, the MCU enters DeepSleep mode immediately when the system exits from the lowest priority interrupt handler.

### Exit DeepSleep Mode

If the WFI instruction is executed to enter DeepSleep mode, any peripheral interrupt that is responded to by the high priority nested vector interrupt controller (operable under DeepSleep peripheral module interrupt) can wake up the system from DeepSleep mode.

For wake-up configuration, refer to the "Interrupt Wakeup Control" section in the "System Controller (SYSCRTL)" chapter.

### Precautions

- The position of SLEEPONEXIT is 1, and the interrupt will automatically enter DeepSleep after execution, and the program does not need to write __wfi();
- SLEEPONEXIT bit is cleared to 0, main() executes __wfi() and enters DeepSleep, interrupt triggers and executes the interrupt program and returns to main(), then executes WFI instruction and enters DeepSleep. Wait for a subsequent interrupt to fire.
- The SLEEPONEXIT bit does not affect the execution of the __wfi() instruction. SLEEPONEXIT = 0: main() executes __wfi() and enters DeepSleep, interrupt triggers and executes the interrupt program and returns to main(), and then continues to execute.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

- If DeepSleep is entered in an interrupt, only interrupts with a priority higher than this interrupt can wake up, and the high priority is executed first, and then the low priority is executed; interrupts with a priority lower than or equal to this interrupt cannot wake up.
- For details on entering and waking up from DeepSleep mode via WFE (Wait For Event) instruction, refer to the Cortex-M0+ relevant technical manual.

## 3.6 Functional/Module Working Status

The comparison of the working status of the functional/modules in the three working modes is shown in the table below:

Table 3-2 Peripheral Working Status Table in Each Mode

| Functional/Module | Active Mode | Sleep Mode | DeepSleep Mode |
|---|---|---|---|
| Cortex-M0+ | ✓ | × | × |
| XTH | ✓ | ✓ | × |
| XTL | ✓ | ✓ | ✓ |
| RCH | ✓ | ✓ | × |
| RCL | ✓ | ✓ | ✓ |
| RESET | ✓ | ✓ | ✓ |
| GPIO | ✓ | ✓ | ✓ |
| I2C0/1 | ✓ | ✓ | × |
| HSI2C | ✓ | ✓ | ✓(3) |
| SPI0/1 | ✓ | ✓ | × |
| CTRIM | ✓ | ✓ | ✓(2) |
| FLASH | ✓ | ✓ | × |
| RAM | ✓ | ✓ | × |
| DMAC | ✓ | ✓ | × |
| TIM0/1/2/3 | ✓ | ✓ | × |
| LPTIM0/1 | ✓ | ✓ | ✓(2) |
| RTC | ✓ | ✓ | ✓(2) |
| IWDT | ✓ | ✓ | ✓ |
| WWDT | ✓ | ✓ | × |
| USART0/1 | ✓ | ✓ | × |
| LPUART0/1/2 | ✓ | ✓ | ✓(2) |
| CRC | ✓ | ✓ | × |
| TRNG | ✓ | ✓ | × |
| AES | ✓ | ✓ | × |
| LCD | ✓ | ✓ | ✓ |
| ADC | ✓ | ✓ | × |
| VC0/1 | ✓ | ✓ | ✓(1) |
| LVD | ✓ | ✓ | ✓ |
| SWD | ✓ | ✓ | × |
| POR/PDR | ✓ | ✓ | ✓ |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

**Notice**

1. If you use the VC wake-up function in DeepSleep mode, the VC filter clock cannot select PCLK.
2. When the working clock is PCLK/high-speed clock, wakeup from DeepSleep mode is not supported.
3. Only slave mode supports wakeup from DeepSleep.
4. Before executing the WFI command to enter DeepSleep mode, all interrupts not used for MCU wakeup must be disabled.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

## 3.7 Register Description

### 3.7.1 System Control Register (SCB_SCR) (Cortex-M0+ Core System Control Register)

Base address: 0xE000ED10

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| - | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | SEVONPEND | RESERVED | SLEEPDEEP | SLEEPONEXIT | RESERVED |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | | | RW | | RW | RW | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:5* | *Reserved* | - | - |
| 4 | SEVONPEND | Interrupt Pending Flag<br>● 0b1: A new event is generated upon each interrupt pending. If WFE is used to enter Sleep or DeepSleep mode, this event can wake up the processor | RW |
| *3* | *Reserved* | - | - |
| 2 | SLEEPDEEP | Sleep Mode Selection<br>● 0b0: Sleep mode<br>● 0b1: DeepSleep mode | RW |
| 1 | SLEEPONEXIT | Auto-Sleep Mark<br>● 0b0: This feature disabled<br>● 0b1: When exiting exception handling and returning to thread mode, the processor automatically enters Sleep or DeepSleep mode (WFI) | RW |
| *0* | *Reserved* | - | - |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

# 4 SYSCTRL

## 4.1 System Clock Introduction

The clock control module primarily provides system clock, peripheral clock, and peripheral reset control functions.

This product provides the following different clock sources as the system clock:

- RCH-Internal high-speed RC clock (output frequency: 4/6/32/48 MHz)
- RCL - Internal low-speed RC clock (configurable to 38.4 kHz or 32.768 kHz)
- XTH - External high-speed crystal oscillator clock
- XTL - External low-speed crystal oscillator clock

**Note**

1. When switching the system clock source, please follow the procedure strictly. See the System Clock Switching section for details.
2. XTL can not be connected to the crystal oscillator, and directly input the 32.768kHz clock signal from the PC14 pin. XTH can not be connected to the crystal oscillator, and directly input a maximum frequency of 32MHz clock signal from the PD00 pin.

This product also contains the following two auxiliary clocks:

- RC10K - Internal 10kHz clock, for IWDT modules
- RC256K - Internal 256kHz clock, for LVD and VC modules.

### 4.1.1 Clock Architecture Diagram

The clock architecture of this product is shown as follows:


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

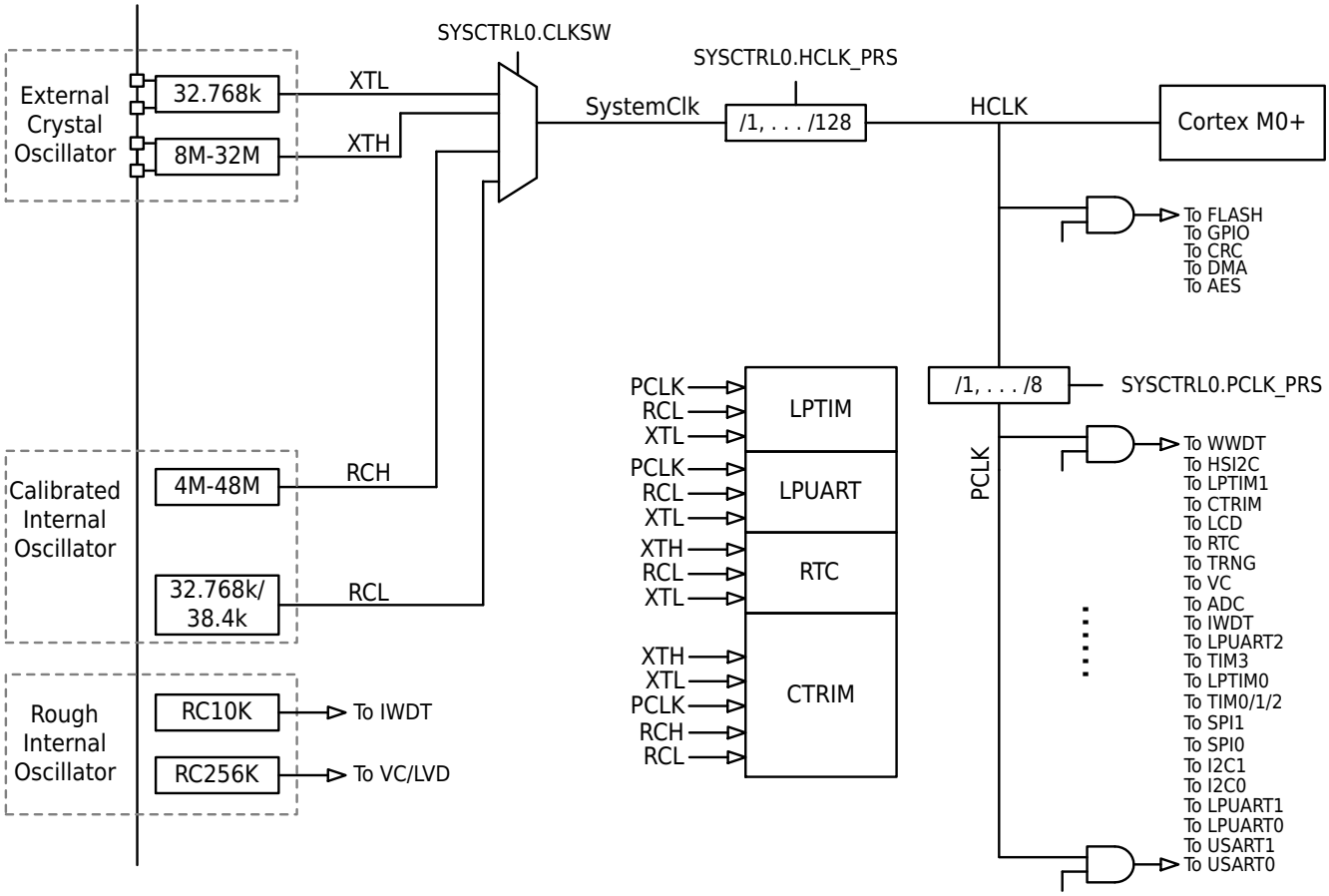



Figure 4-1 Clock Architecture Diagram

## 4.1.2 Clock Types

### 4.1.2.1 RCH

RCH is an internal high-speed clock oscillator with configurable and calibratable frequency, and it is also the default clock source after chip power-up or partial reset, with a default frequency of 4MHz. The clock frequency can be configured through the RCH_CR register. Four preset calibrated frequency configuration values—4MHz, 6MHz, 32MHz and 48MHz before leaving the factory can be directly loaded for use; it can also adjust the calibration value in this register to change to other frequencies or achieve higher-precision calibration according to actual needs.

When the system enters DeepSleep mode, this high-speed clock will be automatically turned off.

For instructions on changing the RCH clock source frequency, please refer to the chapter System Clock Switching.

The internal high-speed clock requires a short time from start up to stabilization. In order to respond to interrupts quickly in DeepSleep mode, it is recommended to switch the system clock source to RCH before entering DeepSleep mode.

### 4.1.2.2 RCL

RCL is an internal low-speed clock with configurable and calibratable frequency.

The clock frequency can be configured through RCL_CR register, two preset calibrated frequency configuration values before leaving the factory: 32.768kHz and 38.4kHz, which can be directly loaded for use; it can also adjust the calibration value in this register to change to other frequencies or achieve higher-precision calibration according to actual needs.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

When the system enters DeepSleep mode, this clock will not be automatically turned off, and the ultra-low power peripheral module can choose RCL as its clock.

#### 4.1.2.3 XTL

XTL external low-speed crystal oscillator clock requires an external 32.768kHz low-power crystal oscillator, which has ultra-high precision and ultra-low power. When the system enters DeepSleep mode, this clock will not shut down automatically. Peripheral modules operating in ultra-low power mode can select XTL as their clock.

XTL supports an external active clock as input, allowing the input clock signal directly from the PC14 pin without connecting a crystal. The method for inputting an active clock signal from PC14 is as follows:

- Configure the PC14 pin as a GPIO input;
- Configure the XTL stable time XTL_CR.STARTUP;
- Set SYSCTRL1.EXTL_EN to 1;
- Wait for the XTL stable flag to be set XTL_CR.STABLE.

**Notice**

The crystal, its matching capacitors, and the external clock source signal must comply with the requirements for the low-speed external clock XTL specified in the Electrical Specifications chapter of the Datasheet.

#### 4.1.2.4 XTH

The external high-speed crystal oscillator clock requires an external high-speed crystal with a frequency of 8-32MHz. When the system enters DeepSleep mode, this clock will shut down automatically.

XTH supports an external active clock as input, allowing the input of a 4-32MHz clock signal directly from the PD00 pin without connecting a crystal. The method for inputting an clock signal from PD00 is as follows:

- Configure the PD00 pin as a GPIO input;
- Configure the XTH stable time XTH_CR.STARTUP;
- Set SYSCTRL1.EXTH_EN to 1;
- Wait for the XTH stable flag to be set XTH_CR.STABLE.

**Notice**

The crystal, its matching capacitors, and the external clock source signal must comply with the requirements for the high-speed external clock XTH specified in the Electrical Specifications section of the Datasheet.

### 4.1.3 Clock Start Process

All of the above clock sources require a startup stabilization time. Taking XTH as example to illustrate the process of the clock startup and stabilization.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

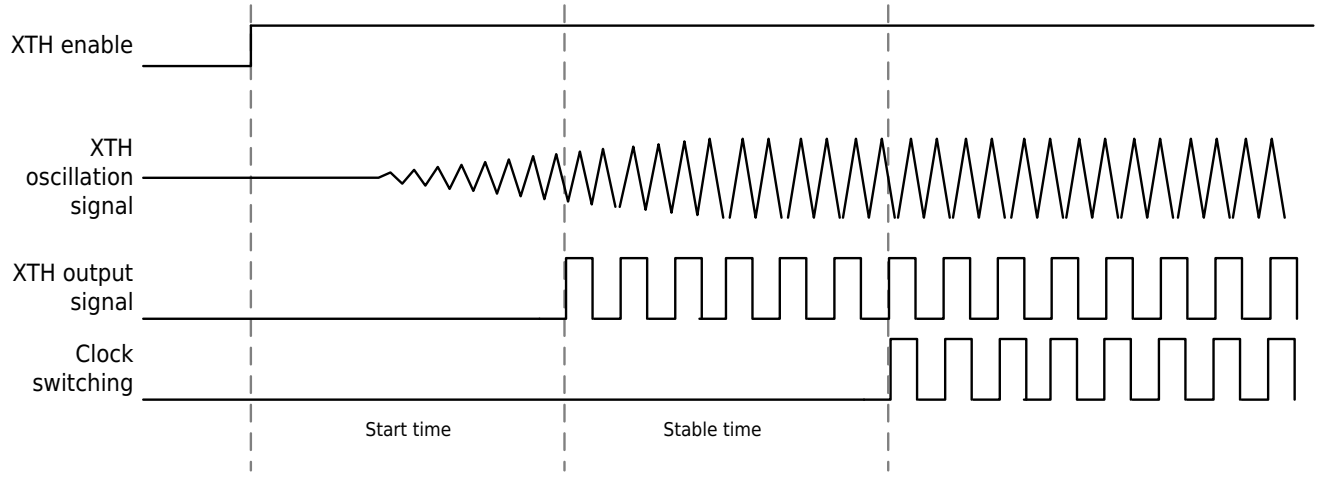



Figure 4-2 Crystal Oscillator Clock Startup Diagram

## 4.2 System Clock Switching

### 4.2.1 System Clock Switching Introduction

The clock source of the system clock can be switched among RCH, RCL, XTH, and XTL through SYSCTRL0.CLKSW. When switching, any two of the clock sources RCH, RCL, XTH, XTL can be switched between each other. The clock switching operation must be performed according to clock switching processes described below, otherwise abnormalities may occur.

The system clock source can be used as HCLK after system division, an appropriate frequency division value can be configured via SYSCTRL0.HCLK_PRS.

When switching the clock, FLASH_CR.WAIT must be configured accordingly based on the current and target HCLK frequencies.

- When HCLK≤24MHz, FLASH_CR.WAIT should be set to 0.
- When HCLK>24MHz, FLASH_CR.WAIT should be set to 1;

**Note**

For detailed instructions on setting FLASH_CR.WAIT, please refer to the chapter "FLASH Controller (FLASH)".

### 4.2.2 Switching Process

#### 4.2.2.1 Standard Clock Switching Process

**Procedure**

**Step1.** If the new clock source requires an external pin, set the pin to the appropriate mode.

**Note**

An analog pin is required when connecting to an external crystal oscillator; GPIO input is required and the external clock input is enabled when connecting to an external clock input.

**Step2.** Configure the oscillation parameters of the new clock source.

**Step3.** Enable the oscillator of the new clock source.

**Step4.** If the frequency of the new clock source is higher than that of the previous one, configure the FLASH_CR.WAIT before switching, based on the new clock source frequency, following the procedure described in the chapter FLASH Controller.

**Step5.** Wait for the new clock source to output a stable frequency.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

**Step6.** Configure SYSCTRL0.CLKSW to select the source of the system clock as the new clock source.

**Step7.** If the frequency of the new clock source is lower than that of the previous one, configure the FLASH_CR.WAIT after switching, based on the new clock source frequency, following the procedure described in the chapter FLASH Controller.

**Step8.** Turn off clock sources that are no longer in use.

#### 4.2.2.2 RCH Switching Process Between Different Oscillation Frequencies

**Context**

Choose one of the following methods according to the usage scenario:

**Procedure**

- Load according to the target frequency: directly load data from the factory preset target frequency calibration value storage address into the RCH_CR.
- Set directly according to actual needs: write the calibration value to the RCH_CR, adjust the target output frequency of the RCH.

### 4.2.3 Examples of Switching

#### 4.2.3.1 Example of Switching from Other Clocks to XTL

**Procedure**

**Step1.** Set PCADS.14 and PCADS.15 to 1, configure PC14/PC15 pins as analog ports.

**Step2.** Configure the parameters in the XTL_CR register according to the features of the crystal and the system.

**Step3.** Write 0x5A5A and 0xA5A5 to SYSCTRL2 register in sequence to enable register rewriting.

**Step4.** Set SYSCTRL0.XTL_EN to 1 to enable the crystal oscillator circuit.

**Step5.** Query and wait for the XTL_CR.STABLE flag to become 1, and the crystal oscillator outputs a stable clock.

**Step6.** Write 0x5A5A and 0xA5A5 to SYSCTRL2 register in sequence to enable register rewriting.

**Step7.** Set SYSCTRL0.CLKSW to 3, switch the system clock to XTL.

**Step8.** See the chapter FLASH Controller to set FLASH_CR.WAIT to 0.

**Step9.** Write 0x5A5A and 0xA5A5 to SYSCTRL2 register in sequence to enable register rewriting.

**Step10.** Turn off the clock source that is no longer used.

#### 4.2.3.2 Example of Switching from Other Clocks to XTH

**Procedure**

**Step1.** Set PDADS.0 and PDADS.1 to 1, configure PD00/PD01 pin as analog ports.

**Step2.** Configure XTH_CR[3:0] according to the features of the crystal oscillator.

**Step3.** Set XTH_CR.STARTUP to 3, then choose the appropriate crystal oscillator stable time.

**Step4.** Write 0x5A5A and 0xA5A5 to SYSCTRL2 register in sequence to enable register rewriting.

**Step5.** Set SYSCTRL0.XTH_EN to 1, then enable the crystal oscillator circuit.

**Step6.** According to the higher frequency of the current clock and XTH, configure FLASH_CR.WAIT according to Flash Controller.

**Step7.** Query and wait for the XTH_CR.STABLE flag to become 1, and the oscillator outputs a stable clock.

**Step8.** Write 0x5A5A and 0xA5A5 to SYSCTRL2 register in sequence to enable register rewriting.

**Step9.** Set SYSCTRL0.CLKSW to 1, switch the system clock to XTH.

**Step10.** According to the frequency of XTH, configure FLASH_CR.WAIT in terms of Flash Controller.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

**Step11.** Write 0x5A5A and 0xA5A5 to SYSCTRL2 register in sequence to enable register rewriting.

**Step12.** Turn off the clock source that is no longer used.

#### 4.2.3.3 Example of Switching from Other Clocks to RCL

**Procedure**

**Step1.** Configure RCL_CR.TRIM and RCL_CR.STARTUP.

**Step2.** Write 0x5A5A and 0xA5A5 to SYSCTRL2 register in sequence to enable register rewriting.

**Step3.** Set SYSCTRL0.RCL_EN to 1 to enable the RCL oscillation circuit.

**Step4.** Query and wait for RCL_CR.STABLE flag to become 1, and the RCL outputs a stable clock.

**Step5.** Write 0x5A5A and 0xA5A5 to SYSCTRL2 register in sequence to enable register rewriting.

**Step6.** Set SYSCTRL0.CLKSW to 2, switch the system clock to RCL.

**Step7.** See the chapter FLASH Controller to set FLASH_CR.WAIT to 0.

**Step8.** Write 0x5A5A and 0xA5A5 to SYSCTRL2 register in sequence to enable register rewriting.

**Step9.** Turn off the clock source that is no longer used.

#### 4.2.3.4 Example of Switching from Other Clocks to RCH

**Procedure**

**Step1.** According to the higher frequency between the current clock and the RCH clock to be configured, configure FLASH_CR.WAIT following the procedure described in chapter FLASH Controller.

**Step2.** Write the preset value to the RCH_CR register to configure the desired clock frequency.

**Step3.** Write 0x5A5A and 0xA5A5 to SYSCTRL2 register in sequence to enable register rewriting.

**Step4.** Set SYSCTRL0.RCH_EN to 1, enable RCH oscillation circuit.

**Step5.** Query and wait for RCH_CR.STABLE flag to become 1, RCH outputs a stable clock.

**Step6.** Write 0x5A5A and 0xA5A5 to SYSCTRL2 register in sequence to enable register rewriting.

**Step7.** Set SYSCTRL0.CLKSW to 0, switch the system clock to RCH.

**Step8.** According to the frequency of RCH, configure FLASH_CR.WAIT according to the procedures in chapter Flash Controller.

**Step9.** Write 0x5A5A and 0xA5A5 to SYSCTRL2 register in sequence to enable register rewriting.

**Step10.** Turn off the clock source that is no longer used.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 4.2.4 Clock Switching Diagram

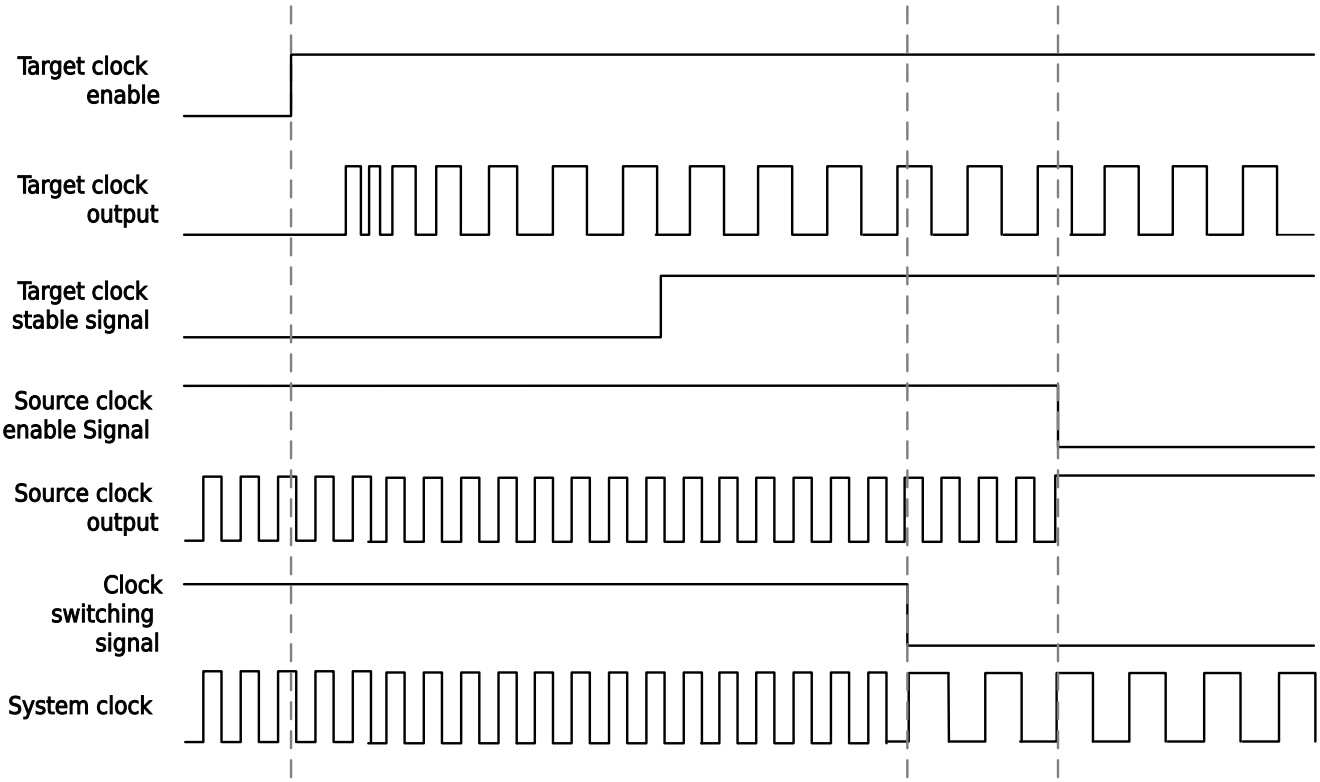



Figure 4-3 Clock Switching Diagram

## 4.3 Clock Calibration Module

This product integrates an internal clock calibration circuit CTRIM modules. For more information, see the relevant chapter descriptions.

## 4.4 Interrupt Wakeup Control

When the processor executes the WFI instruction to enter the Sleep/DeepSleep state, it will stop executing instructions. When an interrupt request (higher priority) occurs during Sleep/DeepSleep and needs to be serviced, the processor is woken up.

The behavior of the processor in the Sleep/DeepSleep state when receiving an interrupt request is shown in the following table:

| PRIMASK Status | WFI Behavior | Wakeup | ISR Execution |
|---|---|---|---|
| 0 | IRQ Priority > Current Class | Y | Y |
| 0 | IRQ Priority≤ Current Class | N | N |
| 1 | IRQ Priority > Current Class | Y | N |
| 1 | IRQ Priority≤ Current Class | N | N |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 4.4.1 Executing Interrupt Service after Waking from DeepSleep mode

#### Procedure

**Step1.** Enable the NVIC corresponding to the module that needs to wake up the processor.

**Step2.** Enable the interrupt corresponding to the module that needs to wake up the processor.

**Step3.** Set SCB->SCR.SLEEPDEEP to 1.

**Step4.** Execute WFI instruction enter the DeepSleep mode.

**Step5.** The system enters the DeepSleep mode and waits for the interrupt to wake up, and executes the interrupt service program after waking up.

#### Examples

```
SCB->SCR |= 0x00000004u;
while(1)
{
  __asm("WFI");
}
```

### 4.4.2 Interrupt Service Not Executed after Waking from DeepSleep mode

#### Procedure

**Step1.** Enable the NVIC corresponding to the module that needs to wake up the processor.

**Step2.** Enable the interrupt corresponding to the module that needs to wake up the processor.

**Step3.** Set PRIMASK to 1.

**Step4.** Set SCB->SCR.SLEEPDEEP to 1.

**Step5.** Execute WFI instruction enter the DeepSleep mode.

**Step6.** The system enters the DeepSleep mode and waits for the interrupt to wake up, and executes the next instruction after waking up;

**Step7.** Clear interrupt flag, clear interrupt pending status.

**Step8.** Execute user-defined operations.

#### Examples

```
__asm("CPSID I");//Set PRIMASK
SCB->SCR |= 0x00000004u;
while(1)
{
  __asm("WFI");
  ...//Clear the interrupt flag of the wake-up interrupt
  ...//Clear the interrupt pending status of the wake-up interrupt
  ...//Perform user-defined operation
}
```


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 4.4.3 Using Sleep-on-exit Feature

#### Context

Sleep-on-exit is ideal for interrupt-driven applications. When this feature is enabled, the processor enters Sleep/DeepSleep mode whenever exception handling is completed and returns to thread mode. With the sleep-on-exit feature, the processor can be in Sleep/DeepSleep mode as much as possible.

Cortex-M0+ take use of the sleep-on-exit feature to enter Sleep/DeepSleep. This situation is similar to the effect of executing WFI immediately after executing an abnormal exit. Attention: in order to avoid the need to push the stack when entering an exception next time, the processor will not perform the process of popping the stack.

#### Procedure

**Step1.** Enable the NVIC corresponding to the module that needs to wake up the processor.

**Step2.** Enable the interrupt corresponding to the module that needs to wake up the processor.

**Step3.** Set SCB->SCR.SLEEPDEEP to 1.

**Step4.** Set SCB->SCR.SLEEPONEXIT to 1.

**Step5.** Execute WFI instruction enter the DeepSleep mode.

**Step6.** The system enters the DeepSleep mode and waits for the interrupt to wake up, and executes the interrupt service subroutine after waking up.

**Step7.** Automatically enters DeepSleep mode when exiting interrupt service.

#### Examples

```
SCB->SCR |= 0x00000004u;
SCB->SCR |= 0x00000002u;
while(1)
{
  __asm("WFI");
}
```

## 4.5 Reset Control

### 4.5.1 Reset Controller Overview

This product has 7 reset signal sources, each of which can make the CPU run again. Most registers will be reset to reset values, and the program will start executing from the reset vector.

- Digital area power-on and power-off reset POR
- External Reset PAD, low level is reset signal
- WWDT reset
- IWDT reset
- LVD low voltage reset
- Cortex-M0+ SYSRESETREQ software reset
- Cortex-M0+ LOCKUP hardware reset

Each reset source is indicated by a corresponding reset flag. Reset flags are set by hardware and need to be cleared by user software. When the chip is reset, if RESET_FLAG.POR15V or RESET_FLAG.POR5V is 1, it is a **power-on reset**. When **power-on a reset**, the user program should clear the RESET_FLAG register to zero. Then, the reset source can be determined by the relevant bits of RESET_FLAG during the next reset.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

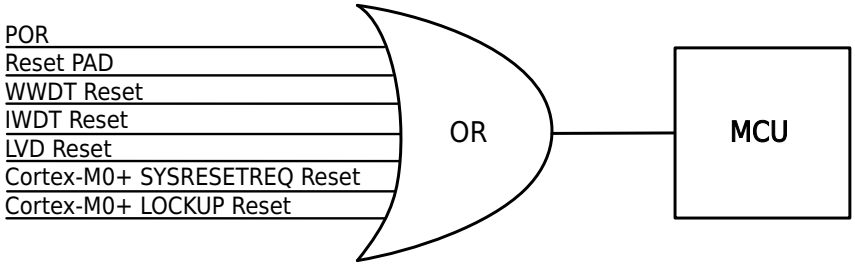



Figure 4-4 Reset Source Diagram

## 4.5.2 Reset Function

### 4.5.2.1 Power-on and Power-off Reset POR

This product has two power supply areas: VCC area and VCAP area. All analog modules and IO work in the VCC area; other modules work in the VCAP area.

- When the VCC area is powered on, when the VCC voltage is lower than the POR threshold voltage, POR5V signal is generated; when the VCC area is powered off, when the VCC voltage is lower than the PDR threshold voltage, POR5V signal is generated.
- When the VCAP area is powered on, when the VCAP voltage is lower than the POR threshold voltage, POR15V signal is generated; when the VCAP area is powered off, when the VCAP voltage is lower than the PDR threshold voltage, POR15V signal is generated.

Both the POR5V signal and the POR15V signal will reset the chip's registers to the initialization state.

### 4.5.2.2 External Reset Pin Reset

A system reset is generated when the external reset pin detects a low level. The reset pin has a built-in pull-up resistor and integrates a glitch filter circuit. The glitch filter circuit filters out glitches shorter than a certain time width. Therefore, the low-level signal applied to the reset pin must exceed this time width to ensure a reliable reset of the chip. For specific timing parameters, see the RESETB Pin Features section under the chapter "Electrical Specifications" in the Datasheet.

### 4.5.2.3 Watchdog Reset

For watchdog reset, refer to the relevant introductions in the chapters "Independent Watchdog (IWDT)" and "Window Watchdog (WWDT)".

### 4.5.2.4 LVD Low Voltage Reset

For LVD low voltage reset, refer to the relevant introduction in the "Low Voltage Detector (LVD)" chapter.

### 4.5.2.5 Cortex-M0+ SYSRESETREQ Reset

Cortex-M0+ software reset.

### 4.5.2.6 Cortex-M0+ LOCKUP Reset

When Cortex-M0+ encounters a serious exception, it will stop its own PC pointer at the current address and lock it, and reset the entire CORE area after a delay of several clock cycles. This reset function is enabled when SYSCTRL1.LOCKUP_EN is set to 1.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

## 4.6 Register Description

### 4.6.1 Register List

Base address: 0x40021800

Table 4-2 System Controller (SYSCTRL) Register Offset

| Offset | Register | Description | Width |
|---|---|---|---|
| 0x00 | SYSCTRL0 | System Control Register 0 | 32 |
| 0x04 | SYSCTRL1 | System Control Register 1 | 32 |
| 0x08 | SYSCTRL2 | System Control Register 2 | 32 |
| 0x0C | RCH_CR | RCH Control Register | 32 |
| 0x10 | XTH_CR | XTH Control Register | 32 |
| 0x14 | RCL_CR | RCL Control Register | 32 |
| 0x18 | XTL_CR | XTL Control Register | 32 |
| 0x1C | RESET_FLAG | Reset Flag Register | 32 |
| 0x20 | PERI_CLKEN0 | Peripheral Module Clock Control Register 0 | 32 |
| 0x24 | PERI_CLKEN1 | Peripheral Module Clock Control Register 1 | 32 |
| 0x28 | PERI_RESET0 | Peripheral Module Reset Control Register 0 | 32 |
| 0x2C | PERI_RESET1 | Peripheral Module Reset Control Register 1 | 32 |
| 0x38 | DEBUG_ACTIVE | Peripheral Debug Function Enable Register | 32 |

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 4.6.2 System Control Register 0 (SYSCTRL0)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00008001 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | WAKEUP_BYRCH | Reserved | | PCLK_PRS | | HCLK_PRS | | | CLKSW | | | Reserved | | XTL_EN | RCL_EN | XTH_EN | RCH_EN |
| Access | | | | | | | | | | | | | | | | | RW | | | RW | | RW | | | RW | | | | RW | RW | RW | RW |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:16* | *Reserved* | - | - |
| 15 | WAKE-UP_BYRCH | System clock source control bit after wake-up<br>● 0b0: After waking up from DeepSleep mode, the system clock source remains unchanged.<br>● 0b1: After waking up from DeepSleep mode, the system clock source switches to RCH, and the original clock source remains enabled. | RW |
| *14:13* | *Reserved* | - | - |
| 12:11 | PCLK_PRS | PCLK clock source selection<br>● 0b00: HCLK<br>● 0b01: HCLK/2<br>● 0b10: HCLK/4<br>● 0b11: HCLK/8 | RW |
| 10:8 | HCLK_PRS | HCLK system frequency division selection<br>● 0b000: SystemClk<br>● 0b001: SystemClk/2<br>● 0b010: SystemClk/4<br>● 0b011: SystemClk/8<br>● 0b100: SystemClk/16<br>● 0b101: SystemClk/32<br>● 0b110: SystemClk/64<br>● 0b111: SystemClk/128 | RW |
| 7:5 | CLKSW | SystemClk clock source selection<br>● 0b000: RCH<br>● 0b001: XTH<br>● 0b010: RCL<br>● 0b011: XTL<br>● Others: Setting prohibited | RW |
| *4* | *Reserved* | - | - |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 3 | XTL_EN | External low-speed crystal oscillator XTL enable control<br>● 0b0: Disable<br>● 0b1: Enable<br>**Note**<br>PC14 and PC15 need to be set as analog ports. | RW |
| 2 | RCL_EN | Internal low-speed clock RCL enable control<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 1 | XTH_EN | External high-speed crystal oscillator XTH enable control<br>● 0b0: Disable<br>● 0b1: Enable<br>**Note**<br>PD00, PD01 need to be set as analog ports.<br>When the system enters DeepSleep, this high-speed clock will be automatically turned off. | RW |
| 0 | RCH_EN | Internal high-speed clockRCH enable control<br>● 0b0: Disable<br>● 0b1: Enable<br>**Note**<br>When the system enters DeepSleep, this high-speed clock will be automatically turned off. | RW |

**Note**

Every time you rewrite the value of SYSCTRL0 or SYSCTRL1, you need to write 0x5A5A and 0xA5A5 to SYSCTRL2 in sequence. Such steps can effectively prevent misoperation of SYSCTRL0 or SYSCTRL1 register.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 4.6.3 System Control Register 1 (SYSCTRL1)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x04 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000008 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | SWD_USE_IO | SPI_PCTRL | LOCKUP_EN | Reserved | CLOCK_FAULT_EN | XTL_ALWAYS_ON | EXTL_EN | EXTH_EN | Reserved |
| Access | | | | | | | | | | | | | | | | | | | | | | | | RW | RW | RW | | RW | RW | RW | RW | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:9* | *Reserved* | - | - |
| 8 | SWD_USE_IO | SWD port function configuration<br>● 0b0: SWD port<br>● 0b1: GPIO port | RW |
| 7 | SPI_PCTRL | SPI port control selection<br>● 0b0: SPI control<br>● 0b1: GPIO control | RW |
| 6 | LOCKUP_EN | Cortex-M0+ LockUp enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| *5* | *Reserved* | - | - |
| 4 | CLOCK_FAULT_EN | Clock error detection enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 3 | XTL_ALWAYS_ON | XTL enable permission control<br>● 0b0: SYSCTRL0.XTL_EN can be set and cleared.<br>● 0b1: SYSCTRL0.XTL_EN can only be set. | RW |
| 2 | EXTL_EN | External XTL clock source control<br>● 0b0: XTL is generated by a crystal.<br>● 0b1: XTL clock source is input from PC14, when using this clock as the system clock, EXTL_EN must not be disabled (the external clock source must comply with the requirements for the low-speed external clock XTL specified in the Electrical Specifications chapter of the Datasheet).<br>**Note**<br>When using PC14 input clock, it is necessary to set SYSCTRL0.XTL_EN to 0. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 1 | EXTH_EN | External XTH clock source control<br>● 0b0: XTH is generated by a crystal.<br>● 0b1: XTH clock source is input from PD00<br>**Note**<br>When using PD00 input clock:<br>1. SYSCTRL0.XTH_EN must be set to 0.<br>2. SYSCTRL0.WAKEUP_BYRCH must be set to 1; otherwise, abnormal current consumption may occur in low-power mode. | RW |
| *0* | *Reserved* | - | - |

**Note**

Every time you rewrite the value of SYSCTRL0 or SYSCTRL1, you need to write 0x5A5A and 0xA5A5 to SYSCTRL2 in sequence. Such steps can effectively prevent misoperation of SYSCTRL0 or SYSCTRL1 registers.

### 4.6.4 System Control Register 2 (SYSCTRL2)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x08 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | SYSCTRL2 | | | | | | | | | |
| Access | | | | | | | | | | | | | | | | | | | | | | | WO | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:16* | *Reserved* | - | - |
| 15:0 | SYSCTRL2 | SYSCTL0 register and SYSCTRL1 protection control register<br>Write 0x5A5A first to SYSCTRL2, and then write 0xA5A5 to start the write operation to SYSCTRL0 and SYSCTRL1 register. As long as the write operation is performed to SYSCTRL0 or SYSCTRL1 register, this protection bit automatically recovery to the protection state. Subsequent write operations to SYSCTRL0 or SYSCTRL1 require repeating the previous steps. | WO |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 4.6.5 RCH Control Register (RCH_CR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x0C | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x0X00XXXX | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | STABLE | Reserved | | | | | | | | | | | | | | | FSEL | TRIMC | | | | | | | TRIMF | | | | | | Reserved | |
| Access | RO | | | | | | | | | | | | | | | | RW | RW | | | | | | | RW | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 31 | STABLE | RCH clock stable flag bit<br>● 0b0: RCH is not stable and cannot be used by internal circuits.<br>● 0b1: RCH is stable and can be used by internal circuits.<br>**Note**<br>The reset value of this bit is 0 upon power-on. During actual operation, after RCH stabilization, read always returns 1. | RO |
| *30:18* | *Reserved* | - | - |
| 17:16 | FSEL | Operating frequency selection<br>This bit does not require independent configuration. For parameter selection, refer to the instructions.<br>● 0b00: 4MHz<br>● 0b01: 6MHz<br>● 0b10: 32MHz<br>● 0b11: 48MHz | RW |
| 15:8 | TRIMC | Coarse adjustment of clock frequency<br>Parameter selection details are provided in the instructions. | RW |
| 7:2 | TRIMF | Clock frequency fine tuning<br>Parameter selection details are provided in the instructions. | RW |
| *1:0* | *Reserved* | - | - |

**Note**

RCH clock frequency adjustment: changing the value of this register allows you to adjust the output frequency of RCH.

The Flash already contains several sets of calibration values and configuration parameters. By reading all the data from the specified addresses in Flash and writing them into the entire RCH_CR register, you can obtain an accurate frequency.

- 48MHz calibration value address: 0x00101BB0 - 0x00101BB3
- 32MHz calibration value address: 0x00101BB4 - 0x00101BB7
- 6MHz calibration value address: 0x00101BB8 - 0x00101BBB
- 4MHz calibration value address: 0x00101BBC - 0x00101BBF


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 4.6.6 XTH Control Register (XTH_CR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x10 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000016 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | FAULT_STATE | DETEN | Reserved | DETIE | STABLE | STARTUP | | XTH_FSEL | | DRIVER | |
| Access | | | | | | | | | | | | | | | | | | | | | | RW | RW | | RW | RO | RW | | RW | | RW | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:11* | *Reserved* | - | - |
| 10 | FAULT_STATE | XTH work flag<br>Writing 0 clears the flag, writing 1 has no effect<br>Read:<br>● 0b0: Works fine<br>● 0b1: Fails to work | RW |
| 9 | DETEN | Oscillation stop detection enable<br>● 0b0: Disable oscillator stop detection<br>● 0b1: Enable oscillator stop detection<br>**Note**<br>1. To enable oscillator stop detection, SYSCTRL0.XTH_EN must be set to 1, and XTH_CR.STABLE must read as 1.<br>2. When entering the DeepSleep interrupt, this function must be disabled. Otherwise, an interrupt request will be generated after wake-up. | RW |
| *8* | *Reserved* | - | - |
| 7 | DETIE | Oscillation stop detection interrupt enable control<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 6 | STABLE | External high-speed crystal oscillator stable flag bit<br>● 0b0: XTH is not stable and cannot be used by internal circuits.<br>● 0b1: XTH is stable and can be used by internal circuits. | RO |
| 5:4 | STARTUP | External high-speed crystal oscillator stable time selection<br>● 0b00: 8192 cycles<br>● 0b01: 32768 cycles<br>● 0b10: 131072 cycles<br>● 0b11: 262144 cycles<br>**Notice**<br>If the XTH stable time is insufficient, the system will not work stably when performing clock switching or waking up from DeepSleep. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 3:2 | XTH_FSEL | External crystal oscillator operating frequency selection<br>● 0b00: 8M<br>● 0b01: 8M-16M<br>● 0b10: 16M-24M<br>● 0b11: 24M-32M | RW |
| 1:0 | DRIVER | External crystal oscillator driving capability selection<br>● 0b00: Weakest driving capability<br>● 0b01: Weaker driving capability<br>● 0b10: Stronger driving capability (Recommended value)<br>● 0b11: Strongest driving capability<br>**Note**<br>It is necessary to select an appropriate driving capability according to the characteristics of the crystal oscillator, the load capacitance and the parasitic parameters of the circuit board. The greater the driving capability, the greater the power consumption; the weaker the driving capability, the smaller the power consumption. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 4.6.7 RCL Control Register (RCL_CR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x14 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x0X000XXX (The TRIM value after power-on reset is the value stored at the 32.768kHz calibration value address) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | STABLE | Reserved | MSEL | TRIM | | | | | | | |
| Access | | | | | | | | | | | | | | | | | | | | RO | | RW | RW | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:13* | *Reserved* | - | - |
| 12 | STABLE | Internal low-speed clock RCL stable flag<br>● 0b0: RCL is not stable and cannot be used by internal circuits.<br>● 0b1: RCL is stable and can be used by internal circuits. | RO |
| *11* | *Reserved* | - | - |
| 10 | MSEL | Pattern selection<br>● 0b0: High performance mode (HPM)<br>● 0b1: Low power mode (LPM) | RW |
| 9:0 | TRIM | Internal low-speed clock frequency adjustment<br>4 sets of frequency calibration values are stored in Flash. The precise frequency can be obtained by reading out the calibration value in Flash and writing it into RCL_CR.<br>● 38.4kHz calibration value address:<br>▶ High-Performance Mode: 0x00101BAC-0x00101BAF<br>▶ Low-Power Mode: 0x00101BC8-0x00101BCB<br>● 32.768kHz calibration value address:<br>▶ High-Performance Mode: 0x00101BA8-0x00101BAB<br>▶ Low-Power Mode: 0x00101BC4-0x00101BC7 | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 4.6.8 XTL Control Register (XTL_CR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x18 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000029 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | FAULT_STATE | DETEN | DETTIME | DETIE | STABLE | STARTUP | | AMP_SEL | | DRIVER | |
| Access | | | | | | | | | | | | | | | | | | | | | | RW | RW | RW | RW | RO | RW | | RW | | RW | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:11* | *Reserved* | - | - |
| 10 | FAULT_STATE | XTL work state flag<br>Writing 0 clears the flag, writing 1 has no effect<br>Read:<br>● 0b0: Works fine<br>● 0b1: Fails to work | RW |
| 9 | DETEN | Oscillation stop detection enable<br>● 0b0: Disable oscillator stop detection<br>● 0b1: Enable oscillator stop detection<br>**Note**<br>To enable oscillator stop detection, SYSCTRL0.XTH_EN must be set to 1, and XTH_CR.STABLE must read as 1. | RW |
| 8 | DETTIME | Crystal failure detection interval selection<br>● 0b0: Gear 1, 2ms detection interval<br>● 0b1: Gear 2, 8ms detection interval<br>**Note**<br>This bit must not be modified when SYSCTRL0.XTL_EN is set to 1. | RW |
| 7 | DETIE | Oscillation stop detection interrupt enable control<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 6 | STABLE | External low-speed crystal oscillator stable flag bit<br>● 0b0: XTL is not stable and cannot be used by internal circuits.<br>● 0b1: XTL is stable and can be used by internal circuits. | RO |
| 5:4 | STARTUP | External low-speed crystal oscillator stable time selection<br>● 0b00: 4096 cycles<br>● 0b01: 8192 cycles<br>● 0b10: 16384 cycles (Recommended value)<br>● 0b11: 32768 cycles | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 3:2 | AMP_SEL | XTL crystal oscillation amplitude adjustment<br>● 0b00: minimum amplitude<br>● 0b01: normal amplitude<br>● 0b10: larger amplitude (Recommended value)<br>● 0b11: maximum amplitude | RW |
| 1:0 | DRIVER | XTL crystal oscillator driving capability selection<br>● 0b00: Weakest driving capability<br>● 0b01: Weaker driving capability (Recommended value)<br>● 0b10: Stronger driving capability<br>● 0b11: Strongest driving capability<br>**Note**<br>It is necessary to select an appropriate driving capability according to the characteristics of the crystal oscillator, the load capacitance and the parasitic parameters of the circuit board. The greater the driving capability, the greater the power consumption; the weaker the driving capability, the smaller the power consumption. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 4.6.9 Reset Flag Register (RESET_FLAG)

**Note**

The reset value is 0x03 at power-on reset. After the program determines the reset source, it should clear the register so that it can determine the reset source the next time a reset occurs.

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x1C | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000003 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | WWDT | RSTB | SYSREQ | LOCKUP | Reserved | IWDT | LVD | POR15V | POR5V |
| Access | | | | | | | | | | | | | | | | | | | | | | | | RW | RW | RW | RW | | RW | RW | RW | RW |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:9* | *Reserved* | - | - |
| 8 | WWDT | WWDT reset flag<br>Write 0 to clear flag, write 1 has no effect<br>Read:<br>● 0b0: No reset occurred<br>● 0b1: Reset occurred | RW |
| 7 | RSTB | RESETB port reset flag<br>Write 0 to clear flag, write 1 has no effect<br>Read:<br>● 0b0: No reset occurred<br>● 0b1: Reset occurred | RW |
| 6 | SYSREQ | Cotrex-M0+ CPU software reset flag<br>Write 0 to clear flag, write 1 has no effect<br>Read:<br>● 0b0: No reset occurred<br>● 0b1: Reset occurred | RW |
| 5 | LOCKUP | Cotrex-M0+ CPU Lockup reset flag<br>Write 0 to clear flag, write 1 has no effect<br>Read:<br>● 0b0: No reset occurred<br>● 0b1: Reset occurred | RW |
| *4* | *Reserved* | - | - |
| 3 | IWDT | IWDT reset flag<br>Write 0 to clear flag, write 1 has no effect<br>Read:<br>● 0b0: No reset occurred<br>● 0b1: Reset occurred | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 2 | LVD | LVD reset flag<br>Write 0 to clear flag, write 1 has no effect<br>Read:<br>● 0b0: No reset occurred<br>● 0b1: Reset occurred | RW |
| 1 | POR15V | VCAP domain reset flag<br>Write 0 to clear flag, write 1 has no effect<br>Read:<br>● 0b0: No reset occurred<br>● 0b1: Reset occurred | RW |
| 0 | POR5V | VCC power domain reset flag<br>Write 0 to clear flag, write 1 has no effect<br>Read:<br>● 0b0: No reset occurred<br>● 0b1: Reset occurred | RW |

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 4.6.10 Peripheral Module Clock Control Register 0 (PERI_CLKEN0)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x20 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x80800000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | FLASH | HSI2C | DMA | GPIO | AES | CRC | Reserved | | | LCD | Reserved | RTC | Reserved | TRNG | VC | ADC | IWDT | Reserved | | | TIM3 | Reserved | LPTIM0 | TIM012 | SPI1 | SPI0 | I2C1 | I2C0 | LPUART1 | LPUART0 | USART1 | USART0 |
| Access | RW | RW | RW | RW | RW | RW | | | | RW | | RW | | RW | RW | RW | RW | | | | RW | | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 31 | FLASH | FLASH controller module clock enable<br>After closing, the FLASH configuration registers cannot be written, and the programs in the FLASH can still run.<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 30 | HSI2C | HSI2C module clock enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 29 | DMA | DMAC module clock enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 28 | GPIO | GPIO module clock enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 27 | AES | AES module clock enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 26 | CRC | CRC module clock enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| *25:23* | *Reserved* | - | - |
| 22 | LCD | LCD module clock enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| *21* | *Reserved* | - | - |
| 20 | RTC | RTC module clock enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| *19* | *Reserved* | - | - |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 18 | TRNG | TRNG module clock enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 17 | VC | VC, LVD module clock enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 16 | ADC | ADC, BGR module clock enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 15 | IWDT | IWDT module clock enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| *14:12* | *Reserved* | - | - |
| 11 | TIM3 | TIM3 module clock enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| *10* | *Reserved* | - | - |
| 9 | LPTIM0 | LPTIM0 module clock enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 8 | TIM012 | TIM0/1/2 module clock enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 7 | SPI1 | SPI1 module clock enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 6 | SPI0 | SPI0 module clock enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 5 | I2C1 | I2C1 module clock enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 4 | I2C0 | I2C0 module clock enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 3 | LPUART1 | LPUART1 module clock enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 2 | LPUART0 | LPUART0 module clock enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 1 | USART1 | USART1 module clock enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 0 | USART0 | USART0 module clock enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |

**Note**

The above peripherals must have their corresponding control bits set to enable before use.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 4.6.11 Peripheral Module Clock Control Register 1 (PERI_CLKEN1)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x24 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | WWDT | Reserved | LPUART2 | Reserved | | | LPTIM1 | Reserved | CTRIM | Reserved | |
| Access | | | | | | | | | | | | | | | | | | | | | | RW | | RW | | | | RW | | RW | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:11* | *Reserved* | - | - |
| 10 | WWDT | WWDT module clock enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| *9* | *Reserved* | - | - |
| 8 | LPUART2 | LPUART2 module clock enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| *7:5* | *Reserved* | - | - |
| 4 | LPTIM1 | LPTIM1 module clock enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| *3* | *Reserved* | - | - |
| 2 | CTRIM | CTRIM module clock enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| *1:0* | *Reserved* | - | - |

**Note**

The above peripherals must have their corresponding control bits set to enable before use.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 4.6.12 Peripheral Module Reset Control Register 0 (PERI_RESET0)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x28 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x7C570BFF | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | HSI2C | DMA | GPIO | AES | CRC | Reserved | | LCD | Reserved | | RTC | Reserved | | TRNG | VC | ADC | Reserved | | | TIM3 | Reserved | | LPTIM0 | TIM012 | SPI1 | SPI0 | I2C1 | I2C0 | LPUART1 | LPUART0 | USART1 | USART0 |
| Access | | RW | RW | RW | RW | RW | | | RW | | | RW | | | RW | RW | RW | | | | RW | | | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31* | *Reserved* | - | - |
| 30 | HSI2C | HSI2C block reset enable<br>● 0b0: In reset state<br>● 0b1: Normal working status | RW |
| 29 | DMA | DMAC module reset enable<br>● 0b0: In reset state<br>● 0b1: Normal working status | RW |
| 28 | GPIO | GPIO module reset enable<br>● 0b0: In reset state<br>● 0b1: Normal working status | RW |
| 27 | AES | AES module reset enable<br>● 0b0: In reset state<br>● 0b1: Normal working status | RW |
| 26 | CRC | CRC module reset enable<br>● 0b0: In reset state<br>● 0b1: Normal working status | RW |
| *25:23* | *Reserved* | - | - |
| 22 | LCD | LCD module reset enable<br>● 0b0: In reset state<br>● 0b1: Normal working status | RW |
| *21* | *Reserved* | - | - |
| 20 | RTC | RTC module reset enable<br>● 0b0: In reset state<br>● 0b1: Normal working status | RW |
| *19* | *Reserved* | - | - |
| 18 | TRNG | TRNG module reset enable<br>● 0b0: In reset state<br>● 0b1: Normal working status | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 17 | VC | VC module reset enable<br>● 0b0: In reset state<br>● 0b1: Normal working status | RW |
| 16 | ADC | ADC, BGR module reset enable<br>● 0b0: In reset state<br>● 0b1: Normal working status | RW |
| *15: 12* | *Reserved* | - | - |
| 11 | TIM3 | TIM3 module reset enable<br>● 0b0: In reset state<br>● 0b1: Normal working status | RW |
| *10* | *Reserved* | - | - |
| 9 | LPTIM0 | LPTIM0 module reset enable<br>● 0b0: In reset state<br>● 0b1: Normal working status | RW |
| 8 | TIM012 | TIM0/1/2 module reset enable<br>● 0b0: In reset state<br>● 0b1: Normal working status | RW |
| 7 | SPI1 | SPI1 module reset enable<br>● 0b0: In reset state<br>● 0b1: Normal working status | RW |
| 6 | SPI0 | SPI0 module reset enable<br>● 0b0: In reset state<br>● 0b1: Normal working status | RW |
| 5 | I2C1 | I2C1 module reset enable<br>● 0b0: In reset state<br>● 0b1: Normal working status | RW |
| 4 | I2C0 | I2C0 module reset enable<br>● 0b0: In reset state<br>● 0b1: Normal working status | RW |
| 3 | LPUART1 | LPUART1 module reset enable<br>● 0b0: In reset state<br>● 0b1: Normal working status | RW |
| 2 | LPUART0 | LPUART0 module reset enable<br>● 0b0: In reset state<br>● 0b1: Normal working status | RW |
| 1 | USART1 | USART1 module reset enable<br>● 0b0: In reset state<br>● 0b1: Normal working status | RW |
| 0 | USART0 | USART0 module reset enable<br>● 0b0: In reset state<br>● 0b1: Normal working status | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 4.6.13 Peripheral Module Reset Control Register 1 (PERI_RESET1)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x2C | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000514 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | WWDT | Reserved | LPUART2 | Reserved | | | LPTIM1 | Reserved | CTRIM | Reserved | |
| Access | | | | | | | | | | | | | | | | | | | | | | RW | | RW | | | | RW | | RW | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:11* | *Reserved* | - | - |
| 10 | WWDT | WWDT module reset enable<br>● 0b0: In reset state<br>● 0b1: Normal working status | RW |
| *9* | *Reserved* | - | - |
| 8 | LPUART2 | LPUART2 module reset enable<br>● 0b0: In reset state<br>● 0b1: Normal working status | RW |
| *7:5* | *Reserved* | - | - |
| 4 | LPTIM1 | LPTIM1 module reset enable<br>● 0b0: In reset state<br>● 0b1: Normal working status | RW |
| *3* | *Reserved* | - | - |
| 2 | CTRIM | CTRIM module reset enable<br>● 0b0: In reset state<br>● 0b1: Normal working status | RW |
| *1:0* | *Reserved* | - | - |

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 4.6.14 Peripheral Debug Function Enable Register (DEBUG_ACTIVE)

**Note**

This register setting is effective only in SWD debug mode.

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x38 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | WWDT | LPTIM1 | TIM3 | Reserved | RTC | IWDT | Reserved | | | | LPTIM0 | TIM2 | TIM1 | TIM0 |
| Access | | | | | | | | | | | | | | | | | | | RW | RW | RW | | RW | RW | | | | | RW | RW | RW | RW |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:14* | *Reserved* | - | - |
| 13 | WWDT | WWDT counting function configuration in SWD debug interface<br>● 0b0: Normal counting<br>● 0b1: Stop counting | RW |
| 12 | LPTIM1 | Configuration of LpTimer1 counting function under SWD debug interface<br>● 0b0: Normal counting<br>● 0b1: Stop counting | RW |
| 11 | TIM3 | Configuration of counting function under SWD debug interface<br>● 0b0: Normal counting<br>● 0b1: Stop counting | RW |
| *10* | *Reserved* | - | - |
| 9 | RTC | Configuration of RTC counting function under SWD debug interface<br>● 0b0: Normal counting<br>● 0b1: Stop counting | RW |
| 8 | IWDT | Configuration of IWDT counting function under SWD debug interface<br>● 0b0: Normal counting<br>● 0b1: Stop counting | RW |
| *7:4* | *Reserved* | - | - |
| 3 | LPTIM0 | Configuration of LPTIM0 counting function under SWD debug interface<br>● 0b0: Normal counting<br>● 0b1: Stop counting | RW |
| 2 | TIM2 | Configuration of Timer2 counting function under SWD debug interface<br>● 0b0: Normal counting<br>● 0b1: Stop counting | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 1 | TIM1 | Configuration of Timer1 counting function under SWD debug interface<br>● 0b0: Normal counting<br>● 0b1: Stop counting | RW |
| 0 | TIM0 | Configuration of Timer0 counting function under SWD debug interface<br>● 0b0: Normal counting<br>● 0b1: Stop counting | RW |

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

# 5 NVIC

## 5.1 Introduction

The Cortex-M0+ processor is internally equipped with a Nested Vector Interrupt Controller (NVIC), which supports up to 32 Interrupt Request (IRQ) inputs and 1 Non Maskable Interrupt (NMI) input (not used in this system). Meanwhile, the processor supports multiple internal exceptions.

Each exception source has a separate exception number, each exception type is assigned a corresponding priority. Some exception priorities are fixed, while others are programmable, as detailed in the table below.

Table 5-1 Cortex-M0+ Processor Interrupt List

| Exception Number | Exception Type | Priority | Description |
|---|---|---|---|
| 1 | Reset | -3 (highest) | Reset |
| 2 | NMI | -2 | Non-maskable interrupt (not used in this system) |
| 3 | Hardware error | -1 | Error handling exception |
| 4-10 | Reserved | - | - |
| 11 | SVC | Programmable | Invoke the hypervisor through the SVC command |
| 12-13 | Reserved | - | - |
| 14 | PendSV | Programmable | Pendable requests for system services |
| 15 | SysTick | Programmable | SysTick timer |
| 16 | Interrupt #0 | Programmable | External interrupt #0 |
| 17 | Interrupt #1 | Programmable | External interrupt #1 |
| ... | ... | ... | ... |
| 47 | Interrupt #31 | Programmable | External interrupt #31 |

This chapter only introduces the processor's 32 external interrupt requests (interrupt #0-interrupt #31). For internal exceptions, see the relevant core documentation and reference guides. In addition, this chapter only discusses the interrupt handling mechanism of the NVIC in the processor core, and does not address the interrupt generation mechanisms of peripheral modules.

## 5.2 Functional Description

### 5.2.1 Interrupt Priority

Each external interrupt corresponds to a priority register, each priority is 2 bits wide, and uses the highest two bits of the interrupt priority register, and each register occupies 1 byte (8 bits). Under this setting, the available priorities are 0x00 (highest), 0x40, 0x80 and 0xC0 (lowest).

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|
| Used | | Not used, read as 0 | | | | | |

When multiple interrupts occur during processor execution:


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

- Preemption occurs if the processor is servicing an interrupt and a higher-priority interrupt is triggered. The running interrupt processing will be pended and the new interrupt will be executed instead. This process is usually called interrupt nesting. After the new interrupt is executed, the previous interrupt processing will continue and return to the program thread after it finishes.
- If the priority of the currently executing interrupt is higher than or equal to that of the new interrupt, the new interrupt will wait and enter the pending state. Once the current interrupt handler completes execution, the processor resumes by servicing the pending interrupt.
- If two interrupts occur at the same time and they have the same priority, the interrupt with the lower interrupt number will be serviced first. For example, if interrupt #0 and interrupt #1 are both enabled, have the same priority, and are triggered at the same time, interrupt #0 will be serviced first.

### 5.2.2 Interrupt Digraph

When the Cortex-M0+ processor is about to handle an interrupt service request, it must first determine the starting address of the exception handler. The required information is called the vector table, as illustrated in the figure below. The vector table is stored at the beginning of the memory space and contains the exception (interrupt) vectors of the exceptions (interrupts) available in the system, as well as the initial value of the main stack pointer (MSP).


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Memory address | | Exception number |
|---|---|---|
| 0x000000BC | Interrupt vector #31 | 47 |
| ... | ... | ... |
| 0x0000004C | Interrupt vector #3 | 19 |
| 0x00000048 | Interrupt vector #2 | 18 |
| 0x00000044 | Interrupt vector #1 | 17 |
| 0x00000040 | Interrupt vector #0 | 16 |
| 0x0000003C | SysTick vector | 15 |
| 0x00000038 | PendSV vector | 14 |
| 0x00000034 | Reserved | 13 |
| 0x00000030 | Reserved | 12 |
| 0x0000002C | SVC vector | 11 |
| 0x00000028 | Reserved | 10 |
| 0x00000024 | Reserved | 9 |
| 0x00000020 | Reserved | 8 |
| 0x0000001C | Reserved | 7 |
| 0x00000018 | Reserved | 6 |
| 0x00000014 | Reserved | 5 |
| 0x00000010 | Reserved | 4 |
| 0x0000000C | Hardware Fault Exception | 3 |
| 0x00000008 | NMI vector | 2 |
| 0x00000004 | Reset vector | 1 |
| 0x00000000 | Initial Value of MSP | 0 |

**Figure 5-1 Interrupt Vector Table**

The storage order of the interrupt vector is consistent with the interrupt number. Since each vector is 1 word (4 bytes), the address of the interrupt vector is the interrupt number multiplied by 4, and each interrupt vector is the starting address of the interrupt processing.

### 5.2.3 Interrupt Input and Pending Behavior

In the NVIC module of the Cortex-M0+ processor, each interrupt input corresponds to a pending state register, and each register has only 1 bit, which is used to save the interrupt request, regardless of whether the request has been confirmed. When the processor starts servicing the interrupt, the hardware will automatically clear the pending status bit.

The peripherals of this system use level-triggered interrupt output. When an interrupt event occurs, the interrupt signal will be confirmed because the peripheral is connected to the NVIC. This signal remains high until the processor executes the interrupt service and clears the peripheral's interrupt signal. Inside the NVIC, when an interrupt is detected, the pending state of the interrupt will be set, and when the


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

processor receives the interrupt and starts executing the interrupt service routine, the pending state will be cleared. The handling process is illustrated in the following figure:

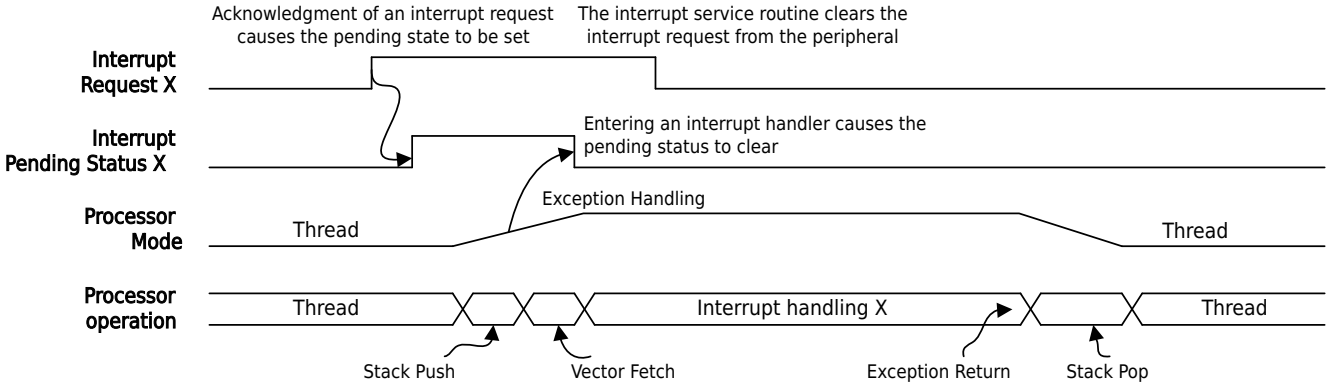



**Figure 5-2 Interrupt Active and Pending States**

If the interrupt request is not executed immediately and is cleared by software before being acknowledged, the processor will ignore the request and does not perform interrupt processing. When configuring a peripheral, the pending interrupt state can be cleared by writing to the NVIC_ICPR register, thereby removing any interrupt requests generated by the peripheral prior to its configuration.

If the peripheral is still holding an interrupt request when software clears the pending state, the pending state is also generated immediately. The handling process is illustrated in the following figure:

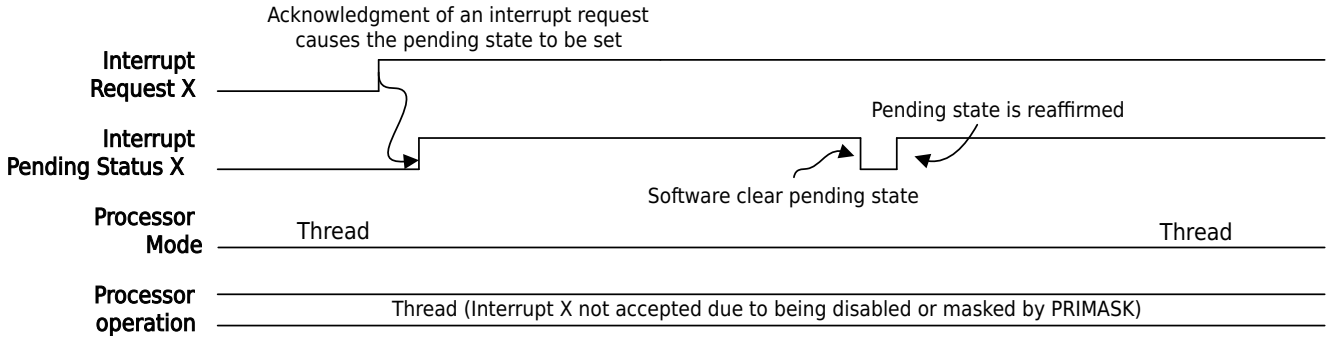



**Figure 5-3 Interrupt Pending State is Cleared and Then Reasserted**

If the interrupt request generated by the peripheral is not cleared when the exception is handled, the pending state will be activated again after the exception returns, so that the interrupt service routine will be executed again. The handling process is illustrated in the following figure:

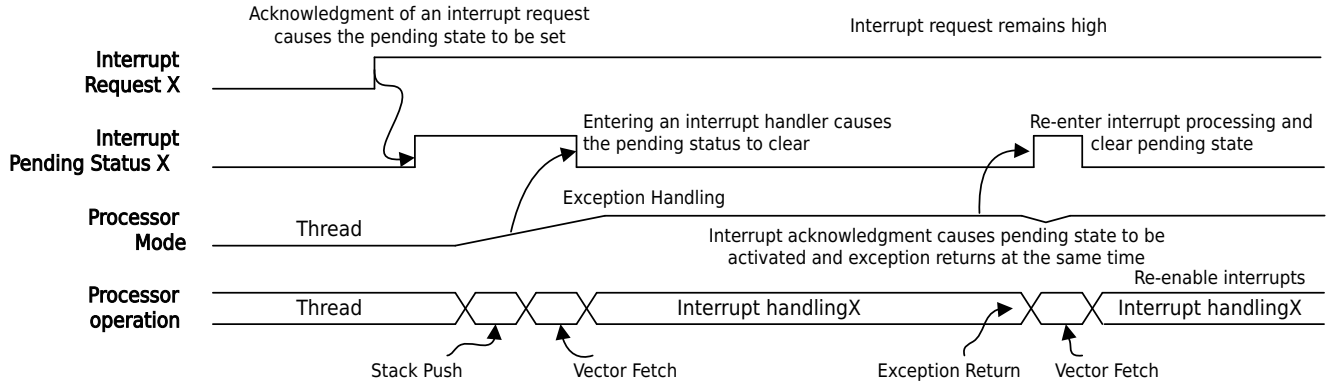



**Figure 5-4 When the Interrupt Exits, If the Interrupt Request Remains High, It Will Cause the Interrupt Processing to Be Executed Again**


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

If a peripheral interrupt request occurs during the execution of the interrupt service routine, the request will be regarded as a new interrupt request, and after this interrupt exits, the interrupt service program will be executed again. The handling process is illustrated in the following figure:

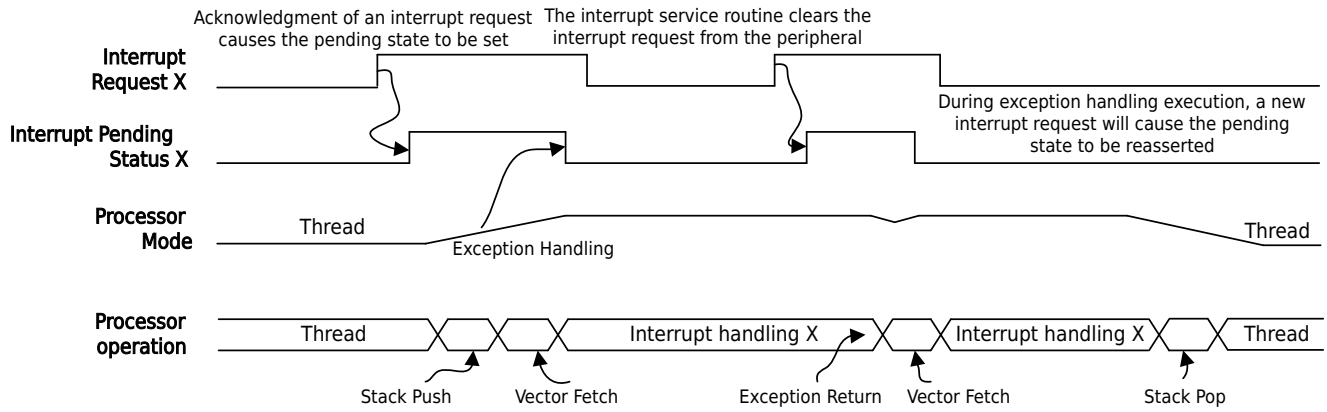



**Figure 5-5 Interrupt Pending Interrupts Generated During Interrupt Processing Can Also Be Acknowledged**

### 5.2.4 Interrupt Waiting

Typically, the interrupt waiting of the NVIC is 16 cycles. This wait time starts from the processor clock cycle when the interrupt is acknowledged, and ends when the interrupt handler starts executing. Computing interrupt waiting requires the following prerequisites:

- The interrupt is enabled and not masked by PRIMASK or any other currently executing exception handler.
- The memory system does not have any wait states. Bus transfers are used for interrupt processing, stack push, vector fetches, or instruction fetches at the start of interrupt processing. If the memory system needs to wait, the wait states generated when bus transfers occur may delay interrupts.

The following situations may cause different interrupt waiting:

- The end of the interrupt is chained. If another interrupt request is generated when the interrupt returns, the processor will skip the process of popping and pushing the stack, thus reducing the interrupt waiting time.
- Delayed arrival. If an interrupt occurs, another low-priority interrupt is being pushed to the stack. Due to the existence of the delayed arrival mechanism, the high-priority interrupt will be executed first, which will also reduce the waiting time of the high-priority interrupt.

### 5.2.5 Interrupt Source

Because the NVIC of the Cortex-M0+ processor supports up to 32 external interrupts, and in this system, there are more than 32 external interrupt sources, some external interrupts are multiplexed on the same NVIC interrupt input, and NMI (Non-Maskable Interrupt) did not use. The corresponding relationship between all external interrupt sources of this system and NVIC interrupt input is shown in the following table:

| NVIC Interrupt Input | External Interrupt Source | Active Mode | Sleep mode | DeepSleep mode |
|---|---|---|---|---|
| Interrupt #0 | PORTA | ✓ | ✓ | ✓ |
| Interrupt #1 | PORTB | ✓ | ✓ | ✓ |
| Interrupt #2 | PORTC | ✓ | ✓ | ✓ |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| NVIC Interrupt Input | External Interrupt Source | Active Mode | Sleep mode | DeepSleep mode |
|---|---|---|---|---|
| Interrupt #3 | PORTD | ✓ | ✓ | ✓ |
| Interrupt #4 | DMAC | ✓ | ✓ | × |
| Interrupt #5 | TIM3 | ✓ | ✓ | × |
| Interrupt #6 | USART0 | ✓ | ✓ | × |
| Interrupt #7 | USART1 | ✓ | ✓ | × |
| Interrupt #8 | LPUART0 | ✓ | ✓ | ✓ |
| Interrupt #9 | LPUART1 | ✓ | ✓ | ✓ |
| Interrupt #10 | SPI0 | ✓ | ✓ | × |
| Interrupt #11 | SPI1 | ✓ | ✓ | × |
| Interrupt #12 | I2C0 | ✓ | ✓ | × |
| Interrupt #13 | I2C1 | ✓ | ✓ | × |
| Interrupt #14 | TIM0 | ✓ | ✓ | × |
| Interrupt #15 | TIM1 | ✓ | ✓ | × |
| Interrupt #16 | TIM2 | ✓ | ✓ | × |
| Interrupt #17 | LPTIM0/1 | ✓ | ✓ | ✓ |
| Interrupt #18 | HSI2C | ✓ | ✓ | ✓(2) |
| Interrupt #19 | LPUART2 | ✓ | ✓ | ✓ |
| Interrupt #20 | - | - | - | - |
| Interrupt #21 | WWDT | ✓ | ✓ | × |
| Interrupt #22 | IWDT | ✓ | ✓ | ✓ |
| Interrupt #23 | RTC | ✓ | ✓ | ✓ |
| Interrupt #24 | ADC | ✓ | ✓ | × |
| Interrupt #25 | - | - | - | - |
| Interrupt #26 | VC0 | ✓ | ✓ | ✓ |
| Interrupt #27 | VC1 | ✓ | ✓ | ✓ |
| Interrupt #28 | LVD | ✓ | ✓ | ✓ |
| Interrupt #29 | LCD | ✓ | ✓ | ✓ |
| Interrupt #30 | FLASH/RAM | ✓ | ✓ | × |
| Interrupt #31 | CTRIM/CLKDET | ✓ | ✓ | ✓ |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

**Notice**

1. Since some module interrupts are multiplexed for the same IRQ interrupt source, when the CPU enters the interrupt operation, it must first determine which module generated the interrupt, and then perform the corresponding interrupt operation.
2. Only slave mode supports wakeup from DeepSleep.

### 5.2.6 Interrupt Structure Diagram

The interrupt structure diagram of this system is shown in the block diagram below. Please note the following:

- The respective interrupt enables of the peripheral interrupt sources are not marked in the figure, and only the logic block diagram of the interrupt signal after the peripheral interrupt is generated is included here.
- IRQ has more than one peripheral interrupt input multiplexing, and the interrupt flag bits of these peripherals must be read separately to determine which peripheral interrupt it is.
- If a peripheral interrupt source is asserted with a high-level signal, the corresponding bit in the NVIC interrupt pending state set register (NVIC_ISPR) will be set regardless of whether the interrupt is enabled in the NVIC interrupt enable set register (NVIC_ISER), indicating that an interrupt has been generated by the peripheral.
- Only when the interrupt enable register NVIC_ISER is set, the corresponding interrupt IRQ will respond to the processor and execute the corresponding interrupt program.
- In the interrupt program, the high-level interrupt signal of the peripheral interrupt source must be cleared, and the interrupt pending register NVIC_ISPR is automatically cleared by the hardware.
- The interrupt priority registers NVIC_IPR0-NVIC_IPR7 set the priority of 32 interrupt sources, 0b00 has the highest priority, and 0b11 has the lowest priority. When the priority is the same, the priority is determined by the interrupt number, and the smaller the number, the higher the priority.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

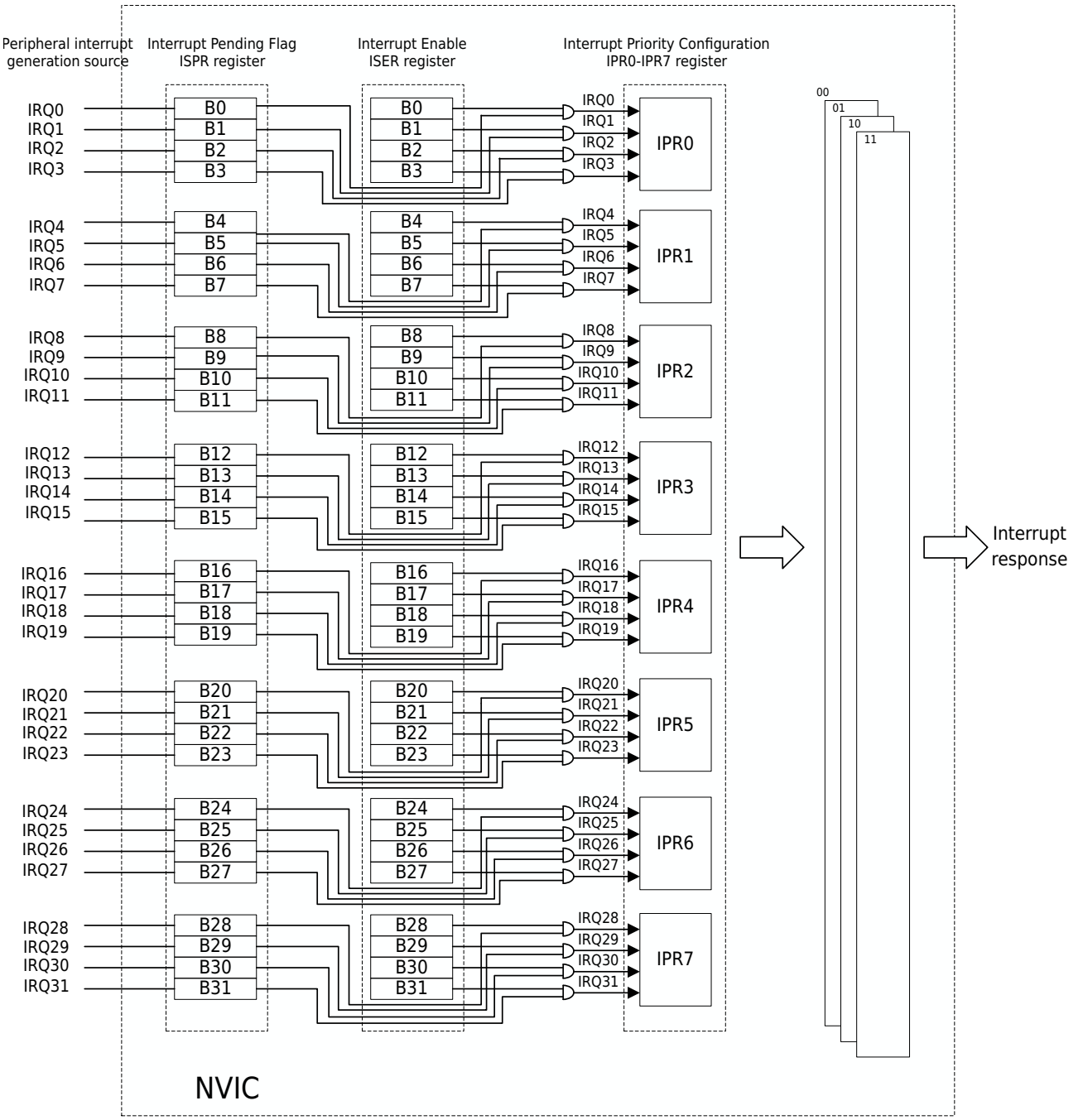



Figure 5-6 Interrupt Structure Diagram


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

## 5.3 Operation Example

### 5.3.1 External Interrupt Enable

Each peripheral module has its own interrupt enable register. When an interrupt operation is required, the peripheral's own interrupt enable must be enabled first. The operation of this enable bit is not discussed in this chapter, please refer to the chapter descriptions of the corresponding peripheral modules.

### 5.3.2 NVIC Interrupt Enable and Clear Enable

The Cortex-M0+ processor supports up to 32 interrupt sources, and each interrupt source corresponds to an interrupt enable bit and a clear-enable bit, forming a 32-bit interrupt enable set register (NVIC_ISER) and a 32-bit interrupt enable clear register (NVIC_ICER). If you want to enable a particular interrupt, set the corresponding bit in the NVIC_ISER register to 1. If you want to clear a particular interrupt, set the corresponding bit in the NVIC_ICER register to 1.

The interrupt enable and clear-enable operations mentioned here apply only to the NVIC processor. Whether an interrupt is generated by each peripheral is determined by the peripheral's own interrupt control registers and is unrelated to NVIC_ISER and NVIC_ICER.

### 5.3.3 NVIC Interrupt Pending and Clear Pending

If an interrupt occurs but cannot be handled immediately, the interrupt request will be pending. The pending state is kept in a register and will remain valid as long as the processor's current priority has not been lowered enough to handle the pending request and the pending state has not been cleared manually.

When the processor starts to enter the interrupt service, the hardware will automatically cause the clearing of the pending state.

You can access or modify the interrupt pending state by manipulating the interrupt pending state set rigister (NVIC_ISPR) and interrupt pending state clear register (NVIC_ICPR). The interrupt pending state register allows software to trigger interrupts.

### 5.3.4 NVIC Interrupt Priority

Setting the NVIC_IPR0-NVIC_IPR7 registers determines the priority of IRQ0-IRQ31. The programming of the interrupt priority register should be done before the interrupt is enabled, which is usually done at the beginning of the program. Changing the interrupt priority after the interrupt is enabled should be avoided, the results of this situation are unpredictable and are not supported by the Cortex-M0+ processor.

### 5.3.5 NVIC Interrupt Mask

Some time-sensitive applications need to disable all interrupts in a short period of time, which can be realized by using the interrupt mask register PRIMASK. Only 1 bit of the special register PRIMASK is valid, and it defaults to 0 after reset. When this register is 0, all interrupts and exceptions are enabled; when set to 1, only NMI (not supported by this system) and hardware error exceptions are enabled. In fact, when PRIMASK is set to 1, the current priority of the processor is reduced to 0 ( highest programmable priority).


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

PRIMASK register can be programmed in a variety of ways. Using assembly language, the MSR instruction can be used to set and clear the PRIMASK register. If using C language and CMSIS device driver library, users can use the following functions to set and clear PRIMASK.

```
void_enable_irq(void);  //Clear PRIMASK
void_disable_irq(void); //Set PRIMASK
```


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

## 5.4 Register Description

### 5.4.1 Register List

Base address: 0xE000E000

Table 5-4 Interrupt Controller (NVIC) Register Offset Address

| Offset | Register | Description | Width |
|---|---|---|---|
| 0x100 | NVIC_ISER | Interrupt Enable Set Register | 32 |
| 0x180 | NVIC_ICER | Interrupt Enable Clear Register | 32 |
| 0x200 | NVIC_ISPR | Interrupt Pending State Set Register | 32 |
| 0x280 | NVIC_ICPR | Interrupt Pending State Clear Register | 32 |
| 0x400 | NVIC_IPR0 | Interrupt #0- Interrupt #3 Priority Registers | 32 |
| 0x404 | NVIC_IPR1 | Interrupt #4- Interrupt #7 Priority Registers | 32 |
| 0x408 | NVIC_IPR2 | Interrupt #8- Interrupt #11 Priority Registers | 32 |
| 0x40C | NVIC_IPR3 | Interrupt #12- Interrupt #15 Priority Registers | 32 |
| 0x410 | NVIC_IPR4 | Interrupt #16- Interrupt #19 Priority Registers | 32 |
| 0x414 | NVIC_IPR5 | Interrupt #20- Interrupt #23 Priority Registers | 32 |
| 0x418 | NVIC_IPR6 | Interrupt #24- Interrupt #27 Priority Registers | 32 |
| 0x41C | NVIC_IPR7 | Interrupt #28- Interrupt #31 Priority Registers | 32 |
| - | PRIMASK | Interrupt Mask Special Register | 32 |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 5.4.2 Interrupt Enable Set Register (NVIC_ISER)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x100 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | SETENA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Access | RW | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 31:0 | SETENA | Interrupt Enable Set Flag<br>Set enable interrupt #0 to interrupt #31<br>● bit 0: IRQ0<br>● bit 1: IRQ1<br>● bit 2: IRQ2<br>● ...<br>● bit 31: IRQ31<br>**Note**<br>Each bit: writing 1 is set, writing 0 has no effect. | RW |

### 5.4.3 Interrupt Enable Clear Register (NVIC_ICER)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x180 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | CLRENA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Access | RW | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 31:0 | CLRENA | Interrupt Clear-Enable<br>Clear-enable interrupt #0 to interrupt #31<br>● bit 0: IRQ0<br>● bit 1: IRQ1<br>● bit 2: IRQ2<br>● ...<br>● bit 31: IRQ31<br>**Note**<br>Each bit: write 1 to clear flag, write 0 has no effect. . | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 5.4.4 Interrupt Pending State Set Register (NVIC_ISPR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x200 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | SETPEND | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Access | RW | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 31:0 | SETPEND | Interrupt Pending State Set Flag<br>Set the pending state of interrupt #0 to interrupt #31<br>● bit 0: IRQ0<br>● bit 1: IRQ1<br>● bit 2: IRQ2<br>● ...<br>● bit 31: IRQ31<br>**Note**<br>Each bit: write 1 is set, writing 0 has no effect. | RW |

### 5.4.5 Interrupt Pending State Clear Register (NVIC_ICPR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x280 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | CLRPEND | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Access | RW | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 31:0 | CLRPEND | Interrupt Pending State Clear<br>Clear the pending state of interrupt #0 to interrupt #31<br>● bit 0: IRQ0<br>● bit 1: IRQ1<br>● bit 2: IRQ2<br>● ...<br>● bit 31: IRQ31<br>**Note**<br>Each bit: write 1 to clear flag, write 0 has no effect. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 5.4.6 Interrupt Priority Register (NVIC_IPR0)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x400 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | IPR0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Access | RW | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 31:0 | IPR0 | Priority of Interrupt #0 to Interrupt #3<br>The corresponding bits for each interrupt priority are as follows:<br>● [31:30]: Priority of interrupt #3<br>● [23:22]: Priority of interrupt #2<br>● [15:14]: Priority of interrupt #1<br>● [7:6]: Priority of interrupt #0<br>In the above configuration, a setting of 0b00 indicates the highest priority, whereas 0b11 indicates the lowest priority. Since the other bits are unused, configuring them has no effect and read always returns 0. | RW |

### 5.4.7 Interrupt Priority Register (NVIC_IPR1)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x404 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | IPR1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Access | RW | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 31:0 | IPR1 | Priority of Interrupt #4 to Interrupt #7<br>The corresponding bits for each interrupt priority are as follows:<br>● [31:30]: Priority of interrupt #7<br>● [23:22]: Priority of interrupt #6<br>● [15:14]: Priority of interrupt #5<br>● [7:6]: Priority of interrupt #4<br>In the above configuration, a setting of 0b00 indicates the highest priority, whereas 0b11 indicates the lowest priority. Since the other bits are unused, configuring them has no effect and read always returns 0. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 5.4.8 Interrupt Priority Register (NVIC_IPR2)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x408 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | IPR2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Access | RW | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 31:0 | IPR2 | Priority of Interrupt #8 to Interrupt #11<br>The corresponding bits for each interrupt priority are as follows:<br>● [31:30]: Priority of interrupt #11<br>● [23:22]: Priority of interrupt #10<br>● [15:14]: Priority of interrupt #9<br>● [7:6]: Priority of interrupt #8<br>In the above configuration, a setting of 0b00 indicates the highest priority, whereas 0b11 indicates the lowest priority. Since the other bits are unused, configuring them has no effect and read always returns 0. | RW |

### 5.4.9 Interrupt Priority Register (NVIC_IPR3)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x40C | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | IPR3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Access | RW | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 31:0 | IPR3 | Priority of Interrupt #12 to Interrupt #15<br>The corresponding bits for each interrupt priority are as follows:<br>● [31:30]: Priority of interrupt #15<br>● [23:22]: Priority of interrupt #14<br>● [15:14]: Priority of interrupt #13<br>● [7:6]: Priority of interrupt #12<br>In the above configuration, a setting of 0b00 indicates the highest priority, whereas 0b11 indicates the lowest priority. Since the other bits are unused, configuring them has no effect and read always returns 0. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 5.4.10 Interrupt Priority Register (NVIC_IPR4)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x410 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | IPR4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Access | RW | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 31:0 | IPR4 | Priority of Interrupt #16 to Interrupt #19<br>The corresponding bits for each interrupt priority are as follows:<br>● [31:30]: Priority of interrupt #19<br>● [23:22]: Priority of interrupt #18<br>● [15:14]: Priority of interrupt #17<br>● [7:6]: Priority of interrupt #16<br>In the above configuration, a setting of 0b00 indicates the highest priority, whereas 0b11 indicates the lowest priority. Since the other bits are unused, configuring them has no effect and read always returns 0. | RW |

### 5.4.11 Interrupt Priority Register (NVIC_IPR5)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x414 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | IPR5 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Access | RW | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 31:0 | IPR5 | Priority of Interrupt #20 to Interrupt #23<br>The corresponding bits for each interrupt priority are as follows:<br>● [31:30]: Priority of interrupt #23<br>● [23:22]: Priority of interrupt #22<br>● [15:14]: Priority of interrupt #21<br>● [7:6]: Priority of interrupt #20<br>In the above configuration, a setting of 0b00 indicates the highest priority, whereas 0b11 indicates the lowest priority. Since the other bits are unused, configuring them has no effect and read always returns 0. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 5.4.12 Interrupt Priority Register (NVIC_IPR6)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x418 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | IPR6 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Access | RW | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 31:0 | IPR6 | Priority of Interrupt #24 to Interrupt #27<br>The corresponding bits for each interrupt priority are as follows:<br>● [31:30]: Priority of interrupt #27<br>● [23:22]: Priority of interrupt #26<br>● [15:14]: Priority of interrupt #25<br>● [7:6]: Priority of interrupt #24<br>In the above configuration, a setting of 0b00 indicates the highest priority, whereas 0b11 indicates the lowest priority. Since the other bits are unused, configuring them has no effect and read always returns 0. | RW |

### 5.4.13 Interrupt Priority Register (NVIC_IPR7)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x41C | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | IPR7 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Access | RW | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 31:0 | IPR7 | Priority of Interrupt #28 to Interrupt #31<br>The corresponding bits for each interrupt priority are as follows:<br>● [31:30]: Priority of interrupt #31<br>● [23:22]: Priority of interrupt #30<br>● [15:14]: Priority of interrupt #29<br>● [7:6]: Priority of interrupt #28<br>In the above configuration, a setting of 0b00 indicates the highest priority, whereas 0b11 indicates the lowest priority. Since the other bits are unused, configuring them has no effect and read always returns 0. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 5.4.14 Interrupt Mask Special Register (PRIMASK)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| - | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | PRIMASK |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | RW |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:1* | *Reserved* | - | - |
| 0 | PRIMASK | Interrupt mask flag<br>Typically used in time-critical application scenarios. The special register can be accessed through MSR and MRS special register operation instructions, and can also be accessed by changing the processor state instruction CPS.<br>● 0b0: All interrupts and exceptions are in the enabled state.<br>● 0b1: Only the NMI and hardware error exceptions are enabled. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

# 6 FLASH

## 6.1 Introduction

This system includes a Flash memory with a capacity of 128KB, which is divided into 256 sectors, and the capacity of each sector is 512 bytes.

This controller supports erasing, programming and reading operations on Flash memory, also supports erasing and writing protection of Flash memory, and writing protection of control registers.

This controller supports data read and write operations of three-bit widths of byte (8 bits), half-word (16 bits) and word (32 bits) of Flash. Note that the target address of the Byte operation must be byte-aligned, the target address of the half-word operation must be half-word-aligned (the lowest bit of the address is 0b0), and the address of the word operation must be word-aligned (the lowest two bits of the address are 0b00). If the target address is not aligned according to the bit width, the operation is invalid and the CPU will enter the Hard Fault interrupt.

## 6.2 Functional Description

### 6.2.1 Capacity Division

The correspondence between the capacity division addresses and sequence numbers is as follows:

Table 6-1 Flash Capacity Division Table

| Address | Serial Number | ... | Address | Serial Number |
|---|---|---|---|---|
| 0x0E00-0x0FFF | Sector7 | ... | 0x1FE00-0x1FFFF | Sector255 |
| 0x0C00-0x0DFF | Sector6 | ... | 0x1FC00-0x1FDFF | Sector254 |
| 0x0A00-0x0BFF | Sector5 | ... | 0x1FA00-0x1FBFF | Sector253 |
| 0x0800-0x09FF | Sector4 | ... | 0x1F800-0x1F9FF | Sector252 |
| 0x0600-0x07FF | Sector3 | ... | 0x1F600-0x1F7FF | Sector251 |
| 0x0400-0x05FF | Sector2 | ... | 0x1F400-0x1F5FF | Sector250 |
| 0x0200-0x03FF | Sector1 | ... | 0x1F200-0x1F3FF | Sector249 |
| 0x0000-0x01FF | Sector0 | ... | 0x1F000-0x1F1FF | Sector248 |

### 6.2.2 Read Wait Period

The fastest instruction fetch frequency supported by the built-in Flash of this device is 24MHz. When the HCLK frequency exceeds 24MHz and is less than or equal to 48MHz, waiting period must be inserted for the Flash read time, that is, set FLASH_CR.WAIT to 1. When the wait cycle is inserted, Flash will complete a read operation every two periods.

### 6.2.3 Flash Security Protection

#### Sector Erase/Write Protection

The entire 128KB Flash memory is divided into 256 sectors, and every 4 sectors share an erase/write protection bit. When the page is protected, the erasing and writing operations on the sector are invalid and an alarm flag and interrupt signal are generated. When any sector in the Flash memory is protected, the whole-chip erasing of the Flash is invalid, and an alarm flag and an interrupt signal are generated.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### PC Address Erase and Write Protection

The CPU runs the program in Flash, if the current PC pointer just falls within the address range of the page to be erased, then the erase operation is invalid and an alarm flag and an interrupt signal are generated.

### Register Write Protection

The important controller of this module shields ordinary write operations, and must be modified by writing sequence.

- **Registers Can Be Changed After a Write Sequence Are As Follows:**
  FLASH_CR, FLASH_SLOCK0, FLASH_SLOCK1
- **Registers Can Be Changed Without a Write Sequence Are As Follows:**
  FLASH_ICLR, FLASH_BYPASS

The specific operation steps to modify the register value by writing sequence are as follows:

**Step1.** Write 0x5A5A to the FLASH_BYPASS register.

**Step2.** Write 0xA5A5 to the FLASH_BYPASS register.

**Step3.** Write the target value to the register to be modified.

**Step4.** Verify whether the current value of the register to be modified is the same as the target, if not, jump to Step1.

**Step5.** Perform other operations.

**Notice**

Write 0x5A5A, 0xA5A5, and write the target register. Do not insert any write operations (write Flash, RAM, REG, etc.) between these three steps of writing operations, otherwise the value of the target register cannot be rewritten. If rewriting fails, you need to perform these three steps again.

### Data Readout Protection

The chip's Flash supports 4 -level read protection, and the current protection level can be read through the FLASH_LOCK_ST register. When programming the chip with a programmer, the read protection level of Flash can be configured.

The functions of the chip under each protection level are shown in the table below.

| Protection Level | Chip Function |
|---|---|
| Level0 | Flash can be read, written and erased through CPU, DMA, ISP and SWD |
| Level1 | Data cannot be read from SWD<br>● Before SWD is inserted, the Flash can be read, written and erased through CPU or DMA;<br>● After SWD is inserted, the Flash denies any access.<br>be downgraded to Level0 through ISP and SWD. After downgrading, the Flash content is all FF |
| Level2 | Data cannot be read<br>Data can be read, written and erased from the Flash through CPU or DMA.<br>Data can be downgraded to Level0 through ISP. After downgrading, the Flash content is all FF<br>The SWD interface is disabled |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Protection Level | Chip Function |
|---|---|
| Level3 | Data cannot be read<br>Data can be read, written and erased from the Flash through CPU or DMA.<br>The ISP and SWD interfaces are both unreadable, that is, the chip can only be programmed once |

## 6.3 Operation Example

**Notice**

- Before performing erase or write operations on FLASH, it is necessary to set FLASH_CR.RO to 0 and configure FLASH as readable, programmable and erasable.
- After the FLASH erase or write operation is completed, it is recommended to set FLASH_CR.RO to 1 and configure FLASH as read-only, non-programmable or non-erasable.

### 6.3.1 Sector Erase

#### Context

Sector erase allows the user to erase one user-specified sector at a time. After the erase operation is completed, the data in the sector are all 0xFF. If the erase operation is performed from Flash, the CPU will stop fetching instructions, and the hardware will automatically wait for the operation to complete(FLASH_CR.BUSY becomes 0); if the erase operation is performed from RAM, the CPU will not stop fetching instructions, user software should wait for the operation to complete(FLASH_CR.BUSY becomes 0).

#### Procedure

**Step1.** Write 0x5A5A and 0xA5A5 to the FLASH_BYPASS register in sequence to enable register rewriting.

**Step2.** Configure FLASH_CR.OP to 2, set the Flash operation mode to Sector Erase.

**Step3.** Check if FLASH_CR.OP is 2, if not, jump to Step1.

**Step4.** Write 0x5A5A and 0xA5A5 to the FLASH_BYPASS register in sequence to enable register rewriting.

**Step5.** Set the corresponding bit of FLASH_SLOCKx to 1 to remove the sector 's erasure protection.

**Step6.** Check whether the corresponding bit of FLASH_SLOCKx is 1, if it is not 1, jump to Step4.

**Step7.** Write arbitrary data to any address in the Sector to be erased, triggering Sector erasure.

```
*((unsigned char *)0x00000200) = 0x00
```

**Step8.** Wait for FLASH_CR.BUSY to become 0, and the Sector erase operation is completed.

**Step9.** To erase other Sectors, repeat Step4-Step8.

### 6.3.2 Chip Erase

#### Context

Chip erase can erase all sector at one time. After the erase operation is completed, the data in all sector are 0xFF. If the erase operation is performed from the Flash, the operation will be prohibited. Because this operation will erase the program segment where the current PC is located. If this happens, the FLASH_IFR.RC will be set; if the erase operation is performed from RAM, the CPU will not stop fetching instructions, and the user software should wait for the operation to complete (FLASH_CR.BUSY becomes 0).


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

#### Procedure

**Step1.** Write 0x5A5A and 0xA5A5 to the FLASH_BYPASS register in sequence to enable register rewriting.

**Step2.** Configure FLASH_CR.OP to 3, set the Flash operation mode to Chip erase.

**Step3.** Check if FLASH_CR.OP is 3, if not, jump to Step1.

**Step4.** Write 0x5A5A and 0xA5A5 to the FLASH_BYPASS register in sequence to enable register rewriting.

**Step5.** Set FLASH_SLOCK0 to 0xFFFFFFFF, remove the erase protection of Sector0-127.

**Step6.** Check whether FLASH_SLOCK0 is 0xFFFFFFFF, if not 0xFFFFFFFF, jump to Step4.

**Step7.** Write 0x5A5A and 0xA5A5 to the FLASH_BYPASS register in sequence to enable register rewriting.

**Step8.** Set FLASH_SLOCK1 to 0xFFFFFFFF, remove the erase protection of Sector128-255.

**Step9.** Check whether FLASH_SLOCK1 is 0xFFFFFFFF, if not 0xFFFFFFFF, jump to Step7.

**Step10.** Write any address in the Chip to be erased to trigger Chip erasure.

```
*((unsigned char *)0x00000000) = 0x00
```

**Step11.** Wait for FLASH_CR.BUSY to become 0, and the Chip erase operation is completed.

### 6.3.3 Program

#### Context

The program can only change the bit data in Flash from 1 to 0, so before writing data, make sure that the data in the address to be written is 0xFF. Support 3 write lengths: Byte (8bits), Half-word (16bits), Word (32bits), the written data is stored in Flash in little-endian mode, that is, the low byte of the data is stored in the low address. If the program is executed from Flash, the CPU will stop fetching instructions, and the hardware will automatically wait for the operation to complete (for FLASH_CR.BUSY to becomes 0); if the program is executed from RAM, the CPU will not stop fetching instructions, and the user Software should wait for the operation to complete (for FLASH_CR.BUSY to goes to 0).

#### Procedure

- **Byte Program**

  **Step1.** Write 0x5A5A and 0xA5A5 to the FLASH_BYPASS register in sequence to enable register rewriting.

  **Step2.** Configure FLASH_CR.OP to 1, set the Flash operation mode to write.

  **Step3.** Check if FLASH_CR.OP is 1, if not, jump to Step1.

  **Step4.** Write 0x5A5A and 0xA5A5 to the FLASH_BYPASS register in sequence to enable register rewriting.

  **Step5.** Set the corresponding bit of FLASH_SLOCK to 1 to remove the erase protection.

  **Step6.** Check whether the corresponding bit of FLASH_SLOCK is 1, if it is not 1, jump to Step4.

  **Step7.** Perform Byte program on the target address to be written to trigger the program.

  ```
  *((unsigned char *)0x00001231) = 0x5A
  ```

  **Step8.** Wait for for FLASH_CR.BUSY to to become 0, and the program is completed.

  **Step9.** To write Byte to other addresses, repeat Step7-Step8.

- **Half-word Program**


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

**Step1.** Write 0x5A5A and 0xA5A5 to the FLASH_BYPASS register in sequence to enable register rewriting.

**Step2.** Configure FLASH_CR.OP to 1, set the Flash operation mode to write.

**Step3.** Check if FLASH_CR.OP is 1, if not, jump to Step1.

**Step4.** Write 0x5A5A and 0xA5A5 to the FLASH_BYPASS register in sequence to enable register rewriting.

**Step5.** Set the corresponding bit of FLASH_SLOCK to 1 to remove the erase protection.

**Step6.** Check whether the corresponding bit of FLASH_SLOCK is 1, if it is not 1, jump to Step4.

**Step7.** Perform a Half-word program on the target address to be written to trigger the program.

```
*((unsigned short int *)0x00001232) = 0xABCD
```

**Step8.** Wait for for FLASH_CR.BUSY to to become 0, and the program is completed.

**Step9.** To write Half-word to other addresses, repeat Step7-Step8.

- **Word Program**

**Step1.** Write 0x5A5A and 0xA5A5 to the FLASH_BYPASS register in sequence to enable register rewriting.

**Step2.** Configure FLASH_CR.OP to 1, set the Flash operation mode to write.

**Step3.** Check if FLASH_CR.OP is 1, if not, jump to Step1.

**Step4.** Write 0x5A5A and 0xA5A5 to the FLASH_BYPASS register in sequence to enable register rewriting.

**Step5.** Set the corresponding bit of FLASH_SLOCK to 1 to remove the erase protection.

**Step6.** Check whether the corresponding bit of FLASH_SLOCK is 1, if it is not 1, jump to Step4.

**Step7.** Perform a Word program on the target address to be written to trigger the program.

```
*((unsigned int *)0x00001234) = 0x55667788
```

**Step8.** Wait for for FLASH_CR.BUSY to to become 0, and the program is completed.

**Step9.** To write Word to other addressed, repeat Step7-Step8.

### 6.3.4 Cont-Program

#### Context

The program can only change the bit data in Flash from 1 to 0, so before writing data, make sure that the data in the address to be written is 0xFF. Support 3 write lengths: Byte (8bits), Half-word (16bits), Word (32bits), The written data is stored in Flash in little-endian mode, that is, the low byte of the data is stored in the low address. Cont-program can only be performed from RAM, and "read " on Flash are prohibited. User software should wait for the operation to complete (for FLASH_CR.BUSY to becomes 0).

It is recommended that HCLK not be lower than 2 MHz during continuous programming.

#### Procedure

- **Byte Program**

**Step1.** Write 0x5A5A and 0xA5A5 to the FLASH_BYPASS register in sequence to enable register rewriting.

**Step2.** Configure FLASH_CR.OP value at 0.

**Step3.** Write 0x5A5A and 0xA5A5 to the FLASH_BYPASS register in sequence to enable register rewriting.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

**Step4.** Configure FLASH_CR.CONTP to 1, set the Flash operation mode to continuous write.

**Step5.** Check whether FLASH_CR.OP and FLASH_CR.CONTP are 0 and 1 respectively, if not, jump to Step1.

**Step6.** Write 0x5A5A and 0xA5A5 to the FLASH_BYPASS register in sequence to enable register rewriting.

**Step7.** Set the corresponding bit of FLASH_SLOCK to 1 to remove the erase protection.

**Step8.** Check whether the corresponding bit of FLASH_SLOCK is 1, if it is not 1, jump to Step6.

**Step9.** Perform Byte program on the target address to be written to trigger the program.

```
*((unsigned char *)0x00001232) = 0xAB
```

**Step10.** Wait for for FLASH_CR.BUSY to to become 0, and the program is completed.

**Step11.** To write Byte to other addresses, repeat Step9-Step10.

**Step12.** If all the writing operation is completed, proceed to the next step.

**Step13.** Write 0x5A5A and 0xA5A5 to the FLASH_BYPASS register in sequence to enable register rewriting.

**Step14.** Configure FLASH_CR.OP at 1.

**Step15.** Write 0x5A5A and 0xA5A5 to the FLASH_BYPASS register in sequence to enable register rewriting.

**Step16.** Configure FLASH_CR.CONTP to 0.

**Step17.** Check whether FLASH_CR.OP and FLASH_CR.CONTP are 1 and 0 respectively, if not, jump to Step13.

**Step18.** Wait FLASH_CR.BUSY to become 0.

**Step19.** Proceed with the following, FLASH_CR.OP can be changed to the value of other modes complete other functional operations.

- **Half-word Program**

**Step1.** Write 0x5A5A and 0xA5A5 to the FLASH_BYPASS register in sequence to enable register rewriting.

**Step2.** Configure FLASH_CR.OP value at 0.

**Step3.** Write 0x5A5A and 0xA5A5 to the FLASH_BYPASS register in sequence to enable register rewriting.

**Step4.** Configure FLASH_CR.CONTP to 1, set the Flash operation mode to continuous write.

**Step5.** Check whether FLASH_CR.OP and FLASH_CR.CONTP are 0 and 1 respectively, if not, jump to Step1.

**Step6.** Write 0x5A5A and 0xA5A5 to the FLASH_BYPASS register in sequence to enable register rewriting.

**Step7.** Set the corresponding bit of FLASH_SLOCK to 1 to remove the erase protection.

**Step8.** Check whether the corresponding bit of FLASH_SLOCK is 1, if it is not 1, jump to Step6.

**Step9.** Perform a Half-word program on the target address to be written to trigger the program.

```
*((unsigned short int *)0x00001232) = 0xABCD
```

**Step10.** Wait for for FLASH_CR.BUSY to to become 0, and the program is completed.

**Step11.** To write Half-word to other addresses, repeat Step9-Step10.

**Step12.** If all the writing operation is completed, proceed to the next step.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

**Step13.** Write 0x5A5A and 0xA5A5 to the FLASH_BYPASS register in sequence to enable register rewriting.

**Step14.** Configure FLASH_CR.OP at 1.

**Step15.** Write 0x5A5A and 0xA5A5 to the FLASH_BYPASS register in sequence to enable register rewriting.

**Step16.** Configure FLASH_CR.CONTP to 0.

**Step17.** Check whether FLASH_CR.OP and FLASH_CR.CONTP are 1 and 0 respectively, if not, jump to Step13.

**Step18.** Wait FLASH_CR.BUSY to become 0.

**Step19.** Proceed with the following, FLASH_CR.OP can be changed to the value of other modes complete other functional operations.

- **Word Program**

**Step1.** Write 0x5A5A and 0xA5A5 to the FLASH_BYPASS register in sequence to enable register rewriting.

**Step2.** Configure FLASH_CR.OP value at 0.

**Step3.** Write 0x5A5A and 0xA5A5 to the FLASH_BYPASS register in sequence to enable register rewriting.

**Step4.** Configure FLASH_CR.CONTP to 1, set the Flash operation mode to continuous write.

**Step5.** Check whether FLASH_CR.OP and FLASH_CR.CONTP are 0 and 1 respectively, if not, jump to Step1.

**Step6.** Write 0x5A5A and 0xA5A5 to the FLASH_BYPASS register in sequence to enable register rewriting.

**Step7.** Set the corresponding bit of FLASH_SLOCK to 1 to remove the erase protection.

**Step8.** Check whether the corresponding bit of FLASH_SLOCK is 1, if it is not 1, jump to Step6.

**Step9.** Perform a Word program on the target address to be written to trigger the program.

```
*((unsigned int *)0x00001234) = 0xABCDABCD
```

**Step10.** Wait for for FLASH_CR.BUSY to to become 0, and the write operation is completed.

**Step11.** To write Word to other addressed, repeat Step9-Step10.

**Step12.** If all the writing operation is completed, proceed to the next step.

**Step13.** Write 0x5A5A and 0xA5A5 to the FLASH_BYPASS register in sequence to enable register rewriting.

**Step14.** Configure FLASH_CR.OP to 1, set the Flash operation mode to write.

**Step15.** Write 0x5A5A and 0xA5A5 to the FLASH_BYPASS register in sequence to enable register rewriting.

**Step16.** Configure FLASH_CR.CONTP to 0.

**Step17.** Check whether FLASH_CR.OP and FLASH_CR.CONTP are 1 and 0 respectively, if not, jump to Step13.

**Step18.** Wait FLASH_CR.BUSY to become 0.

**Step19.** Proceed with the following, FLASH_CR.OP can be changed to the value of other modes complete other functional operations.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 6.3.5 Read

#### Context

Support to 3 kinds read: Byte(8bits), Half-word(16bits), Word(32bits), the read data is little-endian mode, that is, the low byte of the data is stored in the low address. Read need no prior steps, the data in the FLASH can be read at any time.

#### Procedure

- **Byte Read**

```
temp = *((unsigned char *)0x00001231)
```

- **Half-word Read**

```
temp = *((unsigned short int *)0x00001232)
```

- **Word Read**

```
temp = *((unsigned int *)0x00001234)
```


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

## 6.4 Register Description

### 6.4.1 Register List

Base address: 0x40020000

Table 6-3 Flash Controller (FLASH) Register Offset

| Offset | Register | Description |
|---|---|---|
| 0x20 | FLASH_CR | Control Register |
| 0x24 | FLASH_IFR | Interrupt Flag Register |
| 0x28 | FLASH_ICLR | Interrupt Flag Clear Register |
| 0x2C | FLASH_BYPASS | 0x5A5A-0xA5A5 Bypass Sequence Register |
| 0x30 | FLASH_SLOCK0 | Sector0-127 Erase and Write Protection Register |
| 0x34 | FLASH_SLOCK1 | Sector128-255 Erase and Write Protection Register |
| 0x6C | FLASH_LOCK_ST | Protection Status Register |

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 6.4.2 CR Register (FLASH_CR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x20 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0b00000000 00000000 0000X010 00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | RO | CONTP | DPSTB_EN | Reserved | | | IE | | BUSY | WAIT | | OP | |
| Access | | | | | | | | | | | | | | | | | | | | | RW | RW | RW | | | | RW | | RO | RW | | RW | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:12* | *Reserved* | - | - |
| 11 | RO | Read-only enable bit<br>● 0b0: Readable, programmable, erasable<br>● 0b1: Read only, not programmable or erasable | RW |
| 10 | CONTP | Continuous programming<br>● 0b0: Single programming<br>● 0b1: Continuous programming | RW |
| 9 | DPSTB_EN | FLASH dpstb enable mask bit<br>● 0b0: When the system enters DeepSleep mode, FLASH does not enter low power consumption mode<br>● 0b1: When the system enters DeepSleep mode, FLASH enters low power consumption mode | RW |
| *8:7* | *Reserved* | - | - |
| 6:5 | IE | FLASH interrupt enable<br>IE[6]: FLASH erase/write protected address interrupt enable<br>IE[5]: FLASH erase PC Value interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 4 | BUSY | Idle/Busy flag bit<br>● 0b0: Idle State<br>● 0b1: Busy State | RO |
| 3:2 | WAIT | Read FLASH period<br>● 0b00: 1 period, 0-24MHz<br>● 0b01: 2 periods, 24-48MHz<br>● 0b10: Setting prohibited<br>● 0b11: Setting prohibited | RW |
| 1:0 | OP | FLASH operation<br>● 0b00: Read<br>● 0b01: Program<br>● 0b10: Sector erase<br>● 0b11: Chip erase | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 6.4.3 IFR Register (FLASH_IFR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x24 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000010 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | PROT | PC |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | RO | RO |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:2* | *Reserved* | - | - |
| 1 | PROT | FLASH erase/write protection alarm interrupt flag bit | RO |
| 0 | PC | FLASH erase/write PC address alarm interrupt flag bit | RO |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 6.4.4 ICLR Register (FLASH_ICLR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x28 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x0000000F | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | PROT | PC |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | WO | WO |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:2* | *Reserved* | - | - |
| 1 | PROT | Clear protection alarm interrupt flag bit<br>Write 0 to clear flag, write 1 has no effect; read always returns 1. | WO |
| 0 | PC | Clear the PC address alarm interrupt flag bit<br>Write 0 to clear flag, write 1 has no effect; read always returns 1. | WO |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 6.4.5 BYPASS Register (FLASH_BYPASS)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x2C | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | BYSEQ | | | | | | | | | |
| Access | | | | | | | | | | | | | | | | | | | | | | | RW | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:16* | *Reserved* | - | - |
| 15:0 | BYSEQ | FLASH module protection control register<br>Write 0x5A5A first to FLASH_BYPASS, and then write 0xA5A5 to start the program to FLASH module registers. As long as the program is done to module register, this protection bit automatically recovers to the protection state. Subsequent program to the FLASH module registers require repeating the previous procedures. | RW |

### 6.4.6 SLOCK0 Register (FLASH_SLOCK0)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x30 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0X00000000 (Peripheral reset is invalid) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | SLOCK | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Access | RW | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 31:0 | SLOCK | Sector Erase/Write Protection Bit<br>● 0b0: Disable erase/write<br>● 0b1: Enable erase/write | RW |

| Bit | Serial Number |
|---|---|
| 0 | Sector0-3 |
| 1 | Sector4-7 |
| 2 | Sector8-11 |
| 3 | Sector12-15 |
| ... | ... |
| 31 | Sector124-127 |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 6.4.7 SLOCK1 Register (FLASH_SLOCK1)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x34 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0X00000000 (Peripheral reset is invalid) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | SLOCK | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Access | RW | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 31:0 | SLOCK | Sector Erase/Write Protection Bit<br>● 0b0: Disable erase/write<br>● 0b1: Enable erase/write | RW |

| Bit | Serial Number |
|---|---|
| 0 | Sector128-131 |
| 1 | Sector132-135 |
| 2 | Sector136-139 |
| 3 | Sector140-143 |
| ... | ... |
| 31 | Sector252-255 |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 6.4.8 Protection Status Register (FLASH_LOCK_ST)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x6C | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LOCK_ST | |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | RO | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:2* | *Reserved* | - | - |
| 1:0 | LOCK_ST | FLASH Protection Status<br>● 0b00: Level0, ISP can read and write, SWD can read and write<br>● 0b01: Level1, ISP can be downgraded, SWD can be downgraded; data cannot be read out<br>● 0b10: Level2, ISP can be downgraded, SWD has no function; data cannot be read out<br>● 0b11: Level3, ISP has no function, SWD has no function; data cannot be read out | RO |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

# 7 RAM

## 7.1 Introduction

This system includes an SRAM block with a capacity of 16KB, supporting read and write operations in byte (8 bits), halfword (16 bits), and word (32 bits) formats. Read and write operations can be performed at the system clock frequency without waiting for cycles. In addition, this controller also supports parity check, which can perform parity check on each byte of SRAM data, and generate a parity check error interrupt.

## 7.2 Functional Description

### 7.2.1 RAM Address Range

RAM in the system map is shown in the table below:

**Table 7-1 RAM Address Mapping**

| Address Range | Size | Memory Type |
|---|---|---|
| 0x20000000-0x20003FFF | 16KByte | SRAM |

### 7.2.2 Read and Write Bit Width

This controller supports byte (8 bits), halfword (16 bits), word (32 bits) read and write operations with three-bit widths.

- Addresses for byte operations must be byte aligned
- The target address for halfword operations must be halfword-aligned (the least significant bit of the address must be 0b0).
- The address for word operations must be word-aligned (the least significant two bits of the address must be 0b00).

If the target address of the read and write operation is not aligned according to the bit width, the operation is invalid, and the system will generate a Hard Fault error interrupt.

### 7.2.3 Parity Check

This controller supports parity check of SRAM data:

- When writing data to SRAM, do a parity check on each byte of data, and store the 1-bit check value and 8-bit data into the SRAM together.
- When reading data from SRAM, the controller reads 8 bits of data along with a 1-bit parity value and performs a parity check.

If there is an error in the check, the parity check error flag is set, and if the interrupt is enabled, an error interrupt will be generated.

**Notice**

When the parity check is enabled, the SRAM must be initialized before reading the SRAM data, otherwise the parity check alarm flag or interrupt may be triggered by mistake.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

## 7.3 Register Description

### 7.3.1 Register List

Base address: 0x40020400

Table 7-2 RAM Controller (RAM) Register Offset

| Offset | Register | Description | Width |
|---|---|---|---|
| 0x00 | RAM_CR | Control register | 32 |
| 0x04 | RAM_ERRADDR | Parity Check Error Address Register | 32 |
| 0x08 | RAM_IFR | Error Interrupt Flag Register | 32 |
| 0x0C | RAM_ICLR | Error Interrupt Flag Clear Register | 32 |

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 7.3.2 Control Register (RAM_CR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000001 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | IE | Reserved | |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | RW | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:2* | *Reserved* | - | - |
| 1 | IE | Parity check error alarm interrupt enable control<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| *0* | *Reserved* | - | - |

### 7.3.3 Parity Check Error Address Register (RAM_ERRADDR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x04 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | ERRADDR | | | | | | | | | | | | | |
| Access | | | | | | | | | | | | | | | | | | | RO | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:14* | *Reserved* | - | - |
| 13:0 | ERRADDR | Parity check error Byte address<br>After the interrupt flag is cleared, the corresponding address is also cleared. | RO |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 7.3.4 Error Interrupt Flag Register (RAM_IFR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x08 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | ERR |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | RO |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:1* | *Reserved* | - | - |
| 0 | ERR | Parity check error flag<br>● 0b0: when reading RAM, the parity check is correct<br>● 0b1: parity check error when reading RAM | RO |

### 7.3.5 Error Interrupt Flag Clear Register (RAM_ICLR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x0C | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000001 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | ERRCLR |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | WO |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:1* | *Reserved* | - | - |
| 0 | ERRCLR | Error interrupt flag clear bit<br>Write 0 to clear flag, write 1 has no effect; read always returns 1. | WO |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

# 8 GPIO

## 8.1 Introduction

This series supports up to 56 digital general-purpose input/output ports . The input and output signals of the analog module ADC/VC/LVD/LCD, and the input and output signals of various functional modules (such as SPI/USART/I2C/Timer, etc.) can be multiplexed with digital general-purpose input and output ports.

Each port can be configured as internal pull-up/pull-down input, floating input, push-pull output, open-drain output, two levels driving capability output. After the digital port is configured as an analog port, the digital function is isolated, "1" and "0" cannot be output, and the result of reading the port input value register by the CPU is "0".

When each digital port is configured as an input, it can provide an external interrupt. The interrupt type can be configured as high-level trigger, low-level trigger, rising edge trigger, and falling edge trigger. Query the interrupt flag bit of Px_STAT[n] can know the corresponding interrupt trigger port. In addition, each digital port interrupt can wake up the chip from Sleep mode/DeepSleep mode to Active mode.

## 8.2 Features

The general-purpose I/O supports the following features:

- Port input value/output value register supports FAST IO/AHB bus read and write
- Other registers support AHB bus interface read and write
- Analog function pins/digital common pins/digital function pins are multiplexed
- Support pull-up/pull-down/two levels driving/open drain output function selection
- Support interrupt in Active mode/Sleep mode/DeepSleep mode
- Support high level/low level/rising edge/falling edge trigger interrupt
- Support bit set, bit clear, bit set clear function
- Part ports support input filter function

**Note**

For the introduction of FAST IO, please refer to "ARM Cortex-M0+_IntegrationAndImplementationManual.pdf".


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

## 8.3 Functional Description

### 8.3.1 Port Configuration Function

Each port can be configured as an analog port or a digital port through the configuration register (**PxADS**) according to system requirements. When PxADS is 1, the port is configured as an analog port, and when PxADS is 0, the port is configured as a digital port.

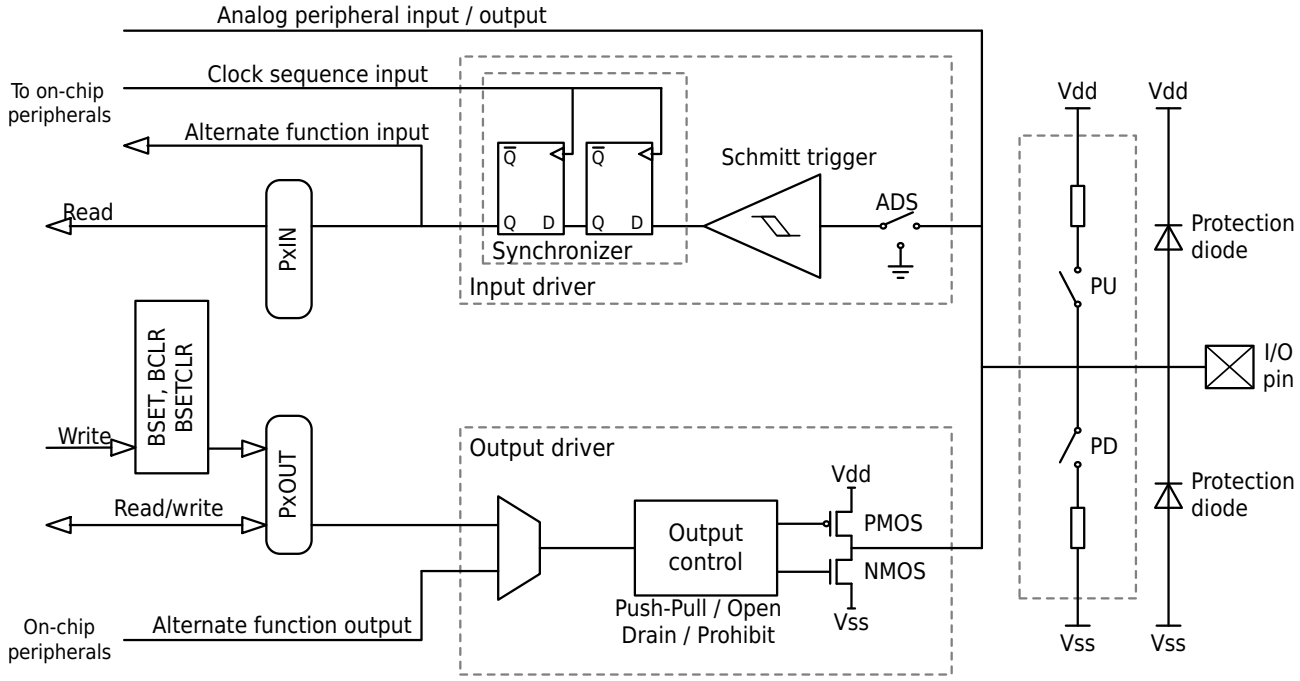



**Figure 8-1 Port Circuit Diagram**

**Note**

The output/input data register (PxOUT/PxIN) supports AHB/FAST IO bus access, other registers support AHB bus access.

When the port is configured as a digital port, it can accept digital general-purpose input and output signals (set the Pxy_SEL register to 0), and accept the input and output signals of various functional modules (such as SPI, USART, I2C, Timer, etc.) through the configuration register Pxy_SEL, for details, refer to the port digital multiplexing function module.

The corresponding registers can also be configured to achieve the following features:

- Internal pull-up (PxPU)/pull-down (PxPD)

  The pull-up register (PxPU) and the pull-down register (PxPD) correspond to the port pull-up enable and port pull-down enable respectively. When the corresponding bit is 1, set the corresponding bit pin pull-up/pull-down enable, When it is 0, the pull-up/pull-down of the corresponding bit pin is prohibited.

- Two levels driving output (PxDR)

  The driving capability can be changed through the PxDR register. When PxDR is 1, it is low driving capability, and when PxDR is 0, it is high driving capability.

- Open-drain output (PxOD)

  Set the pin output state through the PxOD register. When PxOD is '1', the port open-drain output is enabled, and when it is '0', the port push-pull output is enabled. When the open-drain pin is not


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

connected to an external pull-up resistor, it can only output low level. If it needs to output high level at the same time, a pull-up resistor is required.

- Direction selection (PxDIR)

  Used to set the direction of the port pin. When PxDIR is 0, the port is output, and when PxDIR is 1, the port is input.

- Input level status (PxIN)

  The synchronized pin level can be obtained by reading the PxIN register. When PxIN is 1, the pin is high level, and when PxIN is 0, the pin is low level.

- Output high and low level selection (PxOUT)

  When the port pin is configured as an output, if PxOUT is 1, the port pin output is high level, if PxOUT is 0, the output is low level.

- Bit set (PxBSET), bit cleared (PxBCLR), bit set cleared (PxBSETCLR)

  Bit set and bit cleared are applicable to setting the bit that the user wants to change to obtain the corresponding value without changing the value of other bits. When a bit is set, the corresponding value is set to 1, and when a bit is cleared, the corresponding value is set to 0.

**Note**

The above features are invalid when configured as an analog port.

The relationship between port states and register configuration is as follows:


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Table 8-1 Port Status Truth Table

| IO State | IO Direction | PxADS | PxDIR | PxOUT | PxIN | PxBSET | PxBCLR | PxBSETCLR | PxPU | PxPD | PxOD | PxDR | Pxy_SEL |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Simulation | Input/Output | 1 | W | W | 0 | W | W | W | W | W | W | W | W |
| Floating | Input | 0 | 1 | W | X | W | W | W | 0 | 0 | W | W | 0 |
| Pull-down | Input | 0 | 1 | W | 0 | W | W | W | 0 | 1 | W | W | 0 |
| Pull-up | Input | 0 | 1 | W | 1 | W | W | W | 1 | 0 | W | W | 0 |
| Pull-up | Input | 0 | 1 | W | 1 | W | W | W | 1 | 1 | W | W | 0 |
| 1 | Input | 0 | 1 | W | 1 | W | W | W | W | W | W | W | 0 |
| 0 | Input | 0 | 1 | W | 0 | W | W | W | W | W | W | W | 0 |
| 1 | Output | 0 | 0 | 1 | 1 | 0 | 0 | 0 | W | W | 0 | W | 0 |
| 0 | Output | 0 | 0 | 0 | 0 | 0 | 0 | 0 | W | W | 0 | W | 0 |
| 1 | Output | 0 | 0 | W | 1 | 1 | 0 | 0 | W | W | 0 | W | 0 |
| 0 | Output | 0 | 0 | W | 0 | 0 | 1 | 0 | W | W | 0 | W | 0 |
| 1 (SET) | Output | 0 | 0 | W | 1 (SET) | 0 | 0 | 1 | W | W | 0 | W | 0 |
| 0 (CLR) | Output | 0 | 0 | W | 0 (CLR) | 0 | 0 | 1 | W | W | 0 | W | 0 |
| 0 | Output | 0 | 0 | 0 | 0 | 0 | 0 | 0 | W | W | 1 | W | 0 |
| Z | Output | 0 | 0 | 1 | X | 0 | 0 | 0 | 0 | 0 | 1 | W | 0 |
| 0 | Output | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | W | 0 |
| 1 | Output | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | W | 0 |
| 1 | Output | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | W | 0 |
| 0-Logic low, 1-Logic high, W-Whatever 0 or 1, X-unknow state, Z-high impedance | | | | | | | | | | | | | |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 8.3.2 Port Write

The port input value/output value register (PxIN/PxOUT) supports reading and writing of the AHB bus and the FAST IO bus (controlled by the register GPIO_CTRL2.AHB_SEL bit), and only supports reading and writing of the AHB bus for other registers. For these two different buses, the system clock (HCLK) has different processing cycles for these two buses. The following two figures show the fastest timing for two bus port flipping:

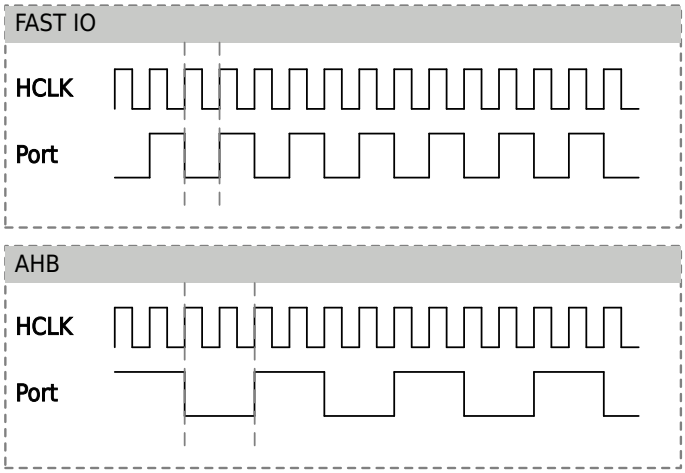



Figure 8-2 AHB/FASTIO Bus Port Changes with System Clock

For the AHB bus, every two HCLK cycles, the IO is flipped once, and for the FAST IO bus, every HCLK cycle, the IO is flipped once.

### 8.3.3 Port Read

Each port can get the port pin level by reading the PxIN register. As Port Circuit Diagram shown, each bit of the PxIN register and the latch in front of it form a synchronizer to avoid the signal instability caused by the pin level change in a short time when the system clock state changes, but it also introduces delay. The synchronization diagram for reading port pin data is as follows:

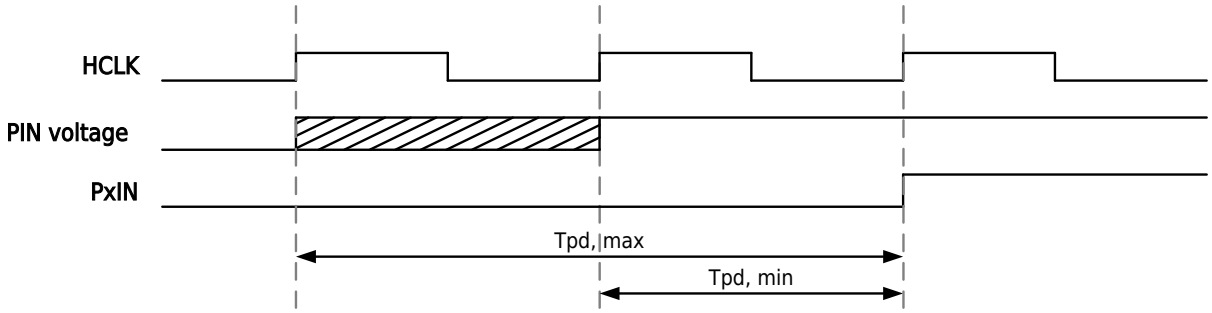



Figure 8-3 Port Pin Data Read Synchronization Diagram

During the clock period after the rising edge of the system clock, the pin level signal will be latched in the internal register, as shown in the shaded part, after the next rising edge of the system clock, the stable pin level signal can be read. Then, when the rising edge of the system clock, the data is latched into the PxIN register. The signal replacement delay Tpd is 1-2 system clocks.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

**Note**

Handling of unconnected pins: If there are unconnected pins during use, in order to avoid floating current consumption caused by pins not having a certain level in other digital input enable modes, it is recommended to assign a certain level to the pin.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 8.3.4 Port Digital Multiplexing Function

Port digital multiplexing is one of the main functions of the general-purpose I/O. Through the configuration register, the port can be flexibly configured as a digital general-purpose port/digital function port.

The Pxy_SEL register is used for digital general-purpose port/digital function port switching, and each port can be independently configured as a functional port required by the system.

**Table 8-2 Port Multiplexing Table**

| AF0 | AF1 | AF2 | AF3 | AF4 | AF5 | AF6 | AF7 |
|---|---|---|---|---|---|---|---|
| PA00 | USART1_CTS | LPUART1_TXD | TIM0_ETR | VC0_OUT | TIM1_CHA | TIM3_ETR | TIM0_CHA |
| PA01 | USART1_RTS | LPUART1_RXD | TIM0_CHB | TIM1_ETR | TIM1_CHB | HCLK_OUT | SPI1_MOSI |
| PA02 | USART1_TXD | TIM0_CHA | VC1_OUT | TIM1_CHA | TIM2_CHA | PCLK_OUT | SPI1_MISO |
| PA03 | USART1_RXD | TIM0_GATE | TIM1_CHB | TIM2_CHB | SPI1_CS | TIM3_CH1A | - |
| PA04 | SPI0_CS | USART1_TXD | LPUART1_RTS | TIM2_ETR | - | LVD_OUT | TIM3_CH2B |
| PA05 | SPI0_SCK | TIM0_ETR | LPUART1_CTS | TIM0_CHA | USART1_RTS | XTL_OUT | XTH_OUT |
| PA06 | SPI0_MISO | LPTIM0_TOG | TIM3_BK | TIM1_CHA | VC0_OUT | TIM3_GATE | LPUART0_CTS |
| PA07 | SPI0_MOSI | LPTIM0_TOGN | HCLK_OUT | TIM3_CH0B | TIM2_CHA | VC1_OUT | LPUART0_RXD |
| PA08 | USART0_TXD | TIM3_CH0A | CTRIM_ETRTOG | LPTIM1_ETR | TIM1_GATE | - | TIM3_BK |
| PA09 | USART0_TXD | TIM3_CH1A | TIM0_BK | I2C0_SCL | LPUART0_TXD | HCLK_OUT | TIM3_CH0A |
| PA10 | USART0_RXD | TIM3_CH2A | TIM2_BK | I2C0_SDA | TIM2_GATE | PCLK_OUT | TIM0_CHB |
| PA11 | USART0_CTS | TIM3_GATE | I2C1_SCL | HSI2C_SCL | VC0_OUT | SPI0_MISO | CTRIM_ETRTOG |
| PA12 | USART0_RTS | TIM3_ETR | I2C1_SDA | HSI2C_SDA | VC1_OUT | SPI0_MOSI | - |
| PA13 | IR_OUT | USART0_RXD | LVD_OUT | TIM3_ETR | RTC_1HZ | I2C1_SDA | USART1_SCK |
| PA14 | USART1_TXD | USART0_TXD | TIM3_CH2A | LVD_OUT | RCH_OUT | RCL_OUT | I2C1_SCL |
| PA15 | SPI0_CS | USART1_RXD | LPUART1_RTS | TIM0_ETR | TIM0_CHA | TIM3_CH1A | USART0_SCK |
| PB00 | - | TIM3_CH1B | LPUART0_TXD | LPTIM1_TOG | RCH_OUT | RCL_OUT | - |
| PB01 | - | PCLK_OUT | TIM3_CH2B | - | LPUART0_RTS | LPTIM1_ETR | TCLK_OUT |
| PB02 | LPTIM0_TOG | - | LPUART1_TXD | - | TIM1_BK | TIM0_BK | TIM2_BK |
| PB03 | SPI0_SCK | TIM0_CHB | TIM1_GATE | TIM3_CH0A | LPTIM0_GATE | XTL_OUT | XTH_OUT |
| PB04 | SPI0_MISO | - | TIM2_BK | USART0_CTS | TIM2_GATE | TIM3_CH0B | LPTIM0_ETR |
| PB05 | SPI0_MOSI | HSI2C_SCL | TIM1_BK | - | LPTIM0_GATE | - | USART0_RTS |
| PB06 | I2C0_SCL | USART0_TXD | TIM1_CHB | TIM0_CHA | LPTIM0_ETR | TIM3_CH0A | LPTIM0_TOG |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| AF0 | AF1 | AF2 | AF3 | AF4 | AF5 | AF6 | AF7 |
|---|---|---|---|---|---|---|---|
| PB07 | I2C0_SDA | USART0_RXD | TIM2_CHB | LPUART1_CTS | TIM0_CHB | LPTIM0_TOGN | - |
| PB08 | I2C0_SCL | TIM1_CHA | USART1_SCK | TIM2_CHA | TIM0_GATE | TIM3_CH2A | USART0_TXD |
| PB09 | I2C0_SDA | IR_OUT | SPI1_CS | TIM2_CHA | HSI2C_SDA | TIM2_CHB | USART0_RXD |
| PB10 | I2C1_SCL | SPI1_SCK | TIM1_CHA | LPUART0_TXD | TIM3_CH1A | LPUART1_RTS | USART1_RTS |
| PB11 | I2C1_SDA | TIM1_CHB | LPUART0_RXD | TIM2_GATE | - | LPUART1_CTS | USART1_CTS |
| PB12 | SPI1_CS | TIM3_BK | LPUART0_TXD | TIM0_BK | USART0_SCK | LPUART0_RTS | - |
| PB13 | SPI1_SCK | I2C1_SCL | TIM3_CH0B | LPUART0_CTS | TIM1_CHA | TIM1_GATE | - |
| PB14 | SPI1_MISO | I2C1_SDA | TIM3_CH1B | TIM0_CHA | RTC_1HZ | LPUART0_RTS | TIM1_BK |
| PB15 | SPI1_MOSI | TIM3_CH2B | TIM0_CHB | TIM0_GATE | - | - | LPUART1_RXD |
| PC00 | LPTIM0_GATE | - | USART1_CTS | HSI2C_SCL | - | - | - |
| PC01 | LPTIM0_TOG | - | USART1_RTS | HSI2C_SDA | - | - | - |
| PC02 | SPI1_MISO | LPTIM0_TOGN | - | - | - | - | - |
| PC03 | SPI1_MOSI | LPTIM0_ETR | LPTIM0_TOGN | - | - | - | - |
| PC04 | LPUART0_TXD | TIM2_ETR | IR_OUT | - | - | - | - |
| PC05 | LPUART0_RXD | - | - | - | - | - | - |
| PC06 | - | - | TIM2_CHA | LPTIM1_GATE | - | LPUART2_RXD | - |
| PC07 | - | - | TIM2_CHB | LPTIM1_ETR | - | LPUART2_TXD | - |
| PC08 | - | - | TIM2_ETR | LPTIM1_TOG | - | LPUART2_CTS | - |
| PC09 | - | - | TIM1_ETR | LPTIM1_TOGN | - | LPUART2_RTS | - |
| PC10 | LPUART1_TXD | LPUART0_TXD | - | - | - | - | - |
| PC11 | LPUART1_RXD | LPUART0_RXD | - | - | - | - | - |
| PC12 | LPUART0_TXD | LPUART1_TXD | - | - | - | - | - |
| PC13 | RTC_TAMP | RTC_1HZ | TIM3_CH1B | - | - | - | - |
| PC14 | - | USART0_SCK | - | LPTIM1_TOGN | - | - | - |
| PC15 | - | USART1_SCK | - | LPTIM1_GATE | - | - | - |
| PD00 | I2C0_SDA | CTRIM_ETRTOG | USART1_TXD | - | - | LPUART2_RXD | - |
| PD01 | I2C0_SCL | - | USART1_RXD | - | - | LPUART2_TXD | - |
| PD02 | - | LPUART0_RTS | TIM1_ETR | HSI2C_SCL | - | - | - |
| PD03 | - | - | - | - | - | - | - |
| PD04 | - | - | LPTIM1_TOGN | - | - | - | - |
| PD05 | USART0_SCK | - | LPTIM1_TOG | HSI2C_SDA | - | - | - |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| AF0 | AF1 | AF2 | AF3 | AF4 | AF5 | AF6 | AF7 |
|---|---|---|---|---|---|---|---|
| PD06 | I2C1_SCL | LPUART1_CTS | USART0_CTS | LPTIM1_GATE | - | LPUART2_CTS | - |
| PD07 | I2C1_SDA | LPUART1_RTS | USART0_RTS | LPTIM1_TOGN | - | LPUART2_RTS | - |

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 8.3.5 Port Interrupt Function

Each digital general-purpose port can be interrupted by an external signal source. The external signal source can be four types of signals: high level/low level/rising edge/falling edge, and the corresponding interrupt enable registers are High Level Interrupts Enable Register (PxHIE)/Low Level Interrupt Enable Register (PxLIE)/Rising Edge Interrupt Enable Register (PxRIE)/Falling Edge Interrupt Enable Register (PxFIE). When an interrupt is triggered, it can be determined which port triggered the interrupt by querying the Interrupt Status Register (Px_STAT), and the corresponding interrupt status flag bit can be cleared by clearing the Interrupt Clear Register (Px_ICLR).

### 8.3.6 Port Input Filter Function

PA00, PA08, PB00 and PB08 support digital input filter function. If the filter function is enabled, the MCU receives the filtered PAD input signal; otherwise, the MCU still receives the original PAD input signal. Both the filter clock and division factor can be configured via the Filter Configuration Register (GPIO_FLTCFG). The timing diagram of the filtering function:

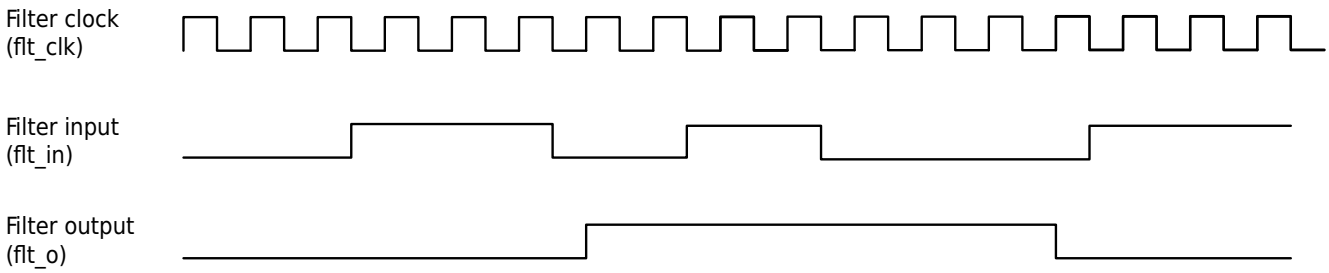



Figure 8-4 GPIO Port Input Filter Function Timing Diagram

**Notice**

If low-power wakeup is required, disable the filter function or switch the filter clock to RC256K.

### 8.3.7 Port Reset Status

During reset and before GPIO registers are configured after reset completion, except for the following I/O ports, all other ports are configured in analog port state; the status of special pins is as follows:

- SWCLK (PA14): Digital input pull-up
- SWDIO (PA13): Digital input pull-up
- PA10: Digital input pull-down
- PA11: Digital input

### 8.3.8 HSI2C Communication Level Selection Function

The HSI2C interface supports level signal communication at different voltages, that is, low voltage level recognition. And only when PAD selects the HSI2C function, this function is supported. Communication level voltage recognition selection is achieved by configuring GPIO_CTRL1.HSI2C_LSEL. When GPIO_CTRL1.HSI2C_LSEL is set to 0, HSI2C and GPIO input levels are recognized normally; When GPIO_CTRL1.HSI2C_LSEL is set to 1, HSI2C and GPIO input low-voltage levels are recognized.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

## 8.4 Operation Example

### 8.4.1 Port Multiplexing Configured as Analog Port Operation Flow

#### Procedure

Set the register PxADS[n] to 1.

### 8.4.2 Port Multiplexing Configured as Digital Universal Port Operation Flow

#### Procedure

**Step1.** Set the register PxADS[n] to 0.

**Step2.** Set the register Pxy_SEL to 0.

**Step3.** Set the register PxDIR[n] to 1: the port direction is input, and the CPU can read the port status PxIN[n].

**Step4.** Set the register PxDIR[n] to 0: the port direction is output.

**Step5.** Set the register PxOUT[n] to 1: the port outputs a high level.

**Step6.** Set the register PxOUT[n] to 0: the port outputs a low level.

### 8.4.3 Port Multiplexing Configured as Digital Function Port Operation Flow

#### Procedure

**Step1.** Set the register PxADS[n] to 0.

**Step2.** According to the system requirement, set the register Pxy_SEL to 1-7, for specific configurations,refer to the Port Multiplexing Table.

**Step3.** According to the system requirement, set the register PxDIR[n].

**Step4.** According to the system requirement, set the register PxPU[n]/PxPD[n]/PxOD[n].

### 8.4.4 Port Multiplexing Configured as Debug Test Port Operation Flow

#### Context

For specific operations, please refer to the relevant introduction in the "SWD Debugging Interface" chapter.

### 8.4.5 Port Multiplexing Configured as Infrared Output Signal Operation Flow

#### Context

Ports PA13, PB09, PC04 can modulate the internal clock signal with a frequency of 38.4kHz into infrared signal output.

> **Note**
> Carrier output is only generated when operating with the 38.4kHz RCL clock enabled.

#### Procedure

**Step1.** Set register PAADS[13]/PBADS[9]/PCADS[4] to 0.

**Step2.** Set register PA13_SEL = 1/PB09_SEL = 2/PC04_SEL = 3.

**Step3.** Set register PADIR[13] = 0/PBDIR[9] = 0/PCDIR[4] = 0, port direction is output.

**Step4.** Set the register GPIO_CTRL1[14] to select the output polarity of the infrared signal.

**Step5.** Set register PAOUT[13] = 1/PBOUT[9] = 1/PCOUT[4] = 1, select the gated infrared signal output.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 8.4.6 Port High Level Interrupt Operation Flow

#### Procedure

**Step1.** Set the register PxADS[n] to 0.

**Step2.** Set the register Pxy_SEL to 0.

**Step3.** Set the register PxDIR[n] to 1.

**Step4.** Set the register PxHIE[n] to 1.

**Step5.** Read the register Px_STAT[n] after the interrupt is triggered.

**Step6.** Set the register Px_ICLR[n] to 0 to clear the register Px_STAT[n].

### 8.4.7 Port Low Level Interrupt Operation Flow

#### Procedure

**Step1.** Set the register PxADS[n] to 0.

**Step2.** Set the register Pxy_SEL to 0.

**Step3.** Set the register PxDIR[n] to 1.

**Step4.** Set the register PxLIE[n] to 1.

**Step5.** Read the register Px_STAT[n] after the interrupt is triggered.

**Step6.** Set the register Px_ICLR[n] to 0 to clear the register Px_STAT[n].

### 8.4.8 Port Rising Edge Interrupt Operation Flow

#### Procedure

**Step1.** Set the register PxADS[n] to 0.

**Step2.** Set the register Pxy_SEL to 0.

**Step3.** Set the register PxDIR[n] to 1.

**Step4.** Set the register PxRIE[n] to 1.

**Step5.** Read the register Px_STAT[n] after the interrupt is triggered.

**Step6.** Set the register Px_ICLR[n] to 0 to clear the register Px_STAT[n].

### 8.4.9 Port Falling Edge Interrupt Operation Flow

#### Procedure

**Step1.** Set the register PxADS[n] to 0.

**Step2.** Set the register Pxy_SEL to 0.

**Step3.** Set the register PxDIR[n] to 1.

**Step4.** Set the register PxFIE[n] to 1.

**Step5.** Read the register Px_STAT[n] after the interrupt is triggered.

**Step6.** Set the register Px_ICLR[n] to 0 to clear the register Px_STAT[n].

### 8.4.10 Port Pull-Up Enable Configuration Operation Flow

#### Procedure

Set the register PxPU[n] to 1.

### 8.4.11 Port Pull-Down Enable Configuration Operation Flow

#### Context

When PxPU[n] and PxPD[n] are set to 1 at the same time, PxPU[n] has higher priority and PxPD[n] is invalid.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

#### Procedure

**Step1.** Set the register PxPU[n] to 0.

**Step2.** Set the register PxPD[n] to 1.

### 8.4.12 Port Enhancement Driver Configuration Operation Flow

#### Procedure

Set the register PxDR[n] to 0.

### 8.4.13 Port Open-Drain Output Configuration Operation Flow

#### Procedure

Set the register PxOD[n] to 1.

### 8.4.14 Port Bit Setting Operation Flow

#### Procedure

Set the register PxBSET[n] to 1.

### 8.4.15 Port Bit Clearing Operation Flow

#### Procedure

Set the register PxBCLR[n] to 1.

### 8.4.16 Port Bit Setting and Clearing Operation Flow

#### Procedure

Set the register PxBSETCLR[n] to 1.

### 8.4.17 Port Input Filtering Operation Flow

#### Procedure

**Step1.** Set the register GPIO_FLTCFG.FCLK_SEL to 0 to select HCLK as the filter clock.

**Step2.** Set the register GPIO_FLTCFG.FCLK_DIV to 0b01 to select the filter clock frequency division factor to 4.

**Step3.** Set the register GPIO_FLTCFG.PA0FLT_EN to 1 to enable the filtering function for PA00.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

## 8.5 Register Description

### 8.5.1 Register List

Base address: 0x40020C00

Table 8-3 General-Purpose I/O (GPIO) Register Offset

| Offset | Register | Description |
|---|---|---|
| 0x00+0x4*y | PAy_SEL | Port PAy Function Configuration Register (y=0-15) |
| 0x40+0x4*y | PBy_SEL | Port PBy Function Configuration Register (y=0-15) |
| 0x80+0x4*y | PCy_SEL | Port PCy Function Configuration Register (y=0-15) |
| 0xC0+0x4*y | PDy_SEL | Port PDy Function Configuration Register (y=0-7) |
| 0x100 | PADIR | Port PA Input/Output Configuration Register |
| 0x104 | PAIN | Port PA Input Value Register |
| 0x108 | PAOUT | Port PA Output Value Configuration Register |
| 0x10C | PAADS | Port PA Digital-to-Analog Configuration Register |
| 0x110 | PABSET | Port PA Bit Set Register |
| 0x114 | PABCLR | Port PA Bit Clear Register |
| 0x118 | PABSETCLR | Port PA Bit Set Clear Register |
| 0x11C | PADR | Port PA Driving Capability Configuration Register |
| 0x120 | PAPU | Port PA Pull-Up Enable Configuration Register |
| 0x124 | PAPD | Port PA Pull-Down Enable Configuration Register |
| 0x12C | PAOD | Port PA Open-Drain Output Configuration Register |
| 0x130 | PAHIE | Port PA High Level Interrupt Enable Configuration Register |
| 0x134 | PALIE | Port PA Low Level Interrupt Enable Configuration Register |
| 0x138 | PARIE | Port PA Rising Edge Interrupt Enable Configuration Register |
| 0x13C | PAFIE | Port PA Falling Edge Interrupt Enable Configuration Register |
| 0x200 | PA_STAT | Port PA Interrupt Status Register |
| 0x210 | PA_ICLR | Port PA Interrupt Clear Register |
| 0x140 | PBDIR | Port PB Input and Output Configuration Register |
| 0x144 | PBIN | Port PB Input Value Register |
| 0x148 | PBOUT | Port PB Output Value Configuration Register |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Offset | Register | Description |
|---|---|---|
| 0x14C | PBADS | Port PB Digital-to-Analog Configuration Register |
| 0x150 | PBBSET | Port PB Bit Set Register |
| 0x154 | PBBCLR | Port PB Bit Clear Register |
| 0x158 | PBBSETCLR | Port PB Bit Set Clear Register |
| 0x15C | PBDR | Port PB Driving Capability Configuration Register |
| 0x160 | PBPU | Port PB Pull-Up Enable Configuration Register |
| 0x164 | PBPD | Port PB Pull-Down Enable Configuration Register |
| 0x16C | PBOD | Port PB Open-Drain Output Configuration Register |
| 0x170 | PBHIE | Port PB High Level Interrupt Enable Configuration Register |
| 0x174 | PBLIE | Port PB Low Level Interrupt Enable Configuration Register |
| 0x178 | PBRIE | Port PB Rising Edge Interrupt Enable Configuration Register |
| 0x17C | PBFIE | Port PB Falling Edge Interrupt Enable Configuration Register |
| 0x240 | PB_STAT | Port PB Interrupt Status Register |
| 0x250 | PB_ICLR | Port PB Interrupt Clear Register |
| 0x180 | PCDIR | Port PC Input and Output Configuration Register |
| 0x184 | PCIN | Port PC Input Value Register |
| 0x188 | PCOUT | Port PC Output Value Configuration Register |
| 0x18C | PCADS | Port PC Digital-to-Analog Configuration Register |
| 0x190 | PCBSET | Port PC Bit Set Register |
| 0x194 | PCBCLR | Port PC Bit Clear Register |
| 0x198 | PCBSETCLR | Port PC Bit Set Clear Register |
| 0x19C | PCDR | Port PC Driving Capability Configuration Register |
| 0x1A0 | PCPU | Port PC Pull-Up Enable Configuration Register |
| 0x1A4 | PCPD | Port PC Pull-Down Enable Configuration Register |
| 0x1AC | PCOD | Port PC Open-Drain Output Configuration Register |
| 0x1B0 | PCHIE | Port PC High Level Interrupt Enable Configuration Register |
| 0x1B4 | PCLIE | Port PC Low Level Interrupt Enable Configuration Register |

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Offset | Register | Description |
|---|---|---|
| 0x1B8 | PCRIE | Port PC Rising Edge Interrupt Enable Configuration Register |
| 0x1BC | PCFIE | Port PC Falling Edge Interrupt Enable Configuration Register |
| 0x280 | PC_STAT | Port PC Interrupt Status Register |
| 0x290 | PC_ICLR | Port PC Interrupt Clear Register |
| 0x1C0 | PDDIR | Port PD Input and Output Configuration Register |
| 0x1C4 | PDIN | Port PD Input Value Register |
| 0x1C8 | PDOUT | Port PD Output Value Configuration Register |
| 0x1CC | PDADS | Port PD Digital-to-Analog Configuration Register |
| 0x1D0 | PDBSET | Port PD Bit Set Register |
| 0x1D4 | PDBCLR | Port PD Bit Clear Register |
| 0x1D8 | PDBSETCLR | Port PD Bit Set Clear Register |
| 0x1DC | PDDR | Port PD Driving Capability Configuration Register |
| 0x1E0 | PDPU | Port PD Pull-Up Enable Configuration Register |
| 0x1E4 | PDPD | Port PD Pull-Down Enable Configuration Register |
| 0x1EC | PDOD | Port PD Open-Drain Output Configuration Register |
| 0x1F0 | PDHIE | Port PD High Level Interrupt Enable Configuration Register |
| 0x1F4 | PDLIE | Port PD Low Level Interrupt Enable Configuration Register |
| 0x1F8 | PDRIE | Port PD Rising Edge Interrupt Enable Configuration Register |
| 0x1FC | PDFIE | Port PD Falling Edge Interrupt Enable Configuration Register |
| 0x2C0 | PD_STAT | Port PD Interrupt Status Register |
| 0x2D0 | PD_ICLR | Port PD Interrupt Clear Register |
| 0x304 | GPIO_CTRL1 | Port Alternate Function Configuration Register 1 |
| 0x308 | GPIO_CTRL2 | Port Alternate Function Configuration Register 2 |
| 0x30C | GPIO_TIMGS | Port Auxiliary Function Timer Gate Selection |
| 0x310 | GPIO_TIMES | Port Auxiliary Function Timer ETR selection |
| 0x314 | GPIO_TIMCPS0 | Port Auxiliary Function Timer Capture Input selection0 |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Offset | Register | Description |
| --- | --- | --- |
| 0x318 | GPIO_TIMCPS1 | Port Auxiliary Function Timer Capture Input selection 1 |
| 0x330 | GPIO_FLTCFG | Port Filter Configuration Register |

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

## 8.5.2 Port General Register

### 8.5.2.1 Port Px Input/Output Configuration Register (PxDIR) (x =A to D)

Offset: 0x100 (PA), 0x140 (PB), 0x180 (PC), 0x1C0 (PD)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| - | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x0000FFFF | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | PxDIR | | | | | | | | | | | | | | | |
| Access | | | | | | | | | | | | | | | | | RW | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:16* | *Reserved* | - | - |
| 15:0 | PxDIR | Port Px input/output configuration register<br>Corresponding to Px15-Px00:<br>● 0b0: Configured as output<br>● 0b1: Configured as input<br>**Note**<br>Each bit corresponds to one port, please refrain from configuring non-existent ports, for example: PxDIR[15] corresponds to port Px15. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 8.5.2.2 Port Px Input Value Register (PxIN) (x =A to D)

Offset: 0x104 (PA), 0x144 (PB), 0x184 (PC), 0x1C4 (PD)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| - | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | - | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | PxIN | | | | | | | | | | | | | | | |
| Access | | | | | | | | | | | | | | | | | RO | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:16* | *Reserved* | - | - |
| 15:0 | PxIN | Port Px input value register<br>Corresponding to Px15-Px00:<br>● 0b0: Input low level<br>● 0b1: Input high level<br>**Note**<br>Each bit corresponds to one port, please refrain from configuring non-existent ports. | RO |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 8.5.2.3 Port Px Output Value Configuration Register (PxOUT) (x=A to D)

Offset: 0x108 (PA), 0x148 (PB), 0x188 (PC), 0x1C8 (PD)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| - | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0xXXXXXXXX | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | PxOUT | | | | | | | | | | | | | | | |
| Access | | | | | | | | | | | | | | | | | RW | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:16* | *Reserved* | - | - |
| 15:0 | PxOUT | Port Px output value register<br>Corresponding to Px15-Px00:<br>● 0b0: Output low level<br>● 0b1: Output high level, if it is configured as an open-drain output, an external pull-up resistor is required to pull it high<br>**Note**<br>Each bit corresponds to one port, please refrain from configuring non-existent ports. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 8.5.2.4 Port Px Digital-to-Analog Configuration Register (PxADS) (x=A to D)

Offset: 0x10C (PA), 0x14C (PB), 0x18C (PC), 0x1CC (PD)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| - | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x000093FF(PA);0x0000FFFF(PB/PC);0x000000FF(PD) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | PxADS | | | | | | | | | | | | | | | |
| Access | | | | | | | | | | | | | | | | | RW | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:16* | *Reserved* | - | - |
| 15:0 | PxADS | Port Px digital-to-analog configuration register<br>Corresponding to Px15-Px00:<br>● 0b0: Configured as a digital port<br>● 0b1: Configured as an analog port<br>**Note**<br>Each bit corresponds to one port, please refrain from configuring non-existent ports. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 8.5.2.5 Port Px Bit Set Register (PxBSET) (x=A to D)

Offset: 0x110 (PA), 0x150 (PB), 0x190 (PC), 0x1D0 (PD)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| - | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | PxBSET | | | | | | | | | | | | | | | |
| Access | | | | | | | | | | | | | | | | | WO | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:16* | *Reserved* | - | - |
| 15:0 | PxBSET | Port Px bit set register<br>Corresponding to Px15-Px00, writing 1 sets the bit, writing 0 has no effect on the port state.<br>**Note**<br>Each bit corresponds to one port, please refrain from configuring non-existent ports. | WO |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 8.5.2.6 Port Px Bit Clear Register (PxBCLR) (x=A to D)

Offset: 0x114 (PA), 0x154 (PB), 0x194 (PC), 0x1D4 (PD)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| - | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | PxBCLR | | | | | | | | | | | | | | | |
| Access | | | | | | | | | | | | | | | | | WO | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:16* | *Reserved* | - | - |
| 15:0 | PxBCLR | Port Px bit clear register<br>Corresponding to Px15-Px00, writing 1 clears the bit, writing 0 has no effect on the port state.<br>**Note**<br>Each bit corresponds to one port, please refrain from configuring non-existent ports. | WO |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 8.5.2.7 Port Px Bit Set Clear Register (PxBSETCLR) (x=A to D)

Offset: 0x118 (PA), 0x158 (PB), 0x198 (PC), 0x1D8 (PD)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| - | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | PxBSET | | | | | | | | | | | | | | | | PxBCLR | | | | | | | | | | | | | | | |
| Access | WO | | | | | | | | | | | | | | | | WO | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 31:16 | PxBSET | Port Px bit set register<br>Corresponding to Px15-Px00, writing 1 sets the bit, writing 0 has no effect on the port state.<br>**Note**<br>Each bit corresponds to one port, please refrain from configuring non-existent ports. | WO |
| 15:0 | PxBCLR | Port Px bit clear register<br>Corresponding to Px15-Px00, writing 1 clears the bit, writing 0 has no effect on the port state.<br>**Note**<br>Each bit corresponds to one port, please refrain from configuring non-existent ports. | WO |

**Note**

When the same bit of PxBSET and PxBCLR is both set to 1, PxBCLR has higher priority, the port is cleared.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 8.5.2.8 Port Px Pull-Up Enable Configuration Register (PxPU) (x=A to D)

Offset: 0x120 (PA), 0x160 (PB), 0x1A0 (PC), 0x1E0 (PD)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| - | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00006000(PA); 0x0000000(PB/PC/PD) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | PxPU | | | | | | | | | | | | | | | |
| Access | | | | | | | | | | | | | | | | | RW | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:16* | *Reserved* | - | - |
| 15:0 | PxPU | Port Px pull-up enable configuration register<br>Corresponding to Px15-Px00:<br>● 0b0: Disable<br>● 0b1: Enable<br>**Note**<br>Each bit corresponds to one port, please refrain from configuring non-existent ports. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 8.5.2.9 Port Px Driving Capability Configuration Register (PxDR) (x=A to D)

Offset: 0x11C (PA), 0x15C (PB), 0x19C (PC), 0x1DC (PD)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| - | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | PxDR | | | | | | | | | | | | | | | |
| Access | | | | | | | | | | | | | | | | | RW | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:16* | *Reserved* | - | - |
| 15:0 | PxDR | Port Px driving capability configuration register<br>Corresponding to Px15-Px00:<br>● 0b0: High driving capability<br>● 0b1: Low driving capability<br>**Note**<br>Each bit corresponds to one port, please refrain from configuring non-existent ports. | RW |

### 8.5.2.10 Port Px Pull-Down Enable Configuration Register (PxPD) (x=A to D)

Offset: 0x124 (PA), 0x164 (PB), 0x1A4 (PC), 0x1E4 (PD)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| - | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000400(PA);0x0000000(PB/PC/PD) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | PxPD | | | | | | | | | | | | | | | |
| Access | | | | | | | | | | | | | | | | | RW | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:16* | *Reserved* | - | - |
| 15:0 | PxPD | Port Px pull-down enable configuration register<br>Corresponding to Px15-Px00:<br>● 0b0: Disable<br>● 0b1: Enable<br>**Note**<br>Each bit corresponds to one port, please refrain from configuring non-existent ports. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 8.5.2.11 Port Px Open-Drain Output Configuration Register (PxOD) (x=A to D)

Offset: 0x12C (PA), 0x16C (PB), 0x1AC (PC), 0x1EC (PD)

| xOffset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| - | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | PxOD | | | | | | | | | | | | | | | |
| Access | | | | | | | | | | | | | | | | | RW | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:16* | *Reserved* | - | - |
| 15:0 | PxOD | Port Px open-drain output configuration register<br>Corresponding to Px15-Px00:<br>● 0b0: Set the port output mode to push-pull output<br>● 0b1: Set the port output mode to open-drain output<br>**Note**<br>Each bit corresponds to one port, please refrain from configuring non-existent ports. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 8.5.2.12 Port Px High Level Interrupt Enable Configuration Register (PxHIE) (x=A to D)

Offset: 0x130 (PA), 0x170 (PB), 0x1B0 (PC), 0x1F0 (PD)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| - | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | PxHIE | | | | | | | | | | | | | | | |
| Access | | | | | | | | | | | | | | | | | RW | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:16* | *Reserved* | - | - |
| 15:0 | PxHIE | Port Px high level interrupt enable configuration register<br>Corresponding to Px15-Px00:<br>● 0b0: Disable<br>● 0b1: Enable<br>**Note**<br>Each bit corresponds to one port, please refrain from configuring non-existent ports. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 8.5.2.13 Port Px Low Level Interrupt Enable Configuration Register (PxLIE) (x=A to D)

Offset: 0x134 (PA), 0x174 (PB), 0x1B4 (PC), 0x1F4 (PD)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| - | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | PxLIE | | | | | | | | | | | | | | | |
| Access | | | | | | | | | | | | | | | | | RW | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:16* | *Reserved* | - | - |
| 15:0 | PxLIE | Port Px low level interrupt enable configuration register<br>Corresponding to Px15-Px00:<br>● 0b0: Disable<br>● 0b1: Enable<br>**Note**<br>Each bit corresponds to one port, please refrain from configuring non-existent ports. | RW |

### 8.5.2.14 Port Px Rising Edge Interrupt Enable Configuration Register (PxRIE) (x=A to D)

Offset: 0x138 (PA), 0x178 (PB), 0x1B8 (PC), 0x1F8 (PD)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| - | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | PxRIE | | | | | | | | | | | | | | | |
| Access | | | | | | | | | | | | | | | | | RW | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:16* | *Reserved* | - | - |
| 15:0 | PxRIE | Port Px rising edge interrupt enable configuration register<br>Corresponding to Px15-Px00:<br>● 0b0: Disable<br>● 0b1: Enable<br>**Note**<br>Each bit corresponds to one port, please refrain from configuring non-existent ports. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 8.5.2.15 Port Px Falling Edge Interrupt Enable Configuration Register (PxFIE) (x=A to D)

Offset: 0x13C (PA), 0x17C (PB), 0x1BC (PC), 0x1FC (PD)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| - | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | PxFIE | | | | | | | | | | | | | | | |
| Access | | | | | | | | | | | | | | | | | RW | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:16* | *Reserved* | - | - |
| 15:0 | PxFIE | Port Px falling edge interrupt enable configuration register<br>Corresponding to Px15-Px00:<br>● 0b0: Disable<br>● 0b1: Enable<br>**Note**<br>Each bit corresponds to one port, please refrain from configuring non-existent ports. | RW |

### 8.5.2.16 Port Px Interrupt Status Register (Px_STAT) (x=A to D)

Offset: 0x200 (PA), 0x240 (PB), 0x280 (PC), 0x2C0 (PD)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| - | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | Px_STAT | | | | | | | | | | | | | | | |
| Access | | | | | | | | | | | | | | | | | RO | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:16* | *Reserved* | - | - |
| 15:0 | Px_STAT | Port Px interrupt status register<br>Corresponding to Px15-Px00:<br>● 0b0: No interrupt trigger<br>● 0b1: Interrupt trigger<br>**Note**<br>Each bit corresponds to one port, please refrain from configuring non-existent ports. | RO |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 8.5.2.17 Port Px Interrupt Clear Register (Px_ICLR) (x=A to D)

Offset: 0x210 (PA), 0x250 (PB), 0x290 (PC), 0x2D0 (PD)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| - | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x0000FFFF | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | Px_ICLR | | | | | | | | | | | | | | | |
| Access | | | | | | | | | | | | | | | | | WO | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:16* | *Reserved* | - | - |
| 15:0 | Px_ICLR | Port Px interrupt clear register<br>Corresponding to Px15-Px00, write 0 to clear flag, write 1 has no effect; read always returns 1.<br>**Note**<br>Each bit corresponds to one port, please refrain from configuring non-existent ports. | WO |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

## 8.5.3 Port Auxiliary Register

### 8.5.3.1 Port Auxiliary Function Configuration Register 1 (GPIO_CTRL1)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x304 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | HSI2C_LSEL | IR_POL | HCLK_EN | PCLK_EN | HCLK_SEL | | PCLK_SEL | | Reserved | | | | | | | |
| Access | | | | | | | | | | | | | | | | | RW | RW | RW | RW | RW | | RW | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:16* | *Reserved* | - | - |
| 15 | HSI2C_LSEL | HSI2C I/O signal active level selection<br>● 0b0: Input level recognized normally<br>● 0b1: Input level low voltage recognition<br>**Note**<br>For specific values, refer to the "Datasheet". | RW |
| 14 | IR_POL | IR output polarity selection<br>● 0b0: Forward output<br>● 0b1: Inverted output | RW |
| 13 | HCLK_EN | HCLK output gating<br>● 0b0: Gating<br>● 0b1: Output | RW |
| 12 | PCLK_EN | PCLK output gating<br>● 0b0: Gating<br>● 0b1: Output | RW |
| 11:10 | HCLK_SEL | HCLK output frequency division selection<br>● 0b00: HCLK<br>● 0b01: HCLK/2<br>● 0b10: HCLK/4<br>● 0b11: HCLK/8 | RW |
| 9:8 | PCLK_SEL | PCLK output frequency division selection<br>● 0b00: PCLK<br>● 0b01: PCLK/2<br>● 0b10: PCLK/4<br>● 0b11: PCLK/8 | RW |
| *7:0* | *Reserved* | - | - |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 8.5.3.2 Port Auxiliary Function Configuration Register 2 (GPIO_CTRL2)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x308 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | AHB_SEL | Reserved | | | | | | | | | TCLK_DIV | | TCLK_SEL | | Reserved | | |
| Access | | | | | | | | | | | | | | | | | RW | | | | | | | | | | RW | | RW | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:16* | *Reserved* | - | - |
| 15 | AHB_SEL | AHB bus access mode<br>● 0b0: Use the FASTIO bus to access PxIN, PxOUT registers<br>● 0b1: Use the AHB bus to access PxIN, PxOUT registers | RW |
| *14:8* | *Reserved* | - | - |
| 7:6 | TCLK_DIV | TCLK output frequency division selection<br>● 0b00: TCLK<br>● 0b01: TCLK/2<br>● 0b10: TCLK/4<br>● 0b11: TCLK/8 | RW |
| 5:4 | TCLK_SEL | TCLK output signal source selection<br>● 0b00: RCH<br>● 0b01: RCL<br>● 0b10: XTH<br>● 0b11: XTL | RW |
| *3:0* | *Reserved* | - | - |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 8.5.3.3 Port Auxiliary Function Timer Gate Selection (GPIO_TIMGS)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x30C | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | LPTIM0_G | | | TIM3_G | | | TIM2_G | | | TIM1_G | | | TIM0_G | | |
| Access | | | | | | | | | | | | | | | | | | RW | | | RW | | | RW | | | RW | | | RW | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:15* | *Reserved* | - | - |
| 14:12 | LPTIM0_G | LPTIM0 timer GATE input selection<br>Parameter selection details (See Table Below) | RW |
| 11:9 | TIM3_G | TIM3 timer GATE input selection<br>Parameter selection details (See Table Below) | RW |
| 8:6 | TIM2_G | TIM2 timer GATE input selection<br>Parameter selection details (See Table Below) | RW |
| 5:3 | TIM1_G | TIM1 timer GATE input selection<br>Parameter selection details (See Table Below) | RW |
| 2:0 | TIM0_G | TIM0 timer GATE input selection<br>Parameter selection details (See Table Below) | RW |

| Value | LPTIM0_G | TIM3_G | TIM2_G | TIM1_G | TIM0_G |
|---|---|---|---|---|---|
| 0b000 | Pxy_SEL | Pxy_SEL | Pxy_SEL | Pxy_SEL | Pxy_SEL |
| 0b001 | LPUART0_RXD | USART0_RXD | USART0_RXD | LPUART0_RXD | USART0_RXD |
| 0b010 | LPUART1_RXD | USART1_RXD | USART1_RXD | LPUART1_RXD | USART1_RXD |
| 0b011 | VC0_OUT | - | VC0_OUT | VC0_OUT | VC0_OUT |
| 0b100 | VC1_OUT | LPUART2_RXD | VC1_OUT | VC1_OUT | VC1_OUT |
| 0b101 | PB03 | VC0_OUT | PA10 | PA08 | PA03 |
| 0b110 | PB05 | PA06 | PB04 | PB03 | PB08 |
| 0b111 | PC00 | PA11 | PB11 | PB13 | PB15 |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 8.5.3.4 Port Auxiliary Function Timer ETR Selection (GPIO_TIMES)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x310 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | LPTIM0_E | | | TIM3_E | | | TIM2_E | | | TIM1_E | | | TIM0_E | | |
| Access | | | | | | | | | | | | | | | | | | RW | | | RW | | | RW | | | RW | | | RW | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:15* | *Reserved* | - | - |
| 14:12 | LPTIM0_E | LPTimer0 timer ETR input selection<br>Parameter selection details (See Table Below) | RW |
| 11:9 | TIM3_E | TIM3 timer ETR input selection<br>Parameter selection details (See Table Below) | RW |
| 8:6 | TIM2_E | TIM2 timer ETR input selection<br>Parameter selection details (See Table Below) | RW |
| 5:3 | TIM1_E | TIM1 timer ETR input selection<br>Parameter selection details (See Table Below) | RW |
| 2:0 | TIM0_E | TIM0 timer ETR input selection<br>Parameter selection details (See Table Below) | RW |

| Value | LPTIM0_E | TIM3_E | TIM2_E | TIM1_E | TIM0_E |
|---|---|---|---|---|---|
| 0b000 | Pxy_SEL | Pxy_SEL | Pxy_SEL | Pxy_SEL | Pxy_SEL |
| 0b001 | - | USART0_RXD | LPUART0_RXD | USART0_RXD | LPUART0_RXD |
| 0b010 | LVD_OUT | USART1_RXD | LPUART1_RXD | USART1_RXD | LPUART1_RXD |
| 0b011 | VC0_OUT | VC1_OUT | VC0_OUT | VC1_OUT | VC0_OUT |
| 0b100 | VC1_OUT | - | - | LVD_OUT | LVD_OUT |
| 0b101 | PA08 | PA00 | PA04 | PA01 | PA00 |
| 0b110 | PB01 | PA12 | PC04 | PC09 | PA05 |
| 0b111 | PC07 | PA13 | PC08 | PD02 | PA15 |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 8.5.3.5 Port Auxiliary Function Timer Capture Input Selection 0 (GPIO_TIMCPS0)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x314 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | TIM3_CB | | TIM3_CA | | | TIM2_CA | | | TIM1_CA | | | TIM0_CA | | |
| Access | | | | | | | | | | | | | | | | | | | RW | | RW | | | RW | | | RW | | | RW | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:15* | *Reserved* | - | - |
| 14:12 | TIM3_CB | TIM3 timer CH0B input selection<br>Parameter selection details (See Table Below) | RW |
| 11:9 | TIM3_CA | TIM3 timer CH0A input selection<br>Parameter selection details (See Table Below) | RW |
| 8:6 | TIM2_CA | TIM2 timer CHA input selection<br>Parameter selection details (See Table Below) | RW |
| 5:3 | TIM1_CA | TIM1 timer CHA input selection<br>Parameter selection details (See Table Below) | RW |
| 2:0 | TIM0_CA | TIM0 timer CHA input selection<br>Parameter selection details (See Table Below) | RW |

| Value | TIM3_CB | TIM3_CA | TIM2_CA | TIM1_CA | TIM0_CA |
|---|---|---|---|---|---|
| 0b000 | Pxy_SEL | Pxy_SEL | Pxy_SEL | Pxy_SEL | Pxy_SEL |
| 0b001 | USART0_RXD | LPUART1_RXD | LPUART0_RXD | USART1_RXD | USART0_RXD |
| 0b010 | USART1_RXD | LPUART0_RXD | VC0_OUT | PA00 | PA00 |
| 0b011 | - | - | PA02 | PA02 | PA02 |
| 0b100 | VC1_OUT | VC0_OUT | PA07 | PA06 | PA05 |
| 0b101 | PA07 | PA08 | PB08 | PB08 | PA15 |
| 0b110 | PB04 | PB03 | PB09 | PB10 | PB06 |
| 0b111 | PB13 | PB06 | PC06 | PB13 | PB14 |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 8.5.3.6 Port Auxiliary Function Timer Capture Input Selection 1 (GPIO_TIMCPS1)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x318 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | LPTIM1_E | | | LPTIM1_G | | | Reserved | | | | | |
| Access | | | | | | | | | | | | | | | | | | | | | RW | | | RW | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:12* | *Reserved* | - | - |
| 11:9 | LPTIM1_E | LPTIM1 timer ETR input selection<br>Parameter selection details (See Table Below) | RW |
| 8:6 | LPTIM1_G | LPTIM1 timer GATE input selection<br>Parameter selection details (See Table Below) | RW |
| *5:0* | *Reserved* | - | - |

| Value | LPTIM1_E | LPTIM1_G |
|---|---|---|
| 0b000 | Pxy_SEL | Pxy_SEL |
| 0b001 | USART1_RXD | USART0_RXD |
| 0b010 | LPUART1_RXD | LPUART0_RXD |
| 0b011 | VC0_OUT | VC1_OUT |
| 0b100 | PC03 | - |
| 0b101 | LPTIM0_OV | PD5 |
| 0b110 | PB04 | - |
| 0b111 | PB06 | PC6 |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 8.5.3.7 Port Filter Configuration Register (GPIO_FLTCFG)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x330 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | FCLK_SEL | FCLK_DIV | | PA0FLT_EN | PA8FLT_EN | PB0FLT_EN | PB8FLT_EN | Reserved | | | |
| Access | | | | | | | | | | | | | | | | | | | | | | RW | RW | | RW | RW | RW | RW | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:11* | *Reserved* | - | - |
| 10 | FCLK_SEL | Filter clock source selection<br>● 0b0: HCLK<br>● 0b1: RC256K | RW |
| 9:8 | FCLK_DIV | Filter clock frequency division configuration<br>● 0b00: FCLK<br>● 0b01: FCLK/4<br>● 0b10: FCLK/16<br>● 0b11: FCLK/64 | RW |
| 7 | PA0FLT_EN | PA00 filter enable<br>● 0b0: Filter disable<br>● 0b1: Filter enable | RW |
| 6 | PA8FLT_EN | PA08 filter enable<br>● 0b0: Filter disable<br>● 0b1: Filter enable | RW |
| 5 | PB0FLT_EN | PB00 filter enable<br>● 0b0: Filter disable<br>● 0b1: Filter enable | RW |
| 4 | PB8FLT_EN | PB08 filter enable<br>● 0b0: Filter disable<br>● 0b1: Filter enable | RW |
| *3:0* | *Reserved* | - | - |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

## 8.5.4 Port Function Selection Register

### 8.5.4.1 Port Pxy Function Configuration Register (Pxy_SEL) (x =A to D)

Offset: 0x00+0x4*y (PA) (y= 0-15);

0x40+0x4*y (PB)(y=0-15);

0x80+0x4*y (PC)(y=0-15);

0xC0+0x4*y (PD)(y=0-7).

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| - | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | SEL | |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | RW | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:3* | *Reserved* | - | - |
| 2:0 | SEL | Port Pxy function selection<br>● 0b000: GPIO<br>● 0b001: AF1<br>● 0b010: AF2<br>● 0b011: AF3<br>● 0b100: AF4<br>● 0b101: AF5<br>● 0b110: AF6<br>● 0b111: AF7<br>**Note**<br>For the specific functions of AF1-AF7 please refer to the "Port Digital Multiplexing Function". | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

# 9 DMAC

## 9.1 Introduction

Direct memory access (DMA) is used to provide high-speed data transmission between peripherals and memory or between memory and memory. The CPU does not need to participate in the transmission process, so the CPU can perform other operations synchronously. The DMAC has 2 independent DMA channels, each channel is dedicated to managing requests from peripherals or memory access. There is also an arbiter to coordinate the priority of each DMA request.

## 9.2 Features

DMA controller supports the following features:

- 2 independent DMA channels, support priority configuration
- 3 data transfer widths: 8-bit, 16-bit, 32-bit
- 4 transmission modes: Software Block, Software Burst, Hardware Block, Hardware Burst
- 2 source address types: peripheral, memory
- 2 target address types: peripheral, memory
- 2 address change modes: fixed, auto-increment
- Transmission address addressing range: 0x00000000-0xFFFFFFFF
- The number of data blocks to be transmitted is configurable: 1-65536
- The size of the data block to be transmitted is configurable: 1-16
- Support address and transfer number reload function


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

## 9.3 Functional Description

### 9.3.1 Function Block Diagram

- DMAC

  DMAC has 2 DMA channels, and when conflicts occur between channels, arbitration is performed by the priority controller.

- Bus Matrix

  CPU and DMAC are connected to the bus matrix. When the CPU and DMAC access different AHB bus devices or AHB bridges, data transfers can be performed simultaneously. When the CPU and DMAC access the same bus device or AHB bridge, the priority of the CPU is higher than that of the DMAC. That is, only when the CPU releases the AHB bus device or the AHB bridge, the DMAC can access the AHB bus device or the AHB bridge.

- DMA Requests

  If the peripheral supports hardware DMA request, the DMA channel can be configured as a hardware trigger, otherwise, it can only be configured as a software trigger.

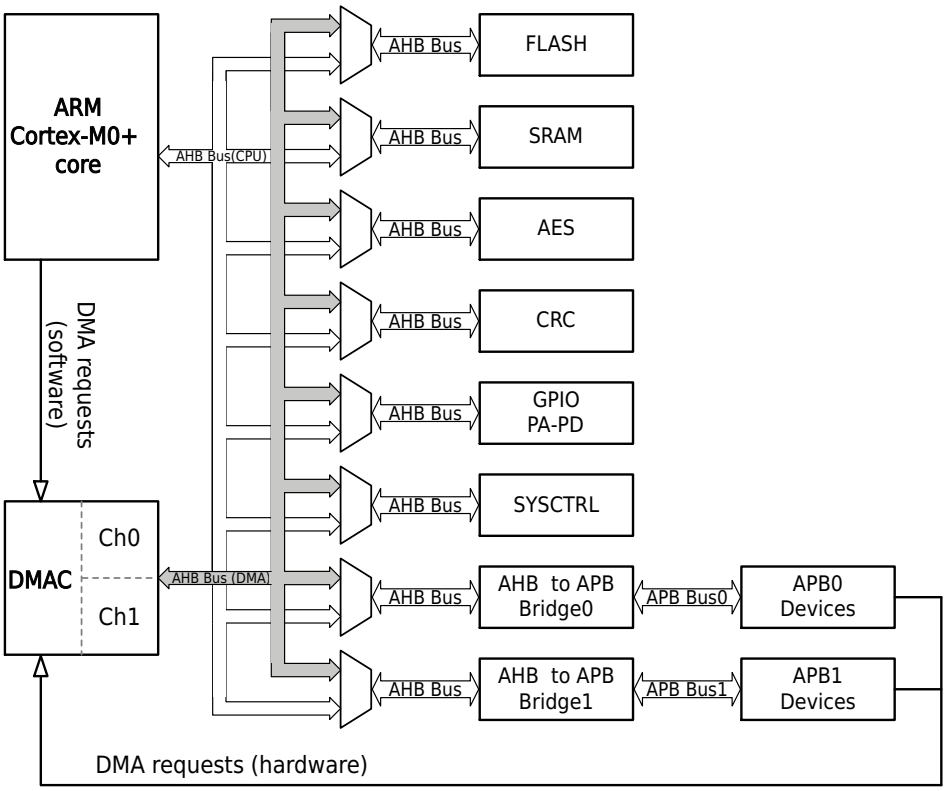



Figure 9-1 DMA Controller Functional Block Diagram


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

## 9.3.2 DMAC Transfer Mode

### 9.3.2.1 DMAC Transfer Mode Overview

DMA controller supports 4 transfer modes: software Block transfer mode, software Burst transfer mode, hardware Block transfer mode, and hardware Burst transfer mode.

**Table 9-1 DMAC Transfer Mode Comparison Table**

| Compare Items | Software Block Transfer | Software Burst Transfer | Hardware Block Transfer | Hardware Burst Transfer |
|---|---|---|---|---|
| Whether interrupted by higher priority DMA channel or CPU | Can | Can't | Can | Can't |
| Conditions that trigger the start of the transfer | Write DMAC register | | Peripheral interrupt flag | |
| The amount of data that a transfer is triggered | (BC+1)*(TC+1) | | BC+1 | (BC+1)*(TC+1) |
| The total amount of data transferred after configuration | (BC+1)*(TC+1) | | | |
| Main application scenarios | ● Memory to memory<br>● Memory to peripherals without DMA requests | | ● Memory to peripherals with DMA requests<br>● Peripherals with DMA requests to memory | ● Memory to memory<br>● Memory to peripherals without DMA requests |

### 9.3.2.2 Software Block Transfer Mode

When configuring DMAC_CONFAx.TRI_SEL=0 and DMAC_CONFBx.MODE=0, DMAC works in software Block transfer mode.

In this mode, the user writes 1 to DMAC_CONFAx.ST, which triggers the DAMC to start the software Block transfer mode, and stops after transferring (BC+1) * (TC+1) data. Each transmission of BC+1 data is completed, the DMAC inserts a transmission gap, and the priority arbiter performs priority arbitration in the gap. If a CPU or higher-priority DMA channel requests the bus during this gap, bus ownership is transferred to the CPU or higher-priority DMA channel. Once released, the remaining data transfer will resume and complete.

The software Block transfer diagram is shown below, where SA represents the source address and DA represents the destination address; The size of the data block is 3 (BC=2), the number of data blocks is 3 (TC=2), the data width is 16-bit, and the address increases automatically.

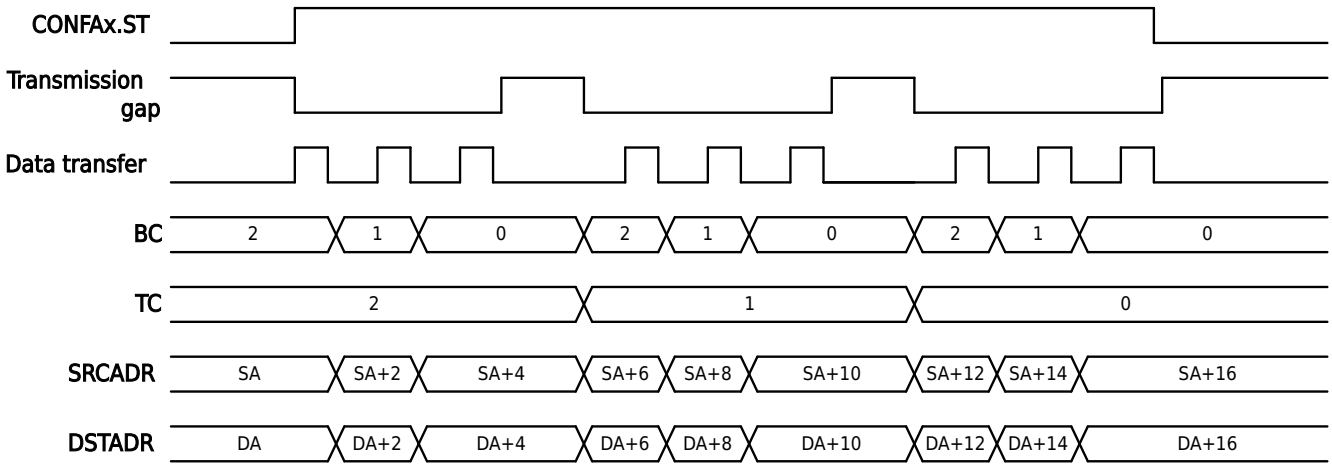



**Figure 9-2 DMA Software Block Transfer Mode**


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

#### 9.3.2.3 Software Burst Transfer Mode

When configuring DMAC_CONFAx.TRI_SEL=0 and DMAC_CONFBx.MODE=1, DMAC works in software Burst transfer mode.

In this mode, the user writes 1 to DMAC_CONFAx.ST, which triggers the DMAC to start the software Burst transfer mode, and stops after transmitting (BC+1) * (TC+1) data. There is no transmission gap in this transfer. Before all data transfers are completed, the DMA channel which is working currently will always occupy the bus. The CPU or a higher priority DMA channel can only access the bus after the operation is completed. It is recommended to avoid transferring excessive data at once, as this may render the CPU completely unable to operate for a period of time.

The software Burst transfer diagram is shown below, where SA represents the source address, DA represents the destination address; the data block size is 3 (BC=2), the number of data blocks is 3 (TC=2), the data width is 16-bit, and the address increases automatically.

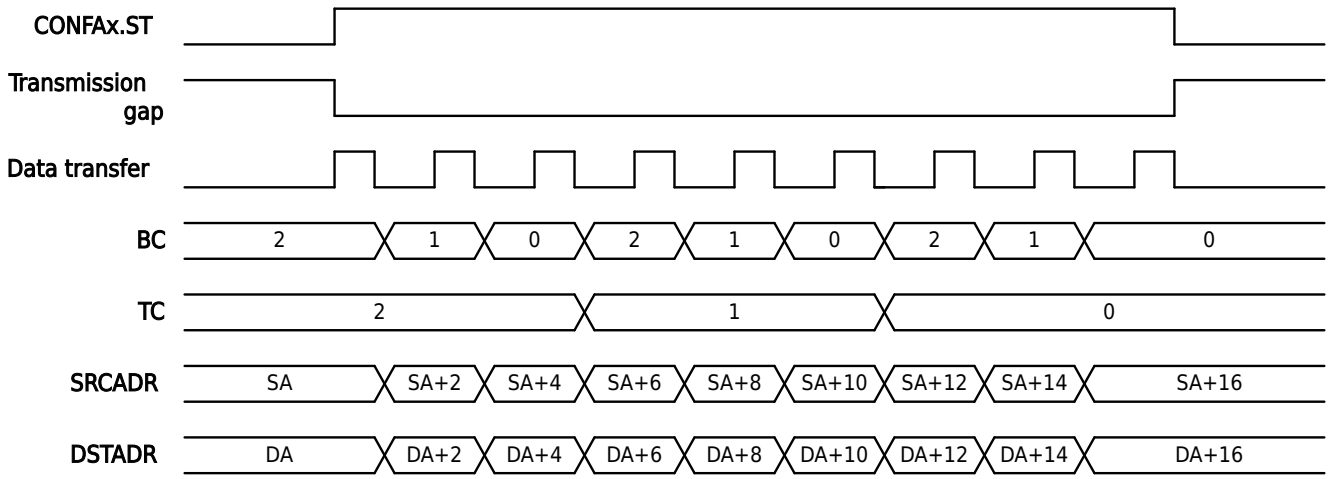



**Figure 9-3 DMA Software Burst Transfer Mode**

#### 9.3.2.4 Hardware Block Transmission Mode

When configuring DMAC_CONFAx.TRI_SEL is not 0 and DMAC_CONFBx.MODE is 0, DMAC works in hardware Block transfer mode.

In this mode, whenever the peripheral DMA request signal appears once, the DMAC is triggered to start a hardware block transfer, and transfer (BC+1) data. It needs to appear (TC+1) peripheral DMA request signal to complete the transmission of all DMA data. Every (BC+1) data is completed, the DMAC inserts a transmission gap, and the priority arbiter performs priority arbitration in the gap. If the CPU or a higher-priority DMA channel requests the bus during this gap, bus ownership is transferred to the CPU or higher-priority DMA channel. Once released, the remaining data transfer will resume and complete.

**Note**

In this mode, generally set BC to 0 to obtain a data from the peripheral or provide a data to the peripheral.

For DMA-supported hardware request signal types, refer to the DMAC_CONFAx.TRI_SEL bit function description.

The hardware block transmission diagram is shown below, where SA represents the source address, DA represents the destination address; the data block size is 2 (BC=1), the number of data blocks is 3 (TC=2), the data width is 8-bit, and the target address increases automatically, the source address is fixed, and the transmission is automatically disabled after completion.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

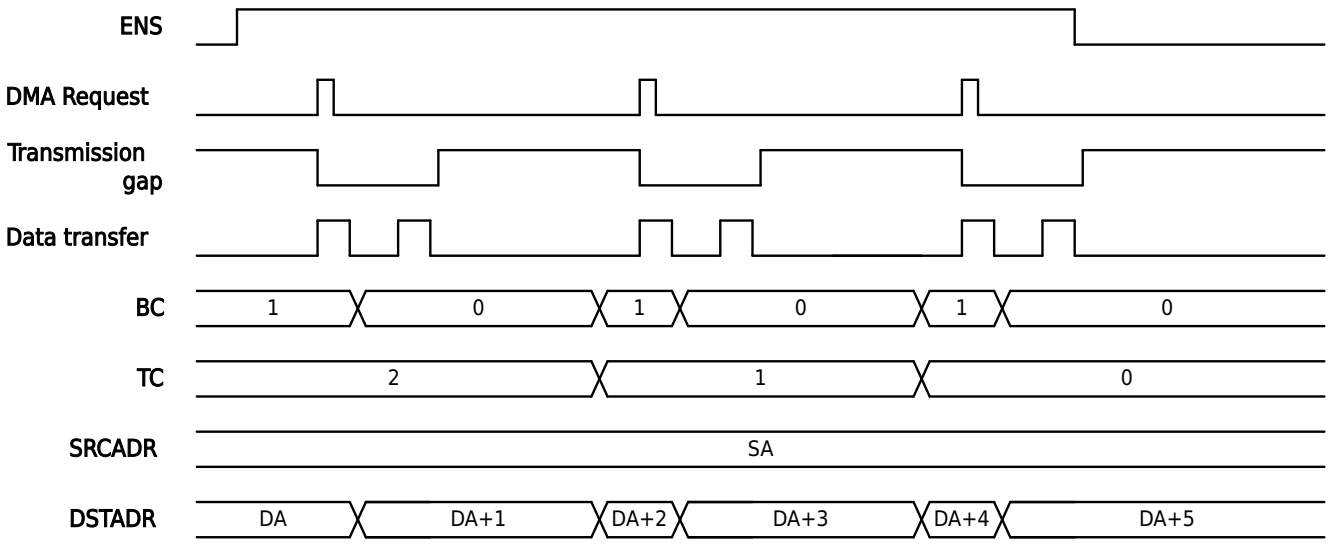



Figure 9-4 DMA Hardware Block Transfer Mode

### 9.3.2.5 Hardware Burst Transfer Mode

When configuring DMAC_CONFAx.TRI_SEL is not 0 and DMAC_CONFBx.MODE is 1, DMAC works in hardware Burst transfer mode.

In this mode, when a peripheral DMA request signal occurs, the DMAC initiates hardware Burst transfer mode, stopping after transmitting (BC+1)*(TC+1) data. There is no transmission gap in this transfer. Before all data transfers are completed, the DMA channel which is working currently will always occupy the bus. The CPU or a higher priority DMA channel can only access the bus after the operation is completed. It is recommended to avoid transferring excessive data at once, as this may render the CPU completely unable to operate for a period of time.

For DMA-supported hardware request signal types, refer to the DMAC_CONFAx.TRI_SEL bit function description.

The hardware Burst transmission diagram is shown below, where SA represents the source address, DA represents the destination address; the data block size is 3 (BC=2), the number of data blocks is 3 (TC=2), the data width is 8-bit, and the source address increases automatically, the target address is fixed, and the transfer is automatically disabled after completion.

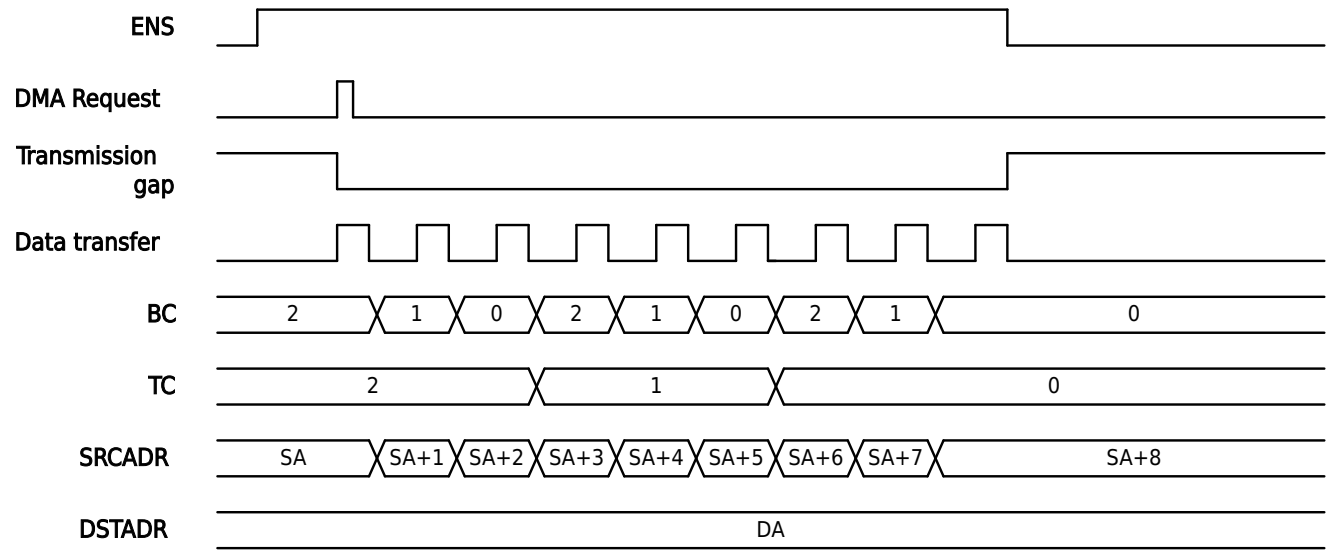



Figure 9-5 DMA Hardware Burst Transfer Mode


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 9.3.3 DMA Transfer Pause

#### Transfer Pause in Software Block Transfer Mode

When DMAC works in software Block transfer mode:

When DMAC_CONF.HALT is set to a non-zero value, all active transfer channels will enter a paused state during transfer intervals and cease data transmission. After setting DMAC_CONF.HALT to 0: Previously active DMAC channels (DMAC_CONFAx.ST=1) will resume unfinished transfers; Inactive DMAC channels (DMAC_CONFAx.ST=0) will wait until DMAC_CONFAx.ST becomes 1 to initiate transfers.

When DMAC_CONFAx.PAS is set to 1, the corresponding active channel to x will pause during transfer intervals and halt data transmission; After setting DMAC_CONFAx.PAS is set to 0, the channel will resume the unfinished transfer.

#### Transfer Pause in Software Burst Transfer Mode

When the DMAC operates in software Burst transfer mode with no active data transmission:

When DMAC_CONF.HALT is set to a non-zero value, DMA channels will enter a paused state and cease data transmission. If a DMAC channel has software trigger enabled (DMAC_CONFAx.ST=1) during the pause, then after setting DMAC_CONF.HALT to 0, the channel will initiate a new complete transfer; if a DMAC channel has software trigger disabled (DMAC_CONFAx.ST=0) during the pause, then after setting DMAC_CONF.HALT to 0, wait until the channel's software trigger is enabled (DMAC_CONFAx.ST=1), and a new complete transfer will be initiated.

When DMAC_CONFAx.PAS is set to 1, DMA channels will enter a paused state and cease data transmission. If the channel's software trigger is enabled (DMAC_CONFAx.ST=1) during the pause, then after setting DMAC_CONFAx.PAS to 0, the channel will initiate a new complete transfer; if the channel's software trigger is disabled (DMAC_CONFAx.ST=0) during the pause, then after setting DMAC_CONFAx.PAS to 0, wait until the channel's software trigger is enabled (DMAC_CONFAx.ST=1), and a new complete transfer will be initiated.

When the DMAC is working in the software Burst transfer mode and data transfer is in progress, writing a non-zero value to DMAC_CONF.HALT or writing 1 to DMAC_CONFAx.PAS cannot pause the ongoing transfer; the DMA channel will not enter the pause state after the current transfer is completed.

#### Transfer Pause in Hardware Block Transfer Mode

When DMAC works in hardware Block transfer mode:

When DMAC_CONF.HALT is set to a non-zero value, all active transfer channels will enter a paused state during transfer intervals and cease data transmission. After setting DMAC_CONF.HALT to 0, each DMAC channel will require the next trigger signal to resume the previously unfinished transfer.

When DMAC_CONFAx.PAS is set to 1, the corresponding active channel to x will pause during transfer intervals and halt data transmission. After setting DMAC_CONFAx.PAS to 0, the channel will require the next trigger signal to resume the previously unfinished transfer.

#### Transfer Pause in Hardware Burst Transfer Mode

When the DMAC operates in hardware Burst transfer mode with no active data transmission:

When DMAC_CONF.HALT is set to a non-zero value, DMA channels will enter a paused state and cease data transmission. After configuring DMAC_CONF.HALT to 0, DMAC will start a new complete transmission when it receives the next trigger signal.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

When DMAC_CONFAx.PAS is set to 1, DMA channels will enter a paused state and cease data transmission. After configuring DMAC_CONFAx.PAS to 0, DMAC will start a new complete transmission when it receives the next trigger signal.

When the DMAC is working in the hardware Burst transfer mode and data transfer is in progress, writing a non-zero value to DMAC_CONF.HALT or writing 1 to DMAC_CONFAx.PAS cannot pause the ongoing transfer; the DMA channel will not enter the pause state after the current transfer is completed.

### 9.3.4 Other Configurations of DMA

#### Data Width

Configure DMAC_CONFBx.WIDTH, you can configure the width of each data to be transferred as 8-bit, 16-bit, or 32-bit, noting that this bit width must exactly match the bit width of the DMA source and destination addresses.

#### Block Size

Configure DMAC_CONFAx.BC, you can configure the number of data to be transmitted in each data block as 1-16. Note: The value of BC should match the number of consecutive operations provided by the source and destination addresses.

- If the source address or destination address can only provide or receive one data at a time, then BC can only be configured as 0. For example, USART/LPUART/SPI can only provide one data to DMA each time, while USART/LPUART/SPI can only receive one data from DMA each time, so BC can only be configured as 0.
- If both the source address and the destination address can provide or receive multiple data, then BC can be configured to a larger value to speed up the transmission speed. For example, when transferring from FLASH to RAM or from FLASH to CRC, the source address and destination address can be operated continuously, so BC can be configured as 15 to achieve fast transfer.

#### Number of Data Blocks

Configure DMAC_CONFAx.TC, you can configure the number of data blocks to be transferred from 1 to 65536 after each DMA is started.

#### Channel Priority

- When DMAC_CONF.PRIO=0b0, DMAC works in fixed priority mode. At this time, CH0 has a higher priority than CH1, Only when the high priority channel does not transmit data, the low priority channel can transmit data.
- When DMAC_CONF.PRIO=0b1, DMAC works in cycle priority mode. At this time, CH0 and CH1 have higher priority in turn, and the two channels will occupy the bus in turn for data transmission.

#### Auto-Reload

If the automatic reload function is enabled, the data block size, number of data blocks, source address, and destination address will be automatically loaded to the last configuration value after the transmission is completed, and there is no need to configure it every time.

### 9.3.5 DMA Interrupt

DMA supports two types of interrupts: transfer error and transfer complete. When an interrupt is triggered, the status register DMAC_CONFBx.STAT can be checked to determine what condition triggered the interrupt, and the built-in interrupt status flag can be cleared by clearing the register DMAC_CONFBx.STAT.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Support conditions that trigger the DMA interrupt are as follows:

- Peripheral request to stop DMA
- Access transfer source address error
- Access transfer destination address error
- Transmission completed

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

## 9.4 Register Description

### 9.4.1 Register List

Base address: 0x40021000

Table 9-2 DMA Controller (DMA) Register Offset

| Offset | Register | Description | Width |
|---|---|---|---|
| 0x00 | DMAC_CONF | All channel configuration registers | 32 |
| 0x10+0x10*x | DMAC_CONFAx | Channel x configuration A register (x=0-1) | 32 |
| 0x14+0x10*x | DMAC_CONFBx | Channel x configuration B register(x=0-1) | 32 |
| 0x18+0x10*x | DMAC_SRCADRx | Channel x transfer source address register (x=0-1) | 32 |
| 0x1C+0x10*x | DMAC_DSTADRx | Channel x transfer destination address register (x=0-1) | 32 |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 9.4.2 All Channel Configuration Register (DMAC_CONF)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | EN | Reserved | | PRIO | HALT | | | | Reserved | | | | | | | | | | | | | | | | | | | | | | | |
| Access | RW | | | RW | RW | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 31 | EN | DMA controller enable<br>● 0b0: Disable<br>● 0b1: Enable<br>**Notice**<br>When the DMA controller is disabled, ongoing transfers will be forcibly terminated, resulting in unpredictable consequences. | RW |
| *30:29* | *Reserved* | - | - |
| 28 | PRIO | DMA channel priority configuration<br>● 0b0: Fixed priority mode, CH0>CH1 priority<br>● 0b1: Cycle priority mode, CH0/CH1 take turns to obtain bus access | RW |
| 27:24 | HALT | DMA all channel transfer pause control<br>● 0b0000: Resume all channel data transmission<br>● 0b0001-0b1111: Suspend all channel data transmission | RW |
| *23:0* | *Reserved* | - | - |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 9.4.3 Channel x Configuration A Register (DMAC_CONFAx) (x=0-1)

Offset: 0x10+0x10*x

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| - | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | ENS | PAS | ST | TRI_SEL | | | | | | | Reserved | | BC | | | | TC | | | | | | | | | | | | | | | |
| Access | RW | RW | RW | RW | | | | | | | | | RW | | | | RW | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 31 | ENS | DMA channel enable control<br>● 0b0: Disable the current DMA channel<br>● 0b1: Enable the current DMA channel; if CONFBx.MSK is 0, this bit is automatically cleared when the transfer is completed<br>**Notice**<br>● When the DMA controller is disabled, ongoing transfers will be forcibly terminated, resulting in unpredictable consequences.<br>● DMA channel enable, the operation is recommended to be performed after all DMA configurations are completed. | RW |
| 30 | PAS | DMA channel data transfer pause control<br>● 0b0: Resume DMA channel data transfer<br>● 0b1: Suspend DMA channel data transfer | RW |
| 29 | ST | DMA channel software trigger control<br>● 0b0: Stop DMA channel software Block/Burst transfer<br>● 0b1: Trigger DMA channel to start software Block/Burst transfer, this bit is automatically cleared when the transfer is completed | RW |
| 28:22 | TRI_SEL | DMA channel trigger source configuration<br>● 0x00: Software trigger<br>● 0x40: SPI0 receiving Buf is not empty<br>● 0x41: SPI0 sending Buf is empty<br>● 0x42: SPI1 receiving Buf is not empty<br>● 0x43: SPI1 sending Buf is empty<br>● 0x44: ADC JQR conversion completed<br>● 0x45: ADC SQR conversion completed<br>● 0x46: LCD timing interrupt<br>● 0x47: Setting prohibited<br>● 0x48: USART0 receiving Buf is not empty<br>● 0x49: USART0 sending Buf is empty<br>● 0x4A: USART1 receiving Buf is not empty<br>● 0x4B: USART1 sending Buf is empty<br>● 0x4C: LPUART0 receiving Buf is not empty<br>● 0x4D: LPUART0 sending Buf is empty<br>● 0x4E: LPUART1 receiving Buf is not empty | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 28:22 | TRI_SEL | ● 0x4F: LPUART1 sending Buf is empty<br>● 0x50: LPUART2 receiving Buf is not empty<br>● 0x51: LPUART2 sending Buf is empty<br>● 0x52: TIM0 channel A, occurrs capture event or compare event<br>● 0x53: TIM0 channel B, occurrs capture event or compare event<br>● 0x54: TIM1 channel A, occurrs capture event or compare event<br>● 0x55: TIM1 channel B, occurrs capture event or compare event<br>● 0x56: TIM2 channel A, occurrs capture event or compare event<br>● 0x57: TIM2 channel B, occurrs capture event or compare event<br>● 0x58: TIM3 channel A, occurrs capture event or compare event<br>● 0x59: TIM3 channel B, occurrs capture event or compare event<br>● 0x5A: HSI2C master or slave receiving Buf is not empty<br>● 0x5B: HSI2C master or slave sending Buf is empty<br>● Others: Setting prohibited | RW |
| *21:20* | *Reserved* | - | - |
| 19:16 | BC | DMA channel transfer data block size<br>Configure the size of the data block to be transmitted by the DMA channel as BC+1<br>**Note**<br>When the source address or destination address can only provide or receive a data every once in a while, the BC value can only be set to 0. | RW |
| 15:0 | TC | Number of data blocks transferred by DMA channel<br>Configure the number of data blocks to be transmitted by the DMA channel as TC+1<br>During transmission, every time a data block is transmitted, the value of this register will be decremented by 1<br>**Note**<br>The total amount of data transferred by DMA is (TC +1)*(BC +1). | RW |

**Note**

The trigger source of TIM0/1/2/3:

- TIM0/1/2/3 channel A: Compare capture A, external trigger
- TIM0/1/2/3 channel B: Compare capture B, event update


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 9.4.4 Channel x Configuration B Register (DMAC_CONFBx) (x=0-1)

Offset: 0x14+0x10*x

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| - | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | MODE | | WIDTH | | FS | FD | RC | RS | RD | ERR_IE | FIS_IE | STAT | Reserved | | | | | | | | | | | | | | | | | MSK |
| Access | | | RW | | RW | | RW | RW | RW | RW | RW | RW | RW | RW | | | | | | | | | | | | | | | | | RW |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:30* | *Reserved* | - | - |
| 29:28 | MODE | DMA channel transfer mode configuration<br>● 0b00: Block transfer<br>● 0b01: Burst transfer<br>● 0b10: Setting prohibited<br>● 0b11: Setting prohibited | RW |
| 27:26 | WIDTH | DMA channel transfer data width configuration<br>● 0b00: 8-bit<br>● 0b01: 16-bit<br>● 0b10: 32-bit<br>● 0b11: Setting prohibited | RW |
| 25 | FS | DMA channel source address auto-increment enable<br>● 0b0: The source address will be incremented automatically after each data transmission is completed<br>● 0b1: The source address remains unchanged during transmission<br>**Note**<br>The auto-increment corresponding to the data width of 8-bit/16-bit/32-bit is 1/2/4 respectively. | RW |
| 24 | FD | DMA channel destination address auto-increment enable<br>● 0b0: The target address will be incremented automatically after each data transmission is completed<br>● 0b1: The destination address remains unchanged during transmission<br>**Note**<br>The auto-increment corresponding to the data width of 8-bit/16-bit/32-bit is 1/2/4 respectively. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 23 | RC | DMA channel BC and TC reload enable<br>● 0b0: Disable BC/TC reloading, and the values of BC/TC are all 0 after the transmission is completed<br>● 0b1: Enable BC/TC reloading, and the values of BC/TC will automatically restore to the written initial value after the transmission is completed | RW |
| 22 | RS | DMA channel source address reload enable<br>● 0b0: Disable source address overloading, the value of the source address is uncertain after the transmission is completed<br>● 0b1: Enable the source address reloading, and the value of the source address is the written initial value after the transmission is completed | RW |
| 21 | RD | DMA channel destination address reload enable<br>● 0b0: Disable destination address overloading, the value of the destination address is uncertain after the transmission is completed<br>● 0b1: Enable target address reloading, the value of the destination address will automatically restore to the written initial value after the transmission is completed | RW |
| 20 | ERR_IE | DMA channel transfer error interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable<br>**Note**<br>In the interrupt service routine, you need to write 0x00 to STAT to clear the interrupt status. | RW |
| 19 | FIS_IE | DMA channel transfer complete interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable<br>**Note**<br>In the interrupt service routine, you need to write 0x00 to STAT to clear the interrupt status. | RW |
| 18:16 | STAT | DMA channel current transfer status<br>● 0b000: Initial value<br>● 0b001: Setting prohibited<br>● 0b010: Transmission error, peripheral request to stop DMA<br>● 0b011: Transmission error, access transmission source address error<br>● 0b100: Transmission error, access transmission destination address error<br>● 0b101: Transmission completed<br>● 0b110: Setting prohibited<br>● 0b111: Transmission pause | RW |
| *15:1* | *Reserved* | - | - |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 0 | MSK | Auto-disable/enable configuration when DMA channel transfer completes<br>● 0b0: CONFAx.ENS is automatically cleared when the DMA channel transfer completes<br>● 0b1: CONFAx.ENS remains unchanged when DMA channel transfer completes | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 9.4.5 Channel x Transfer Source Address Register (DMAC_SRCADRx) (x=0-1)

Offset: 0x18+0x10*x

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| - | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | SRCADR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Access | RW | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 31:0 | SRCADR | DMA channel source address configuration<br>Auto-reload can be configured via CONFBx.RS<br>**Notice**<br>The address alignment should match the data width, otherwise, it will cause address access errors. | RW |

### 9.4.6 Channel x Transfer Destination Address Register (DMAC_DSTADRx) (x=0-1)

Offset: 0x1C+0x10*x

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| - | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | DSTADR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Access | RW | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 31:0 | DSTADR | DMA channel destination address configuration<br>Auto-reload can be configured via CONFBx.RD<br>**Notice**<br>The address alignment should match the data width, otherwise, it will cause address access errors. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

# 10 CTRIM

## 10.1 Introduction

When the module works in calibration mode, it automatically performs real-time calibration of the RCH/RCL output frequency, making the accuracy of the RCH/RCL output frequency is no longer affected by environmental variations. When the module works in timer mode, it has general timing function and can still work normally under DeepSleep mode.

## 10.2 Features

The clock calibration supports the following features:

- Calibration mode: Supports RCH/RCL automatic real-time calibration/single calibration
- Calibration mode: Multiple calibration steps, quickly approaching the target
- Timing mode: Can work in DeepSleep mode
- Timing mode: Supports automatic wake-up
- Timing mode: multiple clock signals available for selection
- Timing mode: Toggle output

## 10.3 Functional Description

### 10.3.1 RCH Clock Calibration Mode

When CTRIM_CR1.MD is configured as 0b00, the module operates in RCH calibration mode. This mode enables automatic real-time calibration of the output frequency of RCH, ensuring the accuracy of RCH remains unaffected by environmental variations. During calibrating RCH, the module requires a precise low-frequency clock signal, which can be either the on-chip PCLK or an external input via the ETR pin.

**Note**

The RCLK frequency must be less than half of the frequency of RCH.

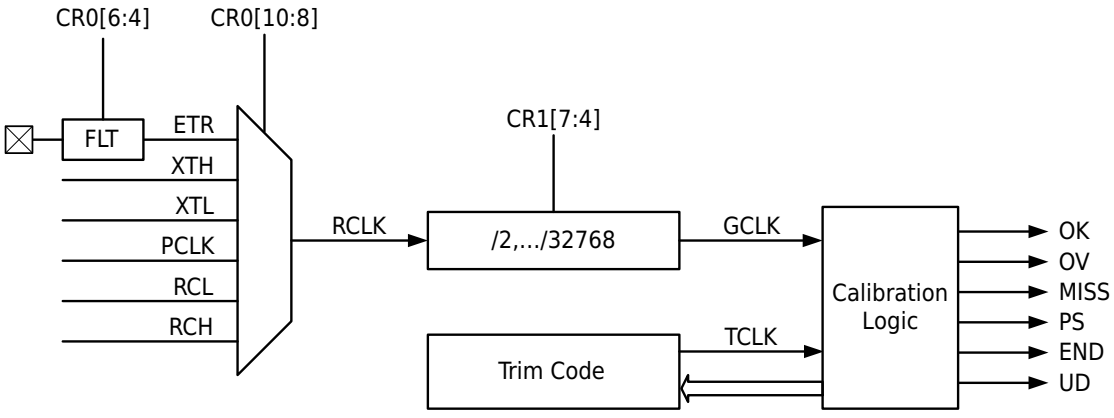



Figure 10-1 RCH Clock Calibration Operational Block Diagram

When automatic calibration begins, the error counter performs down-counting on the TCLK signal ( RCH oscillation frequency) starting from CTRIM_ARR at each GCLK cycle, then switches to up-counting upon reaching zero. If the error counter value within a GCLK cycle is less than the CTRIM_FLIM register value, CTRIM_ISR.OK is set to flag that the output frequency of RCH meets expectations. If the error counter value within a GCLK cycle is greater than the CTRIM_FLIM register value, the TrimCode of RCH is


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

automatically adjusted for the next counting cycle. The auto-adjustment rules are: If counting direction is down-counting, increase the TrimCode of RCH by CTRIM_CR0.STEP; If counting direction is up-counting, decrease the TrimCode of RCH by CTRIM_CR0.STEP. The calibration logic automatically reduces the value of CTRIM_CR0.STEP to achieve gradual convergence of the output frequency of RCH toward the target value.

UD event is generated when the error counter count decrements to zero; MISS event is triggered when the error counter count increments to CTRIM_ARR, stopping further counting. Whenever the rising edge of GCLK arrives, the error counter starts counting again.

- **RCH Single Calibration**

  Configure appropriate STEP, initial value, and FLIM using the same method as above. Set CTRIM_CR1.OST=1 to perform single calibration of RCH, which completes when STEP becomes 1.

### 10.3.2 RCL Clock Calibration Mode

When CTRIM_CR1.MD is set to 0b01, the module operates in RCL calibration mode. This mode can automatically calibrate the output frequency of RCL in real time, so that the accuracy of RCL output frequency will no longer be affected by environmental changes. When calibrating RCL, the module requires a precise high-frequency clock signal, which can be either the on-chip PCLK or an external input via the ETR pin.

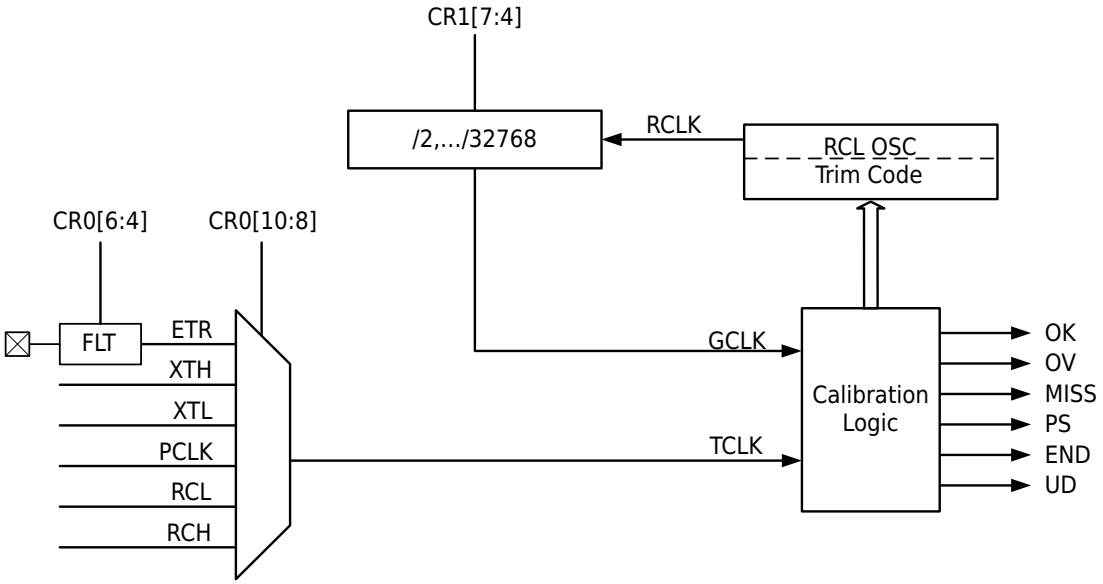



**Figure 10-2 RCL Clock Calibration Operational Block Diagram**

When automatic calibration begins, the error counter performs down-counting on the TCLK clock signal starting from CTRIM_ARR at each GCLK (the oscillation frequency of RCL after frequency division) cycle, then switches to up-counting upon reaching zero. If the error counter value within a GCLK cycle is less than the value of CTRIM_FLIM register, CTRIM_ISR.OK is set to flag that the RCL output frequency meets expectations. If the error counter value within a GCLK cycle is greater than the value of CTRIM_FLIM register, the TrimCode of RCL is automatically adjusted for the next counting cycle. The auto-adjustment rules are: If counting direction is down-counting, decrease TrimCode of RCL by CTRIM_CR0.STEP; If counting direction is up-counting, increase TrimCode of RCL by CTRIM_CR0.STEP. The calibration logic automatically reduces the value of CTRIM_CR0.STEP to achieve gradual convergence of the RCL output frequency toward the target value.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

UD event is generated when the error counter decrements to zero; stopping further counting and MISS event is triggered when the counter increments to CTRIM_ARR. Whenever the rising edge of GCLK arrives, the error counter starts counting again.

- **RCL Single Calibration**

  Configure appropriate STEP, initial value, and FLIM. Set CTRIM_CR1.OST=1 to perform single calibration of RCL, which completes when STEP becomes 1.

### 10.3.3 Low Power Timer Mode

When CTRIM_CR1.MD is set to 0b11, the module operates in general low power timer mode. In this mode, CTRIM_CR0.SRC must be configured to a clock available in low power mode. In this mode, CTRIM has timing and interrupt functions of a general timer. If the timer clock is RCL, it operates normally in DeepSleep mode, supporting automatic wake-up with .

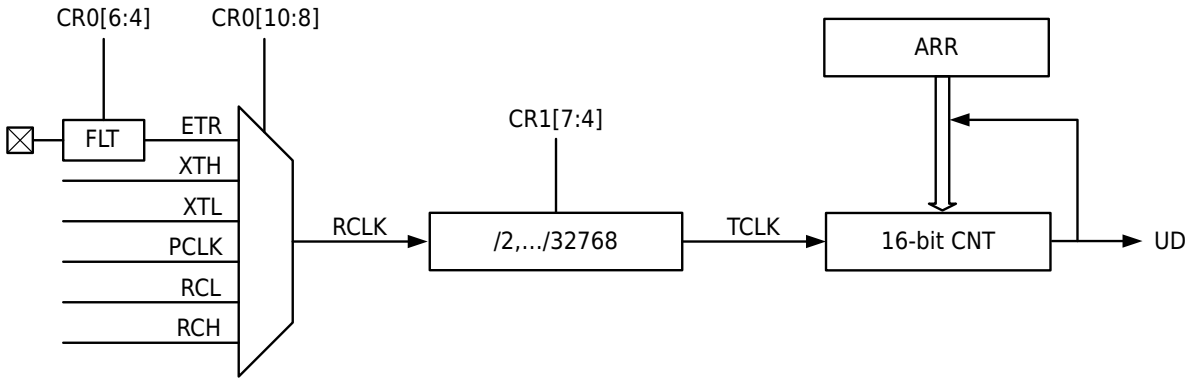



**Figure 10-3 General Low Power Timer Scenario Operational Block Diagram**

The timer counts down from ARR, and when it reaches 0, an underflow is generated. If the interrupt is enabled, underflow interrupt is generated, the counting period is ARR + 1.

TOG output level flips once every time the timer overflows; the user can output the TOG signal to the pin through the port multiplexing function of the GPIO to drive an external circuit.

## 10.4 Operation Example

### 10.4.1 RCH Clock Automatic Calibration Example

#### Procedure

**Step1.** Refer to the "General-Purpose I/O (GPIO)" chapter for related description of pin multiplexing configuration, set the relevant pin function to CTRIM.ETR, and supply it with a 2kHz square wave.

**Step2.** Set CTRIM_CR1.MD to 0 to select RCH clock calibration mode.

**Step3.** Set CTRIM_CR0.SRC to 0, and the external ETR pin input signal is used as RCLK.

**Step4.** Set CTRIM_CR1.PRS to 1, perform 2 frequency division on RCLK, and GCLK frequency is 1kHz.

**Step5.** Configure CTRIM_ARR to 47999, and the target count value in a GCLK count cycle is 48000.

> **Note**
> The ETR pin input signal is a 2kHz square wave, the period of GCLK is 1ms. The theoretical count value of the error counter for the RCH output frequency within 1ms should be 48000.

**Step6.** Set CTRIM_FLIM to 240, select RCH calibration accuracy of 0.5%.
48000*0.005=240.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

**Step7.** Set CTRIM_CR0.STEP to 0, and select the initial calibration step to be 16.

**Step8.** Set CTRIM_CR1.AUTO to 1 to enable automatic calibration.

**Step9.** Set CTRIM_CR1.EN to 1 to start clock calibration.

**Step10.** Wait for CTRIM_ISR.END to be 1, and the automatic calibration is completed.

### 10.4.2 RCL Clock Automatic Calibration Example

#### Procedure

**Step1.** Refer to the "General-Purpose I/O (GPIO)" chapter for related description of pin multiplexing configuration, set the relevant pin function to CTRIM.ETR, and supply it with a 1MHz square wave.

**Step2.** Set CTRIM_CR1.MD to 1, select RCL clock calibration mode.

**Step3.** Set CTRIM_CR0.SRC to 0, use the external ETR pin input signal as TCLK.

**Step4.** Set CTRIM_CR1.PRS to 0x0A, perform 1024 frequency division on RCLK (RCL output frequency is ), and the target frequency of GCLK is 32Hz.

**Step5.** Configure CTRIM_ARR to 31249, and the target count value in a GCLK count cycle is 31250.

**Note**

The target period of GCLK is 31.25ms, during which the theoretical count value of the error counter for TCLK(1MHz) should be 31250.

**Step6.** Set CTRIM_FLIM to 156, select RCL correction accuracy to 0.5%.
31250*0.005=156.

**Step7.** Set CTRIM_CR0.STEP to 0, and select the initial calibration step to be 16.

**Step8.** Set CTRIM_CR1.AUTO to 1 to enable automatic calibration.

**Step9.** Set CTRIM_CR1.EN to 1 to start clock calibration.

**Step10.** Wait for CTRIM_ISR.END to be 1, and the automatic calibration is completed.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

## 10.5 Register Description

### 10.5.1 Register List

Base address: 0x40009000

Table 10-1 Clock Trim Module (CTRIM) Register Offset

| Offset | Register | Description | Width |
|---|---|---|---|
| 0x00 | CTRIM_ARR | Autoload Register | 32 |
| 0x04 | CTRIM_CNT | Error Counter Register | 32 |
| 0x0C | CTRIM_CR0 | Control Register 0 | 32 |
| 0x10 | CTRIM_CR1 | Control Register 1 | 32 |
| 0x14 | CTRIM_IER | Interrupt Enable Register | 32 |
| 0x18 | CTRIM_ISR | Interrupt and Status Register | 32 |
| 0x1C | CTRIM_ICR | Interrupt Flag Clear Register | 32 |
| 0x20 | CTRIM_FCAP | Error Result Register, Counter Value at the End of the GCLK Cycle | 32 |
| 0x24 | CTRIM_TVAL | TRIM Result Register | 32 |
| 0x28 | CTRIM_FLIM | Error Upper Limit Register | 32 |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 10.5.2 Autoload Register (CTRIM_ARR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x0000FFFF | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | ARR | | | | | | | | | | | | | | | |
| Access | | | | | | | | | | | | | | | | | RW | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:16* | *Reserved* | - | - |
| 15:0 | ARR | Error counter reload value | RW |

### 10.5.3 Error Counter Register (CTRIM_CNT)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x04 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | CNT | | | | | | | | | | | | | | | |
| Access | | | | | | | | | | | | | | | | | RO | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:16* | *Reserved* | - | - |
| 15:0 | CNT | Error counter count value | RO |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 10.5.4 Control Register 0 (CTRIM_CR0)

**Note**

This register can only be changed when CTRIM_CR1.EN is 0.

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x0C | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | SRC | | | Reserved | | ETRFLT | | Reserved | | STEP | |
| Access | | | | | | | | | | | | | | | | | | | | | | RW | | | | | RW | | | | RW | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:11* | *Reserved* | - | - |
| 10:8 | SRC | Clock source selection<br>● 0b000: ETR<br>● 0b001: XTH<br>● 0b010: XTL<br>● 0b011: PCLK<br>● 0b100: RCL<br>● 0b101: RCH<br>● 0b110: Setting prohibited<br>● 0b111: Setting prohibited | RW |
| *7* | *Reserved* | - | - |
| 6:4 | ETRFLT | External input pin (ETR) filter configuration<br>● 0b000: No filter<br>● 0b001: $F_{sample}$=PCLK, N=2<br>● 0b010: $F_{sample}$=PCLK, N=4<br>● 0b011: $F_{sample}$=PCLK, N=6<br>● 0b100: $F_{sample}$=PCLK/4, N=4<br>● 0b101: $F_{sample}$=PCLK/4, N=6<br>● 0b110: $F_{sample}$=PCLK/8, N=4<br>● 0b111: $F_{sample}$=PCLK/8, N=6 | RW |
| *3* | *Reserved* | - | - |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 2:0 | STEP | TRIM initial step configuration<br>● 0b000: 0x10<br>● 0b001: 0x02<br>● 0b010: 0x08<br>● 0b011: 0x20<br>● 0b100: 0x80<br>● 0b101: 0x100<br>● 0b110: 0x200<br>● 0b111: 0x400 | RW |

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 10.5.5 Control Register 1 (CTRIM_CR1)

**Note**
This register can only be changed when CTRIM_CR1.EN is 0.

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x10 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | OST | PRS | | | | AUTO | MD | | EN |
| Access | | | | | | | | | | | | | | | | | | | | | | | | RW | RW | | | | RW | RW | | RW |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:9* | *Reserved* | - | - |
| 8 | OST | Calibration mode configuration<br>● 0b0: Real-time calibration mode, the TRIM process continues<br>● 0b1: Single calibration mode, end when STEP becomes 1 | RW |
| 7:4 | PRS | Reference clock prescaler configuration<br>● 0b0000: Setting prohibited<br>● 0b0001: Divide by 2<br>● 0b0010: Divide by 4<br>● 0b0011: Divide by 8<br>● 0b0100: Divide by 16<br>● 0b0101: Divide by 32<br>● 0b0110: Divide by 64<br>● 0b0111: Divide by 128<br>● 0b1000: Divide by 256<br>● 0b1001: Divide by 512<br>● 0b1010: Divide by 1024<br>● 0b1011: Divide by 2048<br>● 0b1100: Divide by 4096<br>● 0b1101: Divide by 8192<br>● 0b1110: Divide by 16384<br>● 0b1111: Divide by 32768 | RW |
| 3 | AUTO | RCH/RCL automatic TRIM enable configuration<br>● 0b0: Disable<br>● 0b1: Enable<br>**Note**<br>In real-time calibration mode, this bit is prohibited from being configured as 0. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 2:1 | MD | Operating mode configuration<br>● 0b00: RCH calibration mode, CR0.SRC needs to select a low-speed precision clock<br>● 0b01: RCL calibration mode, CR0.SRC needs to select a high-speed precision clock<br>● 0b10: Setting prohibited<br>● 0b11: Timer mode | RW |
| 0 | EN | Timer enable control<br>● 0b0: Disable<br>● 0b1: Enable | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 10.5.6 Interrupt Enable Register (CTRIM_IER)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x14 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | OK | OV | MISS | PS | END | UD |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | | RW | RW | RW | RW | RW | RW |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:6* | *Reserved* | - | - |
| 5 | OK | Auto Trim results in range interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 4 | OV | TrimCode overflow interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 3 | MISS | Count failure interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 2 | PS | Error counter completion interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 1 | END | Auto Trim end interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 0 | UD | Error counter underflow interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 10.5.7 Interrupt and Status Register (CTRIM_ISR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x18 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | DIR | OK | OV | MISS | PS | END | UD |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | RO | RO | RO | RO | RO | RO | RO |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:7* | *Reserved* | - | - |
| 6 | DIR | Error counter counting direction<br>● 0b0: Counting up, RCH output frequency is higher than the target frequency, RCL output frequency is lower than the target frequency<br>● 0b1: Counting down, RCH output frequency is lower than the target frequency, RCL output frequency is higher than the target frequency | RO |
| 5 | OK | Trim accuracy flag<br>● 0b0: Accuracy does not meet FLIM requirements when Trim is completed<br>● 0b1: Accuracy meets FLIM requirements when Trim is completed | RO |
| 4 | OV | TrimCode overflow flag<br>● 0b0: The TrimCode of RCH/RCL is not 0x000 or 0x1FF<br>● 0b1: The TrimCode of RCH/RCL is adjusted to 0x000 or 0x1FF | RO |
| 3 | MISS | Counting failure flag<br>● 0b0: The error counter does not count up to ARR during one cycle of GCLK<br>● 0b0: The error counter counts up to ARR during one cycle of GCLK | RO |
| 2 | PS | Error counter completion flag<br>● 0b0: Error counter is counting<br>● 0b1: Error counter has completed one counting cycle and stopped | RO |
| 1 | END | Auto Trim end flag<br>● 0b0: Auto Trim is in progress<br>● 0b1: Auto Trim is completed | RO |
| 0 | UD | Error counter underflow flag<br>● 0b0: The error counter has not counted to 0<br>● 0b1: The error counter has counted to 0 | RO |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 10.5.8 Interrupt Status Flag Register (CTRIM_ICR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x1C | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x0000003F | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | OK | OV | MISS | PS | END | UD |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | | WO | WO | WO | WO | WO | WO |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:6* | *Reserved* | - | - |
| 5 | OK | Clear the calibration success OK flag<br>Write 0 to clear flag, write 1 has no effect; read always returns 1. | WO |
| 4 | OV | Clear TrimCode overflow flag<br>Write 0 to clear flag, write 1 has no effect; read always returns 1. | WO |
| 3 | MISS | Clear count failure flag<br>Write 0 to clear flag, write 1 has no effect; read always returns 1. | WO |
| 2 | PS | Clear error counter completion flag<br>Write 0 to clear flag, write 1 has no effect; read always returns 1. | WO |
| 1 | END | Clear auto Trim end flag<br>Write 0 to clear flag, write 1 has no effect; read always returns 1. | WO |
| 0 | UD | Clear error counter underflow flag<br>Write 0 to clear flag, write 1 has no effect; read always returns 1. | WO |

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 10.5.9 Error Result Register (CTRIM_FCAP)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x20 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x0000FFFF | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | FCAP | | | | | | | | | | | | | | | |
| Access | | | | | | | | | | | | | | | | | RO | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:16* | *Reserved* | - | - |
| 15:0 | FCAP | Frequency error capture value | RO |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 10.5.10 TRIM Result Register (CTRIM_TVAL)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x24 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000100 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | TVAL | | | | | | | | | |
| Access | | | | | | | | | | | | | | | | | | | | | | | RO | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:16* | *Reserved* | - | - |
| 15:0 | TVAL | Auto TRIM result<br>**Note**<br>1. After auto TRIM completes, write the TVAL value to the corresponding TRIM register.<br>2. If reading during TRIM operation, two identical reads are required for validity. | RO |

### 10.5.11 Error Upper Limit Register (CTRIM_FLIM)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x28 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000010 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | FLIM | | | | | | | | | | | |
| Access | | | | | | | | | | | | | | | | | | | | | RW | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:12* | *Reserved* | - | - |
| 11:0 | FLIM | Error tolerance | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

# 11 SYSTICK

## 11.1 Introduction

If the OS wants to support multitasking, it needs to perform context switching periodically, so hardware resources such as timers are needed to interrupt program execution. When the timer interrupt is generated, the processor will perform OS task scheduling in exception handling, and will also perform OS maintenance work at the same time. The Cortex-M0+ processor includes a simple timer called SysTick, which is used to generate periodic interrupt requests.

SysTick is a 24-bit timer and counts down. After the count of the timer is reduced to 0, a programmable value will be reloaded, and a SysTick exception (exception number 15) will be generated at the same time. This abnormal event will cause the execution of SysTick exception handling, which is a part of the OS.

For systems that do not require an OS, the SysTick timer can also be used for other purposes, such as timing, time-counting, or providing an interrupt source for tasks that need to be executed periodically. The generation of SysTick exception is controllable. If the exception is disabled, the SysTick timer can still be used by polling, such as checking the current count value or polling the count flag.

## 11.2 Operation Example

### 11.2.1 Set SysTick

#### Context

Since the reload value and current value of the SysTick timer are undefined at reset, in order to prevent abnormal results, the configuration of SysTick needs to follow a certain process.

#### Procedure

**Step1.** Configure SysTick->CTRL.ENABLE to 0, disable SysTick.

**Step2.** Configure SysTick->CTRL.CLKSOURCE to select the clock source of SysTick.

**Step3.** Configure SysTick->LOAD, select the overflow period of SysTick.

**Step4.** Writing any value to SysTick->VAL, clear both SysTick->VAL and the SysTick->CTRL.COUNTFLAG.

**Step5.** Configure SysTick->CTRL.TICKINT to 1, enable SysTick interrupt.

**Step6.** Configure SysTick->CTRL.ENABLE to 1, enable SysTick.

**Step7.** Read SysTick->CTRL in the interrupt service routine to clear the overflow flag.
The SysTick overflow period is SysTick->LOAD + 1. The configuration example is as follows:

| Clock Source | SysTick->LOAD | Overflow Period |
|---|---|---|
| HCLK 4MHz | 3999 | 1ms |
| XTL 32.768kHz | 327 | 10.01ms |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

## 11.3 Register Description

### 11.3.1 Register List

Base address: 0xE000E010

Table 11-2 SysTick Timer Register Offset

| Offset | Register | CMSIS Symbol | Description | Width |
|---|---|---|---|---|
| 0x00 | SYS_CSR | SysTick->CTRL | SysTick Control and Status Register | 32 |
| 0x04 | SYS_RVR | SysTick->LOAD | SysTick Reload Register | 32 |
| 0x08 | SYS_CVR | SysTick->VAL | SysTick Current Value Register | 32 |
| 0x0C | SYS_CALIR | SysTick->CALIB | SysTick Calibration Value Register | 32 |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 11.3.2 SysTick Control and Status Register (CTRL)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | COUNTFLAG | Reserved | | | | | | | | | | | | | CLKSOURCE | TICKINT | ENABLE |
| Access | | | | | | | | | | | | | | | | RO | | | | | | | | | | | | | | RW | RW | RW |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:17* | *Reserved* | - | - |
| 16 | COUNTFLAG | SysTick timer overflow flag<br>Reading this register can clear COUNTFLAG flag<br>● 0b0: SysTick timer has not overflowed<br>● 0b1: SysTick timer has underflowed | RO |
| *15:3* | *Reserved* | - | - |
| 2 | CLKSOURCE | SysTick clock source selection<br>● 0b0: Use reference clock (XTL)<br>● 0b1: Use core clock (HCLK)<br>**Note**<br>When the reference clock is selected as the SysTick clock source, the XTL clock source must be enabled synchronously. | RW |
| 1 | TICKINT | SysTick interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 0 | ENABLE | SysTick timer enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 11.3.3 SysTick Reload Register (LOAD)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x04 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0xXXXXXXXX | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | RELOAD | | | | | | | | | | | | | | | |
| Access | | | | | | | | | | | | | | | | | RW | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:24* | *Reserved* | - | - |
| 23:0 | RELOAD | SysTick timer reload value | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 11.3.4 SysTick Current Value Register (VAL)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x08 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0xXXXXXXXX | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | CURRENT | | | | | | | | | | | | | | | |
| Access | | | | | | | | | | | | | | | | | RW | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:24* | *Reserved* | - | - |
| 23:0 | CURRENT | Current count value of the SysTick timer<br>Read the register to obtain the current count value of the SysTick timer.<br>Writing any value to the register will clear both the register itself and the COUNTFLAG. | RW |

### 11.3.5 SysTick Calibration Value Register (CALIB)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x0C | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0xXXXXXXXX | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | NOREF | SKEW | Reserved | | | | | | | | | | | | | | TENMS | | | | | | | | | | | | | | | |
| Access | RO | RO | | | | | | | | | | | | | | | RO | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 31 | NOREF | Whether the SysTick count clock supports the reference clock flag<br>● 0b0: Support reference clock (XTL)<br>● 0b1: Does not support reference clock (XTL)<br>**Note**<br>When this bit is set to 1, CTRL.CLKSOURCE must be set to 1. | RO |
| 30 | SKEW | TENMS accuracy indication<br>● 0b0: TENMS value represents an exact 10ms<br>● 0b1: TENMS value represents an approximate 10ms | RO |
| *29:24* | *Reserved* | - | - |
| 23:0 | TENMS | 10 millisecond calibration value | RO |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

# 12 LPTIM

## 12.1 Introduction

The LPTimer is an asynchronous 16-bit timer/counter that can continue timing/counting via the internal low-speed RC oscillator, external low-speed crystal oscillator, or ETR external input even after the system clock is turned off. Wake up the system in low power mode through interrupts.

This module includes 2 low power timers, which can be cascaded internally.

## 12.2 Features

- 16-bit asynchronous counter/timer
- Optional counting/timing clock sources: RCL/PCLK/XTL/ETR
- 4 bits prescaler configuration register, supporting a maximum prescaler value of 32768
- PWM pulse output
- ARR reload configurable
- Overflow interrupt wake-up

## 12.3 Functional Description

### 12.3.1 Function Block Diagram

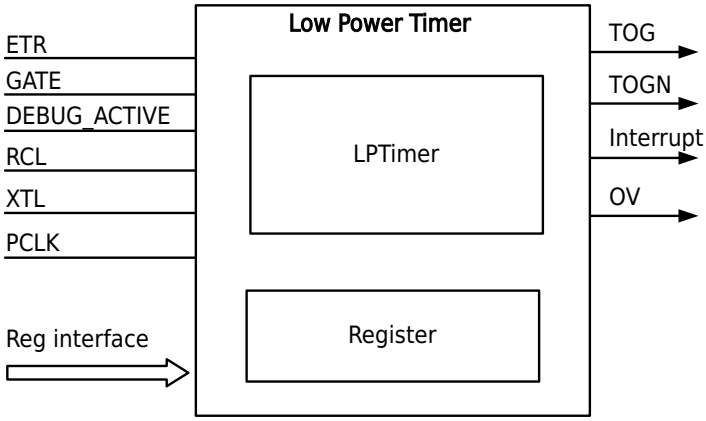



**Figure 12-1 LPTimer Block Diagram**

### 12.3.2 Working Mode

LPTimer supports 2 working modes, each timer/counter has an independent control start signal, as well as external input clock and gating signal.

LPTimer utilizes ETR (External Trigger) and GATE for counting operations:

- ETR serves as an external input clock signal for the counter and can also be used for internal clock counting.
- GATE acts as an active-level counting enable signal.

The timer of LPTimer supports two working modes. The working mode can be selected by setting CR.MD: Mode 1 is a 16-bit free-counting mode, and Mode 2 is a 16-bit overloading mode.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

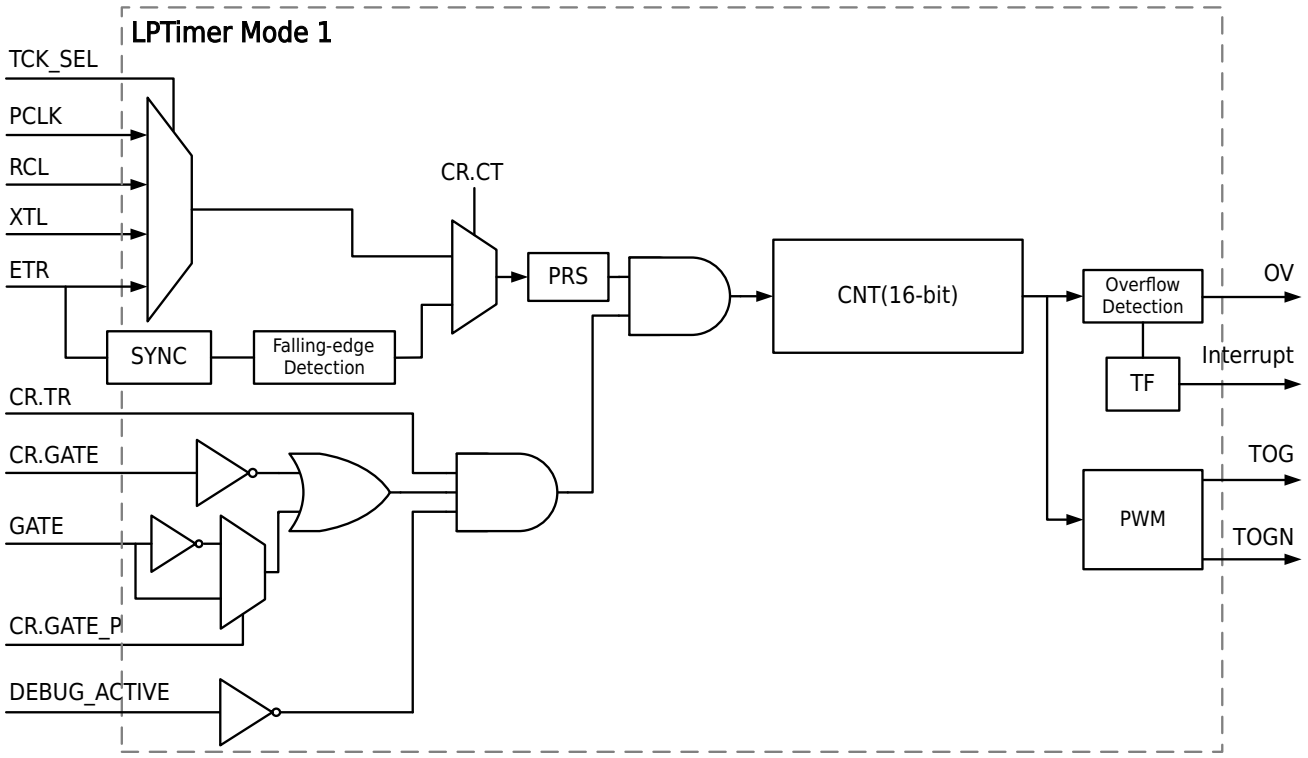



Figure 12-2 LPTimer Mode 1

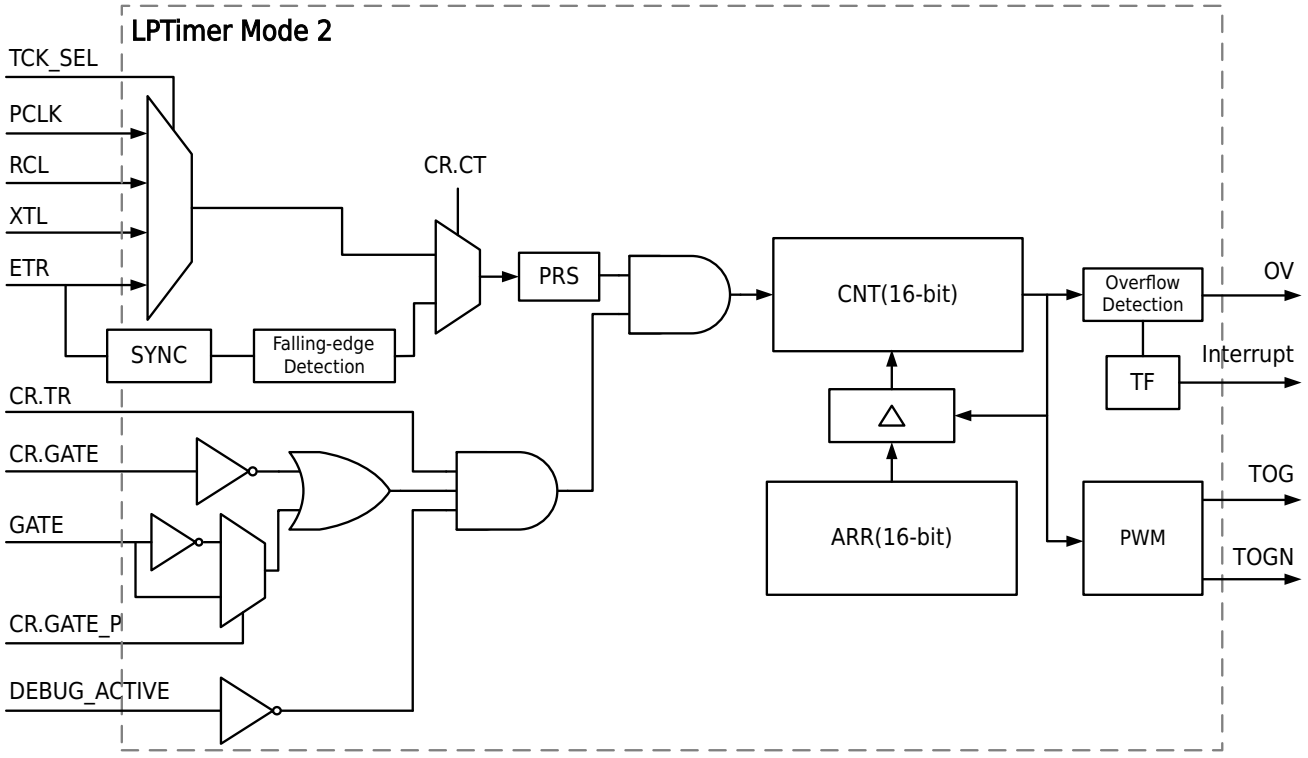



Figure 12-3 LPTimer Mode 2

When LPTimer starts, it will automatically load the value of the reload register ARR into the counter. LPTimer can select 4 clock sources as the timer clock via control register CR.TCK_SEL (default: PCLK). The clock selection is as shown in the table:

| TCK_SEL | Timer Clock | Read Timer Count Value |
|---|---|---|
| 0b00 | PCLK | Read synchronized |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| TCK_SEL | Timer Clock | Read Timer Count Value |
|---|---|---|
| 0b01 | ETR | Read synchronized |
| 0b10 | XTL | Read synchronized |
| 0b11 | RCL | Read synchronized |

When the corresponding timer TR is set to 1, the timer starts to run. The counter starts counting from the set value, and generates an overflow interrupt after counting to a maximum of 0xFFFF. After Mode 1 generates an interrupt, the counter is cleared and continues to count up. After Mode 2 generates an interrupt, the value of the reload register ARR is transferred to the counter and counts up. The initial value of the counter CNT is automatically loaded by ARR when the timer is started.

**Note**

When setting CR.CT=1 to count the falling edge event of ETR, at this time, the counting clock source cannot select the ETR.

### 12.3.3 Counting Function

Counting functions are used to determine the number of times an event occurs. In the counting function, the counter increments every falling edge of the corresponding input clock. The input signal is sampled by the internal count clock, so the external input clock frequency cannot exceed the system count clock. Counting to the maximum value will overflow and generate an interrupt.

### 12.3.4 Timing Function

The timing function is used to generate interval timing. In the timing function, the timer accumulates once per clock, and when the count reaches the maximum value, it will overflow and generate an interrupt.

The timing clock cycle count is (0xFFFF-ARR+0x0001) (when reload mode is enabled).

### 12.3.5 PWM Pulse Output

The duty cycle can be adjusted by setting the value of LPTIM_CMP:

- When the overload mode (LPTIM_CR.MD=1) is enabled, the duty cycle is calculated as (0xFFFF-CMP)/(0xFFFF-ARR+0x0001).
- When the overload mode (LPTIM_CR.MD=0) is disabled, except for the duty cycle of the first cycle after the timer is enabled, which is (0xFFFF-CMP)/(0xFFFF-ARR+0x0001), the other duty cycles are (0xFFFF-CMP)/(0xFFFF+0x0001).

The following figure illustrates an example of disabling the GATE function in the timer mode (LPTIM_CR.CT=0&LPTIM_CR.GATE=0). TOG will start counting from the next valid rising edge after reaching the CMP value, and the counter is cleared to 0 when it reaches the maximum value. TOGN is the logically inverted signal of TOG.

**Note**

When counting ETR with the internal clock, there is a possibility of deviation in PWM accuracy, which is related to the frequency and phase of ETR.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

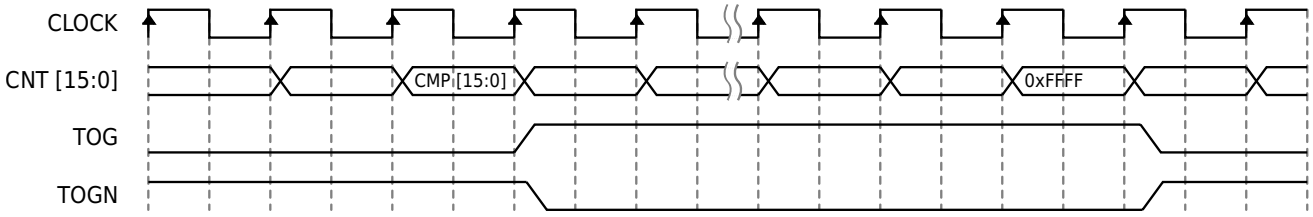



Figure 12-4 PWM Pulse Output Diagram

### 12.3.6 GATE Interconnection

When LPTIMx_G=0b000, the gate control GATE input is the port input selected by Pxy_SEL. When LPTIMx_G=0b001-0b111, it is connected to the input or output of other modules. For details, refer to the descriptions of the registers in the "General-Purpose I/O (GPIO)" chapter: "Port Auxiliary Function Timer Gate Selection (GPIO_TIMGS)", "Port Auxiliary Function Timer Capture Input Selection 1 (GPIO_TIMCPS1)" and "Port Pxy Function Configuration Register (Pxy_SEL)".

### 12.3.7 ETR Interconnection

When LPTIMx_E=0b000, the ETR input is the port input selected by Pxy_SEL. When LPTIMx_E=0b001-0b111, it is connected to the input or output of other modules. For details, refer to the descriptions of the registers in the "General-Purpose I/O (GPIO)" chapter: "Port Auxiliary Function Timer ETR Selection (GPIO_TIMES)", and "Port Auxiliary Function Timer Capture Input Selection 1 (GPIO_TIMCPS1)" and "Port Pxy Function Configuration Register (Pxy_SEL)".

Set the external input LPTIM1_E of the LPTIM1 timer to 0b101, the overflow output LPTIM0_OV signal of LPTIM0 is selected to be output from inside the chip to LPTIM1, as the external clock input of LPTIM1, which is set to external counting mode. In this way, two timers can be used in cascade.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

## 12.4 Register Description

### 12.4.1 Register List

- LPTimer0 base address: 0x40000C00
- LPTimer1 base address: 0x40003C00

Table 12-2 Low Power Timer (LPTIM) Register Offset

| Offset | Register | Description | Width |
|---|---|---|---|
| 0x000 | LPTIM_CNT | Counter Count Value Register | 32 |
| 0x004 | LPTIM_ARR | Reload Register | 32 |
| 0x008 | LPTIM_CMP | Compare Register | 32 |
| 0x00C | LPTIM_CR | Control Register | 32 |
| 0x010 | LPTIM_IFR | Interrupt Flag Register | 32 |
| 0x014 | LPTIM_ICLR | Interrupt Flag Clear Register | 32 |

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 12.4.2 Counter Count Value Register (LPTIM_CNT)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x000 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | CNT | | | | | | | | | | | | | | | |
| Access | | | | | | | | | | | | | | | | | RO | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:16* | *Reserved* | - | - |
| 15:0 | CNT | Timer count value | RO |

### 12.4.3 Reload Register (LPTIM_ARR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x004 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | ARR | | | | | | | | | | | | | | | |
| Access | | | | | | | | | | | | | | | | | RW | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:16* | *Reserved* | - | - |
| 15:0 | ARR | Timer reload register<br>Before writing the ARR register, the CR.WT_FLAG must be read. Data can be written only when WT_FLAG is 1. After writing to the ARR register, CR.WT_FLAG is set to 0.<br>**Note**<br>This value will be loaded into the timer counter when the timer is started or reloaded. The timer counter value can be soft-reset via CR.SYNCCOUNTRST. Setting ARR to 0xFFFF is not recommended to prevent the count counter from remaining at 0xFFFF indefinitely. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 12.4.4 Compare Register (LPTIM_CMP)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x008 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | CMP | | | | | | | | | | | | | | | |
| Access | | | | | | | | | | | | | | | | | RW | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:16* | *Reserved* | - | - |
| 15:0 | CMP | Timer compare register<br>CR.WT_FLAG needs to be read before writing CMP, and data can be written when WT_FLAG is 1. After writing to the CMP register, CR.WT_FLAG is set to 0.<br>**Note**<br>If ARR is configured first causes CR.WT_FLAG to be set to 0, subsequent modifications can be made without constraints. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 12.4.5 Control Register (LPTIM_CR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x00C | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000088 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | SYNCCOUNTRST | PRS | | | | IE | GATE_P | GATE | WT_FLAG | Reserved | TCK_SEL | TOG_EN | CT | MD | TR |
| Access | | | | | | | | | | | | | | | | | RW | RW | | | | RW | RW | RW | RO | | RW | RW | RW | RW | RW |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:16* | *Reserved* | - | - |
| 15 | SYN-CCOUNTRST | Counter synchronization reset<br>Write 1 to reset the counter synchronously, write 0 has no effect<br>Read:<br>● 0b0: Idle<br>● 0b1: In reset process<br>After setting the software to 1, LPTIM_CNT is cleared to 0 (synchronization time takes 2 to 3 counter clock).<br>**Note**<br>● As LPTimer is asynchronous, synchronizing software configurations to the counter clock domain requires 2 to 3 counter clock. It is recommended that when TR is disabled, if LPTIM_CNT.CNT is greater than or equal to 0xFFFC, perform a synchronous reset to the CNT counter using this register before enabling the timer to avoid unexpected overflow phenomena. When TOG output is enabled (LPTIM_CR.TOG_EN=1) and the timer is stopped, if LPTIM_CNT.CNT is within the range of LPTIM_CMP.CMP-3 to LPTIM_CMP.CMP, perform a synchronous reset to the CNT counter before enabling the timer to avoid unexpected PWM phenomena.<br>● When the timer is stopped, ensure this bit is 0 before enabling the timer (LPTIM_CR.TR=1). | RW |
| 14:11 | PRS | Prescaler<br>Cannot be changed while the timer is running<br>● 0b0000: 1<br>● 0b0001: 2<br>● 0b0010: 4<br>● 0b0011: 8<br>● 0b0100: 16<br>● 0b0101: 32<br>● 0b0110: 64<br>● 0b0111: 128<br>● 0b1000: 256<br>● 0b1001: 512<br>● 0b1010: 1024 | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 14:11 | PRS | ● 0b1011: 2048<br>● 0b1100: 4096<br>● 0b1101: 8192<br>● 0b1110: 16384<br>● 0b1111: 32768 | RW |
| 10 | IE | Interrupt enable control<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 9 | GATE_P | GATE polarity control<br>● 0b0: High level is valid<br>● 0b1: Low level is valid | RW |
| 8 | GATE | Timer gating<br>● 0b0: No gate control, the timer works when TR=1<br>● 0b1: The timer operates only when the gate is set to 1 and TR=1 | RW |
| 7 | WT_FLAG | WT write synchronization flag<br>● 0b0: Synchronizing, writing ARR/CMP is invalid<br>● 0b1: Synchronization is complete, ARR/CMP can be changed at this time | RO |
| *6* | *Reserved* | - | - |
| 5:4 | TCK_SEL | LPTimer clock selection<br>● 0b00: PCLK<br>● 0b01: ETR<br>● 0b10: XTL<br>● 0b11: RCL | RW |
| 3 | TOG_EN | PWM output enable<br>● 0b0: TOG, TOGN output 0 at the same time<br>● 0b1: TOG, TOGN output signals with opposite phases | RW |
| 2 | CT | Counter/timer function selection<br>● 0b0: Timer function, the timer uses the clock selected by TCK_SEL to count<br>● 0b1: Counter function, the counter uses the falling edge of the external input to count<br>**Note**<br>The sampling clock uses the clock selected by TCK_SEL, and the external input clock is lower than 1/2 sampling clock. | RW |
| 1 | MD | Timer working mode<br>● 0b0: Mode 1 no reload mode 16-bit counter/timer<br>● 0b1: Mode 2 auto-reload 16-bit counter/timer | RW |
| 0 | TR | Timer run control bit<br>● 0b0: Timer stop<br>● 0b1: Timer run | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 12.4.6 Interrupt Flag Register (LPTIM_IFR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x010 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | OV | |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | RO | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:1* | *Reserved* | - | - |
| 0 | OV | Counter overflow flag<br>● 0b0: No overflow<br>● 0b1: Overflow | RO |

### 12.4.7 Interrupt Flag Clear Register (LPTIM_ICLR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x014 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | OV | |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | WO | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:1* | *Reserved* | - | - |
| 0 | OV | Counter overflow flag cleared<br>Write 0 to clear flag, write 1 has no effect; read always returns 0. | WO |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

# 13 TIM0/1/2/3

## 13.1 Introduction

TIM0/1/2 is a timer composed of 1 counting unit and 2 comparing units respectively. Each timer supports 2 independent PWM outputs or 1 pair of complementary PWM outputs plus 1 independent PWM output, with support for 2 capture inputs. TIM0/1/2 can form 3 pairs of complementary PWM outputs.

TIM3 is a timer composed of 1 counting unit and 8 comparison units, supporting 6 independent PWM outputs or 3 pairs of complementary PWM outputs, with support for 6 capture inputs. Additionally, there are 2 dedicated comparison units that can generate compare interrupts and be used to trigger ADC and DMA operations.

Using timer prescaler, system prescaler and system clock selection, the pulse width and waveform period can be flexibly adjusted; the pulse width can be measured conveniently.

## 13.2 Features

### 13.2.1 TIM0/1/2 Features

- Two independent PWM outputs CHA, CHB
- One complementary PWM output (CHA, CHB)+one independent PWM output (gate)
- One complementary PWM output (CHA, CHB)+one capture function (gate)
- Up to two capture inputs
- Pulse width measurement
- Quadrature encoder counting function
- Single pulse mode
- External counting function
- DMA Trigger
- Dead-time control
- Braking control
- Cycle-by-cycle current limiting
- Edge alignment, symmetric center alignment and asymmetric center alignment PWM output


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

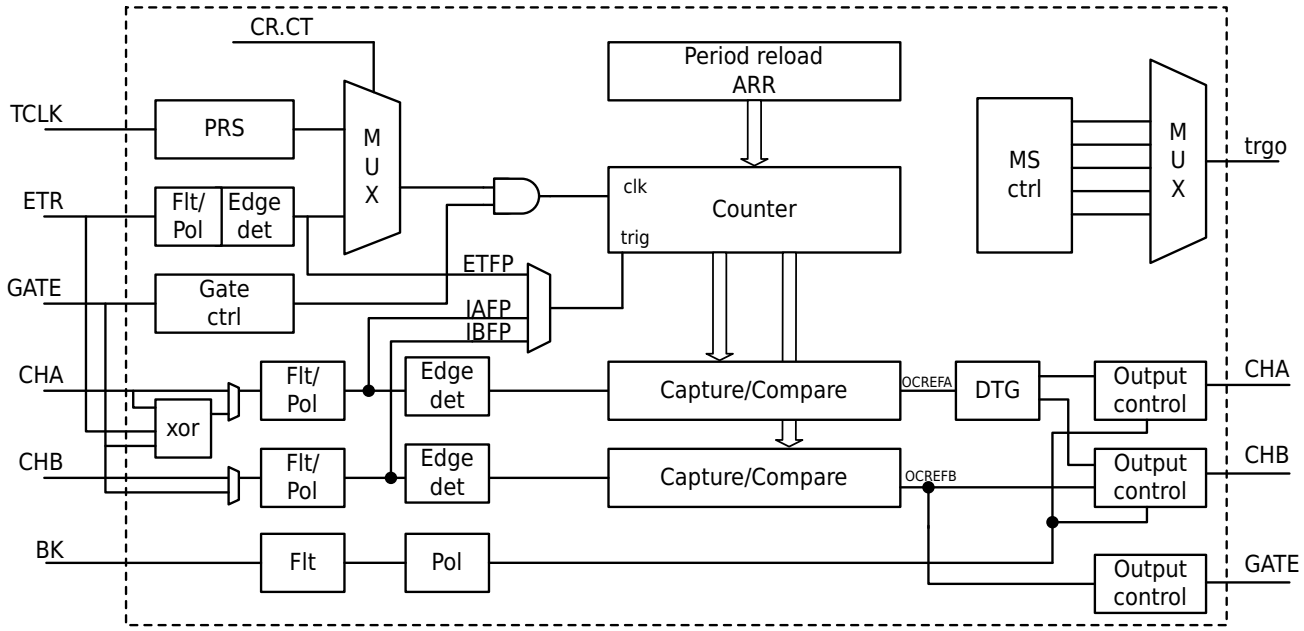



Figure 13-1 TIM0/1/2 Block Diagram

## 13.2.2 TIM3 Features

- 6 independent PWM outputs CH0A, CH0B, CH1A, CH1B, CH2A, CH2B
- 3 complementary PWM outputs(CHxA, CHxB)+1 independent PWM output (gate)
- 3 complementary PWM outputs(CHxA, CHxB)+1 capture function (gate)
- Two dedicated comparison unit, triggers ADC and DMA
- Up to 6 capture inputs
- Pulse width measurement
- Quadrature encoder counting function
- Single pulse mode
- External counting function
- DMA trigger
- Dead-time control
- Braking control
- Cycle-by-cycle current limiting
- Edge alignment, symmetric center alignment and asymmetric center alignment PWM output


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

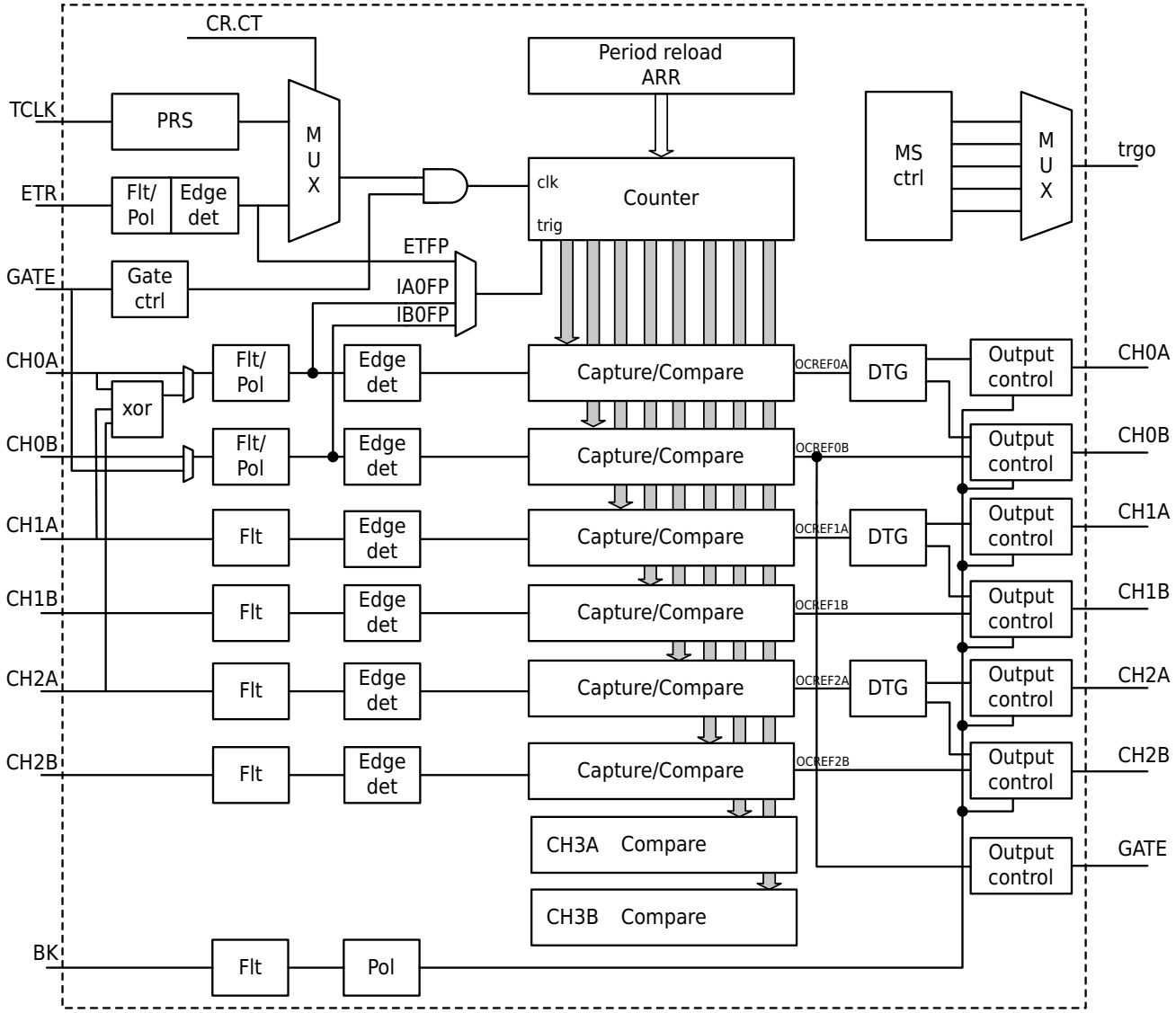



Figure 13-2 TIM3 Block Diagram

## 13.3 Functional Description

### 13.3.1 Timer Clock

The timer uses PCLK as the timer clock, and the timer clock is marked with TCLK below.

### 13.3.2 Timer Counter

The main module of programmable general timer consists of a 16-bit counter and its associated auto-reload register. The counter can count up, count down, or alternately count up and down. The counter clock can be divided by a prescaler.

The counter, auto-reload register and prescaler register can be read and written by software. Read and write operations can be performed even when the counter is running.

### 13.3.3 Timer Prescaler

When using TCLK as the timer clock, the prescaler can be used. The prescaler settings are as follows:


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| PRS | 0b000 | 0b001 | 0b010 | 0b011 | 0b100 | 0b101 | 0b110 | 0b111 |
|---|---|---|---|---|---|---|---|---|
| Frequency division ratio | 1 | 2 | 4 | 8 | 16 | 32 | 64 | 256 |

The prescaler does not have a preloaded buffer, so any changes to the prescaler will take effect immediately.

### 13.3.4 Mode 0 Count Timer Function

The counter counts up in this mode. The counter supports two counting modes, reload mode and free counting mode, and you can choose external clock ETR counting or system clock counting. The gate can control the count mask. Counting to the maximum overflow generates an interrupt. The inverted output signals TOG/TOGN are output through ports CHA/CHB.

The counting range of the reload mode counts up from ARR to 0xFFFF overflow, and then starts counting from ARR, and the counting period is 0xFFFF-ARR+1; the free counting mode overflows after counting from the set count value to 0xFFFFFFFF, and the count value restarts count from 0x0 after overflow.

The counter can be used to control whether the counter counts through the external gate control function. The control relationship is as follows:

| M0CR.GATE | M0CR.GATEP | Port GATE Input | M0CR.CTEN | Counter |
|---|---|---|---|---|
| × | × | × | 0 | Not count |
| 0 | × | × | 1 | Count |
| 1 | 0 | High level | 1 | Count |
| 1 | 0 | Low level | 1 | Not count |
| 1 | 1 | High level | 1 | Not count |
| 1 | 1 | Low level | 1 | Count |

#### Function Block Diagram

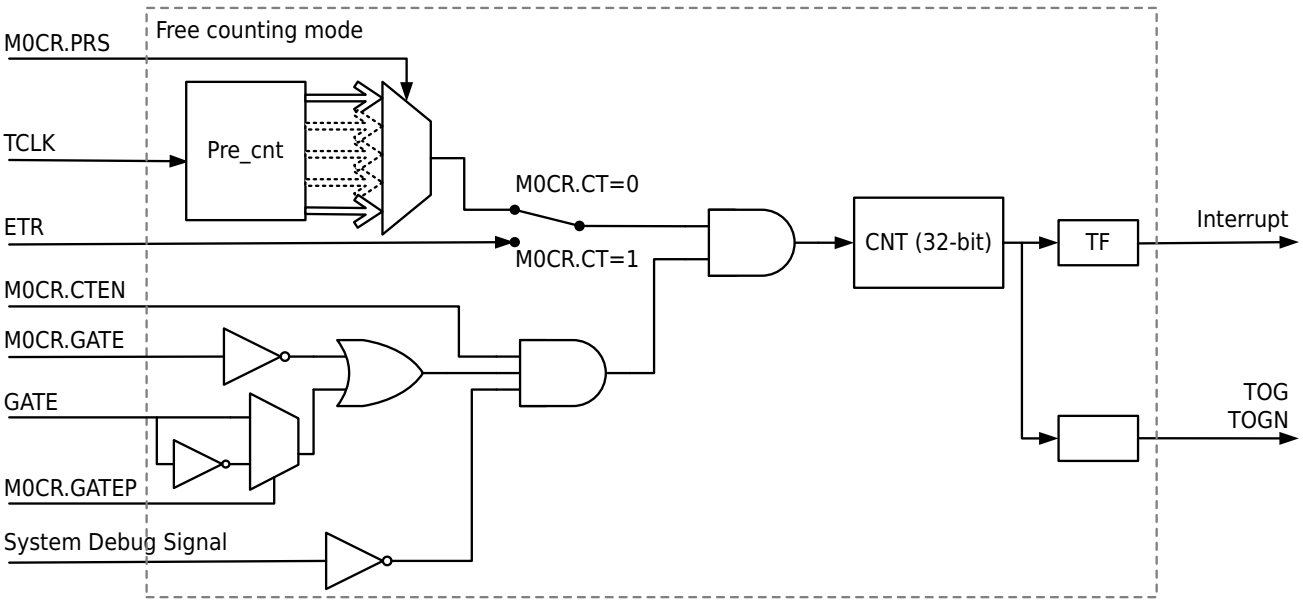



Figure 13-3 Mode0 Count Timer Free Counting Block Diagram


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

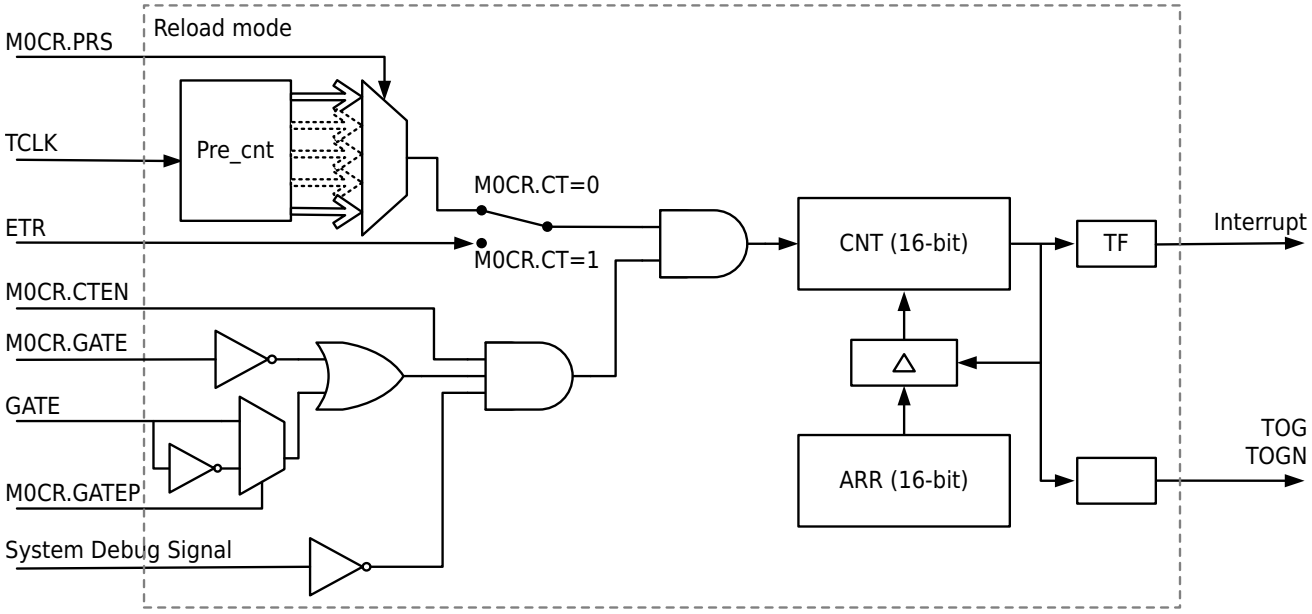



Figure 13-4 Mode0 Count Timer Reload Counting Block Diagram

**Note**

The "System Debug Signal" in the block diagram originates from DEBUG_ACTIVE.

## Count Waveform

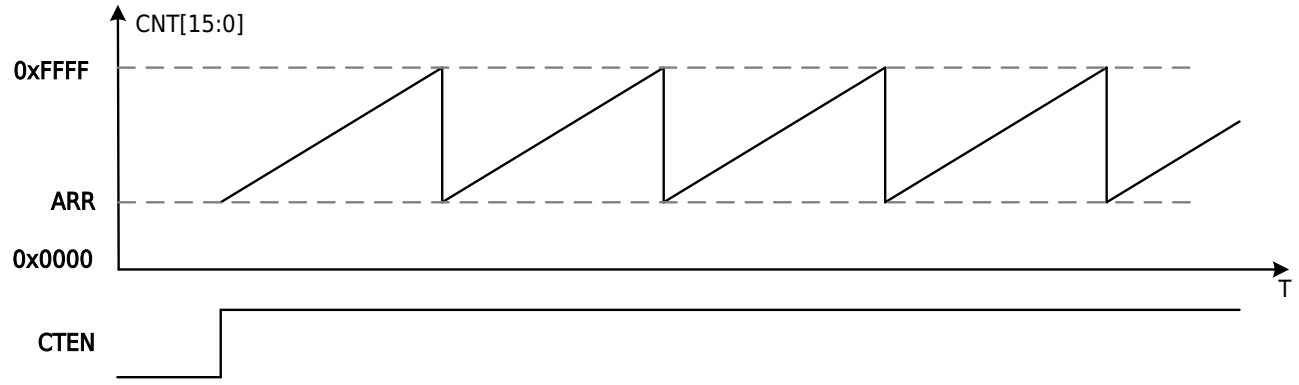



Figure 13-5 Mode0 Count Timer 16-Bit Reload Counting Waveform

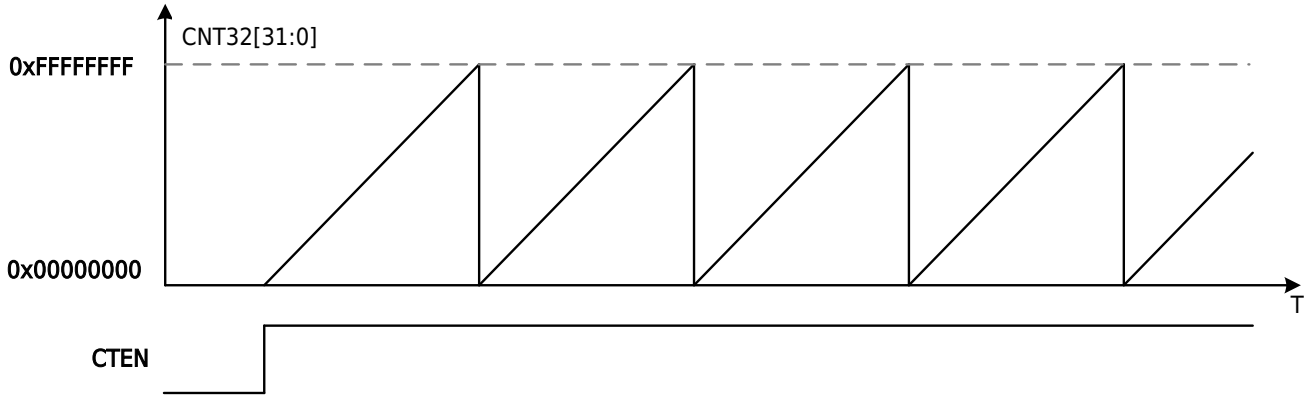



Figure 13-6 Mode0 Count Timer 32-Bit Free Counting Waveform


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### Counting Function

Counting function are used to determine the number of times an event occurs. In the counting function, the counter increments every falling edge of the corresponding input clock. The input signal is sampled by the internal TCLK. Therefore, the external input signal must remain stable for a duration longer than one TCLK clock cycle. Counting to the maximum value will overflow and generate an interrupt. The interrupt flag needs to be cleared by software.

**Notice**

When the ETR input serves as the count clock of the counter, the ETR clock frequency must not exceed 1/2 of the TCLK clock.

### Timing Function

The timing function is used to generate interval timing. In the timing function, the timer has a pre-divided frequency, and the timer accumulates once per clock of each pre-divided frequency. When counting to the maximum value, it will overflow and generate an interrupt. The interrupt flag needs to be cleared by software.

### Timing Diagram

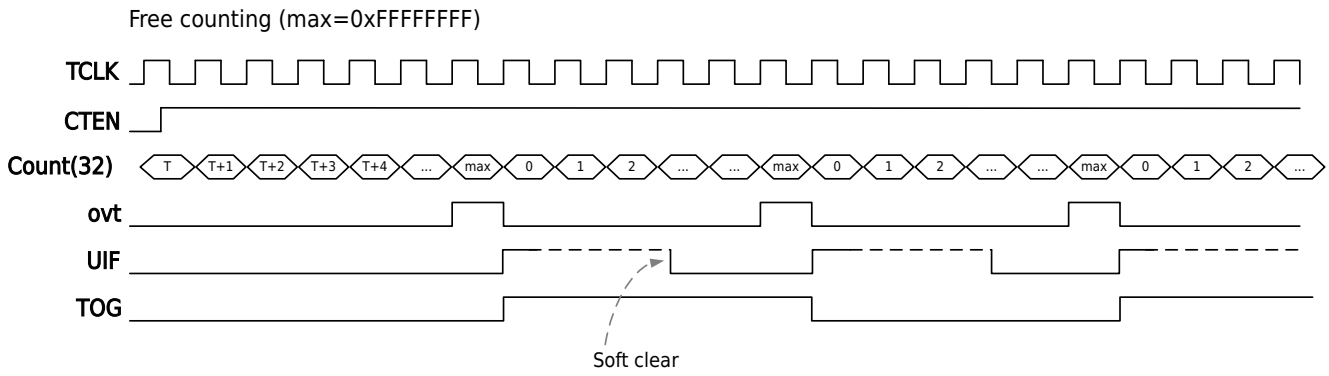



**Figure 13-7 Mode0 Count Timer Free Counting Timing Diagram**

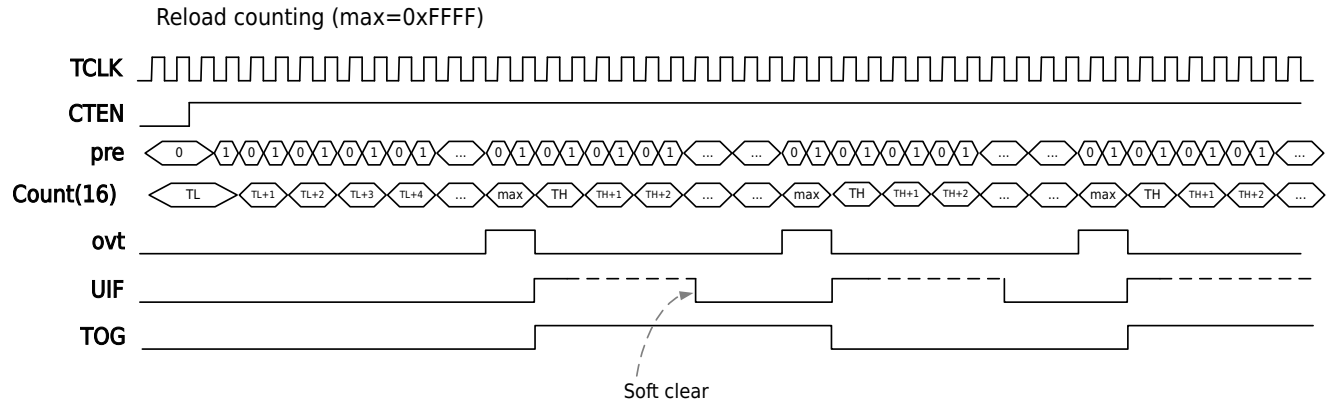



**Figure 13-8 Mode0 Count Timer Reload Counting Timing Diagram (Prescaler=2)**

### Buzzer Function

The Buzzer function can be realized through the toggling output function of the timer. To use toggle output, the DTR.MOE control bit must be enabled. Setting M0CR.TOGEN to 0 simultaneously sets both CHA and CHB outputs to 0. The timer reload mode configurations for Buzzer output at different frequencies under a 4MHz counting clock are as follows:

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Buzzer Frequency | Counter Period | Counter Count Value | Counter Reload Value | CNT Initial Value | ARR Reload Value |
|---|---|---|---|---|---|
| 1000Hz | 0.5ms | 2000 | 63536 | 0xF830 | 0xF830 |
| 2000Hz | 0.25ms | 1000 | 64536 | 0xFC18 | 0xFC18 |
| 4000Hz | 0.125ms | 500 | 65036 | 0xFE0C | 0xFE0C |

### 13.3.5 Mode 1 Pulse Width Measurement PWC

In this mode, the high level and low level or period width of the input pulse can be automatically measured.

The first valid edge counter initializes to 0x0001, the second valid edge will stop counting, store the current count value in CCR0A, generate a capture interrupt CA0F, and if the counter overflows, an overflow flag will be raised. Setting overflow interrupt enable will generate overflow interrupt.

| | | | | |
|---|---|---|---|---|
| M1CR.EDG1ST | 0 | 0 | 1 | 1 |
| M1CR.EDG2ND | 0 | 1 | 0 | 1 |
| Pulse width measurement | Upper edge-upper edge<br>Cycle width | Upper edge-lower edge<br>High level width | Lower edge-upper edge<br>Low level width | Lower edge-lower edge<br>Cycle width |

During period measurement, a period is measured at intervals of one period.

#### PWC Function Block Diagram

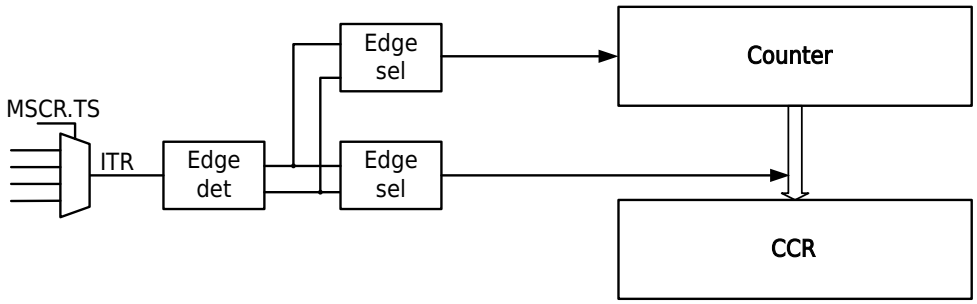



**Figure 13-9 Mode1 Pulse Width Measurement PWC Function Block Diagram**

Select the measurement signal source by registering MSCR.TS.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Trigger Selection | Value | Signal Source |
|---|---|---|
| MSCR.TS | 0b000 | ETFP: ETR filtered external input phase selection signal, selectable between external filtered and input inverted phase |
| | 0b001 | ITR0: Timer internal interconnection signal 0, TRGO output from other timers |
| | 0b010 | ITR1: Timer internal interconnection signal 1, TRGO output from other timers |
| | 0b011 | ITR2: Timer internal interconnection signal 2, TRGO output from other timers |
| | 0b100 | ITR3: Timer internal interconnection signal 3, TRGO output from other timers |
| | 0b101 | IA0ED: Invalid |
| | 0b110 | IAFP: CH0A filtered external input phase selection signal, selectable between external filtered and input inverted phase (polarity selection invalid in pulse width measurement mode) |
| | 0b111 | IBFP: CH0B filtered external input phase selection signal, selectable between external filtered and input inverted phase (polarity selection invalid in pulse width measurement mode) |

For TRGO output, refer to the corresponding register description.

| Timer | ITR0 | ITR1 | ITR2 | ITR3 |
|---|---|---|---|---|
| Timer0 | - | TIM1_TRGO | TIM2_TRGO | TIM3_TRGO |
| Timer1 | TIM0_TRGO | - | TIM2_TRGO | TIM3_TRGO |
| Timer2 | TIM0_TRGO | TIM1_TRGO | - | TIM3_TRGO |
| Timer3 | TIM0_TRGO | TIM1_TRGO | TIM2_TRGO | - |

### PWC Waveform Measurement Timing Diagram

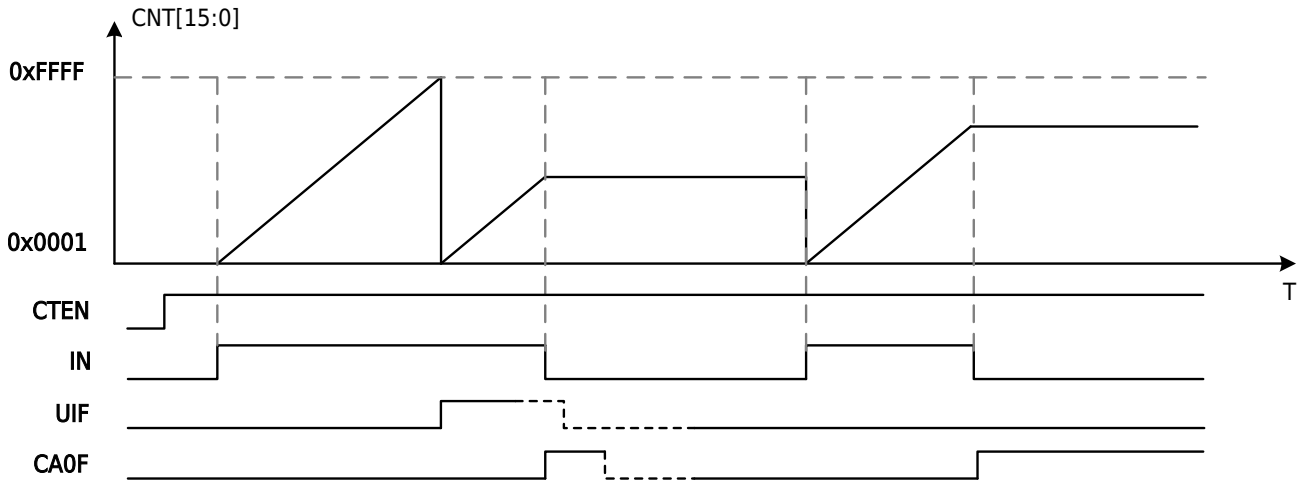



Figure 13-10 Mode1: Pulse Width Measurement PWC High-Level Pulse Width Measurement


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

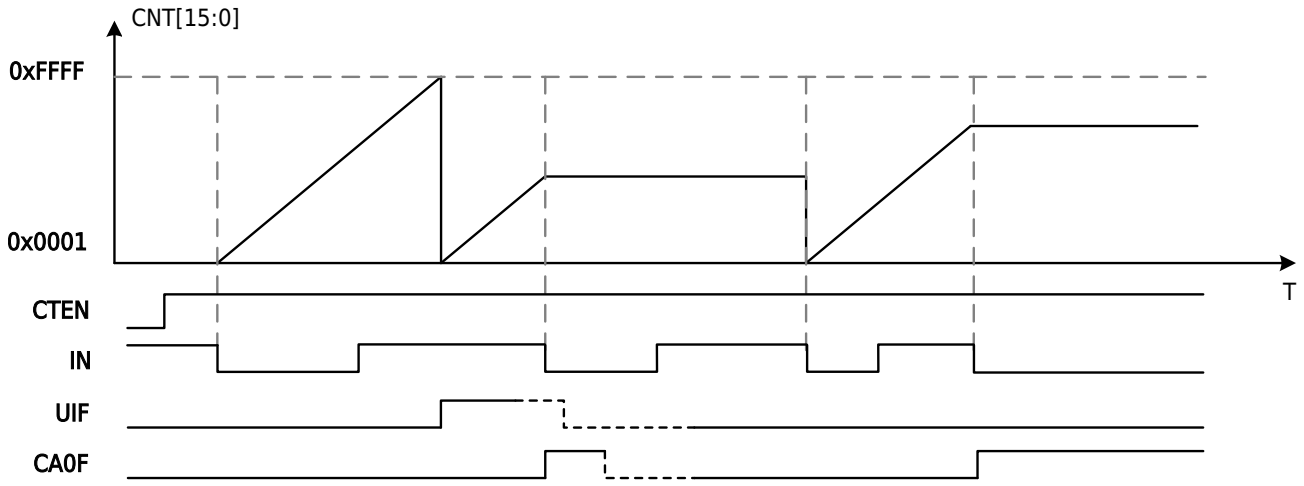



Figure 13-11 Mode1: Pulse Width Measurement PWC Falling Edge to Falling Edge Period Measurement

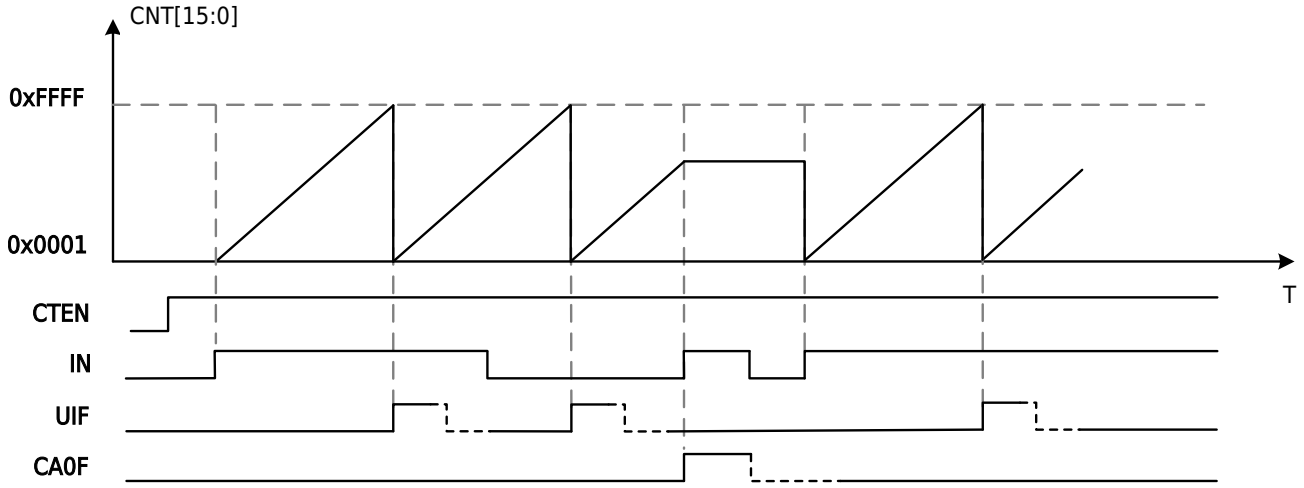



Figure 13-12 Mode1: Pulse Width Measurement PWC Rising Edge to Rising Edge Period Measurement

### PWC Single trigger Mode

Set M1CR.ONESHOT=1 to set PWC single measurement, and CTEN will be cleared after the measurement is completed.

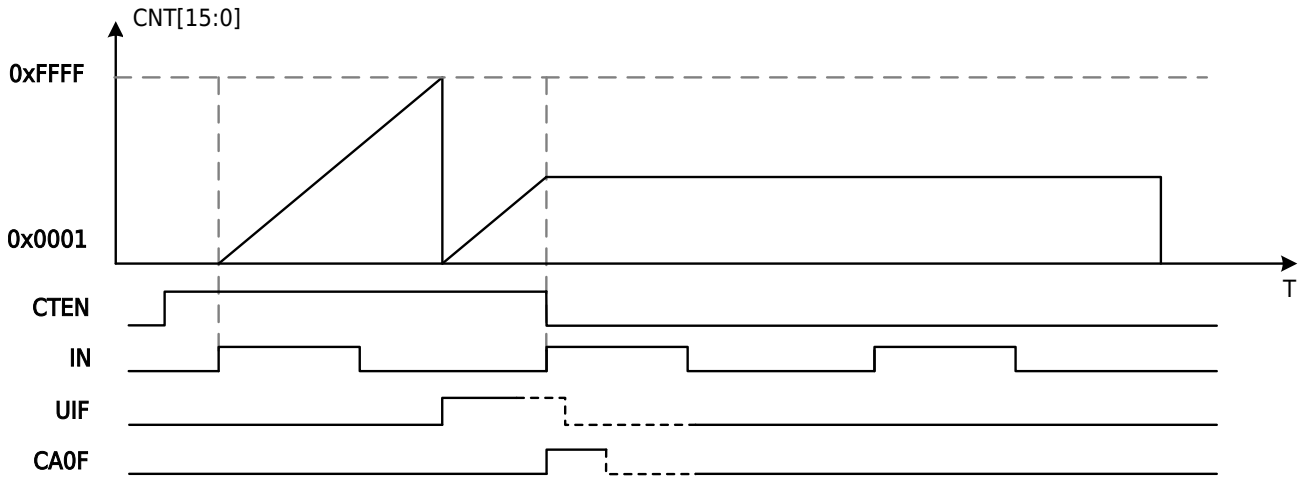



Figure 13-13 Mode1: Pulse Width Measurement PWC Rising Edge to Rising Edge Period Measurement Single Mode


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

## 13.3.6 Mode2/3 Compare Capture Mode

### 13.3.6.1 Counter

The main part of the counter is a 16-bit counter with associated auto-loaded register. This counter can count up (mode 2), count down (mode 2) or count up and down in both directions (mode 3). The clock of the counter can be divided by the prescaler PRS, or ETR can be selected to input an external clock or the external input signal and internal interconnection signal selected by MSCR.TS.

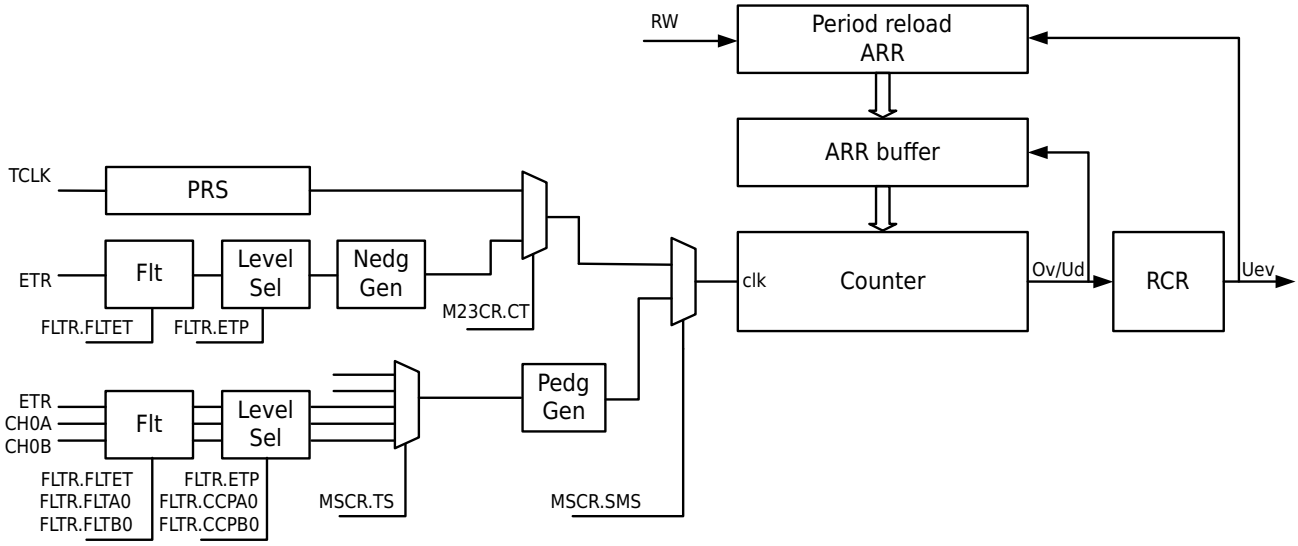



Figure 13-14 Mode 2/3 Compare Capture Mode Counter Block Diagram

The basic counter unit includes:

- Counter register CNT
- Prescaler register M23CR.PRS
- Auto-reload register ARR
- Repeat count register RCR
- Clock selection control registers FLTR, M23CR, MSCR

The auto-loaded register has a cache function, and the reload value is updated from the buffer register to the counter after the counter generates an event update. When the counter is in the stopped state or the cache function is disabled, the auto-reload register is immediately updated to the buffer register. When the timer is in the running state and the cache function is enabled, the value written to the auto-reload register is not immediately updated to the buffer register. The auto-reload register updates to the buffer register only after an event update occurs.

For clock selection and gating functions, trigger functions, and reset functions, refer to the relevant descriptions in the Mode2/3 Slave Mode chapter.

### 13.3.6.2 Counter Waveform

#### Mode 2 Sawtooth Counting Waveform

Mode 2 operates as a sawtooth counting waveform, and the counting direction can be changed by configuring M23CR.DIR.

When M23CR.DIR is set to 0, the counter operates in up-counting mode. In this mode, the counter increments from 0 to the auto-reload value (TIMx_ARR), then start counting from 0 again and generates a counter overflow event. If the repeat counter is used, an update event (UEV) is generated when the


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

number of up-counting repetitions reaches the programmed value in the repeat counter register plus one (TIMx_RCR+1). Otherwise, an update event is generated on every counter overflow.

An update event is also generated when M23CR.UG is set to 1 (either by software or using the slave mode controller).

When the update event occurs, all registers will be updated and the update flag (TIMx_IFR.UIF bit) will be set to 1 (depending on the M23CR.URS bit):

- The auto-reload cache value will be updated with the ARR register value
- The compare buffer values will be updated with the compare register CCRxy.

The following diagrams illustrate the counter waveforms for different counting directions when ARR = 0x2C. The number of count overflow period clocks is ARR+1.

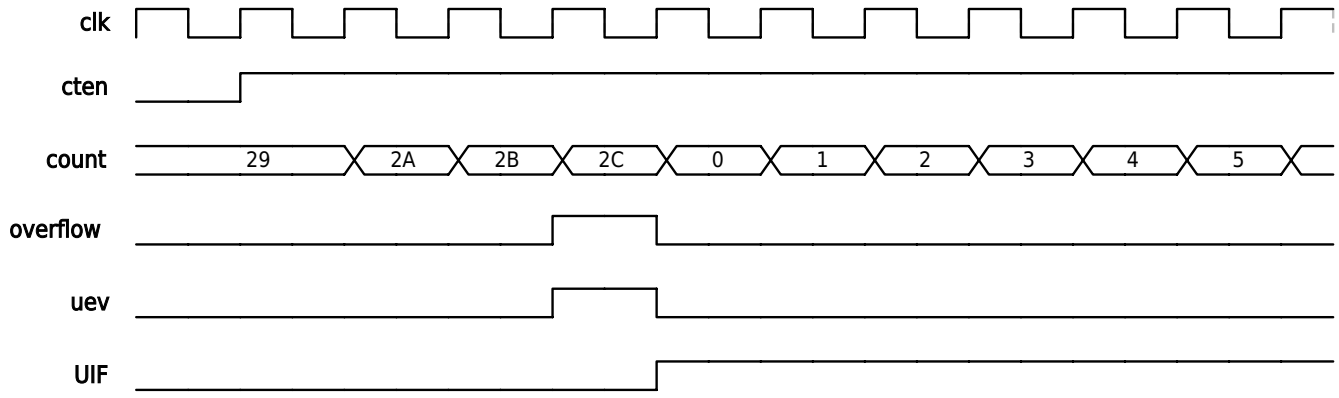



Figure 13-15 Mode 2 Compare Capture Mode Up-Counting Without Prescaler

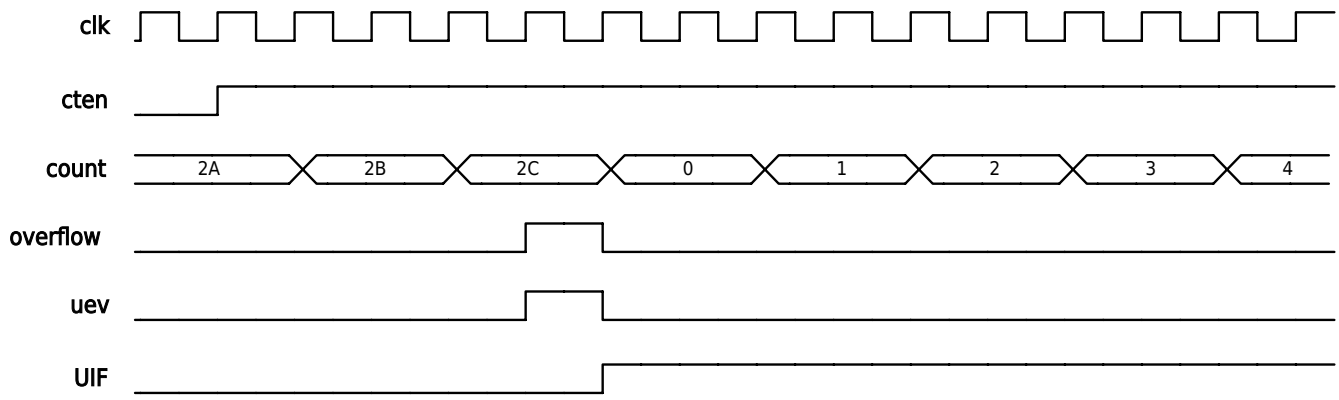



Figure 13-16 Mode 2 Compare Capture Mode Up-Counting With Prescaler

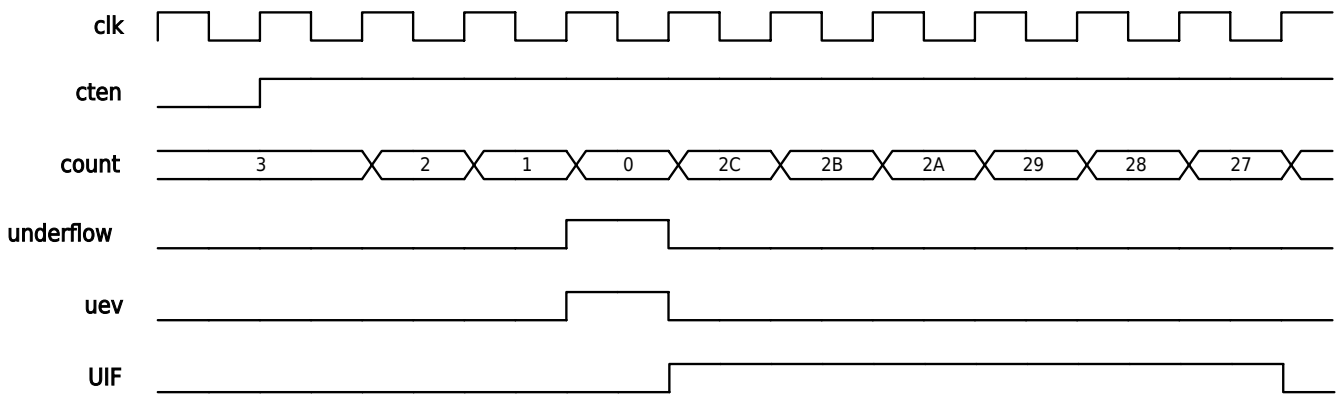



Figure 13-17 Mode 2 Compare Capture Mode Down-Counting Without Prescaler


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

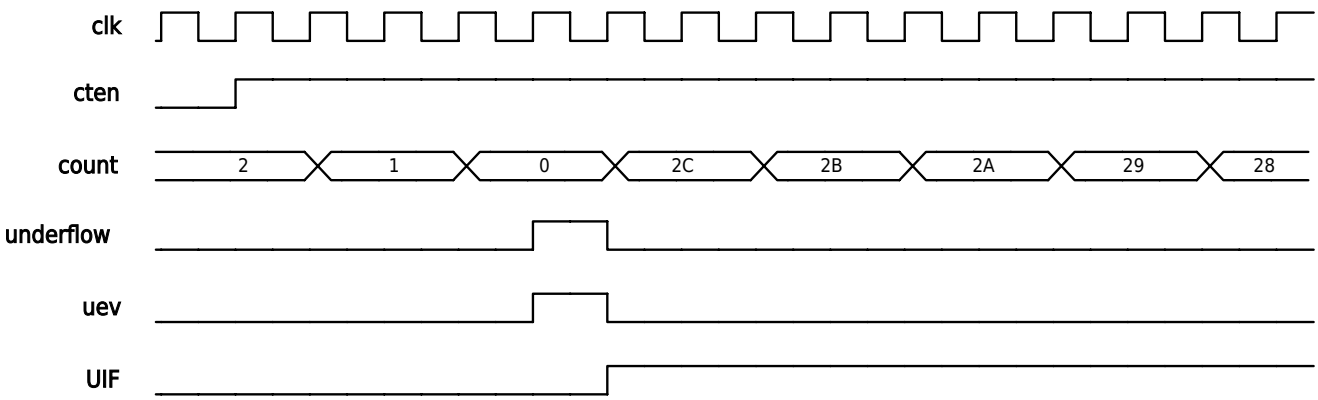



Figure 13-18 Mode 2 Compare Capture Mode Down-Counting With Prescaler

The counting waveforms for mode 2 are shown in the following diagrams:

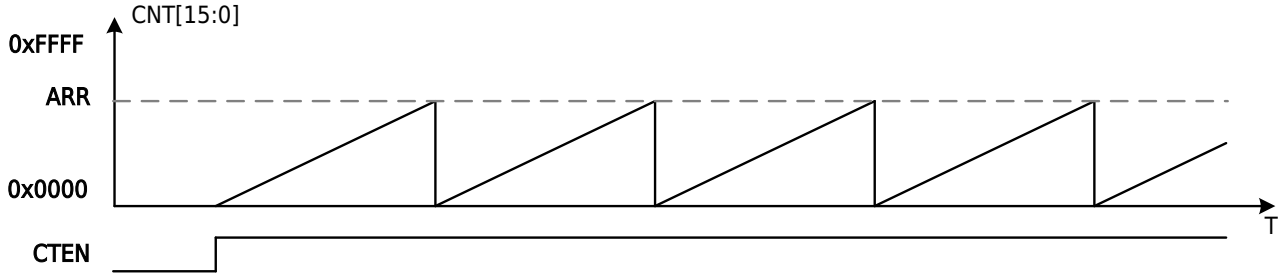



Figure 13-19 Mode2 Edge-Aligned Timer Waveform (M23CR.DIR=0)

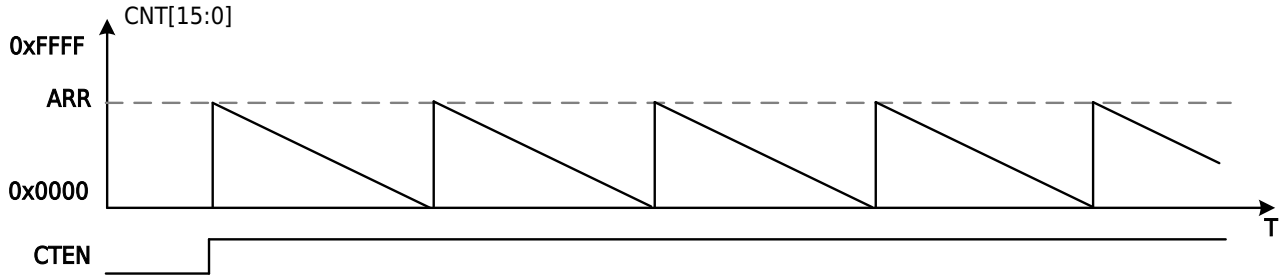



Figure 13-20 Mode2 Edge-Aligned Timer Waveform (M23CR.DIR=1)

### Mode3 Triangular Counting Waveform

Mode 3 is a triangular counting waveform, the counting direction control bit is read-only, and the counting direction cannot be changed.

In center-aligned (triangular wave) mode, the M23CR.DIR direction bit is read-only, and writing to it has no effect. Switching from other modes to center-aligned mode DIR is automatically cleared to 0. DIR is automatically cleared in reset mode by software event update and external trigger in slave mode.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

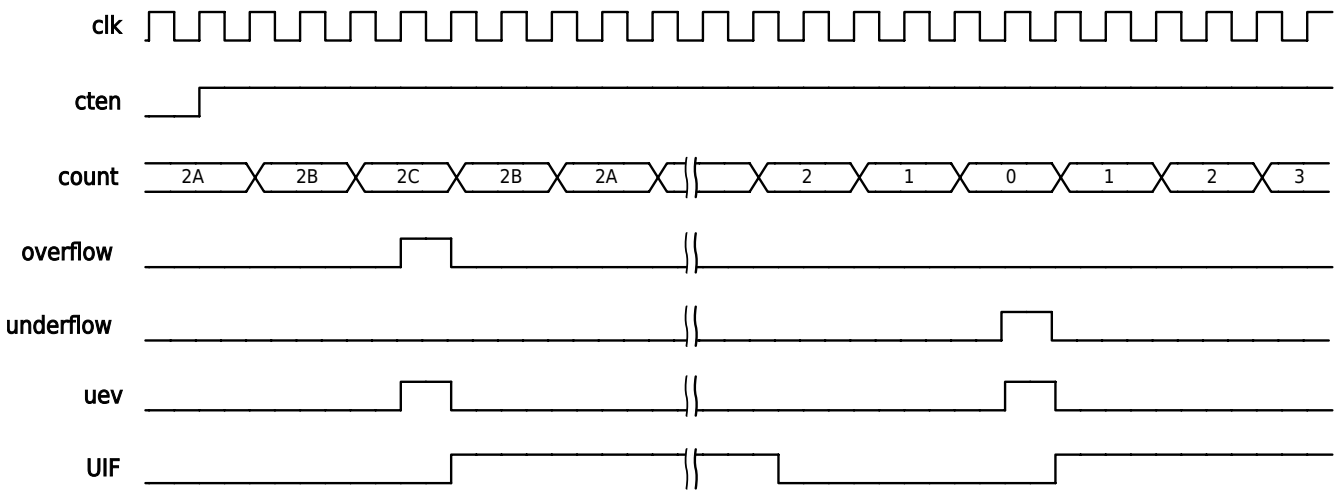



Figure 13-21 Mode3 Compare Capture Mode Up/Down-Counting with Prescaler

The counting waveforms for mode 3 are shown in the following diagrams:

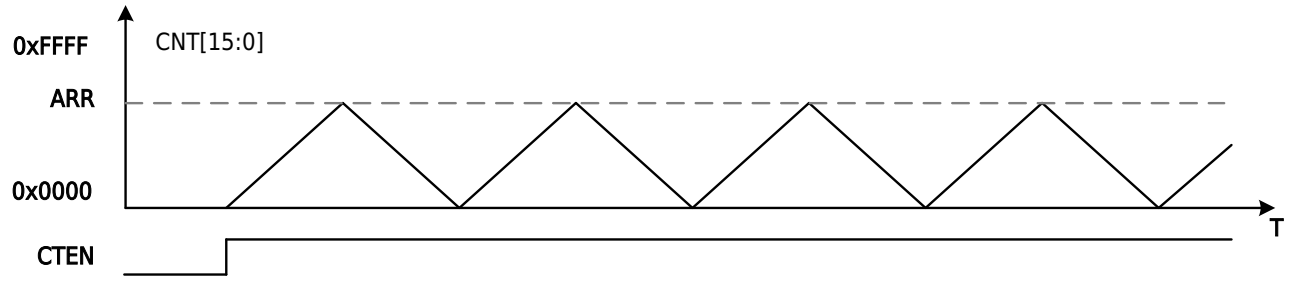



Figure 13-22 Mode3 Center-Aligned Counter Waveform

### 13.3.6.3 Repeat Counting

The repeat counter performs down-counting using counter overflows. When counting reaches 0, that is, when the counter overflows by the value set in the repeat register plus one. If buffer register is enabled, the period reload register updates to the period buffer register. In compare mode, the compare register value updates to the compare buffer register.

The repetition counter decrements when the following conditions are met (when RCR.UND=0 and RCR.OV=0):

- In up-counting mode: upon each counter overflow
- In down-counting mode: upon each counter underflow
- In triangular wave mode: upon each overflow and underflow


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

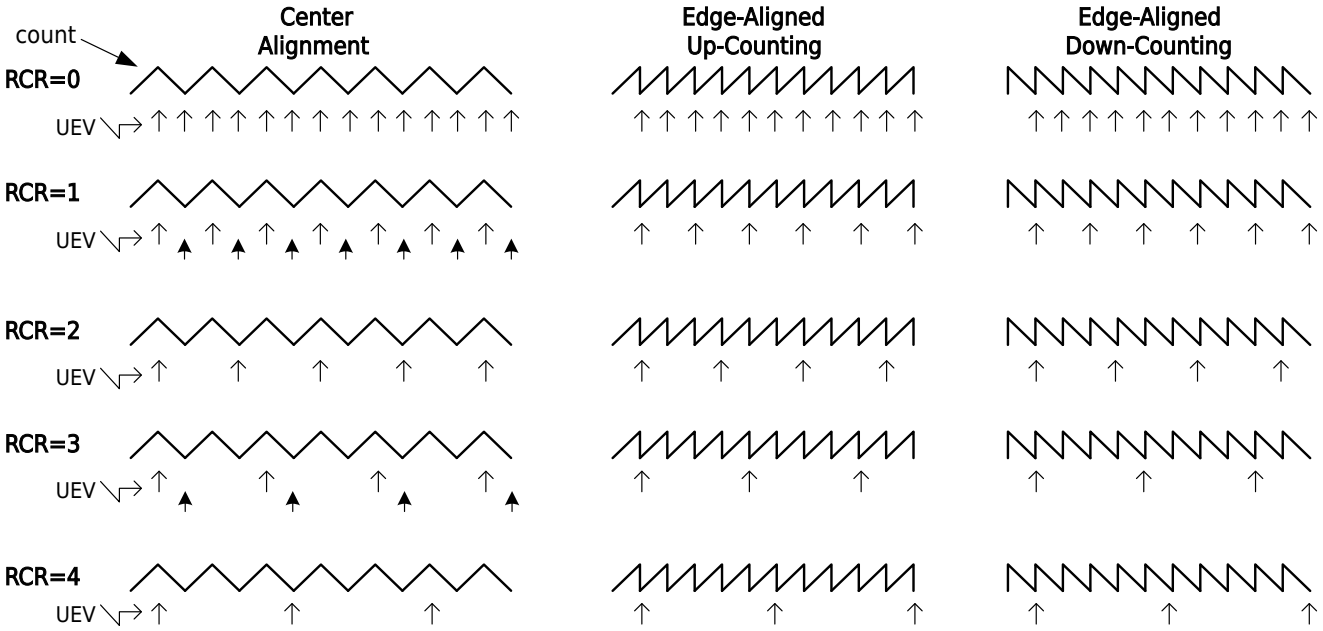



Figure 13-23 Repetition Counter Generates Update Timing (RCR.OV=0, RCR.UND=0)

RCR.OV is the overflow mask control bit for the repeat counter. When RCR.OV=1, overflow events are ignored during repeat counting.

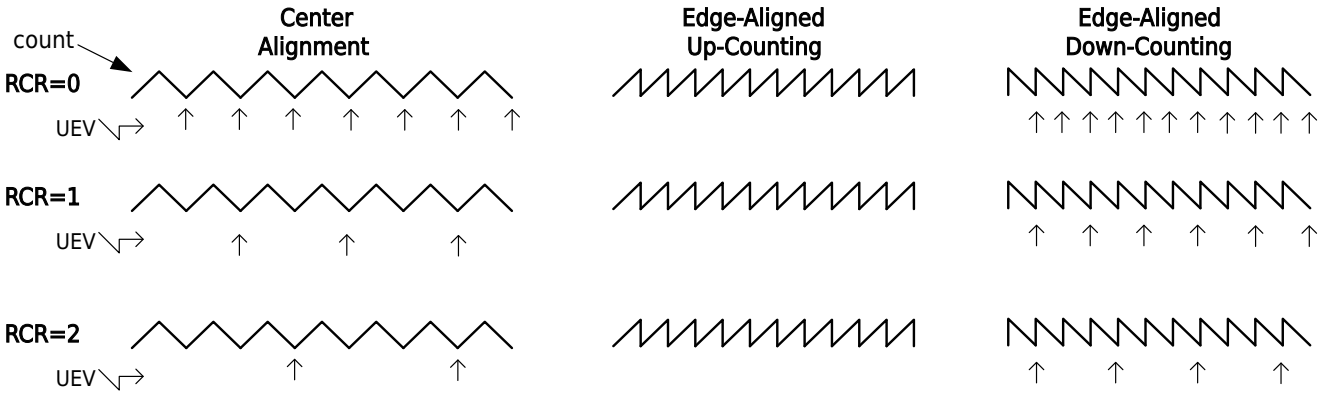



Figure 13-24 Update Timing Generated by Repetition Counter when Overflow Is Masked (RCR.OV=1, RCR.UND=0)

RCR.UND is the underflow mask control bit for the repetition counter. If RCR.UND is 1, underflow events will be ignored during repetition counting.

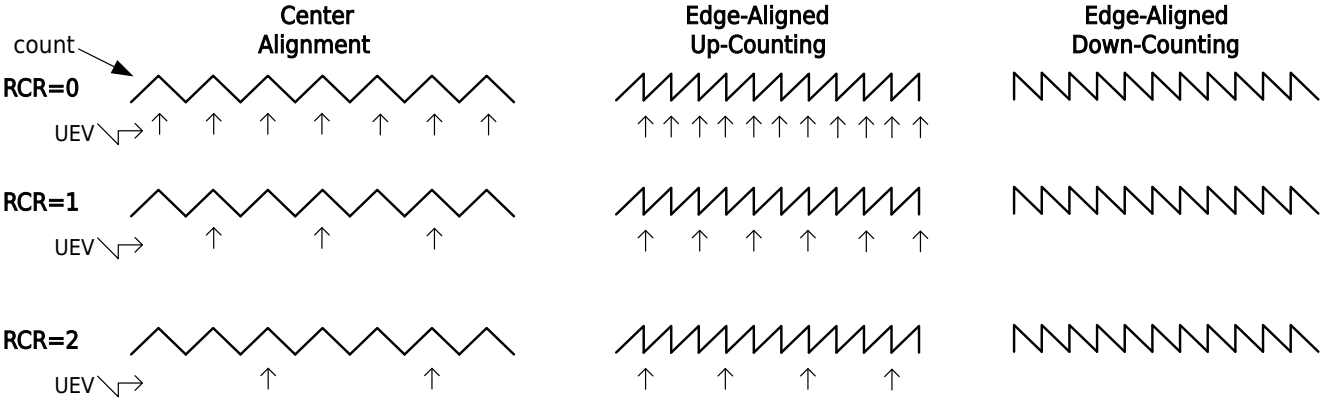



Figure 13-25 Update Timing Generated by Repetition Counter when Underflow Is Masked (RCR.OV=0, RCR.UND=1)

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

In addition to events update (UEV) generated by overflow/underflow through the repeat counter, software-triggered and slave-mode reset events update (UEV) can also be generated by writing to register M23CR.UG. This operation requires configuration of M23CR.URS.

### 13.3.6.4 Data Buffer

Both the auto-reload data (ARR) and compare registers (CCRxy) can be configured with buffering function. When buffering function is enabled, the written period value (ARR) and compare value (CCRxy) only take effect upon a UEV event update.

The update timing diagrams for the auto-reload value under different counting modes are shown below:

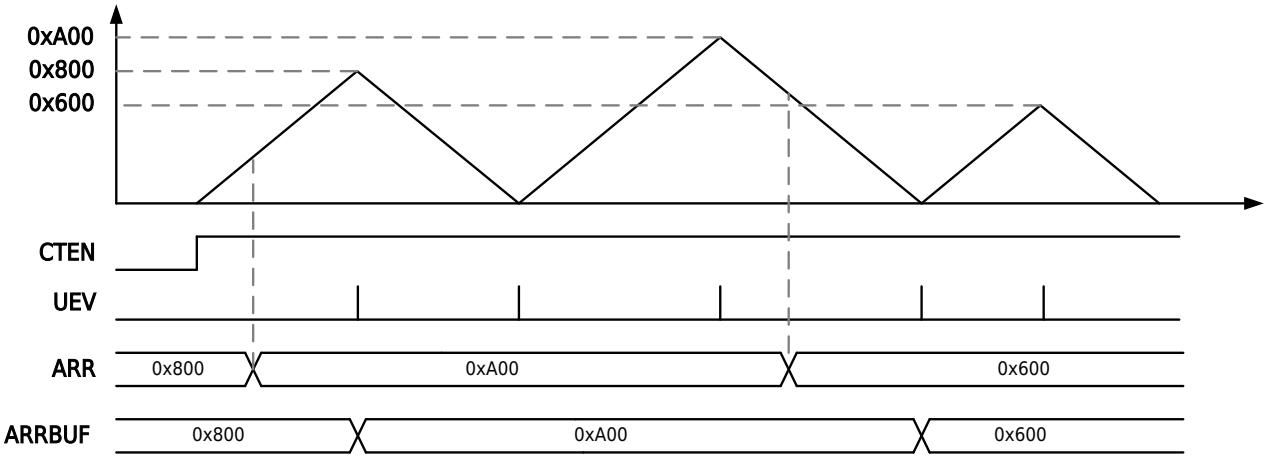



Figure 13-26 Mode 3 Triangular Wave Mode with Cache Enable

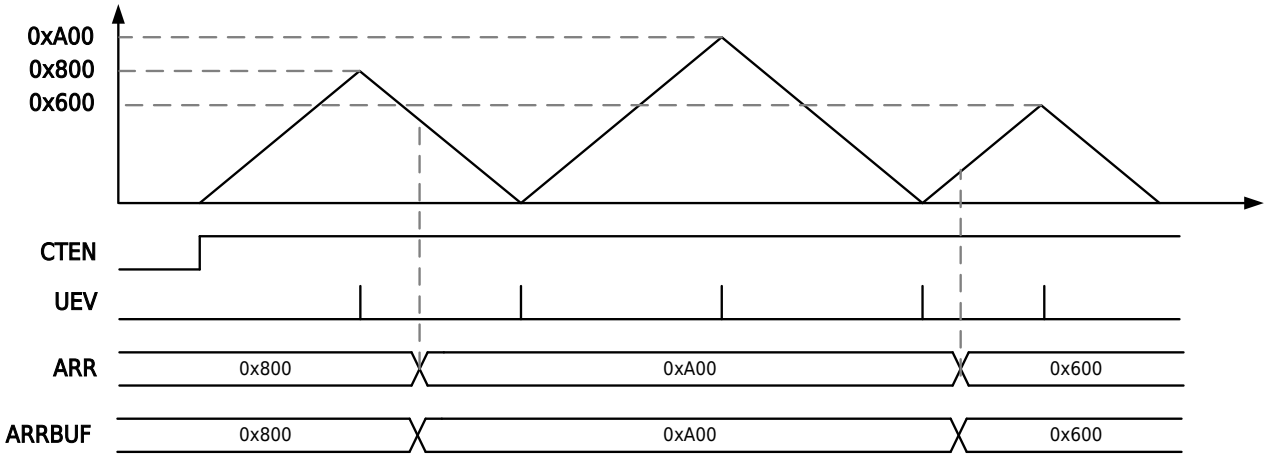



Figure 13-27 Mode 3 Triangular Wave Mode with Cache Invalid


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

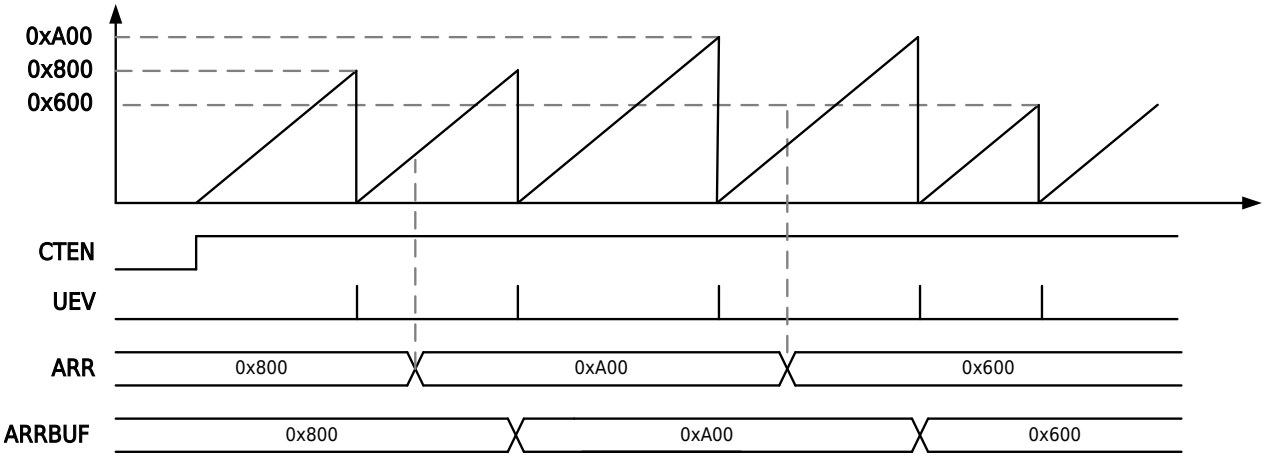



Figure 13-28 Mode 2 Sawtooth Wave Mode Up-Counting with Cache Enable

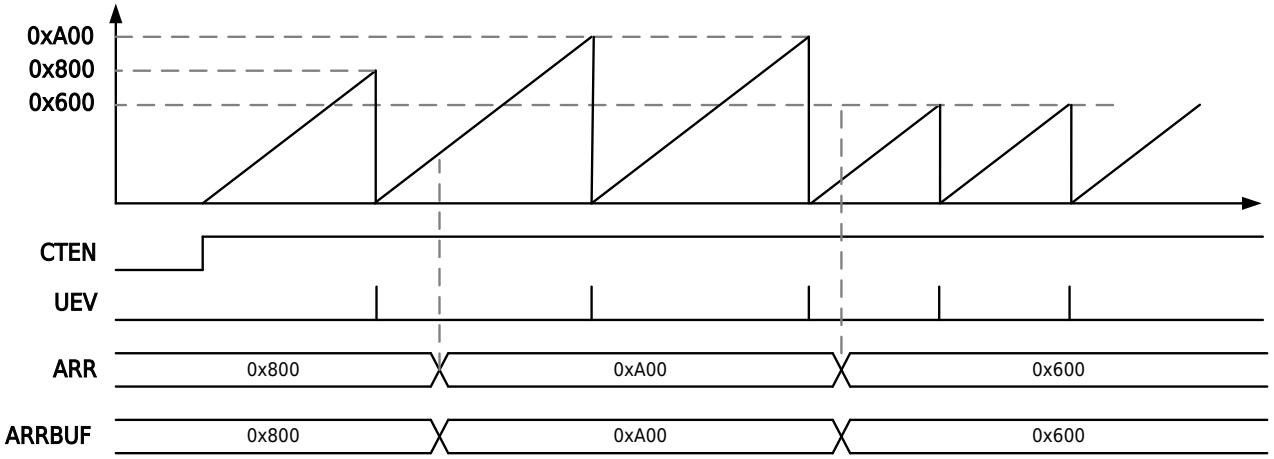



Figure 13-29 Mode 2 Sawtooth Wave Mode Up-Counting with Cache Invalid

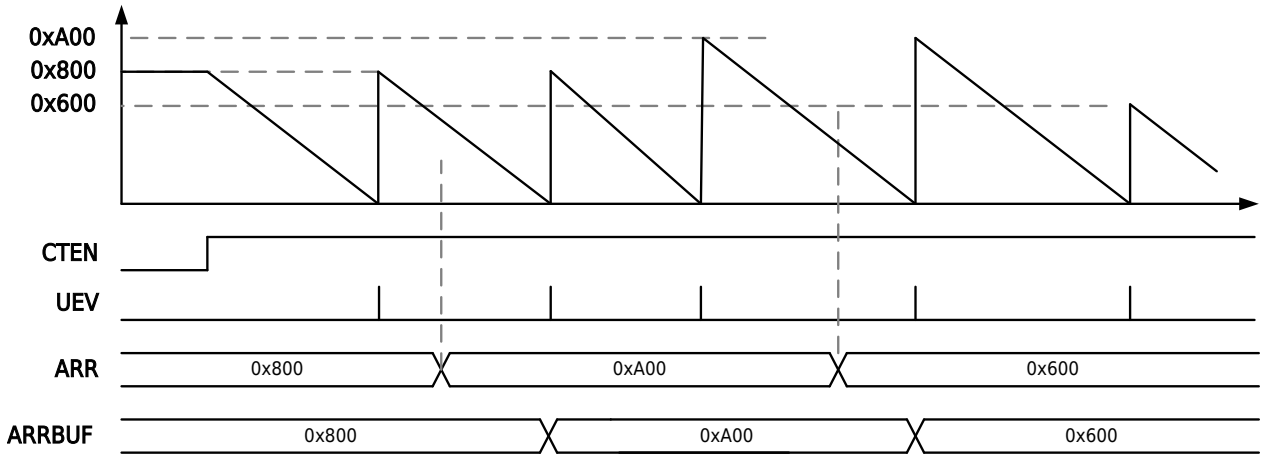



Figure 13-30 Mode 2 Sawtooth Wave Mode Down-Counting with Cache Enable


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

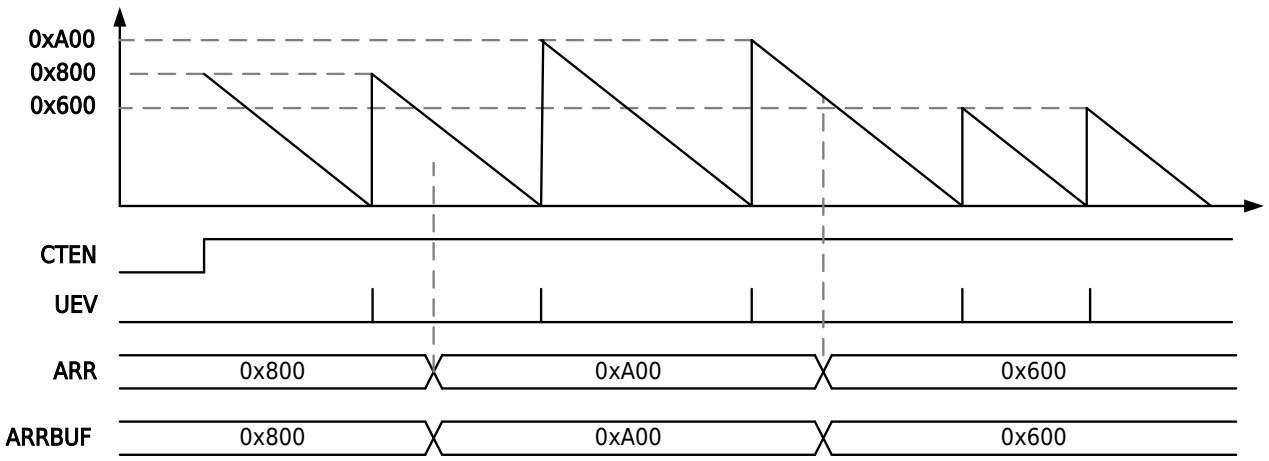



Figure 13-31 Mode 2 Sawtooth Wave Mode Down-Counting with Cache Invalid

**Note**

In triangular wave mode and sawtooth up-counting mode, when cache is disabled and the ARR value is modified, the current counter value must be less than the ARR period value to be changed; otherwise, the current cycle will count up to 0xFFFF.

The update behavior of the compare cache and period cache remains consistent, therefore their timing diagrams are not listed individually here.

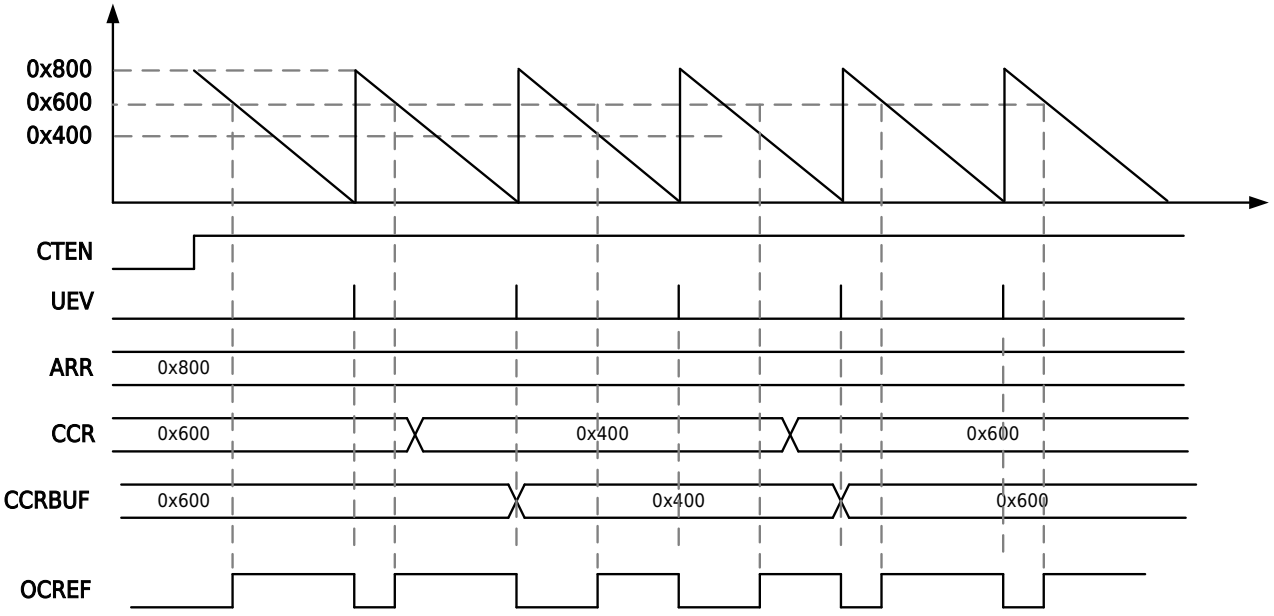



Figure 13-32 Mode2 Sawtooth Wave Down-Counting Mode with Compare Cache Enable

### 13.3.6.5 Compare Output OCREF

The compare output OCREFA can be configured for single point comparison, where the OCREFA output is controlled by compare register CCRA. Alternatively, OCREFA compare output can be configured for dual point comparison, where its output is jointly controlled by compare registers CCRA and CCRB.

OCREFB can only use single point comparison, and the comparison register CCRB is used to control the comparison output of OCREFB.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

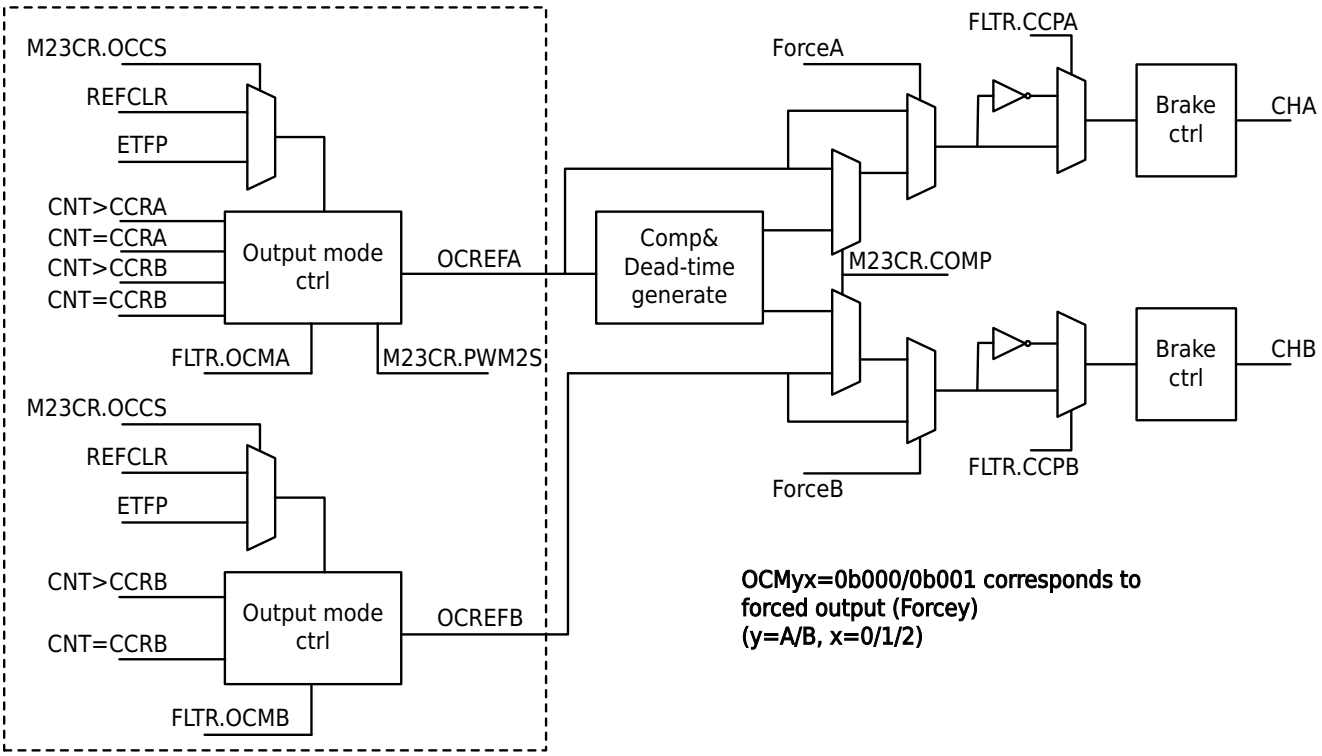



**Figure 13-33 OCREF Output Block Diagram**

OCREF output uses TIMx_FLTR.OCMyx selection:

- 0b000: Force to 0
- 0b001: Force to 1
- 0b010: Force to 0 on compare match
- 0b011: Force to 1 on compare match
- 0b100: Toggle on compare match
- 0b101: Output high level for one clock cycle on compare match
- 0b110: PWM mode 1
  - Single point comparison

    During up-counting, the output is high level when CNT<CCRxy; during down-counting, the output is low level when CNT>CCRxy.
  - Dual point comparison
    - For sawtooth wave up-counting, the output is low level when CCRxA<CNT≤CCRxB
    - For sawtooth wave down-counting, the output is high level when CCRxA<CNT≤CCRxB
    - For triangular wave up-counting, the output is high level when CNT<CCRxA; for down-counting, the output is low level when CNT>CCRxB
- 0b111: PWM mode 2
  - Single point comparison

    During up-counting, the output is low level when CNT<CCRxy; during down-counting, the output is high level when CNT>CCRxy
  - Dual point comparison
    - For sawtooth wave up-counting, the output is high level when CCRxA≤CNT<CCRxB
    - For sawtooth wave down-counting, the output is low level when CCRxA≤CNT<CCRxB
    - For triangular wave up-counting, the output is low level when CNT<CCRxA; for down-counting, the output is high level when CNT>CCRxB


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

**Note**

Forced output has higher priority. When forced output is active, complementary output control becomes invalid.

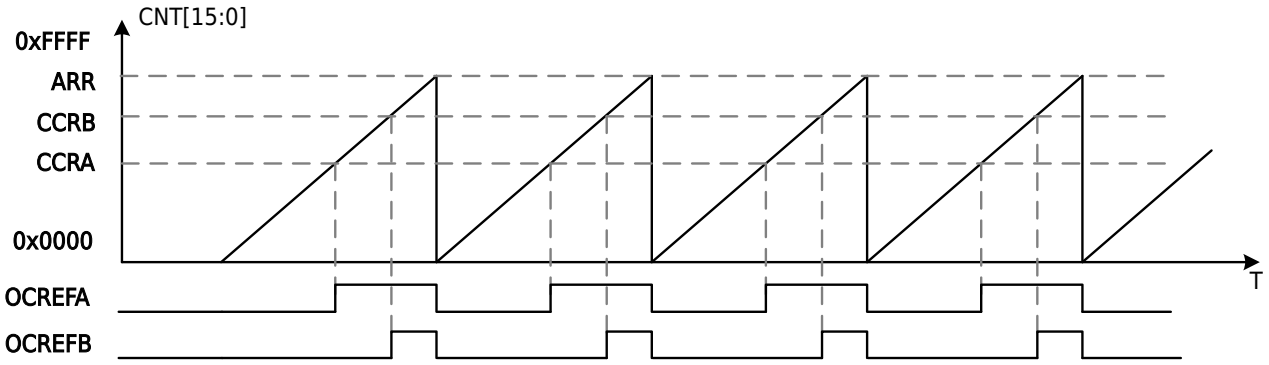



Figure 13-34 Sawtooth Counting Single Point Comparison OCREF Output Waveform (OCMyx=0b111)

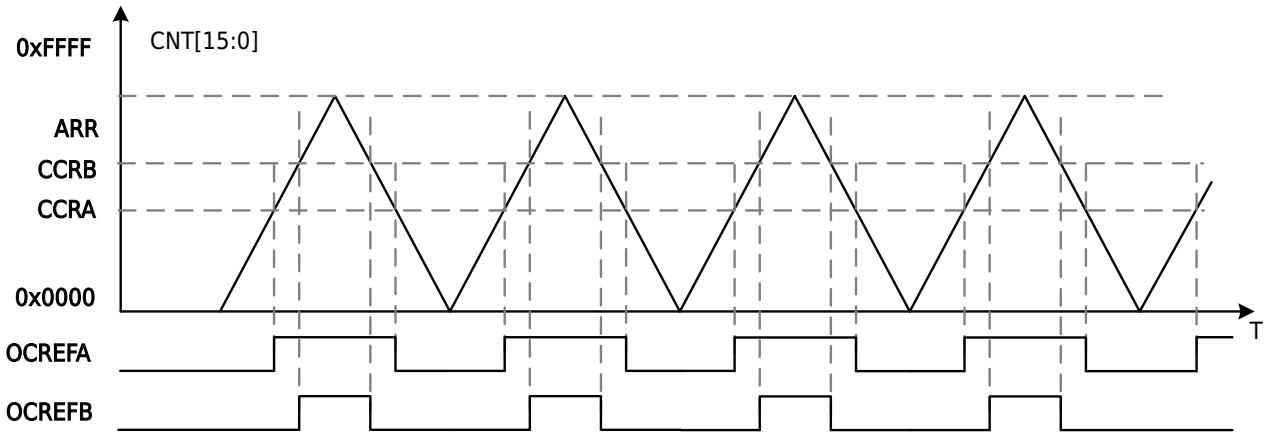



Figure 13-35 Triangular Counting Single Point Comparison OCREF Output Waveform (OCMyx=0b111)

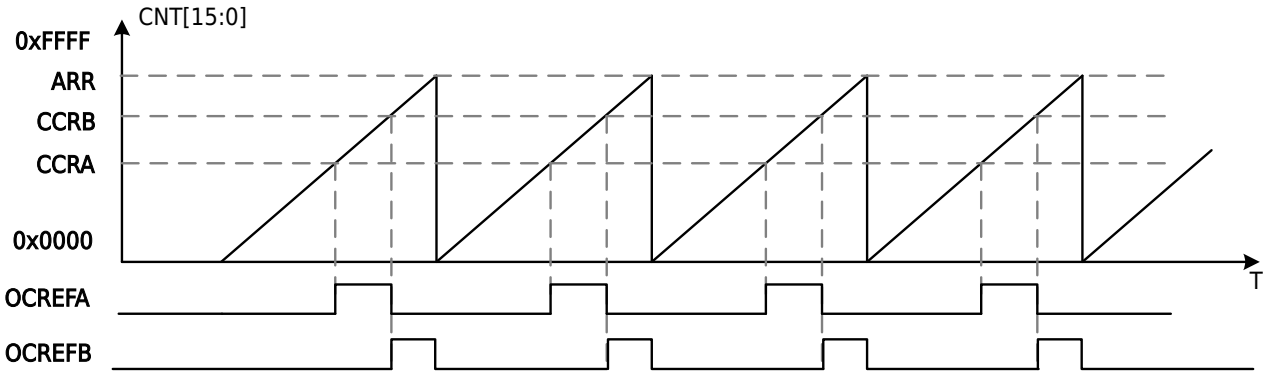



Figure 13-36 Sawtooth Counting Dual Point Comparison OCREF Output Waveform (OCMyx=0b111)


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

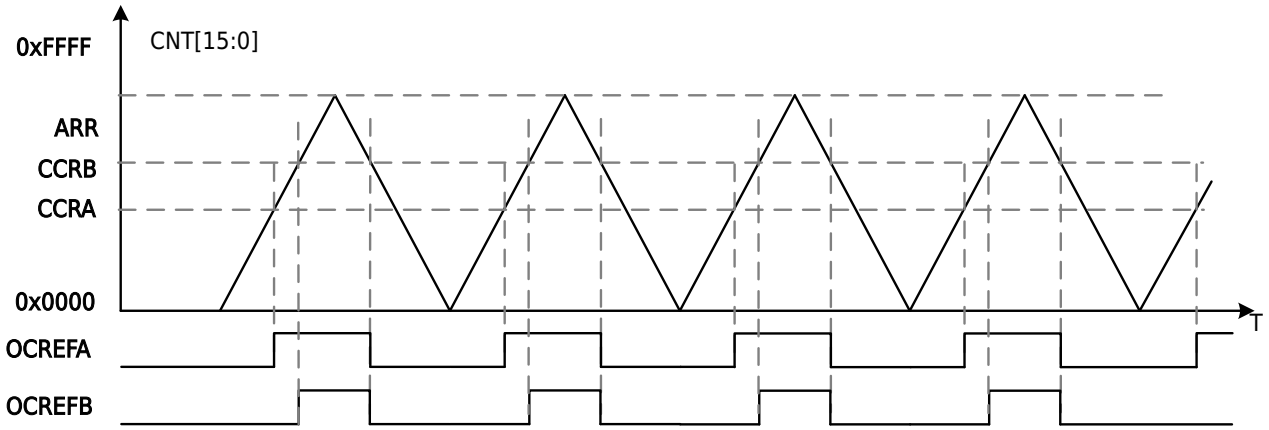



Figure 13-37 Triangular Counting Dual Point Comparison OCREF Output Waveform (OCMyx=0b111)


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 13.3.6.6 Independent PWM Output

The output of CHA is controlled by OCREFA, and the output of CHB is controlled by OCREFB. The inversion of CHA and CHB outputs can be controlled through FLTR.CCPA and FLTR.CCPB respectively.

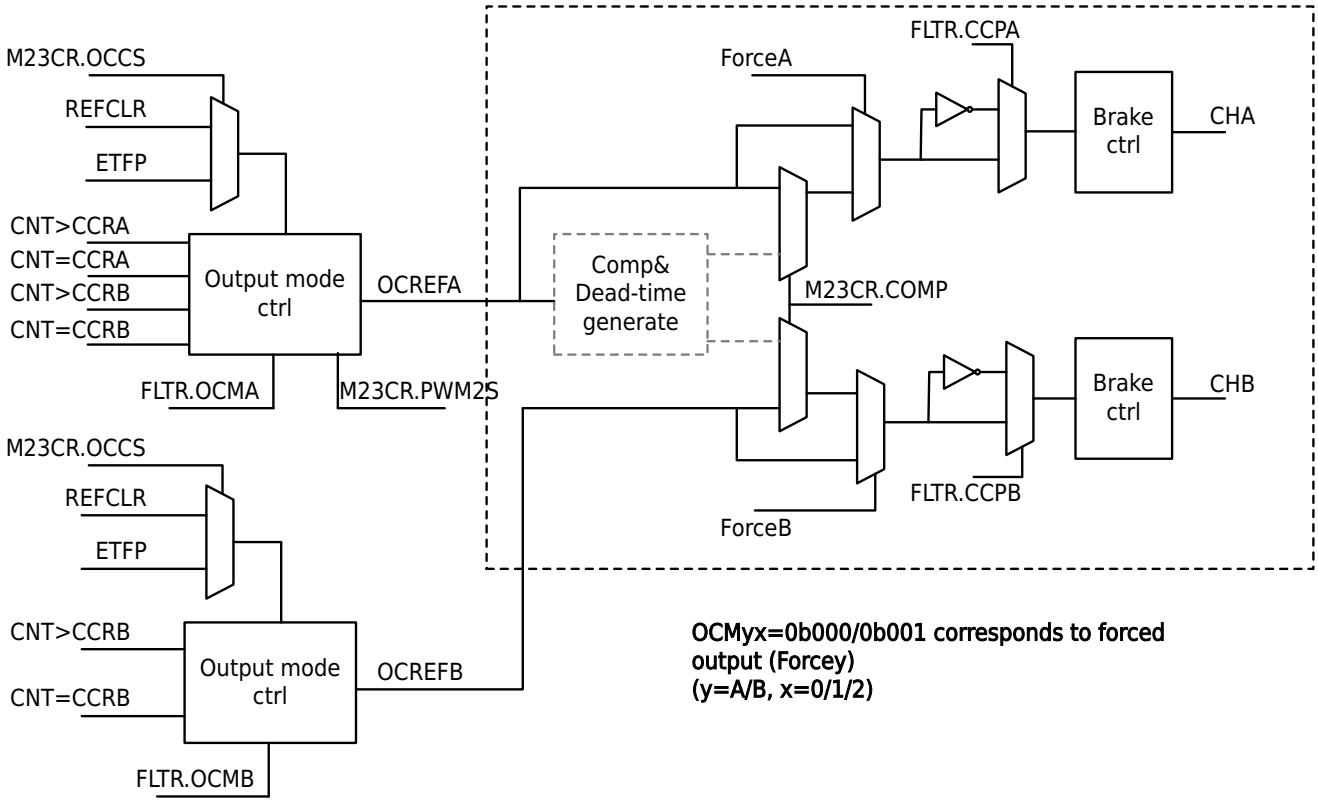



Figure 13-38 Independent PWM Output Block Diagram

PWM output and OCREF relationship, as shown in the following diagram:

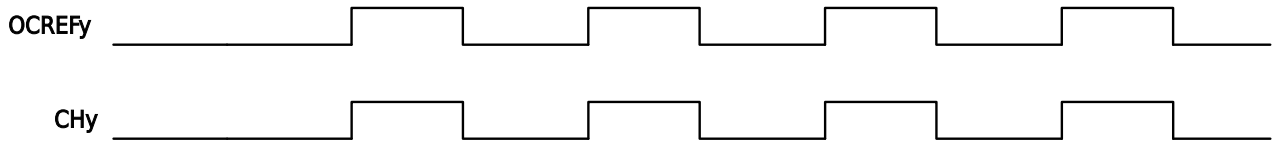



Figure 13-39 PWM Output Waveform When CCPy=0

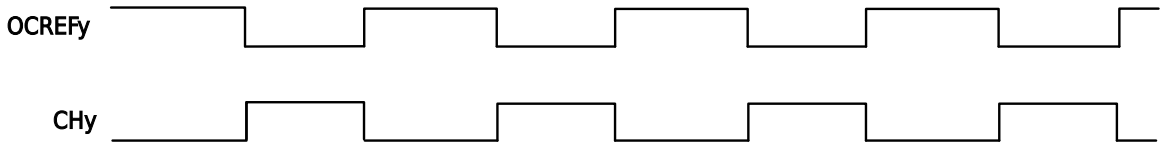



Figure 13-40 PWM Output Waveform When CCPy=1


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 13.3.6.7 Complementary PWM Output

CHA is controlled by OCREFA, and OCREFA controls the output of CHB at the same time. The compare register CCRxB can be used as a dedicated compare control ADC trigger.

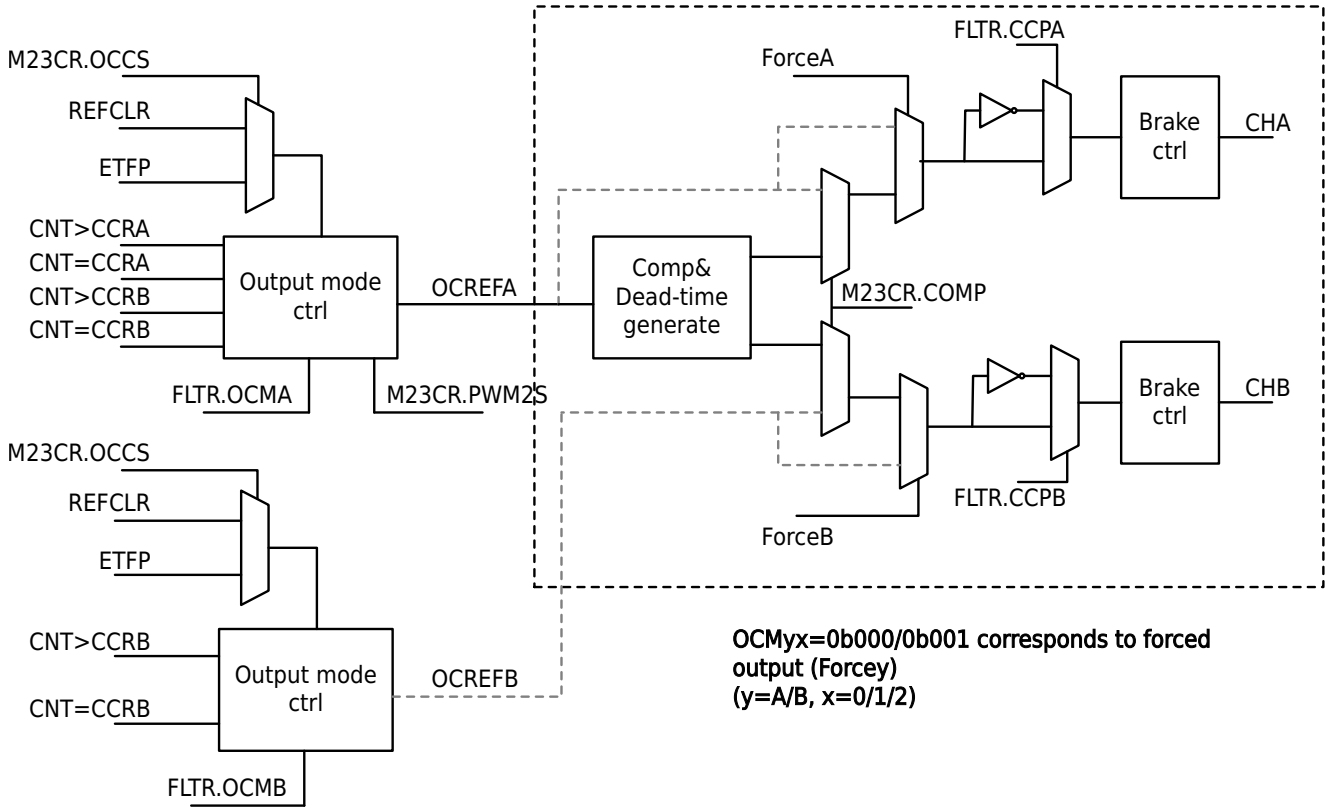



Figure 13-41 Complementary PWM Output Block Diagram

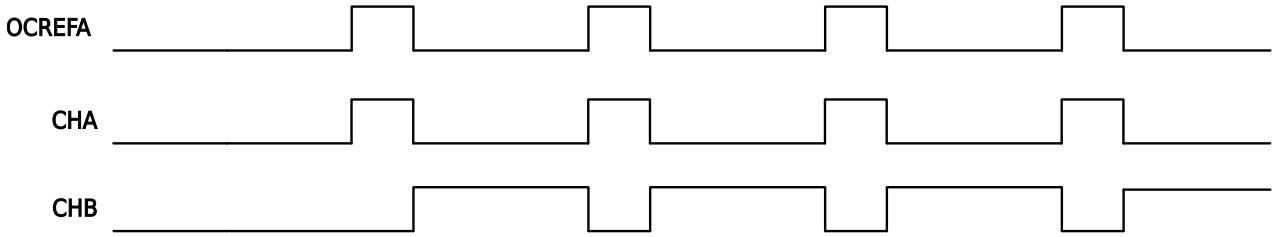



Figure 13-42 Complementary PWM Output Waveform


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 13.3.6.8 PWM Output with Dead Time

The dead time function can be set in complementary PWM output mode.

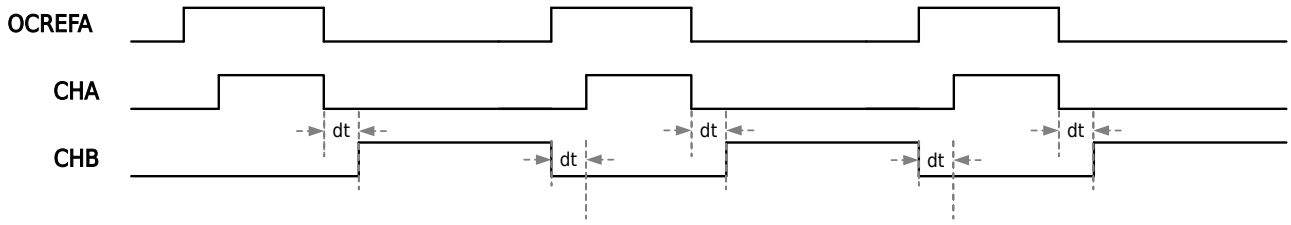



Figure 13-43 Complementary PWM Output Waveform (with Dead Time)

The dead time duration is controlled by 8-bit DTR. The relationship between dead time dt and DTR is as follows:

| DTR | T | dt | step |
|---|---|---|---|
| DTR[7]=0b0 | DTR[6:0] + 2 | 2-129 | 1 |
| DTR[7:6]=0b10 | {DTR[5:0] + 64}* 2 + 2 | 130-256 | 2 |
| DTR[7:5]=0b110 | {DTR[4:0] + 32}* 8 + 2 | 258-506 | 8 |
| DTR[7:5]=0b111 | {DTR[4:0] + 32}* 16 + 2 | 514-1010 | 16 |

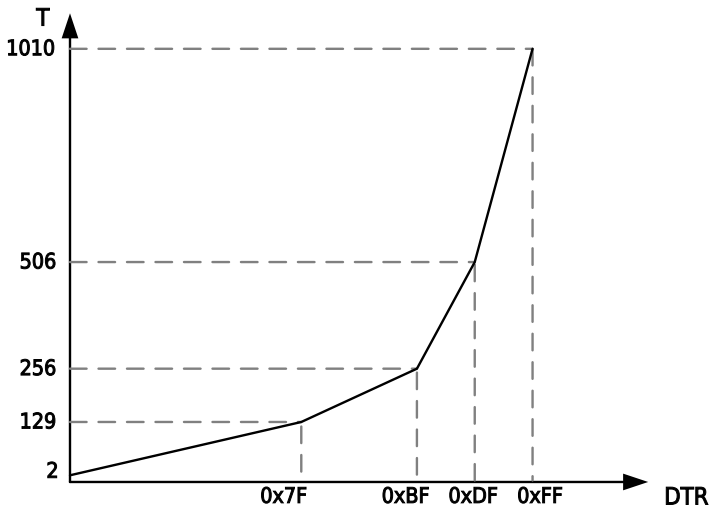



Figure 13-44 Dead Time

**Note**

When TCLK is 48MHz and M23CR.PRS is set to 0, the dead time is adjusted from 5.33μs to 21.04μs.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 13.3.6.9 Single Pulse Output

Single pulse mode (ONESHOT) is a special case of the aforementioned PWM modes. In this mode, the counter can be triggered to start by an excitation signal and generate a programmable width pulse, as shown in the following diagram.

The counter can be started by the slave mode controller. Waveforms can be generated in either output compare mode or PWM mode. Setting the TIMx_M23CR.ONESHOT register to 1 enables single pulse mode. Thus, when the next update event (UEV) occurs, the counter will automatically stop.

Single pulse mode in sawtooth wave down-counting mode, the counter initial value should not be set to 0, and the count value in up-counting mode should not be set to a value greater than or equal to ARR.

Output pulse can only be correctly generated when the compare value differs from the counter initial value. Before starting (when the timer is waiting for a trigger), the following configurations must be made:

- For increment counting: CNT<CCRxy≤ARR (special attention: CCRxy>0)
- For decrement counting: CNT>CCRxy

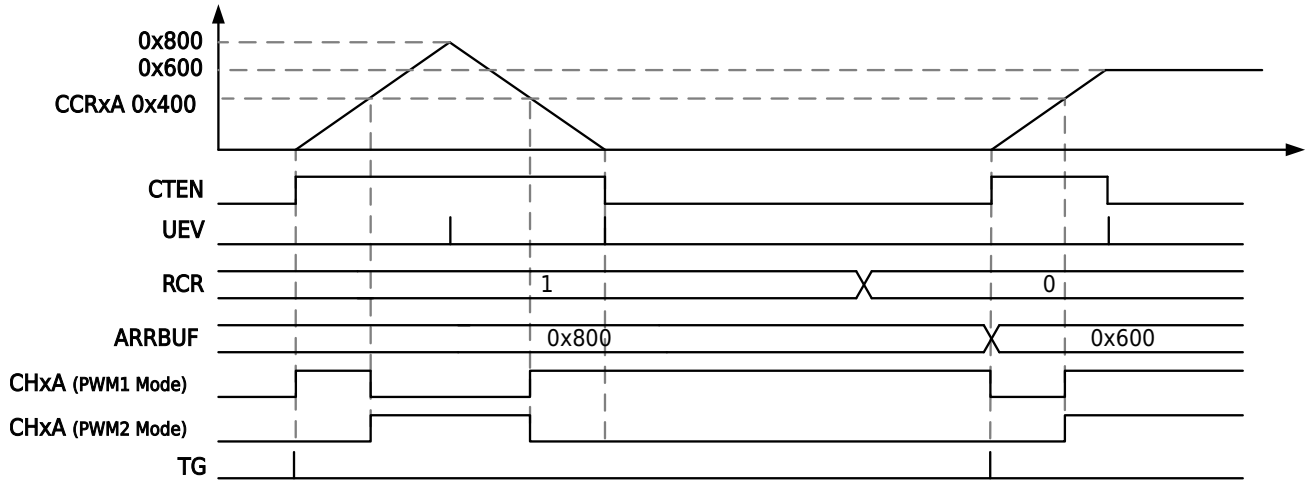



Figure 13-45 Triangular Wave Mode Single Pulse Counting

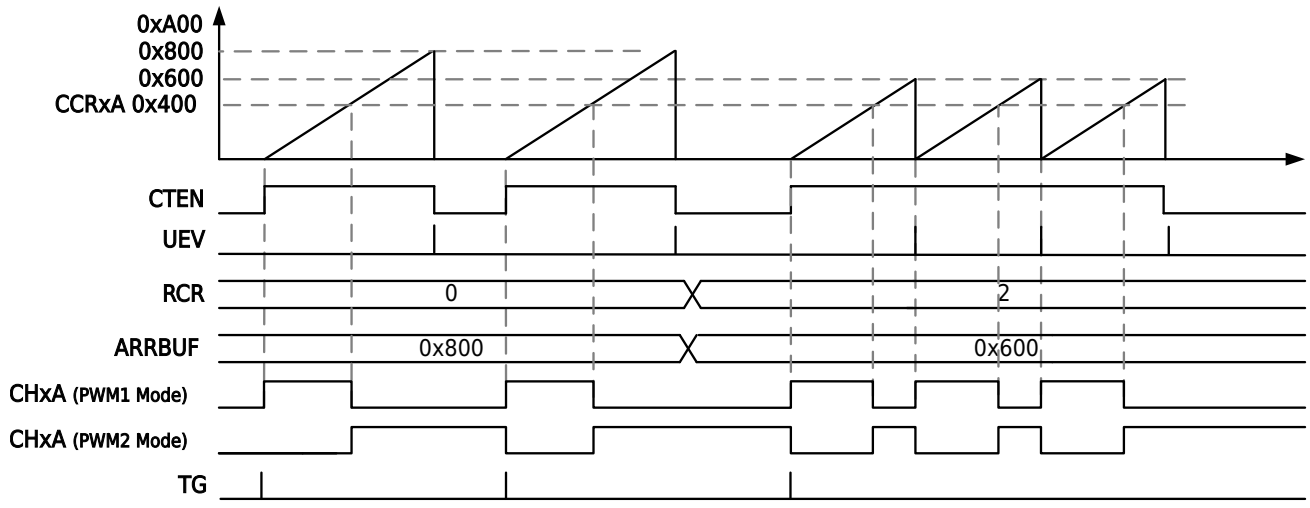



Figure 13-46 Sawtooth Up-Counting Single Pulse Mode


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

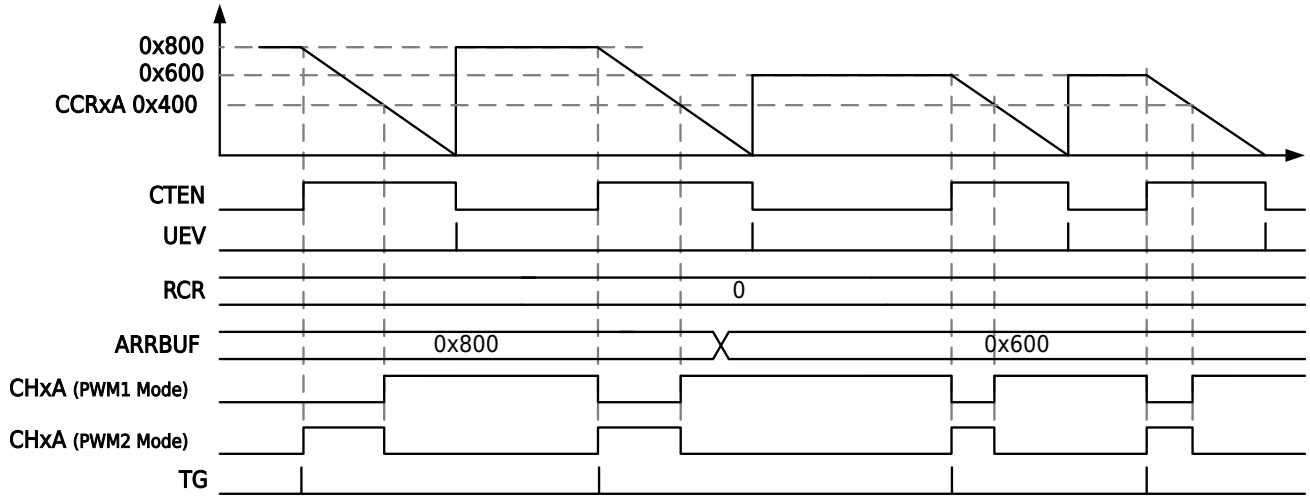



Figure 13-47 Sawtooth Down-Counting Single Pulse Mode

**Notice**

In up-counting, when configured as PWM2 mode, since the channel initial output is low level, the first trigger will generate a rising edge and maintain a high level. Upon the second trigger, a programmable width low level pulse can be output. In down-counting, when configured as PWM1 mode, since the channel initial output is low level, the first trigger will generate a rising edge and maintain a high level. Upon the second trigger, a programmable width low level pulse can be output.

### 13.3.6.10 Compare Interrupt

Sawtooth wave compare match will set the corresponding flag in the IFR register to 1. If the interrupt enables CRCHx.CIEy (x=0/1/2; y=A/B), an interrupt will be triggered.

For triangular wave compare match, it can independently select rising count compare match, falling count compare match, or both.

For compare A match, when the count value equals CCRxA, it is uniformly controlled by M23CR.CIS:

- M23CR.CIS=0b00: no output on compare match
- M23CR.CIS=0b01: compare match during up-counting
- M23CR.CIS=0b10: compare match during down-counting
- M23CR.CIS=0b11: compare match during both up-counting and down-counting

For compare B match, when the count value equals CCRxB, each channel can be individually controlled by CRCHx.CISB:

- When CRCHx.CISB=0b00, channel x has no output on compare match.
- When CRCHx.CISB=0b01, channel x compare match during up-counting.
- When CRCHx.CISB=0b10, channel x compare match during down-counting.
- When CRCHx.CISB=0b11, channel x compare match during both up-counting and down-counting.

The independent control of channel B compare match enables more flexible ADC triggering. For more details, refer to the Timer Trigger ADC section in the documentation.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

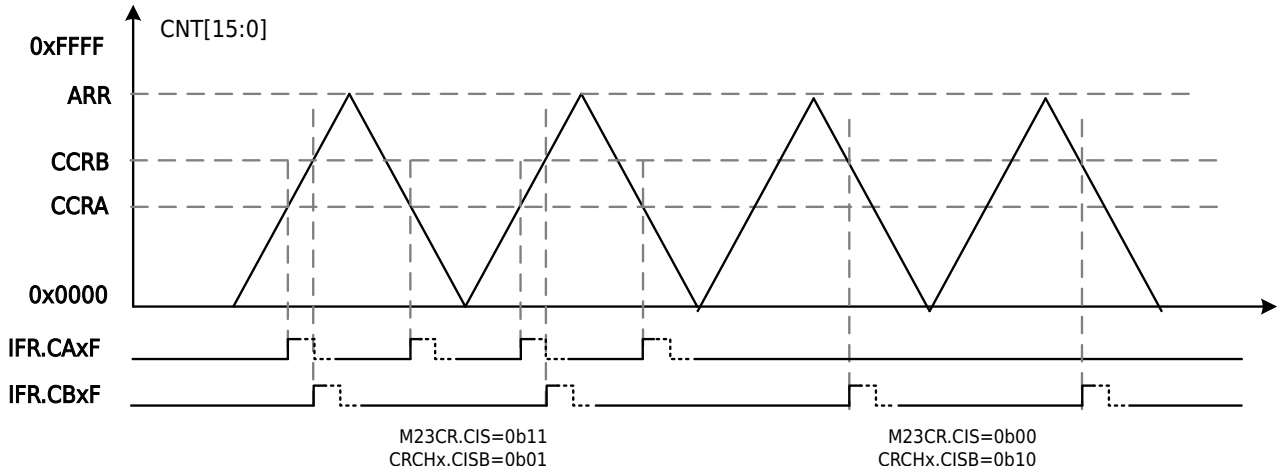



Figure 13-48 Compare Interrupt Diagram

### 13.3.6.11 Capture Input

The capture function (M23CR.MODE=2/3) can be configured in both triangular wave counting mode and sawtooth wave counting mode. The capture level edge can be set, and when a capture occurs, the captured value is stored in the compare capture register and generates a capture interrupt.

The compare capture function for each channel can be individually configured and selected through the CRCHx.CSA/CSB register. The capture edge for each channel can be independently set and via the CRCHx.CFy/CRy (x=0/1/2; y=A/B) register to select the triggering edge.

When the capture action occurs again before the capture flag is cleared after the capture occurs, the capture data overwrite flag will be generated. When the timer is not started, if there is a valid capture edge, the capture flag and capture action will also be generated.

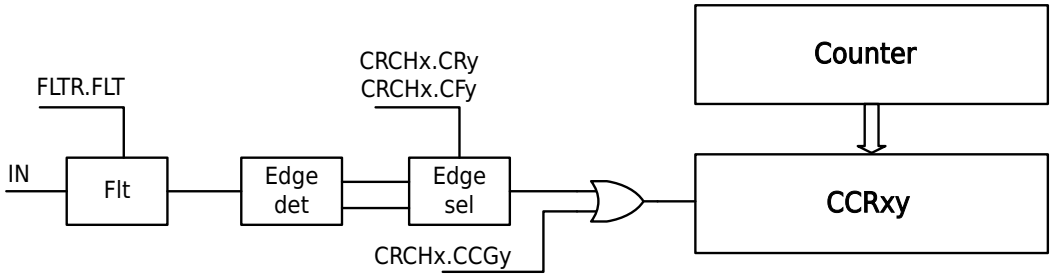



Figure 13-49 Capture Function Block Diagram

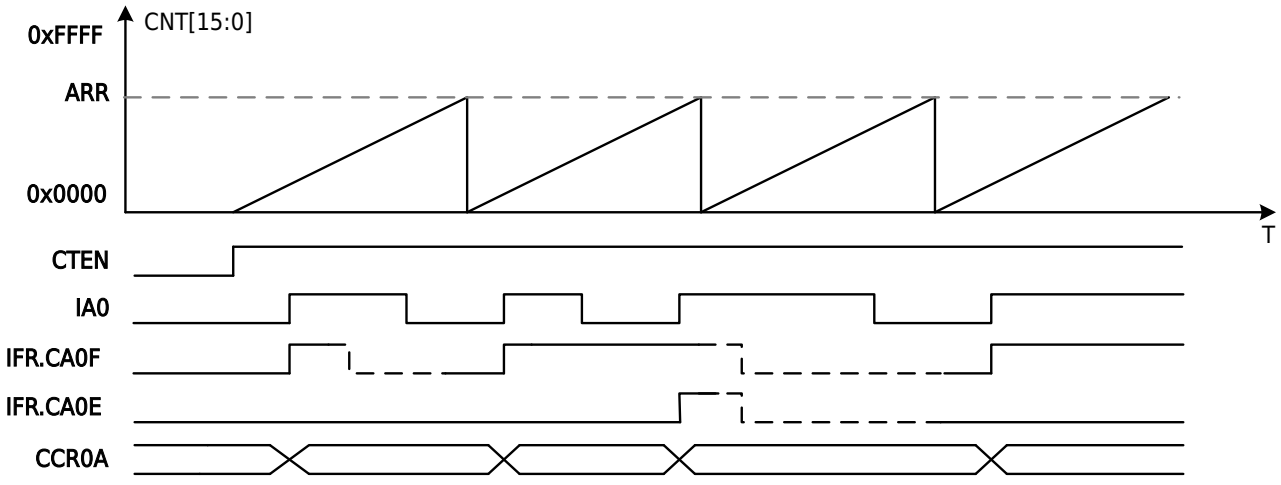



Figure 13-50 Capture Timing Diagram


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### CH0A Capture Input

The capture input signal for CH0A is as shown in the table below.

**Table 13-8 CH0A Capture Input Signal**

| MSCR.IA0S | TIM0/1/2 | TIM3 |
|---|---|---|
| 0b0 | CH0A | CH0A |
| 0b1 | CH0A ETR GATE XOR Input | CH0A CH1A CH2A XOR Input |

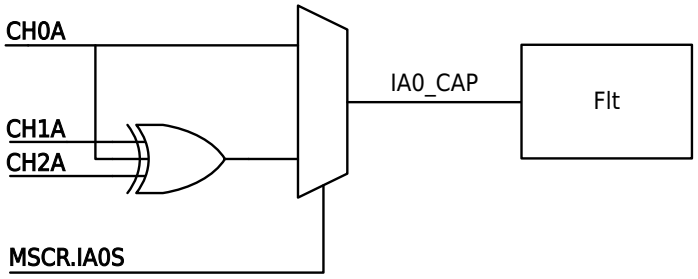



**Figure 13-51 TIM3 CH0A Capture Input Port Selection**

### CH0B Capture Input

The capture input signal for CH0B is as shown in the table below.

| MSCR.IB0S | TIM0/1/2/3 |
|---|---|
| 0b0 | CH0B |
| 0b1 | Internal trigger MSCR.TS select signal |

| MSCR.TS Value | Channel B Capture Input |
|---|---|
| 0b000 | ETR filtered phase selection output signal, filter phase selection configurable |
| 0b001 | Internal interconnection ITR0 |
| 0b010 | Internal interconnection ITR1 |
| 0b011 | Internal interconnection ITR2 |
| 0b100 | Internal interconnection ITR3 |
| 0b101 | CH0A edge |
| 0b110 | CH0A filtered output signal, filter function configurable |
| 0b111 | Reserved |

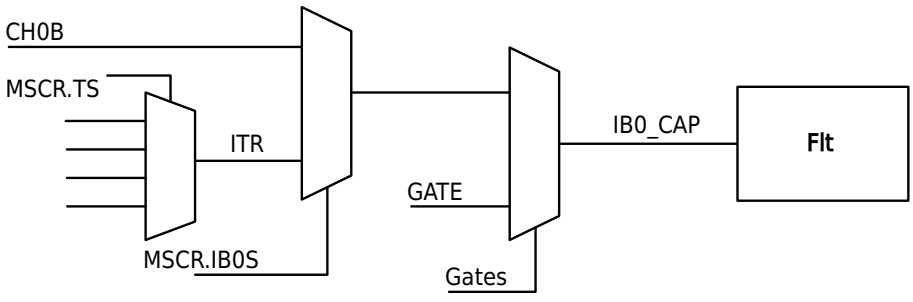



**Figure 13-52 CH0B Capture Input Port Selection**


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

The Gates signal is generated for selecting the PWM complementary output function. When PWM is the single point comparison complementary output, the compare capture register CCR0B is unused, and automatically switches to using GATE as the capture compare output.

**Notice**

When GATE is configured for compare output, its output mode only supports being set to force the output to 0 or 1 and being set to PWM mode valid (that is, OCMB0 is configured as 0/1/6/7).

## 13.3.7 Mode2/3 Slave Mode

The timer can be synchronized to an external trigger in several modes: reset mode, gated mode and triggered mode.

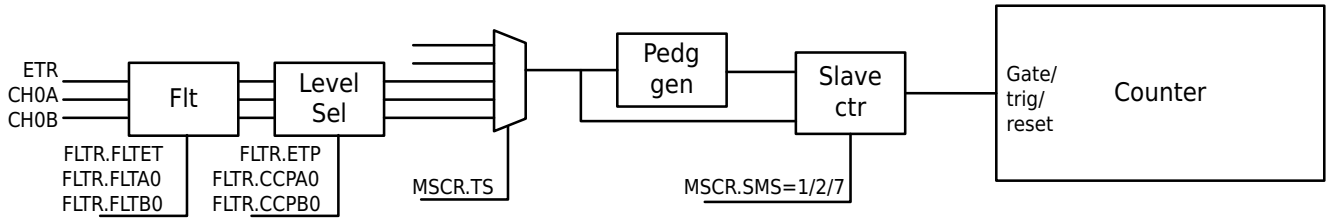



**Figure 13-53 Mode2/3 Slave Mode Diagram**

The slave mode function is selected via MSCR.SMS:

| | Value | Description |
|---|---|---|
| MSCR.SMS | 0b001 | Slave mode function selection: reset mode |
| | 0b010 | Slave mode function selection: trigger mode |
| | 0b111 | Slave mode function selection: gating mode |
| MSCR.TS | 0b000 | The filtered phase selected signal ETFP from port ETR features configurable filtering function |
| | 0b001 | Internal interconnect signal ITR0 |
| | 0b010 | Internal interconnect signal ITR1 |
| | 0b011 | Internal interconnect signal ITR2 |
| | 0b100 | Internal interconnect signal ITR3 |
| | 0b101 | Port CH0A edge signal |
| | 0b110 | The filtered phase selected signal IAFP from port CH0A features configurable filtering function |
| | 0b111 | The filtered phase selected signal IBFP from port CH0B features configurable filtering function |

### Gated Counting

Enable the counter according to the selected input level. In the following example, the counter only counts up when CH0A is low:

- Configure channel CH0A to detect a low level on CH0A. Configure the input filter bandwidth (filtering is not required in this case, therefore maintain FLTR.FLTA0=0b000). Set FLTR.CCPA0=1 to determine the polarity (only detect low level).


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

- Set MSCR.SMS=0b111 to configure the timer in gating mode; set MSCR.TS=0b110 to select CH0A as the input source.
- Set the M23CR.CTEN=1 to start the counter. In gating mode, if CTEN=0, the counter cannot be started regardless of the trigger input level. As long as CH0A is low, the counter starts counting according to the internal clock, and stops counting once CH0A becomes high.

**Note**

The gating input must not be a pulse signal. When gated counting, TS cannot be configured as 0b101; when TS is configured as 0b001/0b010/0b011/0b100, attention must be paid to whether ITR is a pulse signal.

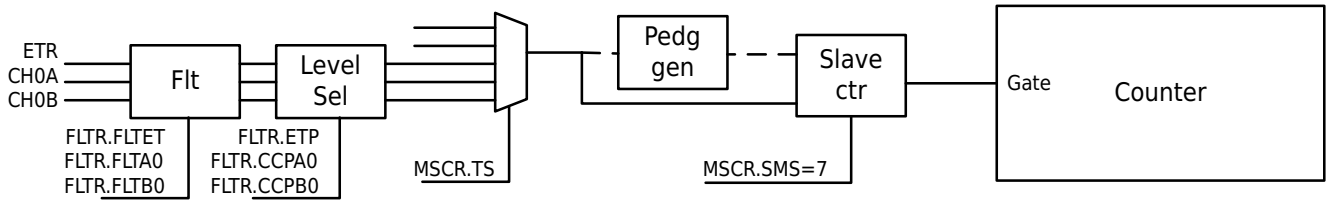



**Figure 13-54 Mode2/3 Slave Mode Gating Function**

### Trigger Function

Using external triggers (CH0A, CH0B, ETR) enables synchronous timer startup. Alternatively, timer synchronization can be configured using the timer internal interconnect signals combined with MSCR.MSM. For related operations, refer to the examples of simultaneous PWM output setup for timers TIM0/1/2 (using TIM0's CTEN to trigger TIM1/2) and simultaneous PWM output setup for timers TIM0/1/2 (using TIM1's UG to trigger TIM0/2). The trigger signal is the rising edge of the input signal. Alternatively, software trigger function can be initiated by writing to M23CR.TG.

A selected event on the input enables the counter. In the example below, the counter starts up counting on the rising edge of the CH0B input:

- Configure channel CH0B to detect the rising edge of CH0B. Configure the input filter bandwidth (in this example, no filter is required, keep FLTR.FLTB0=0b000). Set FLTR.CCPB0=0 to determine polarity (no inversion).
- Set MSCR.SMS=0b010 to configure the timer in trigger mode; set MSCR.TS=0b111 to select CH0B as the input source. When a rising edge appears on CH0B, the counter starts counting driven by the internal clock and sets the TIF flag at the same time. The delay between the rising edge of CH0B and the start of the counter depends on the synchronization circuit at the CH0B input.

**Note**

When using falling edge trigger, first select the trigger polarity and then choose the mode; otherwise, false triggering may occur.

### Reset Counting

When a trigger input event occurs, both the counter and its prescaler can be reinitialized; Simultaneously, if the M23CR.URS is low, an update event UEV is generated; Subsequently, all preload registers (ARR, CCRxy) are updated.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

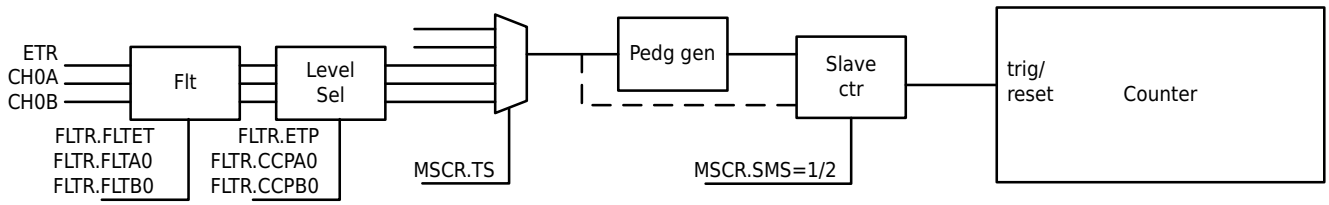



**Figure 13-55 Mode2/3 Slave Mode Trigger and Reset Function**

## 13.3.8 Quadrature Encoder Counting Function

MSCR.SMS=4/5/6, corresponding to the mode 1/2/3 of quadrature encoder mode. At this point, the counter performs encoder counting based on the phase relationship between IAFP and IBFP. IAFP and IBFP are the filtered phase selection signals for port inputs CH0A and CH0B.

Mode 1 uses the edge count of CH0A. Mode 2 uses the edge count of CH0B. Mode 3 uses both the edges count of CH0A and CH0B.

In order to ensure the correctness of the counting phase, it is necessary to ensure that the phase of the A/B input is greater than the phase difference of one pulse width, and the A/B input pulse width needs to be greater than two pulse widths.

As shown in the table below, the edge condition of IAFP corresponds to the level signal of IBFP, while the edge condition of IBFP corresponds to the level signal of IAFP. For example, in MOD1, when IBFP is at a high level and a rising edge is detected on IAFP, the counter counts down; when IBFP is at a high level and a falling edge is detected on IAFP, the counter counts up.

| Mode | Quadrature Signal Level Condition | | Quadrature Signal IAFP Edge Condition | | Quadrature Cignal IBFP Edge Condition | |
|---|---|---|---|---|---|---|
| | IBFP | IAFP | Rising Edge | Falling Edge | Rising Edge | Falling Edge |
| MOD1 | High level | - | Down-counting | Up-counting | - | - |
| | Low level | - | Up-counting | Down-counting | - | - |
| MOD2 | - | High level | - | - | Up-counting | Down-counting |
| | - | Low level | - | - | Down-counting | Up-counting |
| MOD3 | High level | - | Down-counting | Up-counting | - | - |
| | Low level | - | Up-counting | Down-counting | - | - |
| | - | High level | - | - | Up-counting | Down-counting |
| | - | Low level | - | - | Down-counting | Up-counting |

CHAF/CHBF is the signal filtered by port CH0A/CH0B, and IAFP/IBFP is the signal after port filter phase selection.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

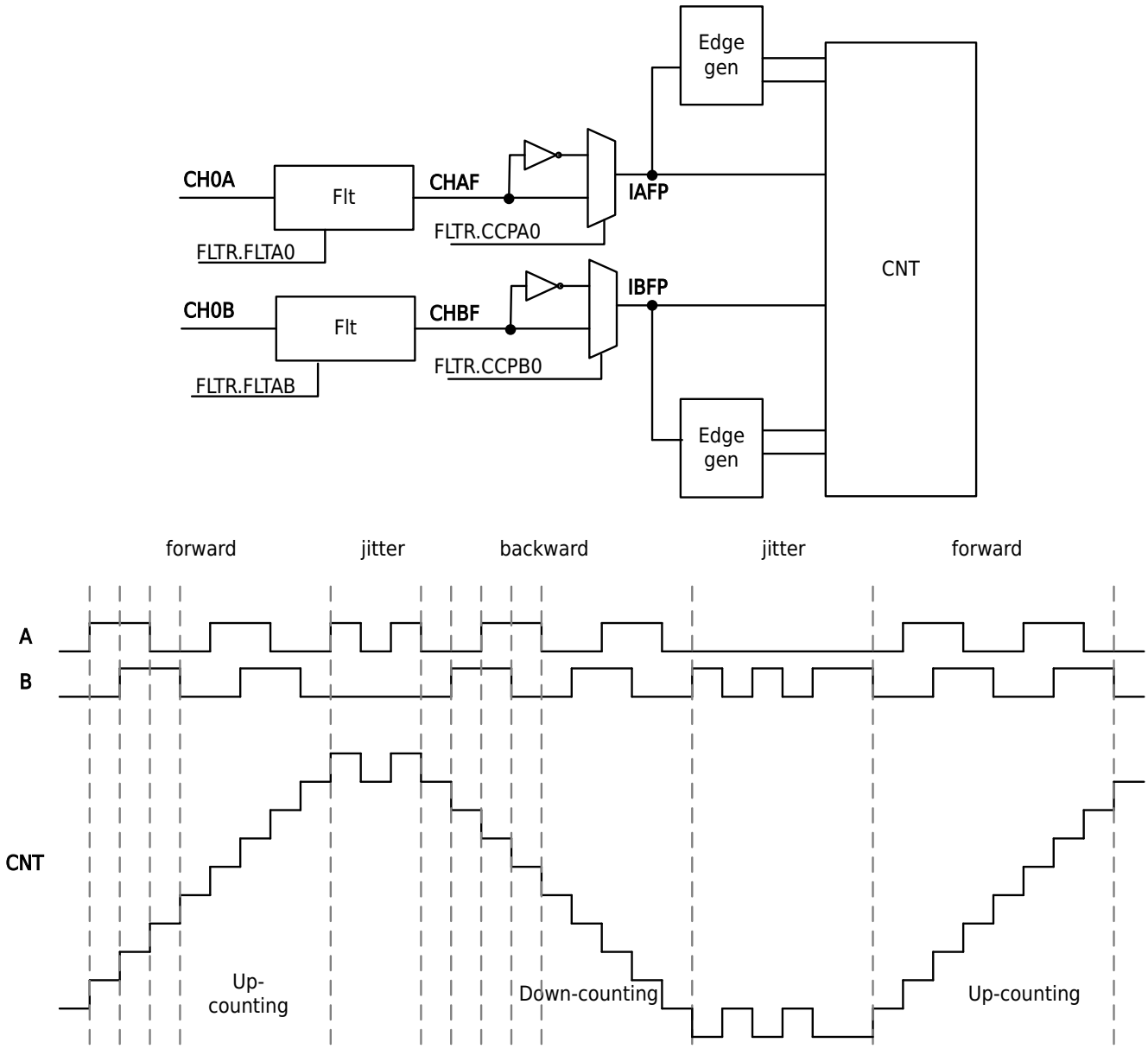



Figure 13-57 Quadrature Encoder Counting Mode3 Counting Diagram

## 13.3.9 Timer Triggers ADC

### UEV Triggers ADC

When a update event (UEV) is generated and both ADTR.UEVE and ADTR.ADTE are enabled, a trigger request can be output to the ADC.

### Compare Match Triggers ADC

CCR0A, CCR1A, and CCR2A compare matches can be configured to trigger the ADC.

- Mode3: The comparison match conditions for CHxA channels are uniformly controlled through the M23CR.CIS control register. You can select to trigger up-counting comparison match, down-counting comparison match, or both up-counting and down-counting comparison matches.
- Mode2: No need to configure M23CR.CIS.

CCR0B, CCR1B, and CCR2B compare matches can be configured to trigger the ADC.

- Mode 3: The comparison match conditions for CHxB channels are individually controlled via the CRCHx.CISB register. Options include triggering up-counting comparison match, down-counting comparison match, or both up-counting and down-counting comparison matches.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

- Mode2: No configuration of CRCHx.CISB is required.

The comparison matches of CCR3A and CCR3B can also be configured to trigger the ADC.

- Mode3: The comparison match conditions for channel 3A and channel 3B are individually controlled by CRCH3A.CIS and CRCH3B.CIS registers respectively. Options include triggering on up-counting comparison match, down-counting comparison match, or both up-counting and down-counting comparison matches.
- Mode2: No configuration of CRCH3A.CIS and CRCH3B.CIS is required.

The TRGO trigger signal from timer can trigger the ADC. For TRGO output configuration, refer to the function description of the TIM_MSCR.MMS register bit.

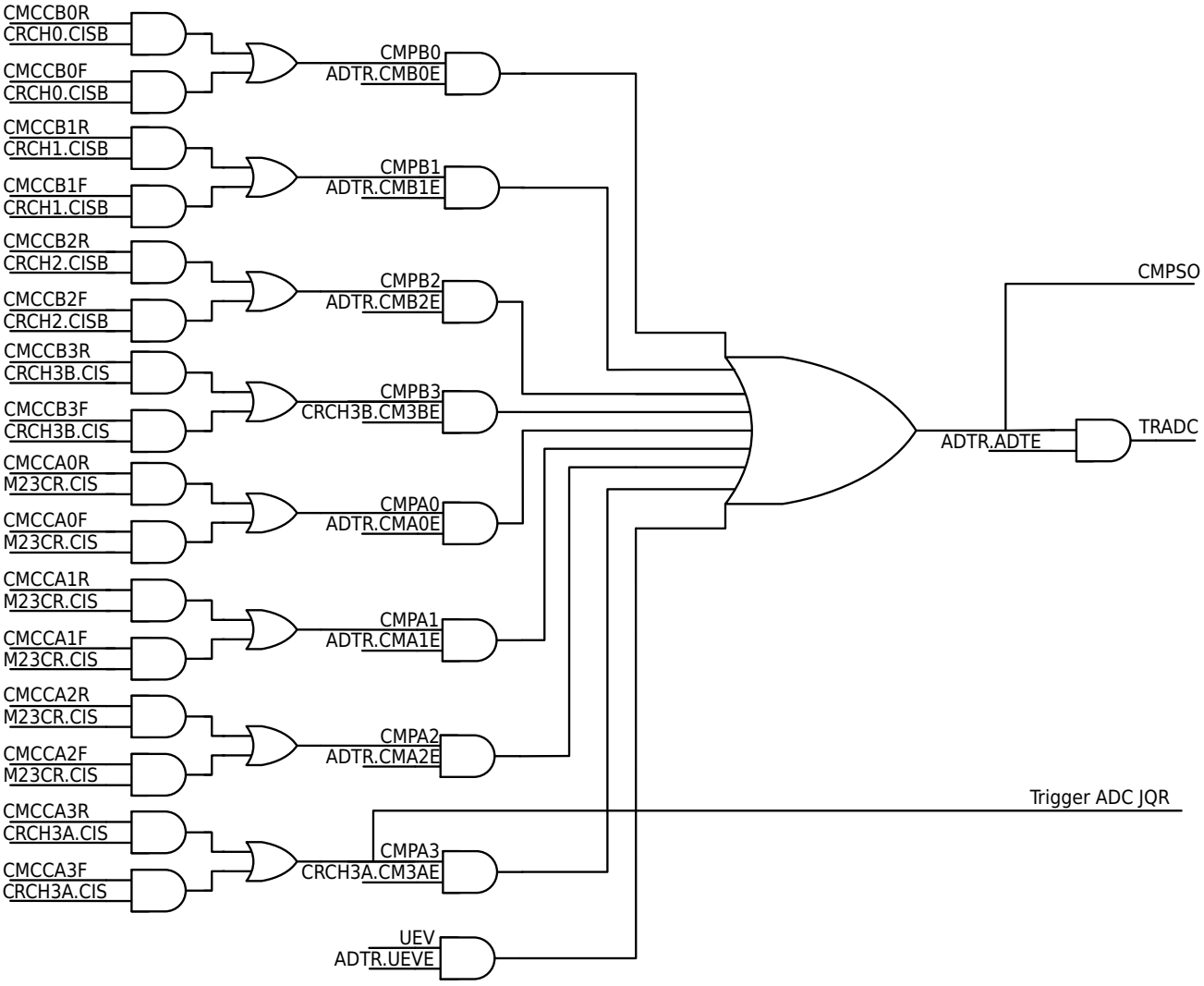



Figure 13-58 Timer Triggers ADC

## 13.3.10 Cycle-By-Cycle Current Limiting and Brake Control

### Cycle-By-Cycle Current Limiting

M23CR.OCCE can select to enable or disable the cycle-by-cycle current limiting function. M23CR.OCCS can select the OCREF_CLR signal from either the VC output signal REFCLR or the filtered phase-selected signal ETFP from the ETR port.

The CR2.TIMxCLR (x=0/1/2/3) of VC0/1 configure whether to enable the REFCLR signal to TIM0/1/2/3. CR2.POL determines whether the REFCLR signal is active-high level (CR2.POL=0) or active-low level


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

(CR2.POL=1). FLTR.ETP configures whether the ETR port input level is active-high level (FLTR.ETP=0) or active-low level(FLTR.ETP=1), while FLTR.FLTET sets the filtering time for ETR.

When DTR.AOE=0 and DTR.MOE=1, after enabling M23CR.OCCE, a valid OCREF_CLR signal can clear the current OCREFxy (x=0/1/2, y=A/B) signal (this clearing action is effective when FLTR.OCMyx>1 (x=0/1/2, y=A/B), while DTR.MOE will not be cleared. When the OCREF_CLR signal becomes invalid, the PWM output resumes after the next event update UEV.

#### Brake Control

Brake control can be implemented through the following methods:

1. VC comparator control: The CR2.POL register of VC0/1 determines whether the VC output brake level signal is active-high level (CR2.POL=0) or active-low level (CR2.POL=1). EnableVCx CR2.TIMBK, and enable DTR.VCE, when the VC output reaches the valid level, a valid brake condition is generated to control the braking function.
2. For external port control: Configure the designated port as the TIMx_BK (x=0/1/2/3) alternate function, set the brake active level (FLTR.BKP=0 for high level braking, FLTR.BKP=1 for low braking) in FLTR.BKP, configure the input filter time in FLTR.FLTBK, and enable DTR.BKE port braking. When a valid level is input from this port, the braking condition becomes active, enabling control of the braking function.
3. System Safety braking (XTH fail, lockup) control.
4. M23CR.BG software brake control.

When DTR.AOE=0 and DTR.MOE=1, after the braking condition takes effect, the DTR.MOE bit will be cleared to 0, and the PWM pins will perform the specified actions according to the preset configuration of CRCHx.BKSA/CRCHx.BKSB (x=0/1/2). To restore PWM output, the DTR.MOE bit must be set again after the braking condition is deactivated.

When DTR.AOE=1 and DTR.MOE=1, after the braking condition takes effect, the DTR.MOE bit will be cleared to 0, and the PWM pins will perform the specified actions according to the preset configuration of CRCHx.BKSA/CRCHx.BKSB. Once the braking condition is deactivated, DTR.MOE will be automatically set, and PWM output will resume.

The TIM0/1/2 outputs of the two channels can either use TIM0 brake port to control the braking function of TIM1/2 or use their respective brake inputs to control their own braking functions. When DTR.BKSEL is set to 1, TIM1/2 share the brake input of TIM0.

### 13.3.11 Interrupt

Table 13-13 TIM0/1/2/3 Interrupt List

| Interrupt Control Register | Interrupt Name |
|---|---|
| MxCR.UIE (x=0/1/23) | Overflow interrupt/event update interrupt |
| CRCHx.CIEA (x=0/1/2) | Channel (0/1/2) A compare capture interrupt |
| CRCHx.CIEB (x=0/1/2) | Channel (0/1/2) B compare capture interrupt |
| CRCH3y.CIE (y=A/B) | Channel 3A/B compare interrupt |
| M23CR.BIE | Brake interrupt |
| M23CR.TIE | Trigger interrupt |
| M23CR.OVE | Overflow interrupt |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Interrupt Control Register | Interrupt Name |
|---|---|
| M23CR.UNDE | Underflow interrupt |

### 13.3.12 Timer Interconnection

The TRGO output signal can be connected to the ITR signal of other timers. The connection relationship is as follows:

Table 13-14 Timer Interconnection Signal List

| | ITR0 | ITR1 | ITR2 | ITR3 |
|---|---|---|---|---|
| Timer0 | - | TIM1_TRGO | TIM2_TRGO | TIM3_TRGO |
| Timer1 | TIM0_TRGO | - | TIM2_TRGO | TIM3_TRGO |
| Timer2 | TIM0_TRGO | TIM1_TRGO | - | TIM3_TRGO |
| Timer3 | TIM0_TRGO | TIM1_TRGO | TIM2_TRGO | - |

### 13.3.13 Input Interconnection

#### 13.3.13.1 GATE Input Interconnection

When GPIO_TIMGS.TIMx_G=0b000, the gating GATE input is the port input selected by Pxy_SEL; when GPIO_TIMGS.TIMx_G=0b001-0b111, it connects to the input or output of other modules. Details see the "Port Auxiliary Function Timer Gate Selection (GPIO_TIMGS)" description in the "General-Purpose I/O (GPIO)" chapter.

#### 13.3.13.2 ETR Input Interconnection

When GPIO_TIMES.TIMx_E=0b000, the ETR input is the port input selected by Pxy_SEL; when GPIO_TIMES.TIMx_E=0b001-0b111, it connects to the input or output of other modules. For detailed information, refer to the description of the register "Port Auxiliary Function Timer ETR Selection (GPIO_TIMES)" in the "General-Purpose I/O (GPIO)" chapter.

#### 13.3.13.3 CHx Capture Input Interconnection

When GPIO_TIMCPS0.TIMx_Cy=0b000, the capture input (Timer0/1/2 CHA, Timer3 CH0A/CH0B) is the port input selected by Pxy_SEL. When GPIO_TIMCPS0.TIMx_Cy=0b001-0b111, it connects to the input or output of other modules. For detailed information, refer to the description of the register "Port Auxiliary Function Timer Capture Input Selection 0 (GPIO_TIMCPS0)" in the "General-Purpose I/O (GPIO)" chapter.

### 13.3.14 DMA

Timer supports software and hardware to trigger DMA for data transfer. Data is supported to be written to the timer from other locations, or read and written from the timer to other locations. It can be applied to the automatic handling of data after data capture and the automatic adjustment of the pulse width changing the period value or duty cycle. Each timer has two DMA requests: request A trigger source can select between compare capture A or external trigger ETR; request B trigger source can select between compare capture B or update event UEV.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

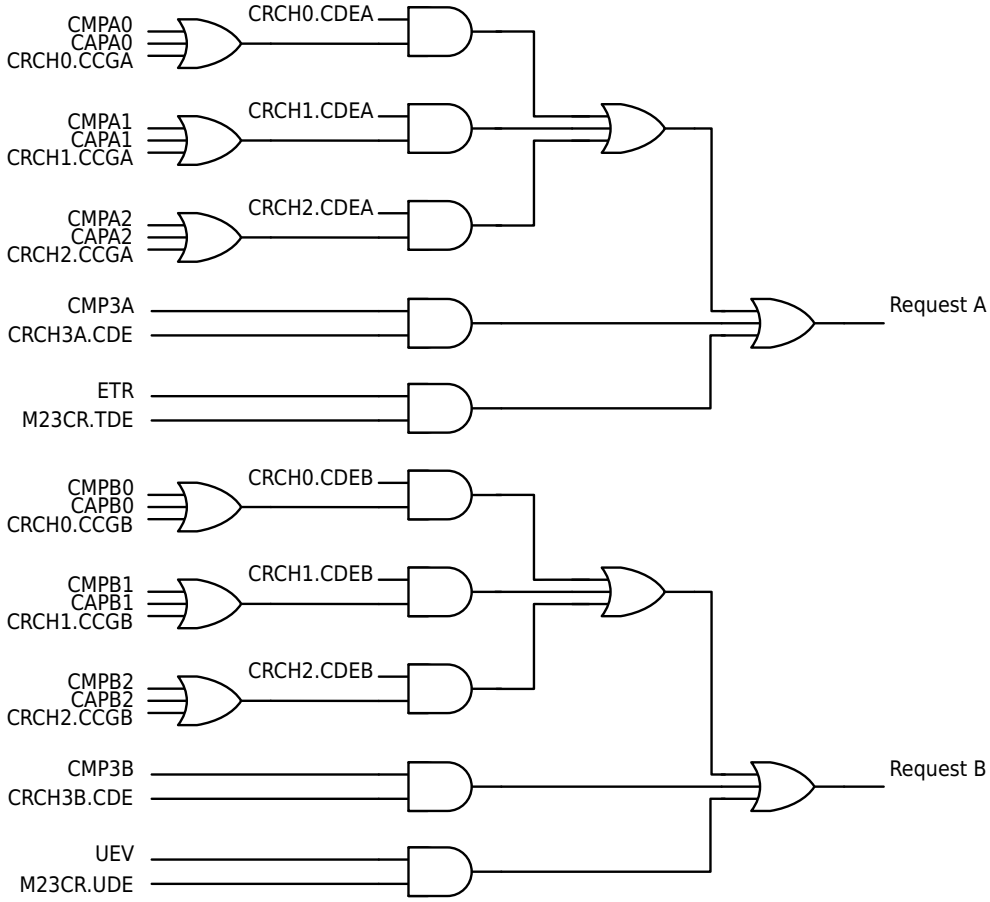



Figure 13-59 DMA Request Trigger Sources

## 13.4 Operation Example

### 13.4.1 Mode0 Counting Timer Function Configuration Example

#### Procedure

- **Reload Timer Setting**
  - **Step1.** Configure timer mode M0CR.MODE=0;
  - **Step2.** Set load value ARR;
  - **Step3.** Set counter initial value CNT;
  - **Step4.** Clear interrupt flag;
  - **Step5.** Enable interrupt M0CR.UIE;
  - **Step6.** Enable reload mode M0CR.MD;
  - **Step7.** Start timer M0CR.CTEN.
- **Gating External Clock Free Counting Setting**
  - **Step1.** Configure timer mode M0CR.MODE=0;
  - **Step2.** Set counter initial value CNT;
  - **Step3.** Enable gating function M0CR.GATE;
  - **Step4.** Select gating active level M0CR.GATEP;
  - **Step5.** Clear interrupt flag;
  - **Step6.** Enable interrupt M0CR.UIE;
  - **Step7.** Enable external clock mode M0CR.CT;
  - **Step8.** Start timer M0CR.CTEN.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

- **BUZZER Output Control**
  - **Step1.** Set appropriate ARR value based on output frequency;
  - **Step2.** Configure timer as reload mode, refer to Reload Timer Settings;
  - **Step3.** Enable output enable DTR.MOE;
  - **Step4.** Start another timer to control M0CR.TOGEN for implementing interval frequency output.

### 13.4.2 Mode1 Pulse Width Measurement PWC Configuration Example

#### Procedure

- **Pulse Low Level Measurement Setting**
  - **Step1.** Configure pulse measurement mode M1CR.MODE=1;
  - **Step2.** Set MSCR.TS to select measurement signal, then configure MSCR.IA0S/IB0S (IB0S cannot be set to 1) based on MSCR.TS setting to select measurement channel;
  - **Step3.** Set M1CR.EDG2ND=0 and M1CR.EDG1ST=1 to measure low level;
  - **Step4.** Clear interrupt flag;
  - **Step5.** Enable overflow interrupt M1CR.UIE;
  - **Step6.** Enable measurement completion interrupt CR0.CIEA;
  - **Step7.** Enable timer M1CR.CTEN;
  - **Step8.** In interrupt service routine, read CCR0A and overflow count then clear interrupt flag;
  - **Step9.** Wait for next measurement.
- **Pulse High Level Single Measurement Setting**
  - **Step1.** Configure pulse measurement mode M1CR.MODE=1;
  - **Step2.** Set MSCR.TS to select measurement signal, then configure MSCR.IA0S/IB0S (IB0S cannot be set to 1) based on MSCR.TS setting to select measurement channel;
  - **Step3.** Set pulse single-shot measurement mode M1CR.ONESHOT=1;
  - **Step4.** Configure M1CR.EDG2ND=1 and M1CR.EDG1ST=0 to measure high level;
  - **Step5.** Clear interrupt flag;
  - **Step6.** Enable overflow interrupt M1CR.UIE;
  - **Step7.** Enable measurement completion interrupt CR0.CIEA;
  - **Step8.** Enable timer M1CR.CTEN;
  - **Step9.** In interrupt service routine, read CCR0A and overflow count then clear interrupt flag;
  - **Step10.** Measurement is completed.

### 13.4.3 Mode2/3 Compare Capture Mode Configuration Example

#### Procedure

- **Edge-aligned Independent PWM Output Setting**
  - **Step1.** Set mode M23CR.MODE=2;
  - **Step2.** Configure sawtooth wave counting direction M23CR.DIR;
  - **Step3.** Set PWM period value ARR;
  - **Step4.** Set counter initial value (must be smaller than period value);
  - **Step5.** Configure PWM compare value CCRxA/CCRxB;
  - **Step6.** Set PWM output compare modes FLTR.OCMAx and FLTR.OCMBx to 6 or 7;
  - **Step7.** Clear relevant interrupt flags;
  - **Step8.** Enable corresponding interrupts;
  - **Step9.** Configure output polarities FLTR.CCPAx and FLTR.CCPBx;


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

**Step10.** Enable output DTR.MOE;

**Step11.** Enable timer M23CR.CTEN.

- **Center-Aligned Complementary PWM Output Setting**

  **Step1.** Set mode M23CR.MODE=3;

  **Step2.** Configure complementary output M23CR.COMP;

  **Step3.** Set PWM period value ARR;

  **Step4.** Set counter initial value (must be smaller than period value);

  **Step5.** Set PWM compare value CCRxA (CCRxB does not need to be set as PWM output is independent of this register);

  **Step6.** Set PWM output compare modes FLTR.OCMAx and FLTR.OCMBx to 6 or 7;

  **Step7.** Clear relevant interrupt flags;

  **Step8.** Enable corresponding interrupts;

  **Step9.** Configure output polarities FLTR.CCPAx and FLTR.CCPBx;

  **Step10.** Enable output DTR.MOE;

  **Step11.** Enable timer M23CR.CTEN.

- **Asymmetric Center-Aligned Complementary PWM Output with Dead-Time Setting**

  **Step1.** Set mode M23CR.MODE=3;

  **Step2.** Configure complementary output M23CR.COMP;

  **Step3.** Enable dual-point comparison enable M23CR.PWM2S;

  **Step4.** Set PWM period value ARR;

  **Step5.** Set counter initial value (must be smaller than period value);

  **Step6.** Set PWM compare values CCRxA and CCRxB (up-counting comparison point is CCRxA, down-counting comparison point is CCRxB);

  **Step7.** Set PWM output compare modes FLTR.OCMAx and FLTR.OCMBx to 6 or 7;

  **Step8.** Clear relevant interrupt flags;

  **Step9.** Enable corresponding interrupts;

  **Step10.** Configure output polarities FLTR.CCPAx and FLTR.CCPBx;

  **Step11.** Set dead time enbale DTR.DTEN;

  **Step12.** Set dead time duration DTR.DTR;

  **Step13.** Enable output DTR.MOE;

  **Step14.** Enable timer M23CR.CTEN.

- **Capture Function Configuration**, setting CH0B for rising edge capture function

  **Step1.** Select counter counting mode, set M23CR.MODE=2;

  **Step2.** Configure CH0B as capture mode CRCH0.CSB=1;

  **Step3.** Select input filter FLTR.FLTB0 as needed;

  **Step4.** Select capture edge CRCH0.CRB=1;

  **Step5.** Set ARR to change the period value;

  **Step6.** Clear relevant interrupt flags;

  **Step7.** Clear the capture flag and enable the corresponding interrupt;

  **Step8.** Start timer M23CR.CTEN;

  **Step9.** After detecting the capture flag, read CCR0B to obtain the captured value.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

> **Note**
> When the timer is not started, if there is a valid capture edge, the capture flag and capture action will also be generated.

- **Complementary PWM Output Uses GATE as PWM Output Function**
  - **Step1.** Configure as PWM complementary output mode, with steps referenced to Center-aligned Complementary PWM Output Setup;
  - **Step2.** Set CCR0B to configure the GATE PWM comparison value;
  - **Step3.** Set FLTR.OCMB0=6/7 to configure the PWM output mode;
  - **Step4.** Enable PWM output DTR.MOE;
  - **Step5.** Start the timer M23CR.CTEN.
- **Complementary PWM Output Uses GATE Falling Edge as Capture Input Function**
  - **Step1.** Configure as PWM complementary output mode, with steps referenced to Center-aligned Complementary PWM Output Setup;
  - **Step2.** Set GATE as capture input M23CR.CSG;
  - **Step3.** Select GATE capture input edge configuration M23CR.CFG (GATE as capture input B channel filter setting is invalid);
  - **Step4.** Clear relevant interrupt flags;
  - **Step5.** Clear the capture flag and enable the corresponding interrupt;
  - **Step6.** Start timer M23CR.CTEN;
  - **Step7.** After detecting the capture flag, read CCR0B to obtain the captured value.

### 13.4.4 Complementary PWM Output B Compare as ADC Trigger Function

#### Procedure

- **Step1.** Configure as PWM complementary output mode, with setup steps referring to Center-Aligned Complementary PWM Output Setting
- **Step2.** Set CCRxB to set ADC trigger comparison point
- **Step3.** Set CRCHx.CISB to select up-counting and down-counting comparison matching (sawtooth wave does not need to be selected)
- **Step4.** Select the source ADTR for ADC trigger comparison
- **Step5.** Enable ADC triggers ADTR.ADTE

### 13.4.5 Simultaneous PWM Output Startup for Timer TIM0/1/2 Configuration (Using TIM0 CTEN to Trigger TIM1/2) Example

#### Procedure

- **Step1.** Refer to the PWM output setting to set the pulse adjustment output of timer 0/1/2
- **Step2.** Configure TIM0 as master mode MSCR.MSM=1
- **Step3.** Set TIM0 CTEN to trigger the other two timers MSCR.MMS=1
- **Step4.** Keep MSCR.MSM=0 of TIM1/2
- **Step5.** Set TIM1/2 as trigger mode MSCR.SMS=2
- **Step6.** Select TIM0 TRGO as trigger source for TIM1/2, MSCR.TS=1
- **Step7.** Enable timer TIM0, startup the timer M23CR.CTEN=1


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 13.4.6 Simultaneous PWM Output Startup for Timer TIM0/1/2 Configuration (Using TIM1 UG to Trigger TIM0/2) Example

#### Procedure

**Step1.** Refer to the PWM output setting to set the pulse adjustment output of timer 0/1/2

**Step2.** Set TIM1 as trigger master mode (MSCR.MSM=1)

**Step3.** Set TIM1 UG to trigger the other two timers MSCR.MMS=0

**Step4.** Keep MSCR.MSM=0 of TIM0/2

**Step5.** Set TIM0/2 as trigger mode MSCR.SMS=2

**Step6.** Select TIM0 TRGO as trigger source for TIM0/2, MSCR.TS=2

**Step7.** Enable timer TIM1, startup the timer M23CR.UG=1

### 13.4.7 DMA Configuration Example

#### Procedure

- **Capture Data DMA Data Transmission**

  **Step1.** Enable DMA

  **Step2.** Enable DMA channel enable

  **Step3.** Select the channel for timer DMA

  **Step4.** Set the transmission type, transmission length, transmission method

  **Step5.** Set source start address, destination start address

  **Step6.** Set the increment method of source address and destination address

  **Step7.** Enable DMA interrupts as needed

  **Step8.** Set up data capture with reference to the data capture process

  **Step9.** Enable CRCHx.CDEy to enable capture data triggering DMA transfer.

  **Note**

  Set the source address to the channel compare capture register, with the source address fixed and the destination address variable.

- **Pulse Adjusted DMA Data Transmission**

  **Step1.** Enable DMA

  **Step2.** Enable DMA channel enable

  **Step3.** Select the channel for timer DMA

  **Step4.** Set the transmission type, transmission length, transmission method

  **Step5.** Set source start address, destination start address

  **Step6.** Set the increment method of source address and destination address

  **Step7.** Enable DMA interrupts as needed

  **Step8.** Reference pulse adjustment output setting pulse modulation output

  **Step9.** Select the condition of hardware triggering DMA according to needs (UEV trigger or comparison match trigger)

  **Note**

  Set the destination address to the channel compare capture register, with the source address variable and the destination address fixed.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

## 13.5 Register Description

### 13.5.1 Register List

- Timer0 base address: 0x40003000
- Timer1 base address: 0x40003400
- Timer2 base address: 0x40003800
- Timer3 base address: 0x40009800

**Table 13-15 Mode0 Timer Register Offset**

| Offset | Register | Description | Width |
|---|---|---|---|
| 0x000 | TIM_ARR | 16-Bit Mode Reload Register/Period | 32 |
| 0x004 | TIM_CNT | 16-Bit Mode Count Register | 32 |
| 0x008 | TIM_CNT32 | 32-Bit Mode Count Register<br>**Note**<br>TIM_CNT32 is the remapping register for TIM_ARR and TIM_CNT. In order to facilitate the 32-bit timer write operation. | 32 |
| 0x00C | TIM_M0CR | Mode0 Control Register | 32 |
| 0x010 | TIM_IFR | Interrupt Flag Register | 32 |
| 0x014 | TIM_ICLR | Interrupt Flag Clear Register | 32 |
| 0x030 | TIM_DTR | Dead Time Register | 32 |

**Table 13-16 Mode1 Pulse Width Measurement (PWM) Register Offset**

| Offset | Register | Description | Width |
|---|---|---|---|
| 0x004 | TIM_CNT | 16-Bit Mode Count Register | 32 |
| 0x00C | TIM_M1CR | Mode1 Control Register | 32 |
| 0x010 | TIM_IFR | Interrupt Flag Register | 32 |
| 0x014 | TIM_ICLR | Interrupt Flag Clear Register | 32 |
| 0x018 | TIM_MSCR | Master-Slave Mode Control Register | 32 |
| 0x01C | TIM_FLTR | Input Filter Register | 32 |
| 0x024 | TIM_CR0 | Control Register | 32 |
| 0x03C | TIM_CCR0A | Compare Capture Register | 32 |

**Table 13-17 Mode2/3 Compare-Capture Mode Register Offset**

| Offset | Register | Description | Width |
|---|---|---|---|
| 0x000 | TIM_ARR | 16-Bit Mode Reload Register/Period | 32 |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Offset | Register | Description | Width |
|---|---|---|---|
| 0x004 | TIM_CNT | 16-Bit Mode Count Register | 32 |
| 0x00C | TIM_M23CR | Mode2/3 Control Register | 32 |
| 0x010 | TIM_IFR | Interrupt Flag Register | 32 |
| 0x014 | TIM_ICLR | Interrupt Flag Clear Register | 32 |
| 0x018 | TIM_MSCR | Master-Slave Mode Control Register | 32 |
| 0x01C | TIM_FLTR | Output Control/Input Filter Register | 32 |
| 0x020 | TIM_ADTR | ADC Trigger Control Register | 32 |
| 0x024 | TIM_CRCH0 | Channel 0 Control Register | 32 |
| 0x028 | TIM_CRCH1 | Channel 1 Control Register (only TIM3 exists) | 32 |
| 0x02C | TIM_CRCH2 | Channel 2 Control Register (only TIM3 exists) | 32 |
| 0x030 | TIM_DTR | Dead Time Register | 32 |
| 0x034 | TIM_RCR | Repeat Cycle Setting Value Register | 32 |
| 0x038 | TIM_ARRDM | Reload Register/Period Mapped Address | 32 |
| 0x03C | TIM_CCR0A | Channel 0A Compare/Capture Register | 32 |
| 0x040 | TIM_CCR0B | Channel 0B Compare/Capture Register | 32 |
| 0x044 | TIM_CCR1A | Channel 1A Compare/Capture Register (only TIM3 exists) | 32 |
| 0x048 | TIM_CCR1B | Channel 1B Compare/Capture Register (only TIM3 exists) | 32 |
| 0x04C | TIM_CCR2A | Channel 2A Compare/Capture Register (only TIM3 exists) | 32 |
| 0x050 | TIM_CCR2B | Channel 2B Compare/Capture Register (only TIM3 exists) | 32 |
| 0x054 | TIM_CCR3A | Channel 3A Compare Register (only TIM3 exists) | 32 |
| 0x058 | TIM_CRCH3A | Channel 3A Control Register (only TIM3 exists) | 32 |
| 0x05C | TIM_CCR3B | Channel 3B Compare Register (only TIM3 exists) | 32 |
| 0x060 | TIM_CRCH3B | Channel 3B Control Register (only TIM3 exists) | 32 |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

## 13.5.2 Mode0 Count Timer Register

### 13.5.2.1 16-Bit Mode Reload Register (TIM_ARR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x000 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | ARR | | | | | | | | | |
| Access | | | | | | | | | | | | | | | | | | | | | | | RW | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:16* | *Reserved* | - | - |
| 15:0 | ARR | 16-bit reload timer reload value register | RW |

### 13.5.2.2 16-Bit Mode Count Register (TIM_CNT)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x004 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | CNT | | | | | | | | | |
| Access | | | | | | | | | | | | | | | | | | | | | | | RW | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:16* | *Reserved* | - | - |
| 15:0 | CNT | 16-bit reload timer count value register | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 13.5.2.3 32-Bit Mode Count Register (TIM_CNT32)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x008 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | CNT32 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Access | RW | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 31:0 | CNT32 | 32-bit timer count value register<br>Valid only in Mode 0 32-bit timer free-counting mode, and writing to this register disables in other modes. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 13.5.2.4 Mode0 Control Register (TIM_M0CR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x00C | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00600008 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | MODE | | Reserved | UIE | GATEP | GATE | Reserved | PRS | | | TOGEN | CT | MD | CTEN |
| Access | | | | | | | | | | | | | | | | | | | RW | | | RW | RW | RW | | RW | | | RW | RW | RW | RW |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:14* | *Reserved* | - | - |
| 13:12 | MODE | Working mode<br>● 0b00: Timer mode 0<br>● 0b01: PWC mode<br>● 0b10: Sawtooth wave mode<br>● 0b11: Triangle wave mode | RW |
| *11* | *Reserved* | - | - |
| 10 | UIE | Interrupt enable control<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 9 | GATEP | Port GATE polarity control<br>● 0b0: Port GATE high level is valid<br>● 0b1: Port GATE low level is valid | RW |
| 8 | GATE | Timer gating enable<br>● 0b0: No gating, timer works when CTEN=1<br>● 0b1: Gating enabled, timer only operates when GATE port is valid and CTEN=1 | RW |
| *7* | *Reserved* | - | - |
| 6:4 | PRS | Internal clock frequency division selection<br>● 0b000: 1<br>● 0b001: 2<br>● 0b010: 4<br>● 0b011: 8<br>● 0b100: 16<br>● 0b101: 32<br>● 0b110: 64<br>● 0b111: 256 | RW |
| 3 | TOGEN | Toggle output enable in mode 0<br>● 0b0: Toggle output disabled, CHA and CHB outputs remain low level<br>● 0b1: Toggle output enabled | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 2 | CT | Count clock selection<br>● 0b0: Internal count clock TCLK<br>● 0b1: External count clock ETR | RW |
| 1 | MD | Mode selection 32 timing/16 timing mode selection<br>● 0b0: 32-bit free counting<br>● 0b1: 16-bit reload counting | RW |
| 0 | CTEN | Timer enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 13.5.2.5 Interrupt Flag Register (TIM_IFR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x010 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | UIF |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | RO |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:1* | *Reserved* | - | - |
| 0 | UIF | Overflow interrupt<br>● 0b0: Counter is not overflowed<br>● 0b1: Counter overflowed | RO |

### 13.5.2.6 Interrupt Flag Clear Register (TIM_ICLR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x014 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x003FFFFF(TIM0/1/2);0x000FFFFF(TIM3) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | UIF |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | WO |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:1* | *Reserved* | - | - |
| 0 | UIF | Overflow interrupt clear<br>Write 0 to clear flag, write 1 has no effect; read always returns 1. | WO |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 13.5.2.7 Dead Time Register (TIM_DTR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x030 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | MOE | Reserved | | | | | | | | | | | |
| Access | | | | | | | | | | | | | | | | | | | | RW | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:13* | *Reserved* | - | - |
| 12 | MOE | Toggle output enable<br>● 0b0: Toggle output in input status<br>● 0b1: Toggle port in output status | RW |
| *11:0* | *Reserved* | - | - |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

## 13.5.3 Mode1 Pulse Width Measurement PWC Register

### 13.5.3.1 16-Bit Mode Count Register (TIM_CNT)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x004 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | CNT | | | | | | | | | | | | | | | |
| Access | | | | | | | | | | | | | | | | | RW | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:16* | *Reserved* | - | - |
| 15:0 | CNT | Count value register | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 13.5.3.2 Mode1 Control Register (TIM_M1CR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x00C | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00600008 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | ONESHOT | MODE | | Reserved | UIE | EDG2ND | EDG1ST | Reserved | PRS | | | Reserved | CT | Reserved | CTEN |
| Access | | | | | | | | | | | | | | | | | | RW | RW | | RW | RW | RW | RW | | RW | | | RW | | RW | RW |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:15* | *Reserved* | - | - |
| 14 | ONESHOT | Single trigger mode selection<br>● 0b0: Cycle measurement<br>● 0b1: Automatically terminates after completing one pulse measurement, re-enable CTEN is required for subsequent measurements | RW |
| 13:12 | MODE | Working mode<br>● 0b00: Timer mode 0<br>● 0b01: PWC mode<br>● 0b10: Sawtooth wave mode<br>● 0b11: Triangle wave mode | RW |
| *11* | *Reserved* | - | - |
| 10 | UIE | Interrupt enable control<br>Counting to 0xFFFF will overflow and generate an overflow flag<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 9 | EDG2ND | Pulse width measurement end edge selection<br>Refer to the EDG1ST function description for details | RW |
| 8 | EDG1ST | Pulse width measurement start edge selection<br>**EDG2ND / EDG1ST / Measurement**<br>0b0 / 0b0 / Up edge-up edge period<br>0b0 / 0b1 / Low level width<br>0b1 / 0b0 / High level width<br>0b1 / 0b1 / Falling edge-falling edge period | RW |
| *7* | *Reserved* | - | - |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 6:4 | PRS | Internal clock frequency division selection<br>● 0b000: 1<br>● 0b001: 2<br>● 0b010: 4<br>● 0b011: 8<br>● 0b100: 16<br>● 0b101: 32<br>● 0b110: 64<br>● 0b111: 256 | RW |
| *3* | *Reserved* | - | - |
| 2 | CT | Count clock selection<br>● 0b0: Internal count clock TCLK<br>● 0b1: External count clock ETR | RW |
| *1* | *Reserved* | - | - |
| 0 | CTEN | Pulse width measurement enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |

### 13.5.3.3 Interrupt Flag Register (TIM_IFR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x010 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | CA0F | Reserved | UIF |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | RO | | RO |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:3* | *Reserved* | - | - |
| 2 | CA0F | Pulse width measurement interrupt flag<br>● 0b0: Pulse width measurement is not completed<br>● 0b1: Pulse width measurement completed | RO |
| *1* | *Reserved* | - | - |
| 0 | UIF | Overflow interrupt flag<br>● 0b0: Counter is not overflowed<br>● 0b1: Counter overflowed | RO |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 13.5.3.4 Interrupt Flag Clear Register (TIM_ICLR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x014 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x003FFFFF(TIM0/1/2);0x000FFFFF(TIM3) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | CA0F | Reserved | UIF |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | WO | | WO |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:3* | *Reserved* | - | - |
| 2 | CA0F | Pulse width measurement interrupt flag clear<br>Write 0 to clear flag, write 1 has no effect; read always returns 1. | WO |
| *1* | *Reserved* | - | - |
| 0 | UIF | Overflow interrupt clear<br>Write 0 to clear flag, write 1 has no effect; read always returns 1. | WO |

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 13.5.3.5 Master-Slave Mode Control Register (TIM_MSCR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x018 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | IB0S | IA0S | Reserved | | | TS | | | Reserved | | | | |
| Access | | | | | | | | | | | | | | | | | | | | RW | RW | | | | RW | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:13* | *Reserved* | - | - |
| 12 | IB0S | CH0B input selection<br>● 0b0: CH0B<br>● 0b1: Internal trigger TS selection signal | RW |
| 11 | IA0S | IA0 input selection<br>● 0b0: CH0A<br>● 0b1:<br>▶ CH0A CH1A CH2A XOR (TIM3)<br>▶ CH0A ETR GATE XOR (TIM0/1/2)<br>**Note**<br>Setting to 1 will cause input changes upon any port transition. | RW |
| *10:8* | *Reserved* | - | - |
| 7:5 | TS | Trigger selection<br>● 0b000: Signal ETFP after port ETR filtered phase selection<br>● 0b001: Internal interconnect signal ITR0<br>● 0b010: Internal interconnect signal ITR1<br>● 0b011: Internal interconnect signal ITR2<br>● 0b100: Internal interconnect signal ITR3<br>● 0b101: Setting prohibited<br>● 0b110: Signal IAFP after port CH0A filtered phase selection<br>● 0b111: Signal IBFP after port CH0B filtered phase selection | RW |
| *4:0* | *Reserved* | - | - |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 13.5.3.6 Input Filter Register (TIM_FLTR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x01C | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | ETP | FLTET | | | Reserved | | | | | | | | | | | | | | | | | | | | | FLTB0 | | | Reserved | FLTA0 | | |
| Access | RW | RW | | | | | | | | | | | | | | | | | | | | | | | | RW | | | | RW | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 31 | ETP | ETR input phase selection<br>● 0b0: Same phase<br>● 0b1: Reverse input | RW |
| 30:28 | FLTET | ETR filter control<br>The sampling clock is PCLK or PCLK frequency division, and the number of sampling points is N<br>● 0b000: No filter<br>● 0b001: $F_{sample}$=PCLK, N=2<br>● 0b010: $F_{sample}$=PCLK, N=4<br>● 0b011: $F_{sample}$=PCLK, N=6<br>● 0b100: $F_{sample}$=PCLK/4, N=4<br>● 0b101: $F_{sample}$=PCLK/4, N=6<br>● 0b110: $F_{sample}$=PCLK/8, N=4<br>● 0b111: $F_{sample}$=PCLK/8, N=6 | RW |
| *27:7* | *Reserved* | - | - |
| 6:4 | FLTB0 | CHB input filter control<br>The sampling clock is PCLK or PCLK frequency division, and the number of sampling points is N<br>● 0b000: No filter<br>● 0b001: $F_{sample}$=PCLK, N=2<br>● 0b010: $F_{sample}$=PCLK, N=4<br>● 0b011: $F_{sample}$=PCLK, N=6<br>● 0b100: $F_{sample}$=PCLK/4, N=4<br>● 0b101: $F_{sample}$=PCLK/4, N=6<br>● 0b110: $F_{sample}$=PCLK/8, N=4<br>● 0b111: $F_{sample}$=PCLK/8, N=6<br>**Note**<br>CR0.CSB needs to be configured as 1. | RW |
| *3* | *Reserved* | - | - |
| 2:0 | FLTA0 | CHA input filter control | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 2:0 | FLTA0 | The sampling clock is PCLK or PCLK frequency division, and the number of sampling points is N<br>● 0b000: No filter<br>● 0b001: $F_{sample}$=PCLK, N=2<br>● 0b010: $F_{sample}$=PCLK, N=4<br>● 0b011: $F_{sample}$=PCLK, N=6<br>● 0b100: $F_{sample}$=PCLK/4, N=4<br>● 0b101: $F_{sample}$=PCLK/4, N=6<br>● 0b110: $F_{sample}$=PCLK/8, N=4<br>● 0b111: $F_{sample}$=PCLK/8, N=6<br>**Note**<br>CR0.CSA needs to be configured as 1. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 13.5.3.7 Control Register (TIM_CR0)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x024 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00003000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | CIEA | Reserved | | CSB | CSA | Reserved | | | |
| Access | | | | | | | | | | | | | | | | | | | | | | | | RW | | | RW | RW | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:9* | *Reserved* | - | - |
| 8 | CIEA | Pulse width measurement complete interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| *7:6* | *Reserved* | - | - |
| 5 | CSB | B channel capture/compare function selection<br>When pulse width measurement uses channel B filtering, CSB must be set to 1<br>● 0b0: Compare mode<br>● 0b1: Capture mode | RW |
| 4 | CSA | A channel capture/compare function selection<br>When pulse width measurement uses channel A filtering, CSA must be set to 1<br>● 0b0: Compare mode<br>● 0b1: Capture mode | RW |
| *3:0* | *Reserved* | - | - |

### 13.5.3.8 Compare Capture Register (TIM_CCR0A)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x03C | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | CCR0A | | | | | | | | | | | | | | | |
| Access | | | | | | | | | | | | | | | | | RO | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:16* | *Reserved* | - | - |
| 15:0 | CCR0A | Pulse width measurement result | RO |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

## 13.5.4 Mode2/3 Compare Capture Mode Register

### 13.5.4.1 16-Bit Mode Reload Register (TIM_ARR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x000 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | ARR | | | | | | | | | | | | | | | |
| Access | | | | | | | | | | | | | | | | | RW | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:16* | *Reserved* | - | - |
| 15:0 | ARR | Reload register/period register<br>With cache function, when the counter is not enabled or the cache is not enabled, the cache register can be updated immediately | RW |

### 13.5.4.2 16-Bit Mode Count Register (TIM_CNT)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x004 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | CNT | | | | | | | | | | | | | | | |
| Access | | | | | | | | | | | | | | | | | RW | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:16* | *Reserved* | - | - |
| 15:0 | CNT | Reload timer count value register<br>When using PWM compare output, triangular wave mode or saw-tooth wave mode up-counting, the initialized CNT value must be less than the ARR value. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 13.5.4.3 Mode2/3 Control Register (TIM_M23CR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x00C | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00600008 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | UNDE | OVE | DIR | BG | UG | TG | OCCE | CIS | | BIE | TIE | TDE | URS | OCCS | CSG | ONESHOT | MODE | | UDE | UIE | CFG | CRG | BUFPEN | PRS | | PWM2S | CT | COMP | CTEN |
| Access | | | RW | RW | RW/RO | WO | WO | WO | RW | RW | | RW | RW | RW | RW | RW | RW | RW | RW | | RW | RW | RW | RW | RW | RW | | RW | RW | RW | RW |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:30* | *Reserved* | - | - |
| 29 | UNDE | Underflow interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 28 | OVE | Overflow interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 27 | DIR | Count direction<br>Write only in sawtooth wave mode. Read only in other modes, write invalid<br>● 0b0: Up-counting<br>● 0b1: Down-counting<br>**Note**<br>DIR is automatically cleared when switching from other modes to center-aligned mode. In the center-aligned mode, DIR is automatically cleared in reset mode by software event update and external trigger in slave mode. | RW/RO |
| 26 | BG | Software brake<br>Auto-clear. Write 1 to generate a software brake, write 0 has no effect. | WO |
| 25 | UG | Software update<br>Auto-clear. Write 1 to generate a software update, write 0 has no effect.<br>Initializes the counter and updates the buffer register to the corresponding registers (buffer enable), the PRS bit will also be cleared. | WO |
| 24 | TG | Software trigger<br>Auto-clear. Write 1 to generate a software trigger, write 0 has no effect.<br>Triggering is available in trigger mode SMS=2 and mode=2/3. | WO |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 23 | OCCE | OCREF clear enable<br>● 0b0: OCREF output unaffected by OCREF_CLR<br>● 0b1: OCREF_CLR signal can clear OCREF output | RW |
| 22:21 | CIS | Channel A compare match mode<br>● 0b00: No match<br>● 0b01: Compare match during up-counting<br>● 0b10: Compare match during down-counting<br>● 0b11: Compare match during both up/down counting<br>**Note**<br>Channel B compare match mode is controlled by the CISB bit in the CRCHx register.<br>This bit does not affect the OCREF output. | RW |
| 20 | BIE | Brake interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 19 | TIE | Trigger interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 18 | TDE | Trigger start DMA enbale<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 17 | URS | Update source<br>● 0b0: Overflow/underflow/software update UG/slave mode reset<br>● 0b1: Overflow/underflow | RW |
| 16 | OCCS | OCREF_CLR signal source selection<br>● 0b0: Voltage comparator VC output, VC selection is configured in the VCx_CR2 register.<br>● 0b1: Signal after ETR port filtered phase selection<br>When OCCE is enabled, OCREF_CLR can clear the OCREF compare output signal (valid when OCMyx>1), and the compare output resumes after the next UEV event. | RW |
| 15 | CSG | GATE capture/compare selection in PWM complementary mode<br>Valid only during PWM complementary output, using CCR0B as the GATE compare or capture channel<br>● 0b0: Compare<br>● 0b1: Capture | RW |
| 14 | ONESHOT | Single trigger mode selection<br>● 0b0: Cycle count<br>● 0b1: Timer stops after an event update occurs | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 13:12 | MODE | Working mode<br>● 0b00: Timer mode 0<br>● 0b01: PWC mode<br>● 0b10: Sawtooth wave mode<br>● 0b11: Triangle wave mode | RW |
| 11 | UDE | Update trigger DMA enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 10 | UIE | Update interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 9 | CFG | Falling edge capture valid control when GATE is used as capture input<br>Only valid when PWM complementary output<br>● 0b0: Lower edge capture is invalid<br>● 0b1: Lower edge capture is valid | RW |
| 8 | CRG | Rising edge capture valid control when GATE is used as capture input<br>Only valid when PWM complementary output<br>● 0b0: Upper edge capture is invalid<br>● 0b1: Upper edge capture is valid | RW |
| 7 | BUFPEN | Auto-reload ARR cache enable<br>● 0b0: Period cache disabled, written value immediately affects the period value<br>● 0b1: Period cache enabled, written value affects the period value after the next event update | RW |
| 6:4 | PRS | Internal clock frequency division selection<br>● 0b000: 1<br>● 0b001: 2<br>● 0b010: 4<br>● 0b011: 8<br>● 0b100: 16<br>● 0b101: 32<br>● 0b110: 64<br>● 0b111: 256 | RW |
| 3 | PWM2S | OCREFA dual-point compare selection<br>● 0b0: Dual-point compare enabled, uses CCRA and CCRB compares to control OCREFA output<br>● 0b1: Single-point compare enabled, only uses CCRA compare to control OCREFA output (default value)<br>**Note**<br>OCREFB remains unaffected and still uses CCRB to control OCREFB output. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 2 | CT | Count clock selection<br>● 0b0: Internal count clock TCLK<br>● 0b1: External count clock ETR | RW |
| 1 | COMP | PWM complementary output mode selection<br>● 0b0: Independent PWM output<br>● 0b1: Complementary PWM output | RW |
| 0 | CTEN | Timer enable<br>● 0b0: Disable<br>● 0b1: Enable<br>It can be enabled by external trigger, and this bit is automatically cleared at the end of ONESHOT mode | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 13.5.4.4 Interrupt Flag Register (TIM_IFR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x010 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | CB3F | CA3F | UNDF | OVF | TIF | BIF | CB2E | CB1E | CB0E | CA2E | CA1E | CA0E | CB2F | CB1F | CB0F | CA2F | CA1F | CA0F | Reserved | UIF |
| Access | | | | | | | | | | | | | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO | | RO |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:20* | *Reserved* | - | - |
| 19 | CB3F | Channel CH3B compare match flag<br>Only TIM3 exists<br>● 0b0: Not occurred<br>● 0b1: Occurred | RO |
| 18 | CA3F | Channel CH3A compare match flag<br>Only TIM3 exists<br>● 0b0: Not occurred<br>● 0b1: Occurred | RO |
| 17 | UNDF | Underflow interrupt flag<br>● 0b0: Not occurred<br>● 0b1: Occurred | RO |
| 16 | OVF | Overflow interrupt flag<br>● 0b0: Not occurred<br>● 0b1: Occurred | RO |
| 15 | TIF | Trigger interrupt flag<br>● 0b0: Not occurred<br>● 0b1: Occurred | RO |
| 14 | BIF | Brake interrupt flag<br>● 0b0: Not occurred<br>● 0b1: Occurred | RO |
| 13 | CB2E | Channel CH2B capture data loss flag<br>Only TIM3 exists<br>● 0b0: No data loss<br>● 0b1: Data loss | RO |
| 12 | CB1E | Channel CH1B capture data loss flag<br>Only TIM3 exists<br>● 0b0: No data loss<br>● 0b1: Data loss | RO |
| 11 | CB0E | Channel CH0B capture data loss flag<br>● 0b0: No data loss<br>● 0b1: Data loss | RO |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 10 | CA2E | Channel CH2A capture data loss flag<br>Only TIM3 exists<br>● 0b0: No data loss<br>● 0b1: Data loss | RO |
| 9 | CA1E | Channel CH1A capture data loss flag<br>Only TIM3 exists<br>● 0b0: No data loss<br>● 0b1: Data loss | RO |
| 8 | CA0E | Channel CH0A capture data loss flag<br>● 0b0: No data loss<br>● 0b1: Data loss | RO |
| 7 | CB2F | Channel CH2B capture/compare match flag<br>Only TIM3 exists<br>● 0b0: Not occurred<br>● 0b1: Occurred | RO |
| 6 | CB1F | Channel CH1B capture/compare match flag<br>Only TIM3 exists<br>● 0b0: Not occurred<br>● 0b1: Occurred | RO |
| 5 | CB0F | Channel CH0B capture/compare match flag<br>● 0b0: Not occurred<br>● 0b1: Occurred | RO |
| 4 | CA2F | Channel CH2A capture/compare match flag<br>Only TIM3 exists<br>● 0b0: Not occurred<br>● 0b1: Occurred | RO |
| 3 | CA1F | Channel CH1A capture/compare match flag<br>Only TIM3 exists<br>● 0b0: Not occurred<br>● 0b1: Occurred | RO |
| 2 | CA0F | Channel CH0A capture/compare match flag<br>● 0b0: Not occurred<br>● 0b1: Occurred | RO |
| *1* | *Reserved* | - | - |
| 0 | UIF | Event update interrupt flag<br>● 0b0: Not occurred<br>● 0b1: Occurred | RO |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 13.5.4.5 Interrupt Flag Clear Register (TIM_ICLR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x014 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x003FFFFF(TIM0/1/2);0x000FFFFF(TIM3) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | CB3F | CA3F | UNDF | OVF | TIF | BIF | CB2E | CB1E | CB0E | CA2E | CA1E | CA0E | CB2F | CB1F | CB0F | CA2F | CA1F | CA0F | Reserved | UIF |
| Access | | | | | | | | | | | | | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | | WO |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:20* | *Reserved* | - | - |
| 19 | CB3F | Channel CH3B compare match flag clear<br>Write 0 to clear flag, write 1 has no effect; read always returns 1.<br>Only TIM3 exists | WO |
| 18 | CA3F | Channel CH3A compare match flag clear<br>Write 0 to clear flag, write 1 has no effect; read always returns 1.<br>Only TIM3 exists | WO |
| 17 | UNDF | Underflow interrupt flag clear<br>Write 0 to clear flag, write 1 has no effect; read always returns 1. | WO |
| 16 | OVF | Overflow interrupt flag clear<br>Write 0 to clear flag, write 1 has no effect; read always returns 1. | WO |
| 15 | TIF | Trigger interrupt flag clear<br>Write 0 to clear flag, write 1 has no effect; read always returns 1. | WO |
| 14 | BIF | Brake interrupt flag clear<br>Write 0 to clear flag, write 1 has no effect; read always returns 1. | WO |
| 13 | CB2E | Channel CH2B capture data loss flag clear<br>Write 0 to clear flag, write 1 has no effect; read always returns 1.<br>Only TIM3 exists | WO |
| 12 | CB1E | Channel CH1B capture data loss flag clear<br>Write 0 to clear flag, write 1 has no effect; read always returns 1.<br>Only TIM3 exists | WO |
| 11 | CB0E | Channel CH0B capture data loss flag clear<br>Write 0 to clear flag, write 1 has no effect; read always returns 1. | WO |
| 10 | CA2E | Channel CH2A capture data loss flag clear<br>Write 0 to clear flag, write 1 has no effect; read always returns 1.<br>Only TIM3 exists | WO |
| 9 | CA1E | Channel CH1A capture data loss flag clear<br>Write 0 to clear flag, write 1 has no effect; read always returns 1.<br>Only TIM3 exists | WO |
| 8 | CA0E | Channel CH0A capture data loss flag clear<br>Write 0 to clear flag, write 1 has no effect; read always returns 1. | WO |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 7 | CB2F | Channel CH2B capture/compare match flag clear<br>Write 0 to clear flag, write 1 has no effect; read always returns 1.<br>Only TIM3 exists | WO |
| 6 | CB1F | Channel CH1B capture/compare match flag clear<br>Write 0 to clear flag, write 1 has no effect; read always returns 1.<br>Only TIM3 exists | WO |
| 5 | CB0F | Channel CH0B capture/compare match flag clear<br>Write 0 to clear flag, write 1 has no effect; read always returns 1. | WO |
| 4 | CA2F | Channel CH2A capture/compare match flag clear<br>Write 0 to clear flag, write 1 has no effect; read always returns 1.<br>Only TIM3 exists | WO |
| 3 | CA1F | Channel CH1A capture/compare match flag clear<br>Write 0 to clear flag, write 1 has no effect; read always returns 1.<br>Only TIM3 exists | WO |
| 2 | CA0F | Channel CH0A capture/compare match flag clear<br>Write 0 to clear flag, write 1 has no effect; read always returns 1. | WO |
| *1* | *Reserved* | - | - |
| 0 | UIF | Event update interrupt flag clear<br>Write 0 to clear flag, write 1 has no effect; read always returns 1. | WO |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 13.5.4.6 Master-Slave Mode Control Register (TIM_MSCR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x018 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | IB0S | IA0S | SMS | | | TS | | | MSM | CCDS | MMS | | |
| Access | | | | | | | | | | | | | | | | | | | | RW | RW | RW | | | RW | | | RW | RW | RW | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:13* | *Reserved* | - | - |
| 12 | IB0S | CH0B input selection<br>● 0b0: CH0B<br>● 0b1: Internal trigger TS selection signal | RW |
| 11 | IA0S | IA0 input selection<br>● 0b0: CH0A<br>● 0b1:<br>▶ CH0A CH1A CH2A XOR (TIM3)<br>▶ CH0A ETR GATE XOR (TIM0/1/2)<br>**Note**<br>Setting to 1 will cause input changes upon any port transition. | RW |
| 10:8 | SMS | Slave mode function selection<br>● 0b000: Use internal clock<br>● 0b001: Reset function<br>● 0b010: Trigger mode<br>● 0b011: External clock mode<br>● 0b100: Quadrature encoder counting mode1<br>● 0b101: Quadrature encoder counting mode2<br>● 0b110: Quadrature encoder counting mode3<br>● 0b111: Gating function | RW |
| 7:5 | TS | Trigger selection<br>● 0b000: Signal ETFP after port ETR filtered phase selection<br>● 0b001: Internal interconnect signal ITR0<br>● 0b010: Internal interconnect signal ITR1<br>● 0b011: Internal interconnect signal ITR2<br>● 0b100: Internal interconnect signal ITR3<br>● 0b101: Edge signal of port CH0A<br>● 0b110: Signal IAFP after port CH0A filtered phase selection<br>● 0b111: Signal IBFP after port CH0B filtered phase selection | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

<table>
<tr><th>Bit</th><th>Name</th><th>Functional Description</th><th>Access</th></tr>
<tr><td>4</td><td>MSM</td><td>Master-slave selection<br>● 0b0: No delay<br>● 0b1: Delay enabled, synchronously starts both master and slave counters<br>Note<br>When using the trigger mode, the slave mode is set to 0, and the master mode is set to 1, so that the master and slave counts can be started at the same time.</td><td>RW</td></tr>
<tr><td>3</td><td>CCDS</td><td>DMA compare trigger selection in compare mode<br>● 0b0: Compare match trigger DMA<br>● 0b1: Compare match does not trigger DMA, event update replaces compare match to trigger DMA</td><td>RW</td></tr>
<tr><td>2:0</td><td>MMS</td><td>Master mode input (TIMx_TRGO) selection<br>For internal interconnection, connected to ITRx of other timers
<table>
<tr><th></th><th>TIM0/1/2</th><th>TIM3</th></tr>
<tr><td>0b000</td><td>Software update UG, write M23CR.UG</td><td>Software update UG, write M23CR.UG<br>Software update TG, write M23CR.TG</td></tr>
<tr><td>0b001</td><td>Timer enable CTEN</td><td>Timer enable CTEN</td></tr>
<tr><td>0b010</td><td>Timer event update UEV</td><td>Timer event update UEV</td></tr>
<tr><td>0b011</td><td>Compare match select output CMPSO</td><td>Compare match select output CMPSO</td></tr>
<tr><td>0b100</td><td>Timer compare parameter output OCREF0A</td><td>Timer compare parameter output OCREF0A</td></tr>
<tr><td>0b101</td><td>Timer compare parameter output OCREF0B</td><td>Timer compare parameter output OCREF1A</td></tr>
<tr><td>0b110</td><td>Timer compare parameter output OCREF0B</td><td>Timer compare parameter output OCREF2A</td></tr>
<tr><td>0b111</td><td>Timer compare parameter output OCREF0B</td><td>Timer compare parameter output OCREF0B</td></tr>
</table>
Note<br>When MMS is configured as 0b000/0b010/0b011, the output is a pulse signal.</td><td>RW</td></tr>
</table>


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 13.5.4.7 Output Control/Input Filter Register (TIM_FLTR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x01C | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | ETP | FLTET | | | BKP | FLTBK | | | CCPB2 | OCMB2/FLTB2 | | | CCPA2 | OCMA2/FLTA2 | | | CCPB1 | OCMB1/FLTB1 | | | CCPA1 | OCMA1/FLTA1 | | | CCPB0/CCPB0 | OCMB0/FLTB0 | | | CCPA0/CCPA0 | OCMA0/FLTA0 | | |
| Access | RW | RW | | | RW | RW | | | RW | RW | | | RW | RW | | | RW | RW | | | RW | RW | | | RW | RW | | | RW | RW | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 31 | ETP | ETR input phase selection<br>● 0b0: Same phase<br>● 0b1: Inverted input | RW |
| 30:28 | FLTET | ETR filter control<br>The sampling clock is PCLK or PCLK frequency division, and the number of sampling points is N<br>● 0b000: No filter<br>● 0b001: $F_{sample}$=PCLK, N=2<br>● 0b010: $F_{sample}$=PCLK, N=4<br>● 0b011: $F_{sample}$=PCLK, N=6<br>● 0b100: $F_{sample}$=PCLK/4, N=4<br>● 0b101: $F_{sample}$=PCLK/4, N=6<br>● 0b110: $F_{sample}$=PCLK/8, N=4<br>● 0b111: $F_{sample}$=PCLK/8, N=6 | RW |
| 27 | BKP | Brake BK input phase selection<br>● 0b0: Same phase<br>● 0b1: Inverted input | RW |
| 26:24 | FLTBK | Brake input filter control<br>The sampling clock is PCLK or PCLK frequency division, and the number of sampling points is N<br>● 0b000: No filter<br>● 0b001: $F_{sample}$=PCLK, N=2<br>● 0b010: $F_{sample}$=PCLK, N=4<br>● 0b011: $F_{sample}$=PCLK, N=6<br>● 0b100: $F_{sample}$=PCLK/4, N=4<br>● 0b101: $F_{sample}$=PCLK/4, N=6<br>● 0b110: $F_{sample}$=PCLK/8, N=4<br>● 0b111: $F_{sample}$=PCLK/8, N=6 | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 23 | CCPB2 | Compare function: CH2B channel compare output phase control<br>● 0b0: Normal output<br>● 0b1: Inverted output | RW |
| 22:20 | OCMB2 | Compare function: CH2B channel compare control<br>Refer to OCMB0 setting | RW |
| | FLTB2 | Capture function: CH2B input channel filter setup<br>Refer to FLTBK setting | RW |
| 19 | CCPA2 | Compare function: CH2A channel compare output phase control<br>● 0b0: Normal output<br>● 0b1: Inverted output | RW |
| 18:16 | OCMA2 | Compare function: CH2A channel compare control<br>Refer to OCMB0 setting | RW |
| | FLTA2 | Capture function: CH2A input channel filter setup<br>Refer to FLTBK setting | RW |
| 15 | CCPB1 | Compare function: CH1B channel compare output phase control<br>● 0b0: Normal output<br>● 0b1: Inverted output | RW |
| 14:12 | OCMB1 | Compare function: CH1B channel compare control<br>Refer to OCMB0 setting | RW |
| | FLTB1 | Capture function: CH1B input channel filter setup<br>Refer to FLTBK setting | RW |
| 11 | CCPA1 | Compare function: CH1A channel compare output phase control<br>● 0b0: Normal output<br>● 0b1: Inverted output | RW |
| 10:8 | OCMA1 | Compare function: CH1A channel compare control<br>Refer to OCMB0 setting | RW |
| | FLTA1 | Capture function: CH1A input channel filter setup<br>Refer to FLTBK setting | RW |
| 7 | CCPB0 | Compare function: output compare mode<br>CCPBx compare output CHBx port polarity control<br>● 0b0: Normal output<br>● 0b1: Inverted output | RW |
| | CCPB0 | Encode count and slave mode gating function: input phase control<br>CCPB0 slave mode gating, reset, external trigger, external clock using CH0B port input polarity control<br>● 0b0: Normal input<br>● 0b1: Inverted input | RW |
| 6:4 | OCMB0 | Compare function: CH0B channel compare control<br>● 0b000: Force to 0<br>● 0b001: Force to 1<br>● 0b010: Force to 0 on compare match<br>● 0b011: Force to 1 on compare match | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 6:4 | OCMB0 | ● 0b100: Toggle on compare match<br>● 0b101: Output high level for one clock cycle on compare match<br>● 0b110: PWM mode 1<br>▶ Single point comparison<br>During up-counting, the output is high level when CNT<CCRxy; during down-counting, the output is low level when CNT>CCRxy.<br>▶ Dual point comparison<br>◆ For sawtooth wave up-counting, the output is low level when CCRxA<CNT≤CCRxB<br>◆ For sawtooth wave down-counting, the output is high level when CCRxA<CNT≤CCRxB<br>◆ For triangular wave up-counting, the output is high level when CNT<CCRxA; for down-counting, the output is low level when CNT>CCRxB<br>● 0b111: PWM mode 2<br>▶ Single point comparison<br>During up-counting, the output is low level when CNT<CCRxy; during down-counting, the output is high level when CNT>CCRxy<br>▶ Dual point comparison<br>◆ For sawtooth wave up-counting, the output is high level when CCRxA≤CNT<CCRxB<br>◆ For sawtooth wave down-counting, the output is low level when CCRxA≤CNT<CCRxB<br>◆ For triangular wave up-counting, the output is low level when CNT<CCRxA; for down-counting, the output is high level when CNT>CCRxB | |
| | FLTB0 | Capture function: CH0B input channel filter setup<br>Refer to FLTBK setting<br>**Note**<br>In order to ensure the PWM output setting of OCMB0, GATE is used as the capture input in PWM complementary mode, and the filter setting is invalid. | RW |
| 3 | CCPA0 | Compare function: output compare mode<br>CCPAx compare output CHAx port polarity control<br>● 0b0: Normal output<br>● 0b1: Inverted output | RW |
| | CCPA0 | Encode count and slave mode gating function: input phase control<br>CCPA0 slave mode gating, reset, external trigger, external clock using CH0A port input polarity control<br>● 0b0: Normal input<br>● 0b1: Inverted input | RW |
| 2:0 | OCMA0 | Compare function: channel A compare control<br>Refer to OCMB0 setting | RW |
| | FLTA0 | Capture function: CH0A input channel filter setup<br>Refer to FLTBK setting | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 13.5.4.8 ADC Trigger Control Register (TIM_ADTR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x020 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | ADTE | CMB2E | CMB1E | CMB0E | CMA2E | CMA1E | CMA0E | UEVE |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | RW | RW | RW | RW | RW | RW | RW | RW |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:8* | *Reserved* | - | - |
| 7 | ADTE | Enable ADC trigger global control<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 6 | CMB2E | Channel 2B compare match trigger ADC enable<br>Only TIM3 exists<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 5 | CMB1E | Channel 1B compare match trigger ADC enable<br>Only TIM3 exists<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 4 | CMB0E | Channel 0B compare match trigger ADC enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 3 | CMA2E | Channel 2A compare match trigger ADC enable<br>Only TIM3 exists<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 2 | CMA1E | Channel 1A compare match trigger ADC enable<br>Only TIM3 exists<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 1 | CMA0E | Channel 0A compare match trigger ADC enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 0 | UEVE | Event update trigger ADC enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 13.5.4.9 Channel x Control Register (TIM_CRCHx) (x=0/1/2)

Offset: 0x024 (CRCH0), 0x028 (CRCH1), 0x02C (CRCH2)

**Note**

CRCH1/CRCH2 only TIM3 exists.

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| - | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00003000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | CCGB | CCGA | CISB | | CDEB | CDEA | CIEB | CIEA | BUFEB | BUFEA | CSB | CSA | CFB_CRB/BKSB | | CFA_CRA/BKSA | |
| Access | | | | | | | | | | | | | | | | | WO | WO | RW | | RW | RW | RW | RW | RW | RW | RW | RW | RW | | RW | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:16* | *Reserved* | - | - |
| 15 | CCGB | Capture compare B software trigger<br>Hardware auto-clear, writing 1 triggers the following actions, writing 0 has no effect:<br>● In compare mode, generates interrupt only<br>● In capture mode, generates interrupt and captures counter value to capture register | WO |
| 14 | CCGA | Capture compare A software trigger<br>Hardware auto-clear, writing 1 triggers the following actions, writing 0 has no effect:<br>● In compare mode, generates interrupt only<br>● In capture mode, generates interrupt and captures counter value to capture register | WO |
| 13:12 | CISB | B channel compare match setting<br>● 0b00: No match<br>● 0b01: Compare match during up-counting<br>● 0b10: Compare match during down-counting<br>● 0b11: Compare match during both up/down counting | RW |
| 11 | CDEB | B capture compare trigger DMA enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 10 | CDEA | A capture compare trigger DMA enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 9 | CIEB | B capture compare trigger interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 8 | CIEA | A capture compare trigger interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 7 | BUFEB | Compare function: B compare cache enable control<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 6 | BUFEA | Compare function: A compare cache enable control<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 5 | CSB | B channel capture/compare function selection<br>● 0b0: Compare mode<br>● 0b1: Capture mode | RW |
| 4 | CSA | A channel capture/compare function selection<br>● 0b0: Compare mode<br>● 0b1: Capture mode | RW |
| 3:2 | CFB_CRB | Capture function: channel B rising falling edge capture enable<br>● 0b00: Disable<br>● 0b01: Rising edge capture<br>● 0b10: Falling edge capture<br>● 0b11: Both rising and falling edge capture | RW |
| | BKSB | Compare function: channel B output brake level control<br>● 0b00: High-impedance output<br>● 0b01: No effect on output<br>● 0b10: Force output low level<br>● 0b11: Force output high level<br>**Note**<br>No DEBUG_STOP mode. | RW |
| 1:0 | CFA_CRA | Capture function: channel A rising falling edge capture enable<br>● 0b00: Disable<br>● 0b01: Rising edge capture<br>● 0b10: Falling edge capture<br>● 0b11: Both rising and falling edge capture | RW |
| | BKSA | Compare function: channel A output brake level control<br>● 0b00: High-impedance output<br>● 0b01: No effect on output<br>● 0b10: Force output low level<br>● 0b11: Force output high level<br>**Note**<br>In DEBUG_STOP mode, the brake level control for both channel A and channel B is governed by BKSA. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 13.5.4.10 Dead Time Register (TIM_DTR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x030 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | VCE | SAFEEN | MOE | AOE | BKE | DTEN | BKSEL | DTR | | | | | | | |
| Access | | | | | | | | | | | | | | | | | | RW | RW | RW | RW | RW | RW | RW | RW | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:15* | *Reserved* | - | - |
| 14 | VCE | VC brake enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 13 | SAFEEN | Safety brake enbale (XTH fail, lockup)<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 12 | MOE | PWM output enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 11 | AOE | PWM output automatically enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 10 | BKE | Port brake enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 9 | DTEN | Dead time control enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 8 | BKSEL | Brake selection<br>● 0b0: Use own brake control<br>● 0b1: TIM1/2 uses brake control of TIM0<br>**Note**<br>TIM0 and TIM3 bit selection is invalid. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

<table>
<tr><th>Bit</th><th>Name</th><th>Functional Description</th><th>Access</th></tr>
<tr><td>7:0</td><td>DTR</td><td>Dead time
<table>
<tr><th>DTR</th><th>T</th><th>dt</th><th>step</th></tr>
<tr><td>DTR[7]=0b0</td><td>DTR[6:0] + 2</td><td>2-129</td><td>1</td></tr>
<tr><td>DTR[7:6]=0b10</td><td>{DTR[5:0] + 64}* 2 + 2</td><td>130-256</td><td>2</td></tr>
<tr><td>DTR[7:5]=0b110</td><td>{DTR[4:0] + 32}* 8 + 2</td><td>258-506</td><td>8</td></tr>
<tr><td>DTR[7:5]=0b111</td><td>{DTR[4:0] + 32}* 16 + 2</td><td>514-1010</td><td>16</td></tr>
</table></td><td>RW</td></tr>
</table>

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 13.5.4.11 Repeat Cycle Setting Value Register (TIM_RCR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x034 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | UND | OV | RCR | | | | | | | |
| Access | | | | | | | | | | | | | | | | | | | | | | | RW | RW | RW | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:10* | *Reserved* | - | - |
| 9 | UND | Underflow mask control for repeating counting<br>● 0b0: Down-counting overflow enable<br>● 0b1: Down-count overflow mask | RW |
| 8 | OV | Overflow mask control for repetiting counting<br>● 0b0: Up-counting overflow enable<br>● 0b1: Up-counting overflow mask | RW |
| 7:0 | RCR | Repetition period count value<br>Sets RCR+1 periods. The overflow controlled by [9:8] generates an update event. When the counter overflows up or down, the internal RCR_CNT decrements by 1. When counting reaches zero, RCR_CNT reloads the RCR value and generates an update event UEV signal. | RW |

### 13.5.4.12 Reload Register/Period Mapped Address (TIM_ARRDM)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x038 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | ARRDM | | | | | | | | | | | | | | | |
| Access | | | | | | | | | | | | | | | | | RW | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:16* | *Reserved* | - | - |
| 15:0 | ARRDM | Reload register/period register<br>With cache function, when the counter is not enabled or the cache is not enabled, the cache register can be updated immediately | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 13.5.4.13 Channel x Compare Capture Register (TIM_CCRxy) (x=0/1/2, y=A/B)

Offset: 0x03C (CCR0A), 0x040 (CCR0B), 0x044 (CCR1A), 0x048 (CCR1B), 0x04C (CCR2A), 0x050 (CCR2B)

**Note**

CCR1A/CCR1B/CCR2A/CCR2B only TIM3 exists.

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| - | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | CCRxy | | | | | | | | | | | | | | | |
| Access | | | | | | | | | | | | | | | | | RW | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:16* | *Reserved* | - | - |
| 15:0 | CCRxy | Compare capture register<br>Comparison with cache function | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 13.5.4.14 Channel 3A Compare Register (TIM_CCR3A) (Only TIM3 Exists)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x054 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | CCR3A | | | | | | | | | | | | | | | |
| Access | | | | | | | | | | | | | | | | | RW | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:16* | *Reserved* | - | - |
| 15:0 | CCR3A | Compare register<br>Comparison with cache function | RW |

### 13.5.4.15 Channel 3A Control Register (TIM_CRCH3A) (Only TIM3 Exists)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x058 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | CM3AE | C3AEN | CIS | | CDE | CIE | BUF |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | RW | RW | RW | | RW | RW | RW |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:7* | *Reserved* | - | - |
| 6 | CM3AE | ADC enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 5 | C3AEN | Channel compare enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 4:3 | CIS | Channel compare match setting<br>● 0b00: No match<br>● 0b01: Compare match during up-counting<br>● 0b10: Compare match during down-counting<br>● 0b11: Compare match during both up/down counting | RW |
| 2 | CDE | Compare trigger DMA enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 1 | CIE | Compare trigger interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 0 | BUF | Compare cache enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 13.5.4.16 Channel 3B Compare Register (TIM_CCR3B) (Only TIM3 Exists)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x05C | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | CCR3B | | | | | | | | | | | | | | | |
| Access | | | | | | | | | | | | | | | | | RW | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:16* | *Reserved* | - | - |
| 15:0 | CCR3B | Compare register<br>Comparison with cache function | RW |

### 13.5.4.17 Channel 3B Control Register (TIM_CRCH3B) (Only TIM3 Exists)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x060 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | CM3BE | C3BEN | CIS | | CDE | CIE | BUF |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | RW | RW | RW | | RW | RW | RW |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:7* | *Reserved* | - | - |
| 6 | CM3BE | ADC enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 5 | C3BEN | Channel compare enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 4:3 | CIS | Channel compare match setting<br>● 0b00: No match<br>● 0b01: Compare match during up-counting<br>● 0b10: Compare match during down-counting<br>● 0b11: Compare match during both up/down counting | RW |
| 2 | CDE | Compare trigger DMA enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 1 | CIE | Compare trigger interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 0 | BUF | Compare cache enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

# 14 RTC

## 14.1 Introduction

The Real Time Counter provides seconds, minutes, hours, day, week, month, and year information, and the number of days in a month and leap year can be automatically adjusted. Clock operation can be done in 24 or 12 hour format via the RTC_CR0. AMPM register bit. It provides a periodic interrupt with a minimum period of 0.5-32s, wake-up timer interrupt with ranging from 60μs-145Hour.

## 14.2 Features

**Table 14-1 Basic Features of RTC**

| Basic Features | Description |
|---|---|
| Clock source | Off-chip low-speed crystal oscillator XTL (32.768kHz)<br>On-chip low-speed oscillator RCL (32kHz)<br>Off-chip high-speed crystal oscillator XTH |
| Basic function | Can calculate seconds, minutes, hours, days, weeks, months, years between 00 and 99 |
| | Automatic leap year adjustment |
| | Configurable to 24 or 12 hour format |
| | Programmable start or stop |
| | Provides two sets of alarm functions |
| | With high-precision 1Hz square wave signal output |
| | Configurable wake-up timer |
| | Timestamp function |
| Interrupt | Supports configurable periodic interrupts with a period ranging from 0.5 to 32 seconds. |
| | Alarm interrupt A/B with down-to-the-second precision |
| | wake-up timer interrupt |
| | Timestamp interrupt |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

## 14.3 Functional Description

The clock source of the Real Time Counter can be configured as an external low-speed crystal oscillator, an external high-speed crystal oscillator, or the internal low-speed clock RCL; by default, the external low-speed crystal oscillator is used. The RTC_SEC, RTC_MIN, RTC_HOUR, RTC_DAY, RTC_WEEK, RTC_MON, RTC_YEAR, RTC_ALMA, and RTC_ALMB registers are only affected by power-on reset. Their states are undefined after power-on and require initialization. Other registers are only controlled by power-on reset and module reset, and other reset sources cannot reset these registers.

All date and time values written to or read from software are in BCD format and do not require hexadecimal-to-decimal conversion.

Any invalid date time will not be able to be written, such as 32nd day, 25th hour, 70th second, month B, etc.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 14.3.1 Function Block Diagram

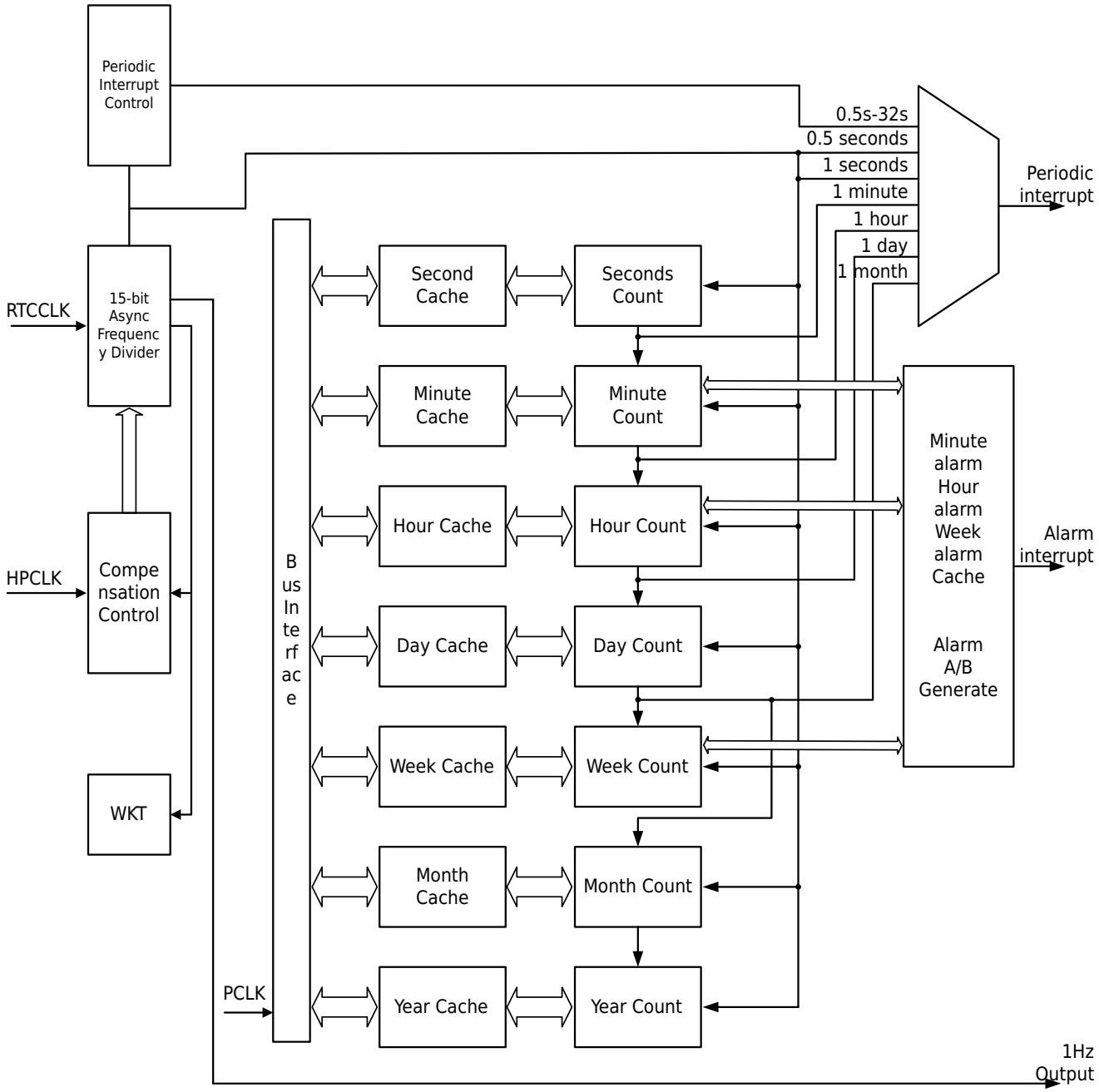



Figure 14-1 RTC Block Diagram

### 14.3.2 Power-on Setting

The RTC is reset once after power-on. When the system remains powered, no external reset requests can reset the RTC. The RTC will continue counting without interruption. After power-on, configure the initial calendar values, alarm settings, error compensation, and interrupts, and then start the RTC.

### 14.3.3 Clock Error Compensation

Due to the inherent error in the external crystal oscillator, compensation is required to achieve high-precision counting results, two levels of compensation accuracy are configurable (0.96ppm or 0.06ppm). Two compensation methods are available: one based on the internal clock RTCCLK error compensation; and the other based on high-speed clock HPCLK error compensation.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### Principle and Calculation of Internal Clock Error Compensation (0.96ppm)

Since the counter uses a 32.768kHz clock to count, if it is necessary to compensate the precision per second, it can only be compensated according to the integer period of 32.768kHz, and the minimum unit of compensation per second is (1/32768) $\times 10^6$=30.5ppm, unable to meet the high precision requirements.

Then, when realizing high-precision clock compensation under the counting clock of 32.768kHz, it is necessary to adjust the algorithm to expand the maximum compensation period by 32 times. Then, when the minimum unit that can only be compensated is 30.5ppm, the average compensation unit per second becomes 30.5ppm/32=0.96ppm. To meet the requirements of higher precision clock compensation and ensure the compensation is distributed evenly within each 32-second interval, this register introduces a setting with 5 fractional bits.

For example, when the 1Hz clock is directly output in the default state, the compensation target value is calculated by measuring the accuracy of the clock.

Assuming that the actual measured value is 1.0000203Hz, then:

Actual Vibration Frequency = 32768 × 1.0000203 ≈ 32768.6651904 (actual oscillation frequency needs to be accurate to at least three decimal places)

$$\begin{aligned}\text{Compensation Target Value} &= (\text{Actual Vibration Frequency} - \text{Target Frequency})/\text{Target Frequency} \times 10^6 \\ &= (32768.6651904 - 32768)/32768 \times 10^6 \\ &= 20.3\text{ppmppm}\end{aligned}$$

Based on the formula:

$$\text{CR}[8:0] = \left(\frac{\text{Compensation Target Value [ppm]} \times 2^{15}}{10^6}\right)_{\text{Take 2's complement}} + 0001.00000B$$

If the compensation target value is +20.3ppm, calculate the corresponding register value as follows:

$$\begin{aligned}\text{CR}[8:0] &= (20.3 \times 2^{15}/10^6)_{\text{Take 2's complement}} + 0001.00000\text{B} \\ &= (0.6651904)_{\text{Take 2's complement}} + 0001.00000\text{B} \\ &= 0000.10101\text{B} + 0001.00000\text{B} \\ &= 0001.10101\text{B}\end{aligned}$$

If the compensation target value is -20.3ppm, calculate the corresponding register value as follows:

$$\begin{aligned}\text{CR}[8:0] &= (-20.3 \times 2^{15}/10^6)_{\text{Take 2's complement}} + 0001.00000\text{B} \\ &= (-0.6651904)_{\text{Take 2's complement}} + 0001.00000\text{B} \\ &= 1111.01011\text{B} + 0001.00000\text{B} \\ &= 0000.01011\text{B}\end{aligned}$$

### Principle and Calculation of Internal Clock Error Compensation (0.06ppm)

Achieving a compensation accuracy of 0.06ppm requires extending the maximum compensation period by a factor of 512. Then, when the minimum unit that can only be compensated is 30.5ppm, the average compensation unit per second becomes 30.5ppm/512=0.06ppm. It meets the clock compensation requirement with higher precision. And the compensation occurs in a relatively uniform range every 512 seconds. Hence, 9 decimal places is introduced in this register.

For example, when the 1Hz clock is directly output in the default state, the compensation target value is calculated by measuring the accuracy of the clock.

Assuming that the actual measured value is 0.9999888Hz, then:


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Actual Vibration Frequency = 32768 × 0.9999888 ≈ 32767.6329984 (actual oscillation frequency needs to be accurate to at least five decimal places)

$$\begin{aligned}\text{Compensation Target Value} &= (\text{Actual Vibration Frequency} - \text{Target Frequency})/\text{Target Frequency} \times 10^6 \\ &= (32767.6329984 - 32768)/32768 \times 10^6 \\ &= -11.29\text{ppm}\end{aligned}$$

Based on the formula:

$$CR[12:0] = \left(\frac{\text{Compensation Target Value [ppm]} \times 2^{15}}{10^6}\right)_{\text{Take 2's complement}} + 0001.000000000B$$

If the compensation target value is -11.29ppm, calculate the corresponding register value as follows:

$$\begin{aligned}CR[12:0] &= (-11.29 \times 2^{15} / 10^6)_{\text{Take 2's complement}} + 0001.000000000B \\ &= (-0.37)_{\text{Take 2's complement}} + 0001.000000000B \\ &= 1111.101000011B + 0001.000000000B \\ &= 0000.101000011B\end{aligned}$$

#### Principle and Calculation of Error Compensation Based on High-Speed Clock:

The calculation method of this method is the same as the error compensation based on its own clock. With the introduction of the high-speed clock HPCLK (XTH external high-speed clock), the 1/32768-second error that originally needed to be accumulated within a maximum of 32 seconds can be dispersed to every 1 second. By applying compensation every 1 second, a high-precision 1Hz clock output is achieved on average.

The correct high-speed compensation parameter value RTC_HCOMP.PARA must be set according to the frequency.

RTC_HCOMP.PARA = f/32768/32

f is the compensation frequency, and it is recommended to configure it as an integer multiple of 32.768 kHz.

- 0.06ppm compensation accuracy: the valid range for f is 16 to 48MHz.
- 0.96ppm compensation accuracy: the valid range for f is 4 to 48MHz.

### 14.3.4 Timestamp Function

Timestamp is enabled by setting RTC_CR2.TSE to 1.

When a timestamp event is detected on the RTC_TS pin, the calendar is saved to the timestamp registers (RTC_TSTR and RTC_TSDR).

When a timestamp event occurs, the RTC_IFR.TS (timestamp flag) is set to 1.

An interrupt can be generated when a timestamp detection event occurs by setting RTC_IER.TS to 1.

If a new timestamp event is detected while the timestamp flag (TS) is already set to 1, the timestamp overflow flag (TSOV) will be set to 1 and the timestamp registers (RTC_TSTR and RTC_TSDR) will hold the results of the previous event.

### 14.3.5 Wake-up Timer

Periodic wake-up flags are generated by a 16-bit programmable auto-reload down counter. This wake-up function can be enabled through the RTC_CR2.WUTE register.

The clock input to the wake-up timer can be:

- RTC clock (RTCCLK) divided by 2, 4, 8 or 16.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

- Second pulse divided by 1, 2, 4 or 8.

After initialization is completed, the timer starts counting down. When the wake-up function is enabled in low power mode, the down count remains active. Additionally, when the counter reaches zero, the RTC_IFR.WU register is set to 1, and the wake-up register automatically reloads using the value of the RTC_WUTR register.

The WU flag must then be cleared in software.

The pause wake-up timer will not generate a reload action. If retiming is required, a write action must be generated to the RTC_WUTR register.

When the periodic wake-up interrupt is enabled by setting the RTC_IER.WU register to 1, it takes the device out of low power mode.

System resets and low power mode (Sleep, DeepSleep) have no effect on the wake-up timer.

After the RTC counting has started, if the system needs to enter a low power mode, it is necessary to ensure that the register information has been synchronized to the RTC. Please perform any of the following confirmations before switching modes.

- After configuring the RTC registers, wait for at least two RTC count clocks before switching modes.
- After configuring the RTC registers, set RTC_CR1.WAIT=1 and check that RTC_CR1.WAITF=1. Then set RTC_CR1.WAIT=0, query RTC_CR1.WAITF=0 and then switch modes.

### 14.3.6 RTC Interrupt

RTC supports 4 interrupt types. Alarm A/B interrupt, periodic interrupt, timestamp interrupt, and wake-up timer interrupt.

#### Alarm A/B Interrupt

When RTC_IER.ALMx=1, if the current calendar time is equal to the alarm register, the alarm interrupt is triggered. By setting the MSK related bits, you can shield the comparison of the corresponding alarm time bits.

#### Periodic Interrupt

When RTC_IER.PRD=1, a period interrupt is triggered after the selected period occurs. Since the interrupt is shared, it is distinguished by flag register bits.

#### Wake-up Timer Interrupt

When RTC_IER.WU=1, an interrupt is triggered when the wake-up timer counter reaches zero. Since the interrupt is shared, it is distinguished by flag register bits.

#### Timestamp Interrupt

When RTC_IER.TS=1, an interrupt is generated upon a timestamp trigger. Since the interrupt line is shared, it is distinguished by flag register bits.

## 14.4 Operation Example

### 14.4.1 RTC Count Start Setting

#### Procedure

**Step1.** Set RTC_CR0.START=0 to stop counting.

**Step2.** Set RTC_CR0.AMPM and RTC_CR0.PRDS; RTC_CR0.PRDX configures the time format and interrupt period.

**Step3.** Set RTC_CR1.CKSEL to select the RTC timing clock.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

**Step4.** Set the calendar counter registers for seconds, minutes, hours, week, day, month, and year.

**Step5.** When clock error compensation is required, set the counting clock error compensation register RTC_COMPEN.

**Step6.** Clear the interrupt flag bits register RTC_IFR, and enable the interrupts.

**Step7.** Set RTC_CR0.START=1 to start counting.

### 14.4.2 Read Count Register

#### Context

There are three methods to read the count register.

#### Procedure

- **Read Method 1 at Any Time**

  **Step1.** Set RTC_CR1.WAIT=1 to stop the calendar register counting and enter read/write mode.

  **Step2.** Query until RTC_CR1.WAITF=1.

  **Step3.** Read the values of the second, minute, hour, week, day, month, and year count registers.

  **Step4.** Set RTC_CR1.WAIT=0 to start the counter counting.

  **Step5.** Query until RTC_CR1.WAITF=0.

- **Read Method 2 at Any Time**

  **Step1.** Read minutes, hours, weeks, days, months, and year count register values.

  **Step2.** Read out the second count register value.

  **Step3.** Read the second count register value again.

  **Step4.** Judging whether the reading values of the two seconds are the same, if they are different, start from the first step again, if the same, end read.

- **Interrupt Read Method**

  Read the second, minute, hour, week, day, month, year count register value in the RTC cycle interrupt service. Because after the interrupt occurs, it will take at least 0.5s for the next data change.

### 14.4.3 Write Count Register

#### Context

Changing the seconds register in the counting mode will reset the seconds counting, writing minutes, hours, weeks, days, months, years count register values will not affect the RTC counting.

> **Note**
>
> The RTC register is write protected. Please refer to the RTC_WPR register to remove the write protection before writing.
>
> WAIT is not required to write seconds, minutes, hours, weeks, days, months, and year count registers in RTC disabled mode.

#### Procedure

**Step1.** Set RTC_CR1.WAIT=1, stop the calendar register counting, and enter the read and write mode.

**Step2.** Query until RTC_CR1.WAITF=1.

**Step3.** Write the second, minute, hour, week, day, month, year count register value.

**Step4.** Set RTC_CR1.WAIT=0, the counter starts counting again.

> **Notice**
>
> All write operations must be completed within 1 second.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

**Step5.** Query until RTC_CR1.WAITF=0.

### 14.4.4 Alarm Clock Setting

#### Procedure

**Step1.** Set RTC_CR2.ALMxE=0 to disable the alarm.

**Step2.** Set RTC_IER.ALMx=1 to enable the alarm interrupt.

**Step3.** Alarm RTC_ALMx setting, configure the alarm time as needed;

**Step4.** Set RTC_CR2.ALMxE=1 to enable the alarm interrupt.

**Step5.** Wait for an interrupt to occur.

**Step6.** Since the alarm clock interrupt and the fixed-period interrupt share the interrupt request signal, when RTC_IFR.ALMx=1, enter the alarm clock interrupt processing; otherwise, enter the fixed-period interrupt processing.

### 14.4.5 1Hz Output

#### Context

RTC can choose to output three types of 1Hz clocks: general precision, higher precision and high precision. When the clock error compensation function is valid, it outputs a high-precision 1Hz clock; when using HPCLK with different frequencies, it outputs a high-precision 1Hz clock. The system control register must be configured according to the HPCLK frequency. The three methods for 1Hz output are as follows:

#### Procedure

- **General Precision 1Hz Output (without Clock Compensation)**

  **Step1.** Set RTC_CR0.START=0 to stop counting;

  **Step2.** RTC output pin setting.

  **Step3.** Set RTC_CR0.HZ1OE=1 to enable clock output.

  **Step4.** Set RTC_CR0.START=1, counting starts.

  **Step5.** Wait for more than 2 count cycles.

  **Step6.** 1Hz output starts.

- **Higher Precision 1Hz Output (Low-speed Compensation)**

  **Step1.** Set RTC_CR0.START=0 to stop counting;

  **Step2.** RTC output pin setting.

  **Step3.** Set RTC_CR0.HZ1OE=1 to enable clock output.

  **Step4.** Configure RTC_COMPEN.CR to set the compensation value.

  **Step5.** Set RTC_COMPEN.EN=1 to enable error compensation.

  **Step6.** Set RTC_CR0.START=1, counting starts.

  **Step7.** Wait for more than 2 count cycles.

  **Step8.** 1Hz output starts.

- **High Precision 1 Hz Output (Based on Higher Precision Output, Providing the RTC with a High-Speed Compensation Clock)**

  **Step1.** Set RTC_CR0.START=0 to stop counting;

  **Step2.** RTC output pin setting.

  **Step3.** Set RTC_CR0.HZ1OE=1 to enable clock output.

  **Step4.** Set RTC_CR0.HZ1SEL=1, choose to output high-precision 1Hz clock.

  **Step5.** Set the high-speed compensation parameter RTC_HCOMP.PARA according to the configured frequency.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

**Step6.** Configure RTC_COMPEN.CR[12:0] to set the compensation.

**Step7.** Set RTC_COMPEN.EN=1 to enable error compensation.

**Step8.** Set RTC_COMPEN.CSTEPS=0 to enable 0.96ppm accuracy compensation; or set RTC_COMPEN.CSTEPS=1 to enable 0.06ppm accuracy compensation.

**Step9.** Set RTC_CR0.START=1, counting starts.

**Step10.** Wait for more than 2 count cycles.

**Step11.** 1Hz output starts.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

## 14.5 Register Description

### 14.5.1 Register List

Base address: 0x40001400

Table 14-2 Real Time Counter (RTC) Register Offset

| Offset | Register | Description | Width |
|---|---|---|---|
| 0x000 | RTC_CR0 | Control Register 0 | 32 |
| 0x004 | RTC_CR1 | Control Register 1 | 32 |
| 0x008 | RTC_SEC | Second Count Register | 32 |
| 0x00C | RTC_MIN | Minute Count Register | 32 |
| 0x010 | RTC_HOUR | Hour Count Register | 32 |
| 0x014 | RTC_WEEK | Week Count Register | 32 |
| 0x018 | RTC_DAY | Day Count Register | 32 |
| 0x01C | RTC_MON | Month Count Register | 32 |
| 0x020 | RTC_YEAR | Year Count Register | 32 |
| 0x030 | RTC_COMPEN | Clock Error Compensation Register | 32 |
| 0x034 | RTC_HCOMP | High-Speed Compensation Accuracy Parameter Register | 32 |
| 0x03C | RTC_CR2 | Control Register 2 | 32 |
| 0x040 | RTC_ALMA | Alarm A Register | 32 |
| 0x044 | RTC_ALMB | Alarm B Register | 32 |
| 0x048 | RTC_TSTR | Timestamp Time Register | 32 |
| 0x04C | RTC_TSDR | Timestamp Date Register | 32 |
| 0x050 | RTC_WUTR | Wake-Up Timer Register | 32 |
| 0x054 | RTC_IER | Interruption Enable Register | 32 |
| 0x058 | RTC_IFR | Interrupt Flag Register | 32 |
| 0x05C | RTC_ICR | Interrupt Clear Register | 32 |
| 0x060 | RTC_WPR | Protection Register | 32 |

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 14.5.2 Control Register 0 (RTC_CR0)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x000 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 (Only power-on reset and module reset are effective for resetting this register) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | PRDSEL | PRDX | | | | | | START | HZ1SEL | HZ1OE | Reserved | AMPM | PRDS | |
| Access | | | | | | | | | | | | | | | | | | RW | RW | | | | | | RW | RW | RW | | RW | RW | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:15* | *Reserved* | - | - |
| 14 | PRDSEL | Periodic interrupt source selection<br>● 0b0: Use the periodic interrupt time interval set by PRDS<br>● 0b1: Use the periodic interrupt time interval set by PRDX | RW |
| 13:8 | PRDX | Set the time interval for generating periodic interrupts<br>The settable range is from 0.5 seconds to 32 seconds, and the step is 0.5 seconds.<br>● 0b000000: 0.5 seconds<br>● 0b000001: 1 second<br>● .....<br>● 0b111110: 31.5 seconds<br>● 0b111111: 32 seconds | RW |
| 7 | START | RTC count enable<br>● 0b0: Stop the RTC count<br>● 0b1: Enable the RTC count | RW |
| 6 | HZ1SEL | 1Hz output selection<br>● 0b0: Normal precision 1Hz output<br>● 0b1: High precision 1Hz output | RW |
| 5 | HZ1OE | 1Hz output enable<br>● 0b0: Disabled 1Hz output<br>● 0b1: Enable 1Hz output | RW |
| *4* | *Reserved* | - | - |
| 3 | AMPM | Time format selection<br>● 0b0: 12-hour time format<br>● 0b1: 24-hour time format | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 2:0 | PRDS | Set the time interval for generating interrupts<br>● 0b000: No periodic interrupt generated<br>● 0b001: 0.5 seconds<br>● 0b010: 1 second<br>● 0b011: 1 minute (At 00 seconds every minute)<br>● 0b100: 1 hour (At 00:00 every hour)<br>● 0b101: 1 day (At 00:00:00 every day)<br>● 0b110/0b111: 1 month (At 00:00:00 on the first day of every month)<br>**Notice**<br>If you need to write and change the time interval of periodic interrupt when START=1, the operation steps are as follows:<br>1. Turn off the RTC interrupt in the NVIC;<br>2. Change the time interval of periodic interrupt;<br>3. Clear the RTC interrupt flag;<br>4. Enable RTC interrupt. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 14.5.3 Control Register 1 (RTC_CR1)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x004 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 (Only power-on reset and module reset are effective for this register reset) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | CKSEL | | | Reserved | | | | WAITF | WAIT |
| Access | | | | | | | | | | | | | | | | | | | | | | RW | | | | | | | RO | RW |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:11* | *Reserved* | - | - |
| 10:8 | CKSEL | RTC clock selection<br>● 0b000/0b001: XTL 32.768k<br>● 0b010/0b011: RCL 32k<br>● 0b100: XTH/128 (select this option when the crystal oscillator is 4M)<br>● 0b101: XTH/256 (select this option when the crystal oscillator is 8M)<br>● 0b110: XTH/512 (select this option when the crystal oscillator is 16M)<br>● 0b111: XTH/1024 (Select this option when the crystal oscillator is 32M.) | RW |
| *7:2* | *Reserved* | - | - |
| 1 | WAITF | Read and write wait valid flag<br>● 0b0: Not write/read status<br>● 0b1: Write/read status<br>**Notice**<br>Flag indicating whether the WAIT bit setting is effective. Please confirm that this bit is set to "1" before writing or reading. During the counting process, the bit is cleared to"0" after the WAIT bit is cleared to"0" and waits for the writing to be completed. | RO |
| 0 | WAIT | Read and write wait setting<br>● 0b0: Normal Counting Mode<br>● 0b1: Write/read mode<br>**Notice**<br>Please set this bit to "1" when writing/reading. Since the counter is counting continuously, please complete the writing/reading operation within 1 second and clear this bit to "0". | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 14.5.4 Second Count Register (RTC_SEC)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x008 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0xXXXXXXXX | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | SECH | | | SECL | | | |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | RW | | | RW | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:7* | *Reserved* | - | - |
| 6:4 | SECH | Second count tens digit<br>BCD code of second tens digit count value | RW |
| 3:0 | SECL | Second count units digit<br>BCD code of second units digit count value | RW |

Indicates 0-59 seconds, using decimal counting. Please write the BCD code in decimal 0 to 59. Values written outside this range will be ignored.

### 14.5.5 Minute Count Register (RTC_MIN)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x00C | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0xXXXXXXXX | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | MINH | | | MINL | | | |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | RW | | | RW | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:7* | *Reserved* | - | - |
| 6:4 | MINH | Minute count tens digit<br>BCD code of minute tens digit count value | RW |
| 3:0 | MINL | Minute count units digit<br>BCD code of minute units digit count value | RW |

Indicates 0-59 points, using decimal counting. Please write the BCD code in decimal 0 to 59. Values written outside this range will be ignored.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 14.5.6 Hour Count Register (RTC_HOUR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x010 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0xXXXXXXXX | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | HOURH | | HOURL | | | |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | | RW | | RW | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:6* | *Reserved* | - | - |
| 5:4 | HOURH | Hour count tens digit<br>BCD code of hour tens digit count value | RW |
| 3:0 | HOURL | Hour count units digit<br>BCD code of hour units digit count value | RW |

In 24-hour format, it means 0-23 hours.

In 12-hour format, it means 0-12 hours. Bit5=0 means AM, then 01-12 means morning; Bit5=1 means PM, then 21-32 means afternoon.

Please set the correct decimal BCD code according to the value of the control bit RTC_CR0.AMPM. The value written out of range will be ignored.

For specific time values, please refer to the following table:

| 24-Hour Clock Format (AMPM=1) | | 12-Hour Clock Format (AMPM=0) | |
|---|---|---|---|
| Time | Register Value | Time | Register Value |
| 00 o'clock | 0x00 | AM 12 o'clock | 0x12 |
| 01 o'clock | 0x01 | AM 01 o'clock | 0x01 |
| 02 o'clock | 0x02 | AM 02 o'clock | 0x02 |
| 03 o'clock | 0x03 | AM 03 o'clock | 0x03 |
| 04 o'clock | 0x04 | AM 04 o'clock | 0x04 |
| 05 o'clock | 0x05 | AM 05 o'clock | 0x05 |
| 06 o'clock | 0x06 | AM 06 o'clock | 0x06 |
| 07 o'clock | 0x07 | AM 07 o'clock | 0x07 |
| 08 o'clock | 0x08 | AM 08 o'clock | 0x08 |
| 09 o'clock | 0x09 | AM 09 o'clock | 0x09 |
| 10 o'clock | 0x10 | AM 10 o'clock | 0x10 |
| 11 o'clock | 0x11 | AM 11 o'clock | 0x11 |
| 12 o'clock | 0x12 | PM 12 o'clock | 0x32 |

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| 24-Hour Clock Format (AMPM=1) | | 12-Hour Clock Format (AMPM=0) | |
|---|---|---|---|
| Time | Register Value | Time | Register Value |
| 13 o'clock | 0x13 | PM 01 o'clock | 0x21 |
| 14 o'clock | 0x14 | PM 02 o'clock | 0x22 |
| 15 o'clock | 0x15 | PM 03 o'clock | 0x23 |
| 16 o'clock | 0x16 | PM 04 o'clock | 0x24 |
| 17 o'clock | 0x17 | PM 05 o'clock | 0x25 |
| 18 o'clock | 0x18 | PM 06 o'clock | 0x26 |
| 19 o'clock | 0x19 | PM 07 o'clock | 0x27 |
| 20 o'clock | 0x20 | PM 08 o'clock | 0x28 |
| 21 o'clock | 0x21 | PM 09 o'clock | 0x29 |
| 22 o'clock | 0x22 | PM 10 o'clock | 0x30 |
| 23 o'clock | 0x23 | PM 11 o'clock | 0x31 |

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 14.5.7 Day Count Register (RTC_DAY)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x018 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0xXXXXXXXX | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | DAYH | | DAYL | | | |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | | RW | | RW | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:6* | *Reserved* | - | - |
| 5:4 | DAYH | Day count tens digit value<br>BCD code of day tens digit count value | RW |
| 3:0 | DAYL | Day count units digit<br>BCD code of day units digit count value | RW |

Means 1 to 31 days, uses decimal counting, automatically calculates leap year and month. Please write the BCD code in decimal 1 to 31. Values written outside the range will be ignored.

Refer to the table below for the specific range of day values.

| Month | Day Count Value |
|---|---|
| February (normal year) | 01-28 |
| February (leap year) | 01-29 |
| April, June, September, November | 01-30 |
| January, March, May, July, August, October , December | 01-31 |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 14.5.8 Week Count Register (RTC_WEEK)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x014 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0xXXXXXXXX | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | WEEK | | |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | RW | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:3* | *Reserved* | - | - |
| 2:0 | WEEK | Week count value<br>BCD code of week count value | RW |

Decimal 0-6 means Sunday-Saturday. Please write the BCD code in decimal 0 to 6. Values written outside the range will be ignored.

For specific week count values, please refer to the following table:

| Week | Week Count Value |
|---|---|
| Sunday | 0x00 |
| Monday | 0x01 |
| Tuesday | 0x02 |
| Wednesday | 0x03 |
| Thursday | 0x04 |
| Friday | 0x05 |
| Saturday | 0x06 |

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 14.5.9 Month Count Register (RTC_MON)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x01C | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0xXXXXXXXX | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | MON | | | | |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | | | RW | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:5* | *Reserved* | - | - |
| 4:0 | MON | Month count value<br>BCD code of month count value | RW |

Indicates 1-12 months, using decimal counting. Please write the BCD code in decimal 1 to 12. Values written outside the range will be ignored.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 14.5.10 Year Count Register (RTC_YEAR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x020 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0xXXXXXXXX | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | YEARH | | | | YEARL | | | |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | RW | | | | RW | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:8* | *Reserved* | - | - |
| 7:4 | YEARH | Year conut tens digit<br>BCD code of year tens digit count value | RW |
| 3:0 | YEARL | Year conut units digit<br>BCD code of year units digit count value | RW |

Indicates 0 to 99 years, using decimal counting, counting according to the month carry. Automatically calculate leap years such as: 00, 04, 08, ..., 92, 96, etc. Please write the BCD code in decimal 0-99. Values written outside the range will be ignored.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 14.5.11 Clock Error Compensation Register (RTC_COMPEN)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x030 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000020 (Only power-on reset and module reset are effective for this register reset) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | EN | CSTEPS | Reserved | CR | | | | | | | | | | | |
| Access | | | | | | | | | | | | | | | | | RW | RW | | RW | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:16* | *Reserved* | - | - |
| 15 | EN | Compensation enable<br>● 0b0: Disable clock error compensation<br>● 0b1: Enable clock error compensation | RW |
| 14 | CSTEPS | Compensation accuracy<br>● 0b0: Compensation accuracy 0.96 ppm<br>● 0b1: Compensation accuracy 0.06 ppm | RW |
| *13* | *Reserved* | - | - |
| 12:0 | CR | Compensation value<br>**CSTEPS=0b0:**<br>By setting the compensation value, an accuracy compensation of ±0.96ppm can be performed per second. The compensation value is 9-bit 2's complement code with a decimal point, and the last 5 bits are the decimal part. Compensation range: 274.6ppm to 212.6ppm, with a minimum differential error of ±0.48ppm and a minimum resolution of 0.96ppm.<br>**CSTEPS=0b1:**<br>By setting the compensation value, an accuracy compensation of ±0.05960ppm can be performed per second. The compensation value is 13-bit 2's complement code with a decimal point, and the last 9 bits are the decimal part. Compensation range: 274.65820ppm to 213.56344ppm, with a minimum differential error of ±0.0298ppm and a minimum resolution of 0.05960ppm. | RW |

For 0.96ppm more compensation accuracy, see the following table:


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Compensation Value Setting | | | | | | | | | | Compensation Value |
|---|---|---|---|---|---|---|---|---|---|---|
| EN | CR[8:0] | | | | | | | | | |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | -274.6ppm |
| | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | -273.7ppm |
| | ... | | | | | | | | | ... |
| | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | -0.96ppm |
| | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0ppm |
| | ... | | | | | | | | | ... |
| | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | +211.7ppm |
| | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | +212.6ppm |
| 0 | X | X | X | X | X | X | X | X | X | Uncompensated |

For 0.06ppm more compensation accuracy, see the following table:

| Compensation Value Setting | | | | | | | | | | | | | | Compensation Value |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| EN&CSTEPS | CR[12:0] | | | | | | | | | | | | | |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | -274.65820ppm |
| | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | -274.59859ppm |
| | ... | | | | | | | | | | | | | ... |
| | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | -0.059605ppm |
| | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0ppm |
| | ... | | | | | | | | | | | | | ... |
| | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | +213.50383ppm |
| | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | +213.56344ppm |
| 0 | X | X | X | X | X | X | X | X | X | X | X | X | X | Uncompensated |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 14.5.12 High-Speed Compensation Accuracy Parameter Register (RTC_HCOMP)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x034 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 (Only power-on reset and module reset are effective for this register reset) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | PARA | | | |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | | RW | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:6* | *Reserved* | - | - |
| 5:0 | PARA | Compensation parameters<br>Set the corresponding high-speed compensation parameter according to the frequency.<br>RTC_HCOMP.PARA = f/32768/32<br>f is the compensation frequency, and it is recommended to configure it as an integer multiple of 32.768 kHz.<br>● 0.06ppm compensation accuracy: the valid range for f is 16 to 48MHz.<br>● 0.96ppm compensation accuracy: the valid range for f is 4 to 48MHz. | RW |

Please refer to the following table for the specific high-speed compensation parameters:

| Compensation Frequency f (MHz) | Compensation Parameter PARA |
|---|---|
| 4 | 0b000100 |
| 6 | 0b000110 |
| 8 | 0b001000 |
| 12 | 0b001100 |
| 16 | 0b001111 |
| 20 | 0b010011 |
| 24 | 0b010111 |
| 32 | 0b011111 |
| 48 | 0b101110 |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 14.5.13 Control Register 2 (RTC_CR2)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x03C | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 (Only power-on reset and module reset are effective for this register reset) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | ALMBE | ALMAE | Reserved | WUTE | TSE | Reserved | | TSEDG | WUCKSEL | | |
| Access | | | | | | | | | | | | | | | | | | | | | | RW | RW | | RW | RW | | | RW | RW | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:11* | *Reserved* | - | - |
| 10 | ALMBE | Alarm B enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 9 | ALMAE | Alarm A enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| *8* | *Reserved* | - | - |
| 7 | WUTE | Wake-up timer enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 6 | TSE | Timestamp enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| *5:4* | *Reserved* | - | - |
| 3 | TSEDG | Timestamp edge selection<br>● 0b0: Rising edge<br>● 0b1: Falling Edge | RW |
| 2:0 | WUCKSEL | Wake-up timer clock source selection<br>● 0b000: RTC/2<br>● 0b001: RTC/4<br>● 0b010: RTC/8<br>● 0b011: RTC/16<br>● 0b100: RTC_Sec/1<br>● 0b101: RTC_Sec/2<br>● 0b110: RTC_Sec/4<br>● 0b111: RTC_Sec/8 | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 14.5.14 Alarm A Register (RTC_ALMA)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x040 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0xXXXXXXXX | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | WEEK | | | | | | | MSKH | Reserved | HOURH | | HOURL | | | | MSKM | MINH | | | MINL | | | | MSKS | SECH | | | SECL | | | |
| Access | | RW | | | | | | | RW | | RW | | RW | | | | RW | RW | | | RW | | | | RW | RW | | | RW | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31* | *Reserved* | - | - |
| 30:24 | WEEK | Alarm weekday bit<br>Bit24-Bit30 correspond to Sunday-Saturday respectively:<br>When the corresponding bit is set to 1, the alarm is active on that day of the week.<br>For example:<br>● Setting it to 0b0000001 indicates that the Sunday alarm is enabled<br>● Setting it to 0b0000011 means the alarm is enabled for Sunday and Monday<br>● Setting it to 0b1000000 indicates that the Saturday alarm is enabled | RW |
| 23 | MSKH | Alarm hour bit mask bit<br>● 0b0: Alarm A will be generated only when the hour bit match<br>● 0b1: The hour bit is invalid in alarm A | RW |
| *22* | *Reserved* | - | - |
| 21:20 | HOURH | Alarm hour tens digit<br>BCD code of the alarm hour tens digit | RW |
| 19:16 | HOURL | Alarm hour units digit<br>BCD code of the alarm hour units digit | RW |
| 15 | MSKM | Alarm minute bit mask bit<br>● 0b0: Alarm A will be generated only when the minute bit match<br>● 0b1: The minute bit is invalid in alarm A | RW |
| 14:12 | MINH | Alarm minute tens digit<br>BCD code of the alarm minute tens digit | RW |
| 11:8 | MINL | Alarm minute units digit<br>BCD code of the alarm minute units digit | RW |
| 7 | MSKS | Alarm second bit mask bit<br>● 0b0: Alarm A will be generated only when the second bit match<br>● 0b1: The second bit is invalid in alarm A | RW |
| 6:4 | SECH | Alarm seconds tens digit<br>BCD code of the alarm second tens digit | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 3:0 | SECL | Alarm seconds units digit<br>BCD code of the alarm second units digit | RW |

**Note**

When performing bit-field operations, note that when configuring the 12-hour format, the HOURL register must not be written with 0.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 14.5.15 Alarm B Register (RTC_ALMB)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x044 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0xXXXXXXXX | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | WEEK | | | | | | | MSKH | Reserved | HOURH | | | HOURL | | | MSKM | MINH | | | MINL | | | MSKS | SECH | | | SECL | | | |
| Access | | RW | | | | | | | RW | | RW | | | RW | | | RW | RW | | | RW | | | RW | RW | | | RW | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31* | *Reserved* | - | - |
| 30:24 | WEEK | Alarm weekday bit<br>Bit24-Bit30 correspond to Sunday-Saturday, respectively:<br>When the corresponding bit is set to 1, the alarm is active on that day of the week.<br>For example:<br>● Setting it to 0b0000001 indicates that the Sunday alarm is enabled<br>● Setting it to 0b0000011 means the alarm is enabled for Sunday and Monday<br>● Setting it to 0b1000000 indicates that the Saturday alarm is enabled | RW |
| 23 | MSKH | Alarm hour mask bit<br>● 0b0: Alarm B will be generated only when the hour bit matches<br>● 0b1: The hour bit is invalid in alarm B | RW |
| *22* | *Reserved* | - | - |
| 21:20 | HOURH | Alarm hour tens digit<br>BCD code of the alarm hour tens digit | RW |
| 19:16 | HOURL | Alarm hour units digit<br>BCD code of the alarm hour units digit | RW |
| 15 | MSKM | Alarm minute bit shielding bit<br>● 0b0: Alarm B will be generated only when the minute bit match<br>● 0b1: The minute bit is invalid in alarm B | RW |
| 14:12 | MINH | Alarm minute tens digit<br>BCD code of the alarm minute tens digit | RW |
| 11:8 | MINL | Alarm minute units digit<br>BCD code of the alarm minute units digit | RW |
| 7 | MSKS | Alarm seconds digit mask bit<br>● 0b0: Alarm B will be generated only when the second bit match<br>● 0b1: The second bit is invalid in alarm B | RW |
| 6:4 | SECH | Alarm seconds tens digit<br>BCD code of the alarm second tens digit | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 3:0 | SECL | Alarm seconds units digit<br>BCD code of the alarm second units digit | RW |

**Note**

When performing bit-field operations, note that when configuring the 12-hour format, the HOURL register must not be written with a value of 0.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 14.5.16 Timestamp Time Register (RTC_TSTR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x048 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 (Only power-on reset and module reset are effective for this register reset) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | HOURH | | HOURL | | Reserved | | MINH | | MINL | | Reserved | | SECH | | | | SECL | | | |
| Access | | | | | | | | | | | | | RO | | RO | | | | RO | | RO | | | | RO | | | | RO | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:22* | *Reserved* | - | - |
| 21:20 | HOURH | Hour count tens digit<br>BCD code of hour tens digit count value<br>**Note**<br>This bit uses 24-hour format and is not affected by RTC_CR0.AMPM. | RO |
| 19:16 | HOURL | Hour count units digit<br>BCD code of hour units digit count value<br>**Note**<br>This bit uses 24-hour format and is not affected by RTC_CR0.AMPM. | RO |
| *15* | *Reserved* | - | - |
| 14:12 | MINH | Minute count tens digit<br>BCD code of minute tens digit count value | RO |
| 11:8 | MINL | Minute count units digit<br>BCD code of minute units digit count value | RO |
| *7* | *Reserved* | - | - |
| 6:4 | SECH | Second count tens digit<br>BCD code of second tens digit count value | RO |
| 3:0 | SECL | Second count units digit<br>BCD code of second units digit count value | RO |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 14.5.17 Timestamp Date Register (RTC_TSDR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x04C | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 (Only power-on reset and module reset are effective for this register) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | WEEK | | | MON | | | | Reserved | | DAYH | | DAYL | | | |
| Access | | | | | | | | | | | | | | | | | RO | | | RO | | | | | | RO | | RO | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:16* | *Reserved* | - | - |
| 15:13 | WEEK | Week count value<br>Week count value, the format is the same as RTC_WEEK | RO |
| 12:8 | MON | Month count value<br>Month count value, the format is the same as RTC_MON | RO |
| *7:6* | *Reserved* | - | - |
| 5:4 | DAYH | Day count tens digit<br>BCD code of day tens digit count value | RO |
| 3:0 | DAYL | Day count units digit<br>BCD code of day units digit count value | RO |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 14.5.18 Wake-Up Timer Register (RTC_WUTR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x050 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x0000FFFF (Only power-on reset and module reset are effective for this register reset) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | ARR | | | | | | | | | | | | | |
| Access | | | | | | | | | | | | | | | | | RW | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:16* | *Reserved* | - | - |
| 15:0 | ARR | Wake-up timer period value | RW |

### 14.5.19 Interrupt Enable Register (RTC_IER)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x054 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 (Only power-on reset and module reset are effective for this register reset) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | PRD | Reserved | | TS | WU | ALMB | ALMA |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | RW | | | RW | RW | RW | RW |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:7* | *Reserved* | - | - |
| 6 | PRD | Periodic interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| *5:4* | *Reserved* | - | - |
| 3 | TS | Timestamp interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 2 | WU | Timing interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 1 | ALMB | Alarm B interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 0 | ALMA | Alarm A interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 14.5.20 Interrupt Flag Register (RTC_IFR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x058 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 (Only power-on reset and module reset are effective for this register reset) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | PRD | Reserved | TSOV | TS | WU | ALMB | ALMA |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | RO | | RO | RO | RO | RO | RO |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:7* | *Reserved* | - | - |
| 6 | PRD | Periodic interrupt flag<br>● 0b0: No interrupt occurred<br>● 0b1: Interrupt is occurred | RO |
| *5* | *Reserved* | - | - |
| 4 | TSOV | Timestamp overflow flag<br>● 0b0: No overflow occurred<br>● 0b1: Overflow is occurred | RO |
| 3 | TS | Timestamp interrupt flag<br>● 0b0: No interrupt occurred<br>● 0b1: Interrupt is occurred | RO |
| 2 | WU | Timer interrupt flag<br>● 0b0: No interrupt occurred<br>● 0b1: Interrupt is occurred | RO |
| 1 | ALMB | Alarm B interrupt flag<br>● 0b0: No interrupt occurred<br>● 0b1: Interrupt is occurred | RO |
| 0 | ALMA | Alarm A interrupt flag<br>● 0b0: No interrupt occurred<br>● 0b1: Interrupt is occurred | RO |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 14.5.21 Interrupt Clear Register (RTC_ICR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x05C | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x000003FF | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | PRD | Reserved | TSOV | TS | WU | ALMB | ALMA |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | WO | | WO | WO | WO | WO | WO |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:7* | *Reserved* | - | - |
| 6 | PRD | Periodic interrupt flag clear<br>Write 0 to clear flag, write 1 has no effect; read always returns 1. | WO |
| *5* | *Reserved* | - | - |
| 4 | TSOV | Timestamp overflow flag clear<br>Write 0 to clear flag, write 1 has no effect; read always returns 1. | WO |
| 3 | TS | Timestamp interrupt flag clear<br>Write 0 to clear flag, write 1 has no effect; read always returns 1. | WO |
| 2 | WU | Timing interrupt flag clear<br>Write 0 to clear flag, write 1 has no effect; read always returns 1. | WO |
| 1 | ALMB | Alarm B interrupt flag clear<br>Write 0 to clear flag, write 1 has no effect; read always returns 1. | WO |
| 0 | ALMA | Alarm A interrupt flag clear<br>Write 0 to clear flag, write 1 has no effect; read always returns 1. | WO |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 14.5.22 Protection Register (RTC_WPR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x060 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | WPR | | | | | | | |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | WO | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:8* | *Reserved* | - | - |
| 7:0 | WPR | Protection sequence<br>To disable RTC register write protection, write 0xCA followed by 0x53 to WPR; to enable RTC register write protection, write 0xCA followed by 0xXX (not equal to 0x53) to WPR.<br>Except for WPR and ICR, all RTC registers are write-protected by default, and cannot be written upon power-on. | WO |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

# 15 IWDT

## 15.1 Introduction

This product features an embedded independent watchdog peripheral for high security and flexibility of use. This independent watchdog detects and resolves failures caused by software errors, triggering an interrupt or reset when its count reaches a programmed timeout value. The watchdog is driven by the IWDTCLK (clock source is RC10K), and it can still work normally even if the main clock fails.

## 15.2 Features

- Independent RC oscillator
- Trigger interrupt or reset on overflow
- A running watchdog can be stopped by a specific write sequence
- During DeepSleep mode, it can be configured to automatically pause counting

## 15.3 Functional Description

### 15.3.1 Function Block Diagram

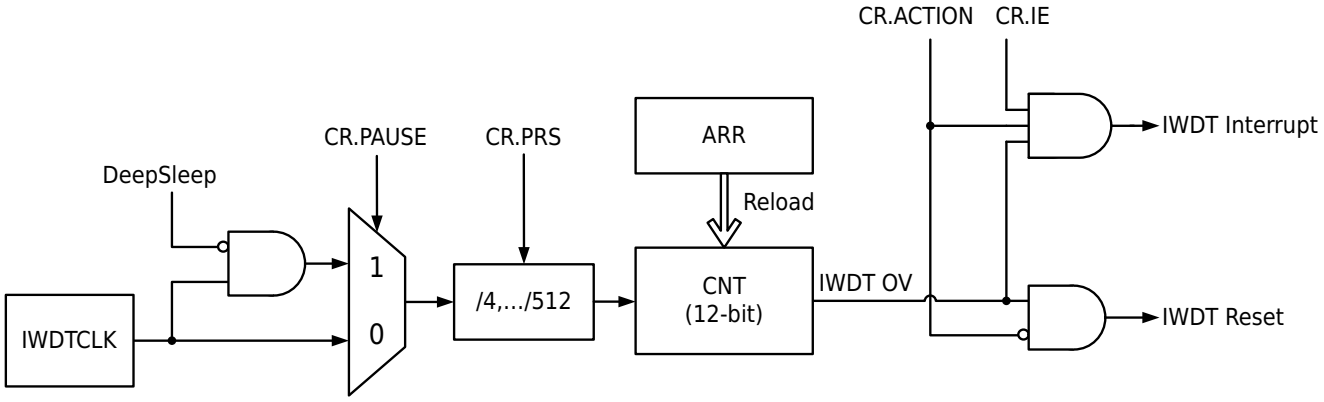



Figure 15-1 IWDT Function Block Diagram

After starting the watchdog, the counter starts counting down from 0xFFF. When 0xAAAA is written to the IWDT_KR register, the value of IWDT_ARR is reloaded to the counter. When the count value is decremented to 0x000, the watchdog will generate an overflow signal; when the counter is reloaded and the count value is greater than the value of the IWDT_WINR register, an overflow signal will also be generated. Configuring the corresponding control bit of IWDT_CR enables the watchdog to generate reset or interrupt after overflow. Configuring the corresponding control bit of IWDT_CR, the watchdog can automatically suspend counting during DeepSleep.

### 15.3.2 Window Watchdog Mode

When the value of the IWDT_WINR register is configured to be less than IWDT_ARR, IWDT works in the window watchdog mode. In this mode, there are two conditions to trigger the watchdog overflow: the count value is decremented to 0x000, and the watchdog is reloaded when the count value is greater than the window value. When changing the IWDT_WINR register value, the hardware will automatically reload the watchdog and reset the down counter value to IWDT_ARR. The user program should reload


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

the watchdog before the count value is less than or equal to the value of the IWDT_WINR register and the count value is decremented to 0x000.

### 15.3.3 Independent Watchdog Mode

When the value of the IWDT_WINR register is configured to be greater than or equal to IWDT_ARR, IWDT works in the independent watchdog mode. In this mode, there is only one condition that triggers the watchdog overflow: the count value is decremented to 0x000. The user program should reload the watchdog before the count value is decremented to 0x000.

### 15.3.4 Watchdog Overflow Handling

The overflow signal of the watchdog can trigger an interrupt or a reset. When IWDT_CR.ACTION is set to 0, the watchdog overflow signal will reset the MCU; when IWDT_CR.ACTION is set to 1 and IWDT_CR.IE is 1, the watchdog overflow signal will trigger an interrupt, by writing 0 to the IWDT_SR register, the IWDT_SR.OV flag can be cleared to 0.

### 15.3.5 Watchdog Pause

If IWDT_CR.PAUSE is set to 1, the watchdog automatically pauses when the MCU enters DeepSleep; when the MCU exits DeepSleep, the watchdog continues to run. Using this feature, the MCU does not need to wake up frequently to feed the dog when it is in a low-power consumption state, and can achieve lower standby power consumption.

### 15.3.6 Watchdog Stop and Run Again

Write 0x1234, 0x5678 to the IWDT_KR register in sequence can stop the running watchdog.

Write 0xCCCC to the IWDT_KR register will enable the watchdog to run again with its configuration unchanged.

### 15.3.7 Register Write Protection

The IWDT_CR, IWDT_ARR, and IWDT_WINR registers are write-protected. If you need to modify these three registers, you should write 0x5555 to the IWDT_KR register to release the write protection of these registers after the watchdog starting; writing other values to the IWDT_KR register will enable the write protection of these registers.

## 15.4 Operation Example

### 15.4.1 Configure IWDT as Window Watchdog

**Step1.** Write 0xCCCC to the IWDT_KR register to enable the IWDT.

**Step2.** Write 0x5555 to the IWDT_KR register to release the write protection of the register.

**Step3.** Configure IWDT_CR, select the counting clock frequency of the watchdog, the action after overflow, and the DeepSleep feature.

**Step4.** Configure IWDT_ARR to select the overflow period of the watchdog.

**Step5.** Wait for IWDT_SR.ARRF to be 0.

**Step6.** Write the required value to IWDT_WINR to configure the window size.

**Step7.** Wait for IWDT_SR.WINRF to become 0, and automatically refresh the counter value to IWDT_ARR.

**Step8.** Before the count value is less than or equal to the IWDT_WINR register value and the count value is decremented to 0, write 0xAAAA to the IWDT_KR register to reload IWDT.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 15.4.2 Configure IWDT as Independent Watchdog

**Step1.** Write 0xCCCC to the IWDT_KR register to enable the IWDT.

**Step2.** Write 0x5555 to the IWDT_KR register to release the write protection of the register.

**Step3.** Configure IWDT_CR, select the counting clock frequency of the watchdog, the action after overflow, and the DeepSleep feature.

**Step4.** Configure IWDT_ARR to select the overflow period of the watchdog.

**Step5.** Wait for IWDT_SR.ARRF to be 0

**Step6.** Refresh the counter value to the value of IWDT_ARR(IWDT_KR=0x0000AAAA).

**Step7.** Before the count value is decremented to 0, write 0xAAAA to the IWDT_KR register to reload the IWDT.

### 15.4.3 Watchdog Timeout Time (10kHz Independent Clock)

**Table 15-1 IWDT Timeout Time (10kHz Independent Clock)**

| Prescaler Factor | PRS[2:0] Bits | Minimum Time (ms) ARR[11:0] = 0x000 | Maximum Time (ms) ARR[11:0] = 0xFFF |
|---|---|---|---|
| 4 frequency division | 0b000 | 0.1 * 4 = 0.4 | 0.1 * 4 * 4096 = 1638.4 |
| 8 frequency division | 0b001 | 0.1 * 8 = 0.8 | 0.1 * 8 * 4096 = 3276.8 |
| 16 frequency division | 0b010 | 0.1 * 16 = 1.6 | 0.1 * 16 * 4096 = 6553.6 |
| 32 frequency division | 0b011 | 0.1 * 32 = 3.2 | 0.1 * 32 * 4096 = 13107.2 |
| 64 frequency division | 0b100 | 0.1 * 64 = 6.4 | 0.1 * 64 * 4096 = 26214.4 |
| 128 frequency division | 0b101 | 0.1 * 128 = 12.8 | 0.1 * 128 * 4096 = 52428.8 |
| 256 frequency division | 0b110 | 0.1 * 256 = 25.6 | 0.1 * 256 * 4096 = 104857.6 |
| 512 frequency division | 0b111 | 0.1 * 512 = 51.2 | 0.1 * 512 * 4096 = 209715.2 |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

## 15.5 Register Description

### 15.5.1 Register List

Base address: 0x40001000

Table 15-2 Independent Watchdog (IWDT) Register Offset

| Offset | Register | Description | Width |
|---|---|---|---|
| 0x80 | IWDT_KR | Key Value Register | 32 |
| 0x84 | IWDT_CR | Control Register | 32 |
| 0x88 | IWDT_ARR | Reload Register | 32 |
| 0x8C | IWDT_SR | Status Register | 32 |
| 0x90 | IWDT_WINR | Window Register | 32 |

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 15.5.2 Key Value Register (IWDT_KR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x80 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | KR | | | | | | | | | | | | | | | |
| Access | | | | | | | | | | | | | | | | | WO | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:16* | *Reserved* | - | - |
| 15:0 | KR | IWDT key value<br>● Write 0xCCCC: Start IWDT counter<br>● Write 0xAAAA: Reload IWDT count value<br>● Write 0x1234, 0x5678 in sequence: Stop watchdog<br>● Write 0x5555: Release the write protection of IWDT_ARR, IWDT_WINR, IWDT_CR<br>● Write non-0x5555: Enable the write protection of IWDT_ARR, IWDT_WINR, IWDT_CR | WO |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 15.5.3 Control Register (IWDT_CR)

**Note**

- This register can only be modified after writing 0x5555 to the IWDT_KR register.
- This register can only be modified when IWDT_SR.CRF is 0.

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x84 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | PAUSE | ACTION | IE | Reserved | PRS | | |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | RW | RW | RW | | RW | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:7* | *Reserved* | - | - |
| 6 | PAUSE | WDT pause control in DeepSleep mode<br>● 0b0: IWDT continues to run in DeepSleep mode<br>● 0b1: IWDT automatically pauses in DeepSleep mode | RW |
| 5 | ACTION | Action configuration after IWDT overflow<br>● 0b0: Reset is generated after watchdog overflow<br>● 0b1: Interrupt is generated after watchdog overflow | RW |
| 4 | IE | IWDT interrupt enable control<br>● 0b0: Disable watchdog interrupt<br>● 0b1: Enable watchdog interrupt | RW |
| *3* | *Reserved* | - | - |
| 2:0 | PRS | Configure the watchdog count clock and the frequency division ratio of RC10K oscillator<br>● 0b000: 4-frequency division<br>● 0b001: 8-frequency division<br>● 0b010: 16-frequency division<br>● 0b011: 32-frequency division<br>● 0b100: 64-frequency division<br>● 0b101: 128-frequency division<br>● 0b110: 256-frequency division<br>● 0b111: 512-frequency division | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 15.5.4 Reload Register (IWDT_ARR)

**Note**

- This register can only be modified after writing 0x5555 to the IWDT_KR register.
- This register can only be modified when IWDT_SR.ARRF is 0.

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x88 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000FFF | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | ARR | | | | | | | | | | | |
| Access | | | | | | | | | | | | | | | | | | | | | RW | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:12* | *Reserved* | - | - |
| 11:0 | ARR | IWDT reload value | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 15.5.5 Status Register (IWDT_SR)

**Note**

Write 0 to this register can clear the IWDT_SR.OV flag bit.

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x8C | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | RUN | OV | WINRF | ARRF | CRF |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | | | RO | RW | RO | RO | RO |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:5* | *Reserved* | - | - |
| 4 | RUN | WDT running flag<br>● 0b0: WDT is not running<br>● 0b1: WDT is running | RO |
| 3 | OV | WDT overflow flag<br>Write 0 to clear the overflow flag, write 1 has no effect.<br>Read:<br>● 0b0: WDT overflow has not occurred<br>● 0b1: WDT overflow has occurred | RW |
| 2 | WINRF | WINR register update flag<br>● 0b0: WINR register update completed<br>● 0b1: WINR register is being updated<br>**Notice**<br>The WINR register can only be updated when the WINRF is 0. | RO |
| 1 | ARRF | ARR register update flag<br>● 0b0: ARR register update completed<br>● 0b1: ARR register is being updated<br>**Notice**<br>The ARR register can only be updated when the ARRF is 0. | RO |
| 0 | CRF | CR register update flag<br>● 0b0: CR register update completed<br>● 0b1: CR register is being updated<br>**Notice**<br>The CR register can only be updated when the CRF is 0. | RO |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 15.5.6 Window Register (IWDT_WINR)

**Note**

- This register can only be modified after writing 0x5555 to the IWDT_KR register.
- This register can only be modified when IWDT_SR.WINRF is 0.

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x90 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000FFF | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | WINR | | | | | | | | | | | |
| Access | | | | | | | | | | | | | | | | | | | | | RW | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:12* | *Reserved* | - | - |
| 11:0 | WINR | Window comparator comparison value | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

# 16 WWDT

## 16.1 Introduction

The window watchdog (WWDT) is often used to monitor software failures triggered by unexpected errors. An interrupt is generated when the counter value is decremented to 0x40, and a reset is generated when the counter value is decremented to 0x3F; if the user program updates the counter value when the counter value is greater than the window value, a reset is generated. So the user program can only update the WWDT counter value within a limited window (the counter value is less than or equal to the window value and greater than 0x3F).

## 16.2 Features

- 7-bit down counter
- PCLK divided clock as count clock
- When PCLK is 48MHz, its timeout time is 85μs - 699ms
- An interrupt is generated when the count value is decremented to 0x40
- A reset is generated when the count value is decremented to 0x3F
- Update the counter when the count value is greater than the window value, a reset is generated

## 16.3 Functional Description

### 16.3.1 Function Block Diagram

After the watchdog is started, the value of the counter WCNT decreases by 1 on each rising edge of WWDTCLK. When the counter is decremented to 0x40, the watchdog will generate a pre-overflow signal POV, enable WWDT_CR1.IE can make the watchdog generate a pre-overflow interrupt; when the counter is decremented to 0x3F, the watchdog will generate a reset signal. User program can write WWDT_CR0.WCNT to update WCNT to prevent watchdog from generating reset signal. If the value of WCNT is greater than the value of WINR when the user updates WCNT, the watchdog will also generate a reset signal.

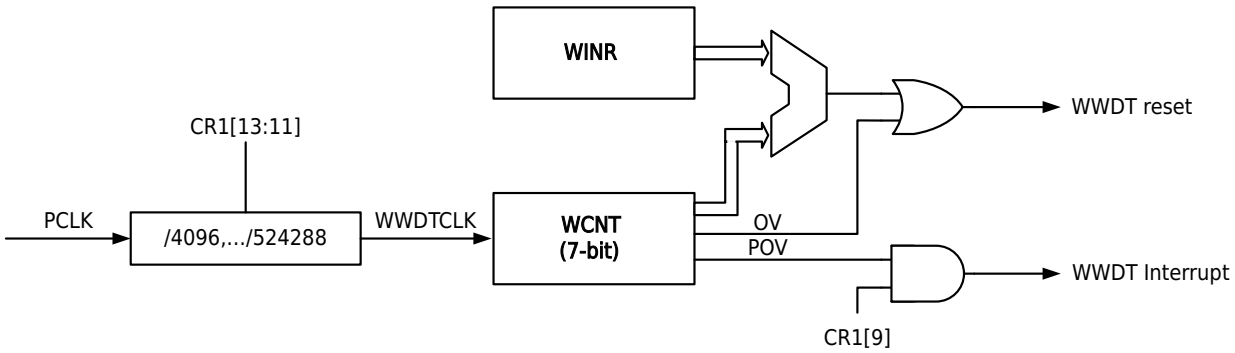



Figure 16-1 WWDT Function Block Diagram


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 16.3.2 Enable Watchdog

After the system is reset, the watchdog is turned off. Set WWDT_CR0.EN to 1 to enable the watchdog. The user program cannot disable the enabled watchdog by writing 0 to WWDT_CR0.EN. It is recommended to enable the watchdog by writing 0xFF to WWDT_CR0 to ensure that the reset is not triggered when the watchdog is enabled.

### 16.3.3 Configure Watchdog Timeout

The required watchdog timeout period can be obtained by properly configuring WWDT_CR0.WCNT and WWDT_CR1.PRS. When PCLK is 48MHz, the range of its timeout period is 85μs - 699ms.

The formula for calculating the timeout is as follows:

$T_{WWDT} = T_{PCLK} \times 4096 \times 2^{CR1.PRS} \times$ (CR0.WCNT-0x3F)

### 16.3.4 Update Down Counter

Write data to WWDT_CR0.WCNT, and the WWDT down counter can be updated. If the written value is less than or equal to 0x3F, a reset will occur directly; if the current count value is greater than the window value when the counter is updated, a reset will occur directly. Therefore, the user program can only update the down counter when the value of the down counter is less than or equal to the window value and greater than 0x3F.

### 16.3.5 Pre-overflow Interrupt

If the down counter decrements to 0x40 and WWDT_CR1.IE is 1, the WWDT will generate a pre-overflow interrupt, the user program can update the down counter in the interrupt service routine to prevent the WWDT from generating a reset; before exiting the interrupt service, the user program should write 0x00 to the WWDT_SR register to clear the WWDT pre-overflow flag.

## 16.4 Operation Example

### 16.4.1 Example of Configuring Watchdog Window Overflow Period

#### Context

The following example configures the overflow period of the watchdog as 786432 PCLK, and the window is 1/3 of the overflow period.

#### Procedure

**Step1.** Configure WWDT_CR1.PRS to 2, and the counting clock of WWDT is PCLK/16384.

**Step2.** Configure WWDT_CR1.WINR to 0x4F, and the update window of WWDT is 0x4F - 0x40.

**Step3.** Configure WWDT_CR0.WCNT to 0x6F, and the overflow period of WWDT is 786432 PCLK.

**Step4.** Set WWDT_CR0.EN to 1 to start the watchdog.

**Step5.** When the value of the down counter is less than or equal to the window value and greater than 0x3F, write 0x6F to WWDT_CR0.WCNT to reload the watchdog counter.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

## 16.5 Register Description

### 16.5.1 Register List

Base address: 0x4000B800

**Notice**

WWDT affects register value after reset.

**Table 16-1 Window Watchdog (WWDT) Register Offset**

| Offset | Register | Description | Width |
|---|---|---|---|
| 0x00 | WWDT_CR0 | Control Register 0 | 32 |
| 0x04 | WWDT_CR1 | Control Register 1 | 32 |
| 0x08 | WWDT_SR | Status Register | 32 |

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 16.5.2 Control Register 0 (WWDT_CR0)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x0000007F | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | EN | WCNT | | | | | | |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | RW | RW | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:8* | *Reserved* | - | - |
| 7 | EN | WWDT enable control<br>Once this bit is set to 1, it cannot be cleared by software<br>● 0b0: Disable WWDT<br>● 0b1: Enable WWDT | RW |
| 6:0 | WCNT | Read: WWDT counter current count value<br>Write: Update WWDT counter current count value | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 16.5.3 Control Register 1 (WWDT_CR1)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x04 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x0000007F | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | PRS | | | Reserved | IE | Reserved | | | | WINR | | | |
| Access | | | | | | | | | | | | | | | | | | | RW | | | | RW | | | | | | RW | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:14* | *Reserved* | - | - |
| 13:11 | PRS | WWDT count clock configuration<br>● 0b000: PCLK/4096<br>● 0b001: PCLK/8192<br>● 0b010: PCLK/16384<br>● 0b011: PCLK/32768<br>● 0b100: PCLK/65536<br>● 0b101: PCLK/131072<br>● 0b110: PCLK/262144<br>● 0b111: PCLK/524288 | RW |
| *10* | *Reserved* | - | - |
| 9 | IE | WWDT pre-overflow interrupt enable control<br>● 0b0: Disable pre-overflow interrupt<br>● 0b1: Enable pre-overflow interrupt | RW |
| *8:7* | *Reserved* | - | - |
| 6:0 | WINR | Watchdog window value<br>The user program can only update the value of the WWDT counter when the counter value is less than or equal to the window value and greater than 0x3F. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 16.5.4 Status Register (WWDT_SR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x08 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | POV |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | RW |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:1* | *Reserved* | - | - |
| 0 | POV | WWDT pre-overflow flag<br>Write 0 to clear the pre-overflow flag, write 1 has no effect.<br>Read:<br>● 0b0: No pre-overflow has occurred in WWDT<br>● 0b1: Pre-overflow has occurred in WWDT | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

# 17 LPUART

## 17.1 Introduction

The Low Power Universal Asynchronous Receiver/Transmitter (LPUART) is a UART that allows full-duplex communication in the low power mode of MCU. When the MCU is in DeepSleep mode, LPUART can also receive UART data frames normally and wake up the MCU through the reception completion interrupt.

## 17.2 Features

- Configuration clock PCLK
- Transmission clock SCLK (SCLK can select XTL, RCL and PCLK)
- Support synchronous half-duplex, asynchronous full-duplex and single-wire half-duplex transmission.
- Programmable serial communication function
  - 2 character lengths: 8 bits, 9 bits
  - 3 check methods: no check, odd check, even check
  - 3 stop lengths: 1 bit, 1.5 bits, 2 bits
- Supports transmitting and receiving data in low power mode
- 16-bit baud rate counter
- Support hardware flow control (RTS, CTS)
- Support multi-machine communication and automatic address recognition
- Support data transfer via DMA


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

## 17.3 Functional Description

### 17.3.1 Function Block Diagram

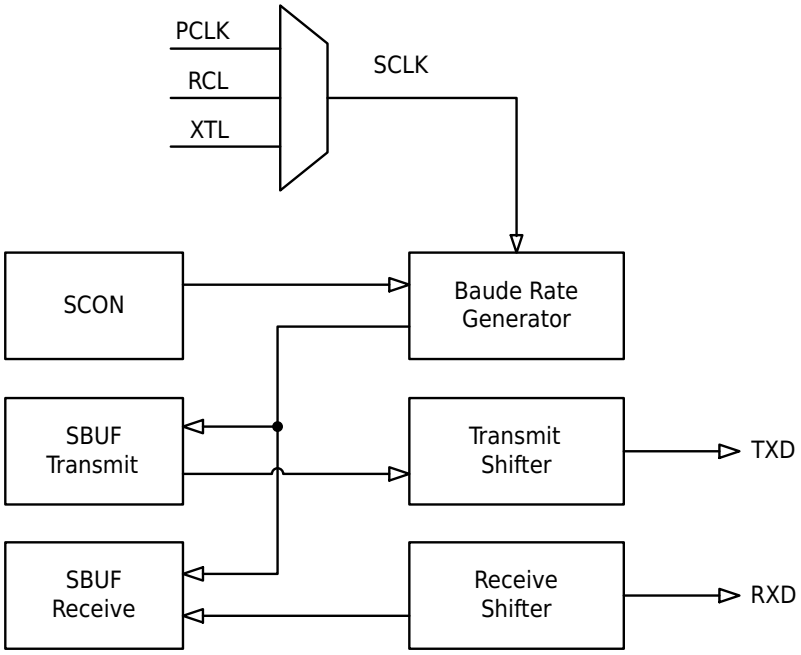



Figure 17-1 LPUART Function Block Diagram

### 17.3.2 Clock Description

The LPUART module has two clocks:

- Configuration Clock PCLK

  The configuration clock is used for register configuration of the LPUART module by the system APB bus, and the configuration clock is fixed as the APB bus clock PCLK. When the system enters DeepSleep mode, the PCLK clock will stop.

- Transmission Clock SCLK

  The transmission clock is used for LPUART data transceiver logic work, and the clock source can be selected from the external low-speed crystal oscillator clock (XTL), the internal low-speed clock (RCL), and the PCLK clock.

  When the system enters DeepSleep mode, if SCLK selects XTL or RCL, the LPUART can still perform normal data transmission and reception without being affected by the system's DeepSleep mode.

### 17.3.3 Working Mode

LPUART supports multiple working modes: synchronous half-duplex mode, asynchronous full-duplex mode, and single-wire half-duplex mode.

By combining LPUART_SCON.SM with LPUART_SCON.HDSEL, various required working modes can be configured.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 17.3.3.1 Mode0-Mode3 Function Comparison

Configure LPUART_SCON.SM to select different transmission modes: Mode0-Mode3. The main function comparison of these four working modes is shown in the table below:

**Table 17-1 LPUART Mode0/1/2/3 Data Structure**

| Working Mode | | Transmission Bit Width | Data Composition | Baud Rate |
|---|---|---|---|---|
| Mode0 | Synchronous half-duplex mode | 8-bit | Data (8bits) | $BaudRate = \frac{f_{SCLK}}{12}$ |
| Mode1 | Asynchronous full-duplex mode | 10-11-bit | Start (1bit) + Data (8bits) + Stop (1-2bits) | $BaudRate = \frac{f_{SCLK}}{OVER \times SCNT}$ |
| Mode2 | Asynchronous full-duplex mode | 11-12-bit | Start (1bit) + Data (8bits) + B8 (1bit) + Stop (1-2bits) | $BaudRate = \frac{f_{SCLK}}{OVER}$ |
| Mode3 | Asynchronous full-duplex mode | 11-12-bit | Start (1bit) + Data (8bits) + B8 (1bit) + Stop (1-2bits) | $BaudRate = \frac{f_{SCLK}}{OVER \times SCNT}$ |

- When SCLK is selected as XTL (with the frequency of 32.768 kHz), Mode 1/2/3 can support baud rates of 9600/4800. The generation of the baud rate is detailed in the registers LPUART_BSEL/ LPUART_MODU.
- Mode0 can only be used as a master to transmit LPUART synchronous shift clock, and cannot be used as a slave to receive LPUART synchronous shift clock input from the outside.
- $f_{SCLK}$ is the clock frequency of SCLK.
- OVER value:
  - When LPUART_SCON.OVER=0b00, the value of OVER in Mode1/3 mode is 16; and the value of OVER in Mode2 mode is 32.
  - When LPUART_SCON.OVER=0b01, the value of OVER in Mode1/3 mode is 8; and the value of OVER in Mode2 mode is 16.
  - When LPUART_SCON.OVER=0b10, the value of OVER in Mode1/3 mode is 4; and the value of OVER in Mode2 mode is 8.
- The definition of SCNT is detailed in register LPUART_SCNT introduction.
- B8 data bit is special and has different meanings in different applications, please refer to the following table:

**Table 17-2 B8 Data Meaning**

| Application Scenario | LPUART_SCON.ADRDET | LPUART_SCON.B8CONT[1:0] | B8 Data Meaning |
|---|---|---|---|
| Parity check | - | 0b01/0b10 | When receiving, B8 is the parity check bit of the 8bits of data received.<br>When transmitting, B8 is the parity check bit of the 8bits of data transmitted. |
| Multi-machine communication | 0b1 | - | B8=1, it means that it is currently an address frame.<br>B8=0, it means that it is currently a data frame. |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Application Scenario | LPUART_SCON.ADRDET | LPUART_SCON.B8CONT[1:0] | B8 Data Meaning |
|---|---|---|---|
| Other | 0b0 | 0b00/0b11 | Bit8 receiving/transmitting data. |

When the multi-machine communication mode is turned on, the parity check of the received data is automatically closed; the parity check of the transmitted data is still controlled by B8CONT.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 17.3.3.2 Mode0 Data Transmitting and Reception

- Mode0 Data Transmitting

  When LPUART_SCON.REN set to 0, the LPUART operates in the transmission mode. The data to be transmitted is written into the LPUART_SBUF register, and then the data will be output from the RXD (lower bits first and higher bits later), while the synchronous shift clock is output from the TXD.

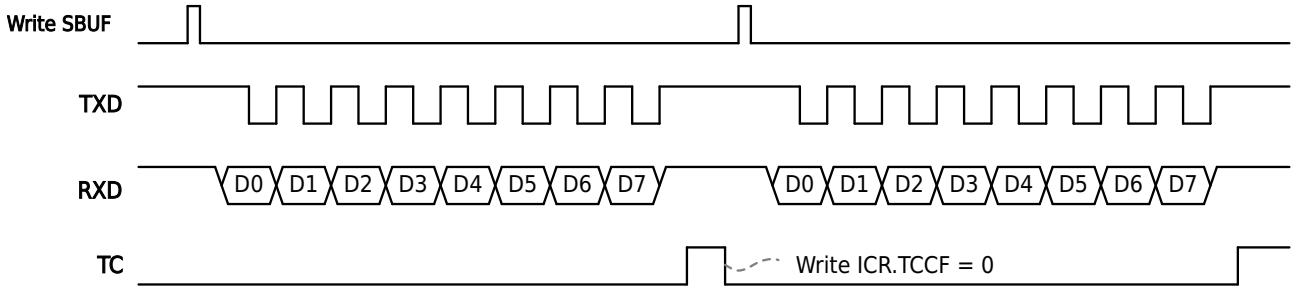



Figure 17-2 Mode0 Data Transmitting

- Mode0 Data Receiving

  When LPUART_SCON.REN is set to 1, the LPUART is in the receiving state. When the LPUART_ISR.RC flag is 0, the TXD pin will output 8 synchronous shift clocks, and data will be input from RXD (lower bits first and higher bits later), each time the LPUART_ISR.RC flag is cleared, the next byte can be received.

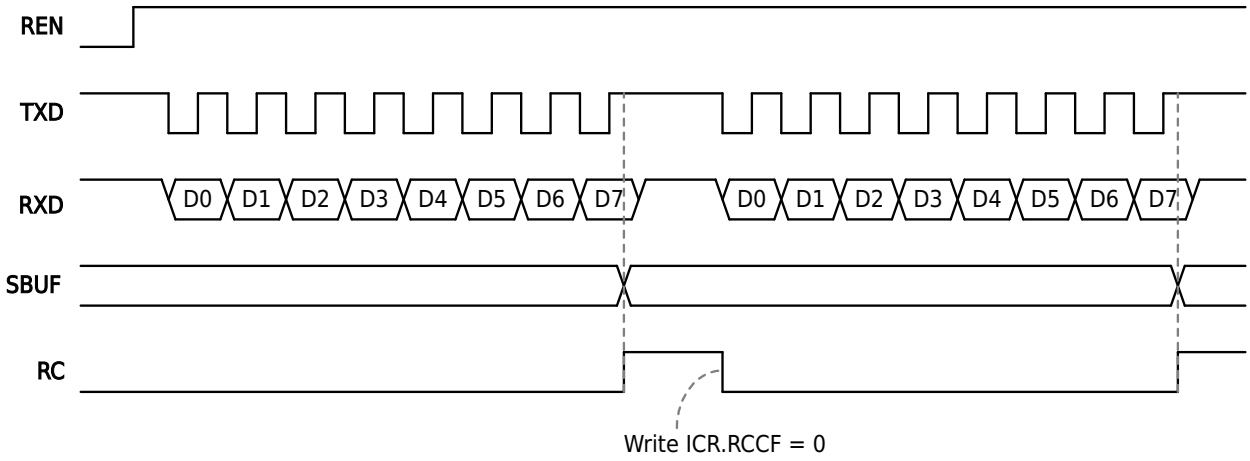



Figure 17-3 Mode0 Data Receiving

### 17.3.3.3 Mode1 Data Transmitting and Reception

- Mode1 Data Transmitting

  Whether LPUART_SCON.REN is 0 or 1, data can be transmitted normally.

  Write data to LPUART_SBUF, the hardware automatically loads the data into the transmit shift register and shifts the output from the TXD pin (lower bits first and higher bits later), and the LPUART_ISR.TXE flag is set to 1 by the hardware.

  When the user queries that the LPUART_ISR.TXE flag is 1, the next data to be transmitted can be written to the LPUART_SBUF register.

  Whenever the TXD pin completes transmitting a frame of data, the LPUART_ISR.TC flag will be set to 1 by hardware to indicate the current transmitting progress.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

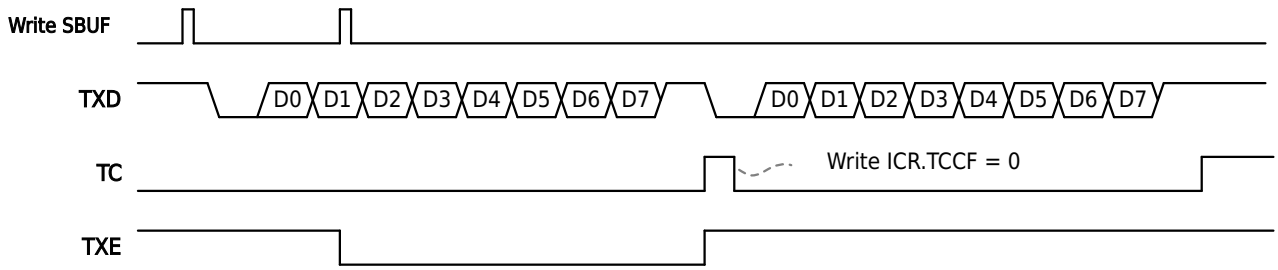



Figure 17-4 Mode1 Data Transmitting

- Mode1 Data Receiving

  When receiving data, set LPUART_SCON.REN to 1, and clear the LPUART_ISR.RC bit to 0.

  When a frame of data reception is completed, LPUART_ISR.RC flag will be set to 1 by hardware, and the received data can be read from LPUART_SBUF register.

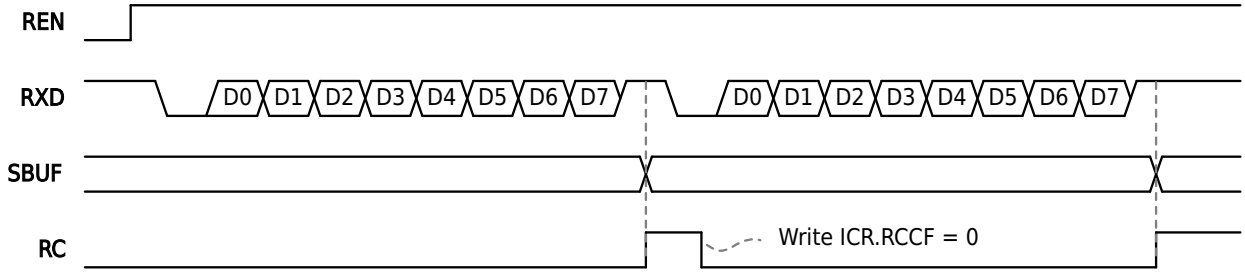



Figure 17-5 Mode1 Data Receiving

### 17.3.3.4 Mode2/Mode3 Data Transmitting and Reception

- Mode2/3 Data Transmitting

  Whether LPUART_SCON.REN is 0 or 1, data can be transmitted normally.

  Write data to LPUART_SBUF, the hardware automatically loads the data into the transmit shift register and shifts the output from the TXD pin (lower bits first and higher bits later), and the LPUART_ISR.TXE flag is set to 1 by the hardware.

  When the user queries that the LPUART_ISR.TXE flag is 1, the next data to be transmitted can be written to the LPUART_SBUF register.

  Whenever the TXD pin completes transmitting a frame of data, the LPUART_ISR.TC flag will be set to 1 by hardware to indicate the current transmitting progress.

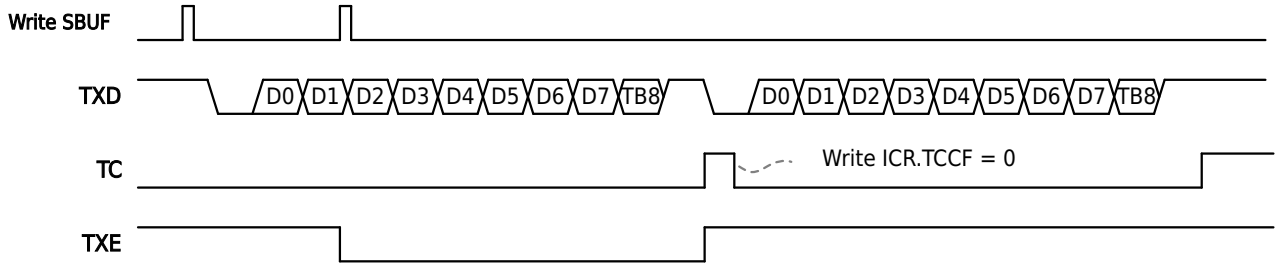



Figure 17-6 Mode2/Mode3 Data Transmitting

- Mode2/3 Data Receiving

  When receiving data, set LPUART_SCON.REN to 1, and clear the LPUART_ISR.RC bit to 0.

  When a frame of data reception is completed, LPUART_ISR.RC flag will be set to 1 by hardware, and the received data can be read from LPUART_SBUF rigester.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

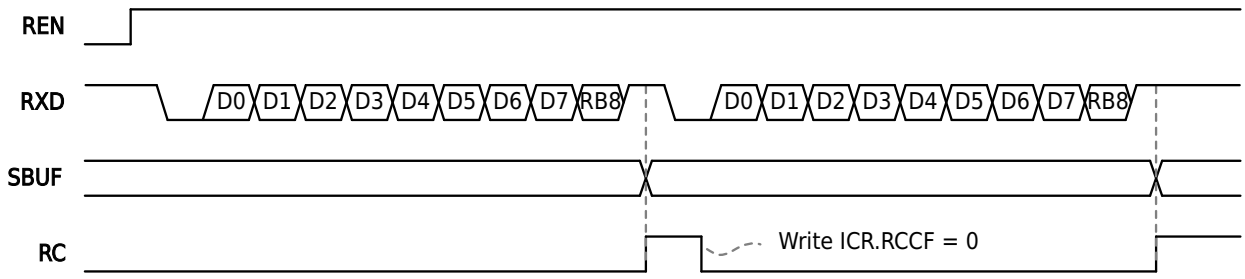



Figure 17-7 Mode2/Mode3 Data Receiving

#### 17.3.3.5 Single-Wire Half-Duplex Data Transmitting and Reception

When LPUART_SCON.HDSEL=0b1, it can enter the single-wire half-duplex transmission mode:

- The TX and RX signal lines are connected inside the module.
- The RX signal is no longer used, and the received and transmitted data are all completed through the TX signal. TX signal is done by hardware logic without software control.
- After configuring LPUART_SCON.HDSEL to 0b1, must wait for the transmission duration of one data bit (i.e., after 1/BaudRate seconds) before normal reception can occur.
- When the transmit buffer is empty, the TX signal is always input (receive state). Once a data is filled into the transmit buffer, the TX signal becomes output (transmit state). When the transmission is complete, the transmit buffer becomes empty and the TX signal returns to input (receive state).
- When there is no data transmission, the TXD pin is always in the input state. The user program needs to configure this pin as an open-drain output and connect an external pull-up resistor.

**Notice**

1. Even if the module is in the process of receiving, as long as the transmit buffer is filled with data, the TX signal will immediately become output (transmit state). The receiving process will be interrupted and a wrong received data will be obtained.
2. Similarities and differences between single-wire half-duplex mode and Mode0 mode:
   - The same: both use a signal to time-division multiplex to complete data transmission and reception.
   - Different: Mode0 is synchronous transmission, with synchronous clock. The single-wire half-duplex mode is asynchronous transmission, and all baud rate calculations still follow the baud rate formula of Mode1-Mode3, only the transceiver function is multiplexed on the TX signal.

Single-wire half-duplex transmission supports baud rates of 9600/4800. However, it is important to note that when receiving data in single-wire half-duplex mode at baud rates of 9600/4800, the following steps must be followed:

**Step1.** Refer to the relevant descriptions of pin digital multiplexing functions in the "General-Purpose I/O (GPIO)" chapter to map TXD to the required pin; configure TXD as an open-drain output and enable the pull-up resistor.

**Step2.** Configure SYSCTRL module register PERI_CLKEN0.LPUARTx to 1 to enable LPUART configuration clock and working clock.

**Step3.** Configure LPUART_SCON.SCLKSEL to select the transmission clock source for LPUART.

**Step4.** Set LPUART_SCON.HDSEL to 0b1 to enable single-wire mode.

**Step5.** Add a delay of at least 500 μs and initialize TXD to the receiving state.

**Step6.** Configure the LPUART_BSEL and LPUART_MODU registers to enable 9600/4800 baud rate modulation.

**Step7.** For subsequent steps, refer to the "Receive Data Example" chapter.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

For other scenarios, such as transmitting data in single-wire half-duplex mode at baud rates of 9600/4800, or modulating the baud rate by configuring the LPUART_SCNT register, simply follow the steps outlined in the "Transmit Data Example" or "Receive Data Example" chapters for normal configuration.

### 17.3.4 Baud Rate Generation

The formula for generating the baud rate in Mode0-Mode3 is as follows:

- Mode0 baud rate generation formula: $\text{BaudRate} = \frac{f_{SCLK}}{12}$
- Mode1 baud rate generation formula: $\text{BaudRate} = \frac{f_{SCLK}}{\text{OVER} \times \text{SCNT}}$
- Mode2 baud rate generation formula: $\text{BaudRate} = \frac{f_{SCLK}}{\text{OVER}}$
- Mode3 baud rate generation formula: $\text{BaudRate} = \frac{f_{SCLK}}{\text{OVER} \times \text{SCNT}}$

**Note**

- $f_{SCLK}$ is the clock frequency of SCLK.
- OVER value:
  - When LPUART_SCON.OVER = 0b00, the value of OVER in Mode1/3 mode is 16; and the value of OVER in Mode2 mode is 32.
  - When LPUART_SCON.OVER = 0b01, the value of OVER in Mode1/3 mode is 8; and the value of OVER in Mode2 mode is 16.
  - When LPUART_SCON.OVER = 0b10, the value of OVER in Mode1/3 mode is 4; and the value of OVER in Mode2 mode is 8.
- The definition of SCNT is detailed in register LPUART_SCNT introduction.

#### Mode1/Mode3 Baud Rate Setting Example

**Table 17-3 SCLK = 4MHz Baud Rate Calculation Table**

| Baud Rate | SCLK=4MHz | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | OVER=4 | | | OVER=8 | | | OVER=16 | | |
| | SCNT | Actual Baud Rate | Error% | SCNT | Actual Baud Rate | Error% | SCNT | Actual Baud Rate | Error% |
| 2400 | 417 | 2398.08 | -0.08% | 208 | 2403.85 | 0.16% | 104 | 2403.85 | 0.16% |
| 4800 | 208 | 4807.69 | 0.16% | 104 | 4807.69 | 0.16% | 52 | 4807.69 | 0.16% |
| 9600 | 104 | 9615.38 | 0.16% | 52 | 9615.38 | 0.16% | 26 | 9615.38 | 0.16% |
| 19200 | 52 | 19230.77 | 0.16% | 26 | 19230.77 | 0.16% | 13 | 19230.77 | 0.16% |
| 38400 | 26 | 38461.54 | 0.16% | 13 | 38461.54 | 0.16% | 7 | 35714.29 | -6.99% |
| 57600 | 17 | 58823.53 | 2.12% | 9 | 55555.56 | -3.55% | 4 | 62500.00 | 8.51% |
| 76800 | 13 | 76923.08 | 0.16% | 7 | 71428.57 | -6.99% | 3 | 83333.33 | 8.51% |
| 115200 | 9 | 111111.11 | -3.55% | 4 | 125000.00 | 8.51% | 2 | 125000.00 | 8.51% |
| 128000 | 8 | 125000.00 | -2.34% | 4 | 125000.00 | -2.34% | 2 | 125000.00 | -2.34% |
| 256000 | 4 | 250000.00 | -2.34% | 2 | 250000.00 | -2.34% | 1 | 250000.00 | -2.34% |
| 1000000 | 1 | 1000000.00 | 0.00% | 1 | / | / | 0 | / | / |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Table 17-4 SCLK = 8MHz Baud Rate Calculation Table

| Baud Rate | SCLK=8MHz | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | OVER=4 | | | OVER=8 | | | OVER=16 | | |
| | SCNT | Actual Baud Rate | Error% | SCNT | Actual Baud Rate | Error% | SCNT | Actual Baud Rate | Error% |
| 2400 | 833 | 2400.96 | 0.04% | 417 | 2398.08 | -0.08% | 208 | 2403.85 | 0.16% |
| 4800 | 417 | 4796.16 | -0.08% | 208 | 4807.69 | 0.16% | 104 | 4807.69 | 0.16% |
| 9600 | 208 | 9615.38 | 0.16% | 104 | 9615.38 | 0.16% | 52 | 9615.38 | 0.16% |
| 19200 | 104 | 19230.77 | 0.16% | 52 | 19230.77 | 0.16% | 26 | 19230.77 | 0.16% |
| 38400 | 52 | 38461.54 | 0.16% | 26 | 38461.54 | 0.16% | 13 | 38461.54 | 0.16% |
| 57600 | 35 | 57142.86 | -0.79% | 17 | 58823.53 | 2.12% | 9 | 55555.56 | -3.55% |
| 76800 | 26 | 76923.08 | 0.16% | 13 | 76923.08 | 0.16% | 7 | 71428.57 | -6.99% |
| 115200 | 17 | 117647.06 | 2.12% | 9 | 111111.11 | -3.55% | 4 | 125000.00 | 8.51% |
| 128000 | 16 | 125000.00 | -2.34% | 8 | 125000.00 | -2.34% | 4 | 125000.00 | -2.34% |
| 256000 | 8 | 250000.00 | -2.34% | 4 | 250000.00 | -2.34% | 2 | 250000.00 | -2.34% |
| 1000000 | 2 | 1000000.00 | 0.00% | 1 | 1000000.00 | 0.00% | 1 | / | / |
| 2000000 | 1 | 2000000.00 | 0.00% | 1 | / | / | 0 | / | / |

Table 17-5 SCLK = 16MHz Baud Rate Calculation Table

| Baud Rate | SCLK=16MHz | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | OVER=4 | | | OVER=8 | | | OVER=16 | | |
| | SCNT | Actual Baud Rate | Error% | SCNT | Actual Baud Rate | Error% | SCNT | Actual Baud Rate | Error% |
| 2400 | 1667 | 2399.52 | -0.02% | 833 | 2400.96 | 0.04% | 417 | 2398.08 | -0.08% |
| 4800 | 833 | 4801.92 | 0.04% | 417 | 4796.16 | -0.08% | 208 | 4807.69 | 0.16% |
| 9600 | 417 | 9592.33 | -0.08% | 208 | 9615.38 | 0.16% | 104 | 9615.38 | 0.16% |
| 19200 | 208 | 19230.77 | 0.16% | 104 | 19230.77 | 0.16% | 52 | 19230.77 | 0.16% |
| 38400 | 104 | 38461.54 | 0.16% | 52 | 38461.54 | 0.16% | 26 | 38461.54 | 0.16% |
| 57600 | 69 | 57971.01 | 0.64% | 35 | 57142.86 | -0.79% | 17 | 58823.53 | 2.12% |
| 76800 | 52 | 76923.08 | 0.16% | 26 | 76923.08 | 0.16% | 13 | 76923.08 | 0.16% |
| 115200 | 35 | 114285.71 | -0.79% | 17 | 117647.06 | 2.12% | 9 | 111111.11 | -3.55% |
| 128000 | 31 | 129032.26 | 0.81% | 16 | 125000.00 | -2.34% | 8 | 125000.00 | -2.34% |
| 256000 | 16 | 250000.00 | -2.34% | 8 | 250000.00 | -2.34% | 4 | 250000.00 | -2.34% |
| 1000000 | 4 | 1000000.00 | 0.00% | 2 | 1000000.00 | 0.00% | 1 | 1000000.00 | 0.00% |
| 2000000 | 2 | 2000000.00 | 0.00% | 1 | 2000000.00 | 0.00% | 1 | / | / |
| 4000000 | 1 | 4000000.00 | 0.00% | 1 | / | / | 0 | / | / |

Table 17-6 SCLK = 24MHz Baud Rate Calculation Table

| Baud Rate | SCLK=24MHz | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | OVER=4 | | | OVER=8 | | | OVER=16 | | |
| | SCNT | Actual Baud Rate | Error% | SCNT | Actual Baud Rate | Error% | SCNT | Actual Baud Rate | Error% |
| 2400 | 2500 | 2400.00 | 0.00% | 1250 | 2400.00 | 0.00% | 625 | 2400.00 | 0.00% |
| 4800 | 1250 | 4800.00 | 0.00% | 625 | 4800.00 | 0.00% | 313 | 4792.33 | -0.16% |
| 9600 | 625 | 9600.00 | 0.00% | 313 | 9584.66 | -0.16% | 156 | 9615.38 | 0.16% |
| 19200 | 313 | 19169.33 | -0.16% | 156 | 19230.77 | 0.16% | 78 | 19230.77 | 0.16% |
| 38400 | 156 | 38461.54 | 0.16% | 78 | 38461.54 | 0.16% | 39 | 38461.54 | 0.16% |
| 57600 | 104 | 57692.31 | 0.16% | 52 | 57692.31 | 0.16% | 26 | 57692.31 | 0.16% |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Baud Rate | SCLK=24MHz | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | OVER=4 | | | OVER=8 | | | OVER=16 | | |
| | SCNT | Actual Baud Rate | Error% | SCNT | Actual Baud Rate | Error% | SCNT | Actual Baud Rate | Error% |
| 76800 | 78 | 76923.08 | 0.16% | 39 | 76923.08 | 0.16% | 20 | 75000.00 | -2.34% |
| 115200 | 52 | 115384.62 | 0.16% | 26 | 115384.62 | 0.16% | 13 | 115384.62 | 0.16% |
| 128000 | 47 | 127659.57 | -0.27% | 23 | 130434.78 | 1.90% | 12 | 125000.00 | -2.34% |
| 256000 | 23 | 260869.57 | 1.90% | 12 | 250000.00 | -2.34% | 6 | 250000.00 | -2.34% |
| 1000000 | 6 | 1000000.00 | 0.00% | 3 | 1000000.00 | 0.00% | 2 | / | / |
| 2000000 | 3 | 2000000.00 | 0.00% | 2 | / | / | 1 | / | / |
| 6000000 | 1 | 6000000.00 | 0.00% | 1 | / | / | 0 | / | / |

Table 17-7 SCLK = 32MHz Baud Rate Calculation Table

| Baud Rate | SCLK=32MHz | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | OVER=4 | | | OVER=8 | | | OVER=16 | | |
| | SCNT | Actual Baud Rate | Error% | SCNT | Actual Baud Rate | Error% | SCNT | Actual Baud Rate | Error% |
| 2400 | 3333 | 2400.24 | 0.01% | 1667 | 2399.52 | -0.02% | 833 | 2400.96 | 0.04% |
| 4800 | 1667 | 4799.04 | -0.02% | 833 | 4801.92 | 0.04% | 417 | 4796.16 | -0.08% |
| 9600 | 833 | 9603.84 | 0.04% | 417 | 9592.33 | -0.08% | 208 | 9615.38 | 0.16% |
| 19200 | 417 | 19184.65 | -0.08% | 208 | 19230.77 | 0.16% | 104 | 19230.77 | 0.16% |
| 38400 | 208 | 38461.54 | 0.16% | 104 | 38461.54 | 0.16% | 52 | 38461.54 | 0.16% |
| 57600 | 139 | 57553.96 | -0.08% | 69 | 57971.01 | 0.64% | 35 | 57142.86 | -0.79% |
| 76800 | 104 | 76923.08 | 0.16% | 52 | 76923.08 | 0.16% | 26 | 76923.08 | 0.16% |
| 115200 | 69 | 115942.03 | 0.64% | 35 | 114285.71 | -0.79% | 17 | 117647.06 | 2.12% |
| 128000 | 63 | 126984.13 | -0.79% | 31 | 129032.26 | 0.81% | 16 | 125000.00 | -2.34% |
| 256000 | 31 | 258064.52 | 0.81% | 16 | 250000.00 | -2.34% | 8 | 250000.00 | -2.34% |
| 1000000 | 8 | 1000000.00 | 0.00% | 4 | 1000000.00 | 0.00% | 2 | 1000000.00 | 0.00% |
| 2000000 | 4 | 2000000.00 | 0.00% | 2 | 2000000.00 | 0.00% | 1 | 2000000.00 | 0.00% |
| 4000000 | 2 | 4000000.00 | 0.00% | 1 | 4000000.00 | 0.00% | 1 | / | / |
| 8000000 | 1 | 8000000.00 | 0.00% | 1 | / | / | 0 | / | / |

Table 17-8 SCLK = 48MHz Baud Rate Calculation Table

| Baud Rate | SCLK=48MHz | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | OVER=4 | | | OVER=8 | | | OVER=16 | | |
| | SCNT | Actual Baud Rate | Error% | SCNT | Actual Baud Rate | Error% | SCNT | Actual Baud Rate | Error% |
| 2400 | 5000 | 2400.00 | 0.00% | 2500 | 2400.00 | 0.00% | 1250 | 2400.00 | 0.00% |
| 4800 | 2500 | 4800.00 | 0.00% | 1250 | 4800.00 | 0.00% | 625 | 4800.00 | 0.00% |
| 9600 | 1250 | 9600.00 | 0.00% | 625 | 9600.00 | 0.00% | 313 | 9584.66 | -0.16% |
| 19200 | 625 | 19200.00 | 0.00% | 313 | 19169.33 | -0.16% | 156 | 19230.77 | 0.16% |
| 38400 | 313 | 38338.66 | -0.16% | 156 | 38461.54 | 0.16% | 78 | 38461.54 | 0.16% |
| 57600 | 208 | 57692.31 | 0.16% | 104 | 57692.31 | 0.16% | 52 | 57692.31 | 0.16% |
| 76800 | 156 | 76923.08 | 0.16% | 78 | 76923.08 | 0.16% | 39 | 76923.08 | 0.16% |
| 115200 | 104 | 115384.62 | 0.16% | 52 | 115384.62 | 0.16% | 26 | 115384.62 | 0.16% |
| 128000 | 94 | 127659.57 | -0.27% | 47 | 127659.57 | -0.27% | 23 | 130434.78 | 1.90% |
| 256000 | 47 | 255319.15 | -0.27% | 23 | 260869.57 | 1.90% | 12 | 250000.00 | -2.34% |
| 1000000 | 12 | 1000000.00 | 0.00% | 6 | 1000000.00 | 0.00% | 3 | 1000000.00 | 0.00% |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Baud Rate | SCLK=48MHz | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | OVER=4 | | | OVER=8 | | | OVER=16 | | |
| | SCNT | Actual Baud Rate | Error% | SCNT | Actual Baud Rate | Error% | SCNT | Actual Baud Rate | Error% |
| 2000000 | 6 | 2000000.00 | 0.00% | 3 | 2000000.00 | 0.00% | 2 | 1500000.00 | -25.00% |
| 4000000 | 4 | 3000000.00 | 0.00% | 2 | 3000000.00 | 0.00% | 1 | 3000000.00 | 0.00% |
| 6000000 | 3 | 4000000.00 | 0.00% | 2 | / | / | 1 | / | / |
| 12000000 | 1 | 12000000.00 | 0.00% | 1 | / | / | 0 | / | / |

### 17.3.5 Frame Error Detection

When LPUART works in Mode1/2/3, it has frame error detection function:

- When receiving data, if the stop bit is not recognized within the expected time, it is a frame error.
- When a frame error is detected, LPUART_ISR.FE will be set to 1 by hardware, and the user program should clear this flag in time.

### 17.3.6 Multi-Machine Communication

When working in Mode2/3, set the LPUART_SCON.ADRDET to 1 to enable the multi-machine communication address automatic identification function. The master can transmit address frames and data frames, and the slave can receive data frames only after receiving matching address frames.

#### Master Transmitting Address and Data Frame

When LPUART_SBUF.DATA8 is 1, the address frame is transmitted, LPUART_SBUF.DATA is the address of the target slave.

When LPUART_SBUF.DATA8 is 0, the data frame is transmitted, LPUART_SBUF.DATA is the data to be transmitted.

#### Slave Receiving Address and Data Frame

When LPUART_SCON.ADRDET is 1, the slave only receives address frames that match its own address, ignoring data frames and unmatched address frames. When the slave receives the address frame from RXD, the hardware automatically compares the address in the address frame with the local address. Only when the two match, the received address is stored in the LPUART_SBUF register and LPUART_ISR.RC is set.

When LPUART_SCON.ADRDET is 0, the slave only receives data frames and ignores address frames. The user program should set LPUART_SCON.ADRDET to 0 after the address matches to receive subsequent data frames transmitted by the master.

#### Slave Address Configuration

The slave address consists of the broadcast address and the user configuration address:

- Broadcast Address: LPUART_SADDR | LPUART_SADEN
- User Configuration Address: LPUART_SADDR & LPUART_SADEN

When an address frame is received, if the address received from the bus, after being ANDed with the value of LPUART_SADEN, matches the user-configured address, or if the received address, after being ANDed with the broadcast address, matches the broadcast address, it is considered that an address frame matching the slave address has been received.

Example: LPUART_SADDR=0x55, LPUART_SADEN=0xF0, then the slave can match addresses 0xF5, 0xF7, 0xFD, 0xFF (broadcast address) and 0x50~0x5F (user-configured address).


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 17.3.7 Hardware Flow Control

The hardware flow control function can be realized by adding CTS and RTS signals, the LPUART module will automatically control the transmitting and receiving of data based on the high and low levels of CTS and RTS without the need for software judgment.

The hardware flow control diagram between two LPUART modules is as follows:

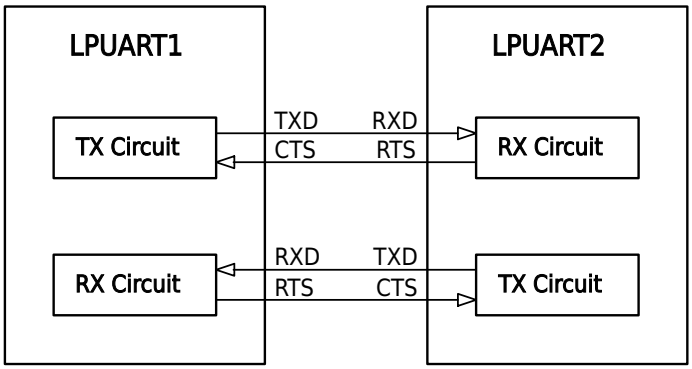



Figure 17-8 LPUART Hardware Flow Control

#### RTS Flow Control

When RTS flow control is enabled (LPUART_SCON.RTSEN is 1), the RTS pin indicates the receiving status of this module:

- When the receive buffer is empty, the RTS pin turns low level to notify the transmitting end to send the next frame of data.
- When the receive buffer is full, the RTS pin turns to high level, notifying the transmitting end to pause sending the next frame of data.

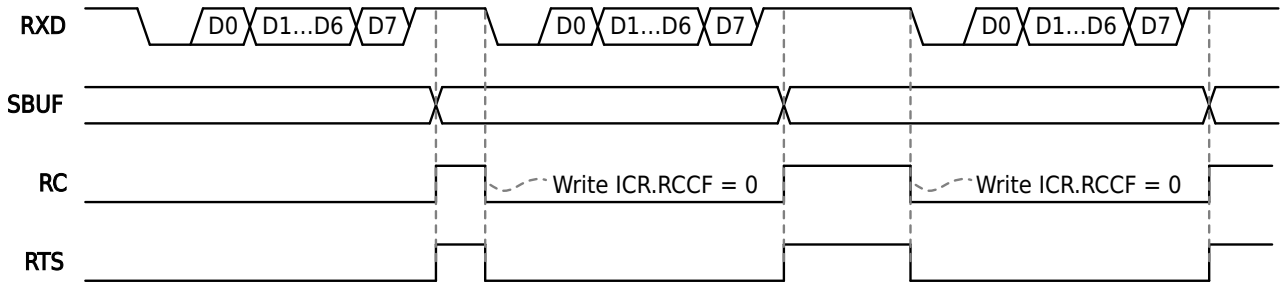



Figure 17-9 RTS Hardware Flow Control Signal

#### CTS Flow Control

When CTS flow control is enabled (LPUART_SCON.CTSEN is set to 1), the CTS pin instructs this module to transmit the next frame of data:

- If CTS is low level, LPUART transmits the next frame of data.
- If CTS is high level, after LPUART finishes transmitting the current frame, it waits for CTS to become low level before sending the next frame.

When the CTS level changes, LPUART_ISR.CTSIF is automatically set to 1. Read LPUART_ISR.CTS to obtain the current CTS pins level value.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

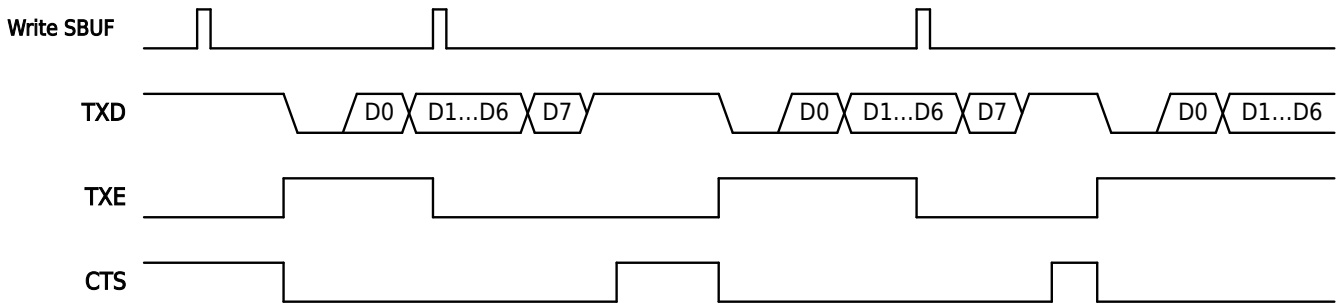



Figure 17-10 CTS Hardware Flow Control Signal

### 17.3.8 DMA Transmission

The LPUART module supports the DMA handshake protocol and can transmit data through DMA.

- When LPUART_SCON.DMATXEN is set to 1, DMA can be used to control the LPUART to transmit data externally.

  Whenever LPUART_ISR.TXE is 1, DMA will write a frame of data to be transmit to LPUART_SBUF until all data configured in DMA is transferred. When starting to use this function, please first configure the various registers of DMA and enable DMA, and then configure the various registers of LPUART. When you want to end this function, please first disable LPUART_SCON.DMATXEN, and then disable DMA (if the disabling of DMA uses the DMA hardware self-clearing, then as soon as it is detected that the enable of DMA has been cleared by hardware, immediately disable LPUART_SCON.DMATXEN).
- When setting, LPUART_SCON.DMARXEN to 1, DMA can be used to read and save the data received by LPUART.

  Whenever LPUART_ISR.RC is 1, DMA will read out one frame of received data from LPUART_SBUF, until all data configured in DMA is transferred.

### 17.3.9 Receive/Transmit Buffer

#### Receive Buffer

The receive end of the LPUART module has a frame (8bits/9bits) receive buffer, which retains the received data frame until the Stop bit of the next data frame is received before updating the current data frame.

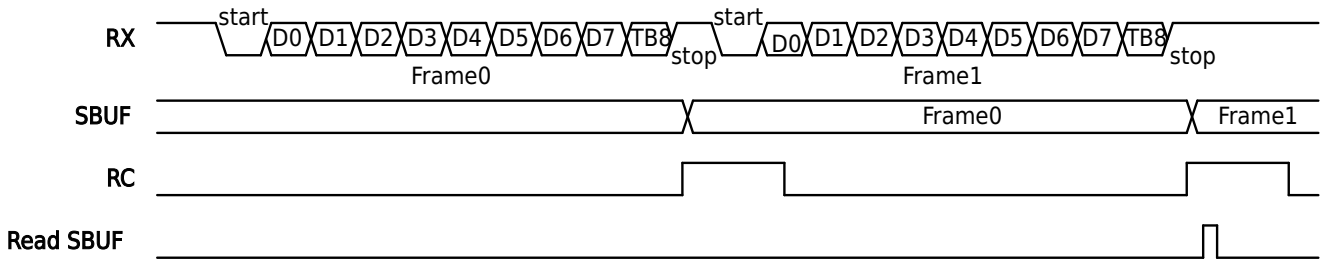



Figure 17-11 Receive Buffer

#### Transmit Buffer

The transmit end of LPUART module has a frame (8bits/9bits) transmit buffer. When the LPUART is transmitting the current frame, the software for the next transmitting data will be written into LPUART_SBUF.

- When LPUART_ISR.TXE = 0, it indicates that the current transmit buffer is full, LPUART_SBUF cannot write the next transmit data.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

- When LPUART_ISR.TXE = 1, it indicates that the current transmit buffer is empty, LPUART_SBUF can write the next transmit data. After the current data transmission is completed, the hardware automatically loads the data in the transmit buffer into the shift register and sends it out.

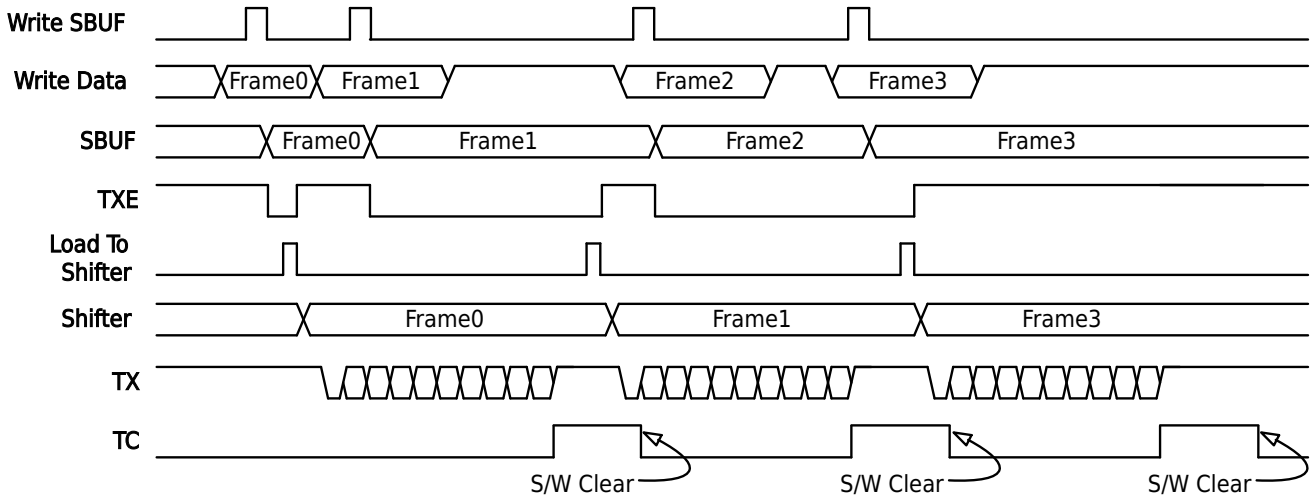



Figure 17-12 Transmit Buffer


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

## 17.4 Operation Example

### 17.4.1 Transmit Data Example

**Procedure**

**Step1.** Refer to the relevant description of the port digital multiplexing function in the "General-Purpose I/O (GPIO)" chapter, map TXD to the required pin; and configure TXD to CMOS output.

**Step2.** Configure SYSCTRL module register PERI_CLKEN0.LPUARTx to 1 to enable LPUART configuration clock and working clock.

**Step3.** Configure LPUART_SCON.SCLKSEL to select the transmission clock source for LPUART.

**Step4.** Configure LPUART_SCON.SM to select the working mode of LPUART.

**Step5.** Configure LPUART_SCON.OVER and LPUART_SCNT register to configure the communication baud rate.

**Step6.** Configure LPUART_SCON.STOPBIT, LPUART_SCON.B8CONT, to configure the STOP length and parity mode of LPUART.

**Step7.** Write 0x00 to the ICR register to clear all status flags.

**Step8.** When the value of ISR.TXE is is queried for 1, write a frame of data to be transmitted to the LPUART_SBUF register.

**Step9.** The query waits for ISR.TC to become 1 and the data frame written this time is sent.

**Step10.** Write 0x00 to ICR.TCCF to clear the transmitting completion flag.

**Step11.** If the data to be transmitted is not completed, then jump to step8.

### 17.4.2 Receive Data Example

**Procedure**

**Step1.** Refer to the relevant description of the port digital multiplexing function in the "General-Purpose I/O (GPIO)" chapter. Map the RXD to the required pin; and configure RXD as an input and enable the pull-up resistor.

**Step2.** Configure SYSCTRL module register PERI_CLKEN0.LPUARTx to 1 to enable LPUART configuration clock and working clock.

**Step3.** Configure LPUART_SCON.SCLKSEL to select the transmission clock source for LPUART.

**Step4.** Configure LPUART_SCON.SM to select the working mode of LPUART.

**Step5.** Configure LPUART_SCON.OVER and LPUART_SCNT register to configure the communication baud rate.

**Step6.** Configure LPUART_SCON.STOPBIT, LPUART_SCON.B8CONT, to configure the STOP length and parity mode of LPUART.

**Step7.** Write 0x00 to the ICR register to clear all status flags.

**Step8.** Set LPUART_SCON.REN to 1 to enable RXD data reception.

**Step9.** The query waits for ISR.RC to become 1, and one frame of data is received.

**Step10.** Query the ISR.PE and ISR.FE to confirm whether the received data frame is valid. If it is invalid, perform error handling, if it is valid, read the data from the LPUART_SBUF register and save it .

**Step11.** Write 0x00 to ICR.RCCF to clear the reception completion flag.

**Step12.** If the data to be received is not completed, jump to step9.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

## 17.5 Register Description

### 17.5.1 Register List

- LPUART0 base address: 0x40000000
- LPUART1 base address: 0x40008000
- LPUART2 base address: 0x4000B400

Table 17-9 Low Power Universal Asynchronous Receiver/Transmitter (LPUART) Register Offset

| Offset | Register | Description | Width |
|---|---|---|---|
| 0x00 | LPUART_SBUF | Data Register | 32 |
| 0x04 | LPUART_SCON | Control Register | 32 |
| 0x08 | LPUART_SADDR | Address Register | 32 |
| 0x0C | LPUART_SADEN | Address Mask Register | 32 |
| 0x10 | LPUART_ISR | Interrupt Flag Bit Register | 32 |
| 0x14 | LPUART_ICR | Interrupt Flag Bit Clear Register | 32 |
| 0x18 | LPUART_SCNT | Baud Rate Register | 32 |
| 0x1C | LPUART_BSEL | Baud Rate Generate Register | 32 |
| 0x20 | LPUART_MODU | 9600/4800 Baud Rate Modulator Register | 32 |

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 17.5.2 Data Register (LPUART_SBUF)

**Note**

When there is data in the SBUF register, the modification of the control bits related to the transmitting data function is prohibited.

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | DATA8 | DATA | | | | | | | |
| Access | | | | | | | | | | | | | | | | | | | | | | | | RW | RW | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:9* | *Reserved* | - | - |
| 8 | DATA8 | Bit8 Data Bit<br>● In Mode 0/1, read this bit is 0, write this bit is invalid<br>● Under Mode 2/3, there are such situations:<br>▶ When the hardware parity bit is turned on, this bit is the parity bit of the received data when receiving, and the verification is performed by the hardware. If the verification error occurs, the verification error flag bit PE is set to 1; this bit is invalid when transmitting, the parity bit of the transmitted data is calculated and transmitted by the hardware.<br>▶ When the hardware parity bit is turned off, this bit is received data Bit8 when receiving; this bit is sent data Bit8 when transmitting.<br>**Note**<br>When the multi-machine communication mode is turned on, the parity check of the received data is automatically closed; the parity check of the transmitted data is still controlled by B8CONT. | RW |
| 7:0 | DATA | Data<br>When transmitting data, write the data to be transmitted into this register; when receiving data, read the received data from this register. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 17.5.3 Control Register (LPUART_SCON)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x04 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | HDSEL | FEIE | CTSIE | CTSEN | RTSEN | DMATXEN | DMARXEN | STOPBIT | | PEIE | SCLKSEL | | OVER | TXEIE | SM | ADRDET | REN | B8CONT | | TCIE | RCIE | | |
| Access | | | | | | | | | | RW | RW | RW | RW | RW | RW | RW | RW | | RW | RW | | RW | RW | RW | RW | RW | RW | | RW | RW | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:23* | *Reserved* | - | - |
| 22 | HDSEL | Single-wire half-duplex function enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 21 | FEIE | Frame error interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 20 | CTSIE | CTS signal flip interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 19 | CTSEN | CTS signal enable control<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 18 | RTSEN | RTS signal enable control<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 17 | DMATXEN | DMA transmit data enable control<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 16 | DMARXEN | DMA receive data enable control<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 15:14 | STOPBIT | STOPBIT duration configuration<br>● 0b00: 1-bit<br>● 0b01: 1.5-bit<br>● 0b10: 2-bit<br>● 0b11: Setting prohibited<br>**Notice**<br>In Mode 0, this bit must be set to 0b00. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 13 | PEIE | Parity error interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 12:11 | SCLKSEL | Transmission clock SCLK source selection<br>● 0b00: PCLK<br>● 0b01: PCLK<br>● 0b10: XTL<br>● 0b11: RCL | RW |
| 10:9 | OVER | Oversampling control<br>● Mode0: Invalid<br>● Mode1/3:<br>▶ 0b00: 16 sampling frequency division<br>▶ 0b01: 8 sampling frequency division<br>▶ 0b10: 4 sampling frequency division<br>▶ 0b11: Setting prohibited<br>● Mode2:<br>▶ 0b00: 32 sampling frequency division<br>▶ 0b01: 16 sampling frequency division<br>▶ 0b10: 8 sampling frequency division<br>▶ 0b11: Setting prohibited | RW |
| 8 | TXEIE | TX Buf empty interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 7:6 | SM | Working mode<br>● 0b00: Mode0, synchronous half-duplex, Data (8bits)<br>● 0b01: Mode1, asynchronous full-duplex, Start (1bit) + Data (8bits) + Stop (1-2bits)<br>● 0b10: Mode2, asynchronous full-duplex, Start (1bit) + Data (8bits) + B8 (1bit) + Stop (1-2bits)<br>● 0b11: Mode3, asynchronous full-duplex, Start (1bit) + Data (8bits) + B8 (1bit) + Stop (1-2bits) | RW |
| 5 | ADRDET | Multi-machine address automatic recognition enable<br>● 0b0: Disabled<br>● 0b1: Enabled | RW |
| 4 | REN | Receive enable<br>● Mode0:<br>▶ 0b0: Transmit<br>▶ 0b1: Receivie<br>● Mode1/2/3:<br>▶ 0b0: Transmit<br>▶ 0b1: Receive/Transmit<br>**Note**<br>When this bit is set to 1, modifying receive data control bits is disabled. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 3:2 | B8CONT | DATA8 data bit function configuration<br>● 0b00: Deciding by software read/write LPUART_SBUF.DATA8<br>● 0b01: Hardware even parity check<br>● 0b10: Hardware odd parity check<br>● 0b11: Setting prohibited | RW |
| 1 | TCIE | Transmit complete interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 0 | RCIE | Receive complete interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 17.5.4 Address Register (LPUART_SADDR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x08 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | SADDR | | | | | | | |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | RW | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:8* | *Reserved* | - | - |
| 7:0 | SADDR | Slave device address register | RW |

### 17.5.5 Address Mask Register (LPUART_SADEN)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x0C | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | SADEN | | | | | | | |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | RW | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:8* | *Reserved* | - | - |
| 7:0 | SADEN | Slave device address mask register<br>Each bit corresponds to an address bit:<br>● 0b0: Does not participate in address matching<br>● 0b1: Participates in address matching | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 17.5.6 Interrupt Flag Bit Register (LPUART_ISR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x10 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000008 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | CTS | CTSIF | PE | TXE | FE | TC | RC |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | RO | RO | RO | RO | RO | RO | RO |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:7* | *Reserved* | - | - |
| 6 | CTS | CTS signal flag<br>Can only be cleared through hardware<br>● 0b0: CTS signal is low level<br>● 0b1: CTS signal is high level | RO |
| 5 | CTSIF | CTS signal flip flag<br>● 0b0: CTS signal does not flip<br>● 0b1: CTS signal fliped | RO |
| 4 | PE | Parity check error flag<br>● 0b0: No parity errors occurred<br>● 0b1: A parity error occurred | RO |
| 3 | TXE | Tx Buffer empty flag<br>Can only be cleared through hardware<br>● 0b0: Tx Buffer is not empty<br>● 0b1: Tx Buffer is empty | RO |
| 2 | FE | Frame error flag<br>● 0b0: No frame error occurred<br>● 0b1: A frame error occurred | RO |
| 1 | TC | Transmit completion flag<br>● 0b0: Transmission not completed<br>● 0b1: Transmission completed | RO |
| 0 | RC | Receive completion flag<br>● 0b0: Reception not completed<br>● 0b1: Reception completed | RO |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 17.5.7 Interrupt Flag Bit Clear Register (LPUART_ICR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x14 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000037 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | CTSIFCF | PECF | Reserved | FECF | TCCF | RCCF |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | | WO | WO | | WO | WO | WO |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:6* | *Reserved* | - | - |
| 5 | CTSIFCF | ISR.CTSIF flag clear<br>Write 0 to clear flag, write 1 has no effect; read always returns 1. | WO |
| 4 | PECF | ISR.PE flag clear<br>Write 0 to clear flag, write 1 has no effect; read always returns 1. | WO |
| *3* | *Reserved* | - | - |
| 2 | FECF | ISR.FE flag clear<br>Write 0 to clear flag, write 1 has no effect; read always returns 1. | WO |
| 1 | TCCF | ISR.TC flag clear<br>Write 0 to clear flag, write 1 has no effect; read always returns 1. | WO |
| 0 | RCCF | ISR.RC flag clear<br>Write 0 to clear flag, write 1 has no effect; read always returns 1. | WO |

### 17.5.8 Baud Rate Register (LPUART_SCNT)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x18 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | SCNT | | | | | | | | | | | | | | | |
| Access | | | | | | | | | | | | | | | | | RW | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:16* | *Reserved* | - | - |
| 15:0 | SCNT | Baud rate counter | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 17.5.9 Baud Rate Generate Register (LPUART_BSEL)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x1C | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | BSEL | |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | RW | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:2* | *Reserved* | - | - |
| 1:0 | BSEL | Baud rate generation way<br>● 0b00-0b01: Generate baud rate using OVER and SCNT (the same as before)<br>● 0b10: When SCLK is selected as XTL (LPUART_SCON.SCLKSEL=0b10), generate a baud rate of 9600 using modulation method<br>● 0b11: When SCLK is selected as XTL (LPUART_SCON.SCLKSEL=0b10), generate a baud rate of 4800 using modulation method | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 17.5.10 9600/4800 Baud Rate Modulator Register (LPUART_MODU)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x20 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x0000054A | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | MODU | | | | | | | | | | | |
| Access | | | | | | | | | | | | | | | | | | | | | RW | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:12* | *Reserved* | - | - |
| 11:0 | MODU | 9600/4800 baud rate modulator<br>When SCLK is selected as XTL (LPUART_SCON.SCLKSEL=0b10) and BSEL.BSEL = 0b10/0b11, this register is valid:<br>● When BSEL.BSEL = 0b10, the modulation settings required for a 9600 baud rate are generated, and the value to be set in this register should be 0x54A.<br>● When BSEL.BSEL = 0b11, the modulation settings required for a 4800 baud rate are generated, and the value to be set in this register should be 0xEFB. | RW |

#### Principle of Generating 9600 Baud Rate

When SCLK is selected as XTL (LPUART_SCON.SCLKSEL=0b10), and the SCLK frequency is 32768. The baud rate of LPUART is 9600 (BSEL.BSEL = 0b10). Each bit requires 32768/9600 = 3.413 SCLKs. If each bit uses 3 SCLKs, the error of each bit is -0.413 SCLKs. If each bit uses 4 SCLKs, the error of each bit is +0.587 SCLKs.

| Bit | Name | Number of SCLK | LPUART_MODU | Final Inaccuracy (Number of SCLK) |
|---|---|---|---|---|
| 0 | Start | 3 | 0 | -0.413 |
| 1 | DATA[0] | 4 | 1 | +0.174(=-0.413+0.587) |
| 2 | DATA[1] | 3 | 0 | -0.239(=+0.174-0.413) |
| 3 | DATA[2] | 4 | 1 | +0.348(=-0.239+0.587) |
| 4 | DATA[3] | 3 | 0 | -0.065(=+0.348-0.413) |
| 5 | DATA[4] | 3 | 0 | -0.478(=-0.065-0.413) |
| 6 | DATA[5] | 4 | 1 | +0.109(=-0.478+0.587) |
| 7 | DATA[6] | 3 | 0 | -0.304(=+0.109-0.413) |
| 8 | DATA[7] | 4 | 1 | +0.283(=-0.304+0.587) |
| 9 | Parity/stop1 | 3 | 0 | -0.130(=+0.283-0.413) |
| 10 | Stop1/stop2 | 4 | 1 | +0.457(=-0.13+0.587) |
| 11 | Stop2 | 3 | 0 | +0.044(=+0.457-0.413) |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### Principle of Generating 4800 Baud Rate

When SCLK is selected as XTL (LPUART_SCON.SCLKSEL=0b10), and the SCLK frequency is 32768. The baud rate of LPUART is 4800 (BSEL.BSEL = 0b11). Each bit requires 32768/4800 = 6.827 SCLKs. If each bit uses 6 SCLKs, the error of each bit is -0.827 SCLKs. If each bit uses 7 SCLKs, the error of each bit is +0.173 SCLKs.

| Bit | Name | Number of SCLK | LPUART_MODU | Final Inaccuracy (Number of SCLK) |
|---|---|---|---|---|
| 0 | Start | 7 | 1 | +0.173 |
| 1 | DATA[0] | 7 | 1 | +0.346(=+0.173+0.173) |
| 2 | DATA[1] | 6 | 0 | -0.481(=+0.346-0.827) |
| 3 | DATA[2] | 7 | 1 | -0.308(=-0.481+0.173) |
| 4 | DATA[3] | 7 | 1 | -0.135(=-0.308+0.173) |
| 5 | DATA[4] | 7 | 1 | +0.038(=-0.135+0.173) |
| 6 | DATA[5] | 7 | 1 | +0.211(=+0.038+0.173) |
| 7 | DATA[6] | 7 | 1 | +0.384(=+0.211+0.173) |
| 8 | DATA[7] | 6 | 0 | -0.443(=+0.384-0.827) |
| 9 | Parity/stop1 | 7 | 1 | -0.270(=-0.443+0.173) |
| 10 | Stop1/stop2 | 7 | 1 | -0.097(=-0.270+0.173) |
| 11 | Stop2 | 7 | 1 | +0.076(=-0.097+0.173) |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

# 18 USART

## 18.1 Introduction

This product is equipped with 2 Universal serial Synchronous Asynchronous Transmitter modules (USART), which can flexibly exchange data with external devices.

The module supports universal asynchronous serial communication, universal synchronous serial communication, smart card protocol, infrared communication protocol and LIN protocol.

## 18.2 Features

- Two baud rate generation methods: integer frequency division, fractional frequency division
- Two data sequence order: MSB first, LSB first
- Built-in multifunctional timer/counter, supporting funtions such as receive timeout.
- Support TXD, RXD pin swap
- Support TXD, RXD level inversion
- Support DMA continuous communication

### 18.2.1 Asynchronous Serial Port Features

- Transfer data asynchronously
- 2 frame lengths: 8 bits, 9 bits
- Frame check: odd check, even check, no check
- 2 stop bits: 1 bit, 2 bits
- Support hardware flow control (CTS/RTS)
- Support single-wire half-duplex communication
- Support baud rate automatic detection
- Support for multiprocessor communication

### 18.2.2 Synchronous Serial Port Features

- Transfer data synchronously
- The data length is fixed at 8 bits
- Support hardware flow control (CTS/RTS)
- 2 communication clock sources: internal clock source, external clock source

### 18.2.3 Smart Card Interface Features

- Communication protocol: ISO/IEC7816-3
- 8 ETU lengths: 16-512
- Support hardware error detection and retransmission
- Support master interface

### 18.2.4 Infrared Communication Features

- Support 3/16 no carrier mode
- Support 38K carrier modulation mode, adjustable duty cycle, configurable modulation level


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 18.2.5 LIN Features

- Support synchronous interval transmission and detection
- Support synchronous segment transmission and baud rate automatic detection

## 18.3 Functional Description

Details of the asynchronous serial port mode, synchronous serial port mode, smart card mode, infrared mode and LIN mode are described in this section.

Asynchronous serial port, synchronous serial port, smart card, infrared and LIN functions can only activate one of them at a time.

**Note**

This module supports output signal inversion control (USART_CR2.TXINV) and input signal inversion control (USART_CR2.RXINV). Unless otherwise specified, the figures, tables and texts involved in this section are for output inversion control and input inversion control turned off.

### 18.3.1 USART General Function

#### 18.3.1.1 Communication Baud Rate Programming

**Note**

Communication baud rate programming is applicable to asynchronous serial port mode, synchronous serial port mode, smart card mode, infrared mode and LIN mode.

This module supports two baud rate generation methods: integer baud rate generation method and fractional baud rate generation method.

The formulas for the two baud rate generation methods are as follows:

$$Baud_{Integer} = \frac{PCLK}{4^{PSC} \times Sample\ count \times (DIVI+1)}$$

$$Baud_{Fractional} = \frac{PCLK \times (128 + DIVF)}{4^{PSC} \times Sample\ count \times (DIVI+1) \times 256}$$

**Table 18-1 Each Mode Sample Count and Maximum Baud Rate**

| Functional Mode | Sample Count | Maximum Baud Rate (Unit: bps) |
|---|---|---|
| Asynchro-nous serial port | Sample count=8*(2-OVER8) | PCLK/8 |
| Synchronous serial port | Sample count=4 | When internal clock source: PCLK/8<br>When external clock source: PCLK/6<br>(The maximum frequency requirement of the external input clock is PCLK/6) |
| Smart card interface | Sample count=2*BCN | PCLK/32 |
| Infrared communica-tion | Sample count=8*2 | Determined by infrared modulation mode |
| LIN | Sample count=8*(2-OVER8) | 20k |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Envolves:

- Baud: baud rate
- PCLK: clock source
- PSC: USART_BRR.PSC setting value
- OVER8: USART_CR1.OVER8 setting value
- BCN: USART_CR3.BCN setting value
- DIVI: USART_BRR.DIVI setting value
- DIVF: USART_BRR.DIVF setting value

Knowing the communication baud rate, the steps to calculate the USART_BRR register value are as follows:

**Step1.** Configure USART_BRR.PSC so that the calculation result of the following expression is less than or equal to 256.

$$\frac{PCLK}{4^{PSC} \times Baud \times Sample\ count}$$

**Step2.** Calculate the value of USART_BRR.DIVI according to the formula below, where the INT function represents rounding down.

$$DIVI = INT\left(\frac{PCLK}{4^{PSC} \times Baud \times Sample\ count} - 1\right)$$

**Step3.** Calculate the value of USART_BRR.DIVF according to the formula below, where the INT function represents rounding down.

$$DIVF = INT\left(\frac{Baud \times Sample\ count \times (DIVI + 1) \times 256 \times 4^{PSC} \times 1.0}{PCLK} - 128 + 0.5\right)$$

**Step4.** When DIVF is less than 127.5, set USART_BRR.FBME to 1 to enable the fractional baud rate generator;

When DIVF is greater than 127.5, set USART_BRR.FBME to 0 to disable the fractional baud rate generator.

## Asynchronous Serial Communication Baud Rate Programming Example

**Table 18-2 Baud Rate Programming Example with PCLK at 4MHz**

| Target Baud Rate | PCLK=4MHz | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | OVER8=1 | | | | | OVER8=0 | | | | |
| | PSC | DIVI | DIVF | Actual Baud Rate | Error% | PSC | DIVI | DIVF | Actual Baud Rate | Error% |
| 2400 | 0 | 207 | NA | 2404 | 0.16% | 0 | 103 | NA | 2404 | 0.16% |
| 4800 | 0 | 103 | NA | 4808 | 0.16% | 0 | 51 | NA | 4808 | 0.16% |
| 9600 | 0 | 51 | NA | 9615 | 0.16% | 0 | 25 | NA | 9615 | 0.16% |
| 19200 | 0 | 25 | NA | 19231 | 0.16% | 0 | 12 | NA | 19231 | 0.16% |
| 76800 | 0 | 5 | 108 | 76823 | 0.03% | 0 | 2 | 108 | 76823 | 0.03% |
| 115200 | 0 | 3 | 108 | 115234 | 0.03% | 0 | 1 | 108 | 115234 | 0.03% |
| 230400 | 0 | 1 | 108 | 230469 | 0.03% | 0 | 0 | 108 | 230469 | 0.03% |
| 250000 | 0 | 1 | NA | 250000 | 0.00% | 0 | 0 | NA | 250000 | 0.00% |
| 500000 | 0 | 0 | NA | 500000 | 0.00% | | | | | |

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Table 18-3 Baud Rate Programming Example with PCLK at 8MHz

| Target Baud Rate | PCLK=8MHz | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | OVER8=1 | | | | | OVER8=0 | | | | |
| | PSC | DIVI | DIVF | Actual Baud Rate | Error% | PSC | DIVI | DIVF | Actual Baud Rate | Error% |
| 2400 | 1 | 103 | NA | 2404 | 0.16% | 0 | 207 | NA | 2404 | 0.16% |
| 4800 | 0 | 207 | NA | 4808 | 0.16% | 0 | 103 | NA | 4808 | 0.16% |
| 9600 | 0 | 103 | NA | 9615 | 0.16% | 0 | 51 | NA | 9615 | 0.16% |
| 19200 | 0 | 51 | NA | 19231 | 0.16% | 0 | 25 | NA | 19231 | 0.16% |
| 76800 | 0 | 12 | NA | 76923 | 0.16% | 0 | 5 | 108 | 76823 | 0.03% |
| 115200 | 0 | 7 | 108 | 115234 | 0.03% | 0 | 3 | 108 | 115234 | 0.03% |
| 230400 | 0 | 3 | 108 | 230469 | 0.03% | 0 | 1 | 108 | 230469 | 0.03% |
| 250000 | 0 | 3 | NA | 250000 | 0.00% | 0 | 1 | NA | 250000 | 0.00% |
| 500000 | 0 | 1 | NA | 500000 | 0.00% | 0 | 0 | NA | 500000 | 0.00% |
| 750000 | 0 | 0 | 64 | 750000 | 0.00% | | | | | |
| 1000000 | 0 | 0 | NA | 1000000 | 0.00% | | | | | |

Table 18-4 Baud Rate Programming Example with PCLK at 16MHz

| Target Baud Rate | PCLK=16MHz | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | OVER8=1 | | | | | OVER8=0 | | | | |
| | PSC | DIVI | DIVF | Actual Baud Rate | Error% | PSC | DIVI | DIVF | Actual Baud Rate | Error% |
| 2400 | 2 | 51 | NA | 2404 | 0.16% | 1 | 103 | NA | 2404 | 0.16% |
| 4800 | 1 | 103 | NA | 4808 | 0.16% | 0 | 207 | NA | 4808 | 0.16% |
| 9600 | 0 | 207 | NA | 9615 | 0.16% | 0 | 103 | NA | 9615 | 0.16% |
| 19200 | 0 | 103 | NA | 19231 | 0.16% | 0 | 51 | NA | 19231 | 0.16% |
| 76800 | 0 | 25 | NA | 76923 | 0.16% | 0 | 12 | NA | 76923 | 0.16% |
| 115200 | 0 | 16 | 123 | 115349 | 0.13% | 0 | 7 | 108 | 115234 | 0.03% |
| 230400 | 0 | 7 | 108 | 230469 | 0.03% | 0 | 3 | 108 | 230469 | 0.03% |
| 250000 | 0 | 7 | NA | 250000 | 0.00% | 0 | 3 | NA | 250000 | 0.00% |
| 500000 | 0 | 3 | NA | 500000 | 0.00% | 0 | 1 | NA | 500000 | 0.00% |
| 750000 | 0 | 1 | 64 | 750000 | 0.00% | 0 | 0 | 64 | 750000 | 0.00% |
| 1000000 | 0 | 1 | NA | 1000000 | 0.00% | 0 | 0 | NA | 1000000 | 0.00% |
| 1500000 | 0 | 0 | 64 | 1500000 | 0.00% | | | | | |
| 2000000 | 0 | 0 | NA | 2000000 | 0.00% | | | | | |

Table 18-5 Baud Rate Programming Example with PCLK at 24MHz

| Target Baud Rate | PCLK=24MHz | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | OVER8=1 | | | | | OVER8=0 | | | | |
| | PSC | DIVI | DIVF | Actual Baud Rate | Error% | PSC | DIVI | DIVF | Actual Baud Rate | Error% |
| 2400 | 2 | 77 | NA | 2404 | 0.16% | 2 | 38 | NA | 2404 | 0.16% |
| 4800 | 2 | 38 | NA | 4808 | 0.16% | 1 | 77 | NA | 4808 | 0.16% |
| 9600 | 1 | 77 | NA | 9615 | 0.16% | 0 | 155 | NA | 9615 | 0.16% |
| 19200 | 0 | 155 | NA | 19231 | 0.16% | 0 | 77 | NA | 19231 | 0.16% |
| 76800 | 0 | 38 | NA | 76923 | 0.16% | 0 | 18 | 121 | 76789 | -0.01% |
| 115200 | 0 | 25 | NA | 115385 | 0.16% | 0 | 12 | NA | 115385 | 0.16% |
| 230400 | 0 | 12 | NA | 230769 | 0.16% | 0 | 5 | 108 | 230469 | 0.03% |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Target Baud Rate | PCLK=24MHz | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | OVER8=1 | | | | | OVER8=0 | | | | |
| | PSC | DIVI | DIVF | Actual Baud Rate | Error% | PSC | DIVI | DIVF | Actual Baud Rate | Error% |
| 250000 | 0 | 11 | NA | 250000 | 0.00% | 0 | 5 | NA | 250000 | 0.00% |
| 500000 | 0 | 5 | NA | 500000 | 0.00% | 0 | 2 | NA | 500000 | 0.00% |
| 750000 | 0 | 3 | NA | 750000 | 0.00% | 0 | 1 | NA | 750000 | 0.00% |
| 1000000 | 0 | 2 | NA | 1000000 | 0.00% | 0 | 0 | 43 | 1001953 | 0.20% |
| 1500000 | 0 | 1 | NA | 1500000 | 0.00% | 0 | 0 | NA | 1500000 | 0.00% |
| 2000000 | 0 | 0 | 43 | 2003906 | 0.20% | | | | | |
| 2500000 | 0 | 0 | 85 | 2496094 | -0.16% | | | | | |
| 3000000 | 0 | 0 | NA | 3000000 | 0.00% | | | | | |

Table 18-6 Baud Rate Programming Example with PCLK at 32MHz

| Target Baud Rate | PCLK=32MHz | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | OVER8=1 | | | | | OVER8=0 | | | | |
| | PSC | DIVI | DIVF | Actual Baud Rate | Error% | PSC | DIVI | DIVF | Actual Baud Rate | Error% |
| 2400 | 2 | 103 | NA | 2404 | 0.16% | 2 | 51 | NA | 2404 | 0.16% |
| 4800 | 2 | 51 | NA | 4808 | 0.16% | 1 | 103 | NA | 4808 | 0.16% |
| 9600 | 1 | 103 | NA | 9615 | 0.16% | 0 | 207 | NA | 9615 | 0.16% |
| 19200 | 0 | 207 | NA | 19231 | 0.16% | 0 | 103 | NA | 19231 | 0.16% |
| 76800 | 0 | 51 | NA | 76923 | 0.16% | 0 | 25 | NA | 76923 | 0.16% |
| 115200 | 0 | 33 | 123 | 115349 | 0.13% | 0 | 16 | 123 | 115349 | 0.13% |
| 230400 | 0 | 16 | 123 | 230699 | 0.13% | 0 | 7 | 108 | 230469 | 0.03% |
| 250000 | 0 | 15 | NA | 250000 | 0.00% | 0 | 7 | NA | 250000 | 0.00% |
| 500000 | 0 | 7 | NA | 500000 | 0.00% | 0 | 3 | NA | 500000 | 0.00% |
| 750000 | 0 | 4 | 112 | 750000 | 0.00% | 0 | 1 | 64 | 750000 | 0.00% |
| 1000000 | 0 | 3 | NA | 1000000 | 0.00% | 0 | 1 | NA | 1000000 | 0.00% |
| 1500000 | 0 | 1 | 64 | 1500000 | 0.00% | 0 | 0 | 64 | 1500000 | 0.00% |
| 2000000 | 0 | 1 | NA | 2000000 | 0.00% | 0 | 0 | NA | 2000000 | 0.00% |
| 2500000 | 0 | 0 | 32 | 2500000 | 0.00% | | | | | |
| 3000000 | 0 | 0 | 64 | 3000000 | 0.00% | | | | | |
| 4000000 | 0 | 0 | NA | 4000000 | 0.00% | | | | | |

Table 18-7 Baud Rate Programming Example with PCLK at 48MHz

| Target Baud Rate | PCLK=48MHz | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | OVER8=1 | | | | | OVER8=0 | | | | |
| | PSC | DIVI | DIVF | Actual Baud Rate | Error% | PSC | DIVI | DIVF | Actual Baud Rate | Error% |
| 2400 | 2 | 155 | NA | 2404 | 0.16% | 2 | 77 | NA | 2404 | 0.16% |
| 4800 | 2 | 77 | NA | 4808 | 0.16% | 2 | 38 | NA | 4808 | 0.16% |
| 9600 | 2 | 38 | NA | 9615 | 0.16% | 1 | 77 | NA | 9615 | 0.16% |
| 19200 | 1 | 77 | NA | 19231 | 0.16% | 0 | 155 | NA | 19231 | 0.16% |
| 76800 | 0 | 77 | NA | 76923 | 0.16% | 0 | 38 | NA | 76923 | 0.16% |
| 115200 | 0 | 51 | NA | 115385 | 0.16% | 0 | 25 | NA | 115385 | 0.16% |
| 230400 | 0 | 25 | NA | 230769 | 0.16% | 0 | 12 | NA | 230769 | 0.16% |
| 250000 | 0 | 23 | NA | 250000 | 0.00% | 0 | 11 | NA | 250000 | 0.00% |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Target Baud Rate | PCLK=48MHz | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | OVER8=1 | | | | | OVER8=0 | | | | |
| | PSC | DIVI | DIVF | Actual Baud Rate | Error% | PSC | DIVI | DIVF | Actual Baud Rate | Error% |
| 500000 | 0 | 11 | NA | 500000 | 0.00% | 0 | 5 | NA | 500000 | 0.00% |
| 750000 | 0 | 7 | NA | 750000 | 0.00% | 0 | 3 | NA | 750000 | 0.00% |
| 1000000 | 0 | 5 | NA | 1000000 | 0.00% | 0 | 2 | NA | 1000000 | 0.00% |
| 1500000 | 0 | 3 | NA | 1500000 | 0.00% | 0 | 1 | NA | 1500000 | 0.00% |
| 2000000 | 0 | 2 | NA | 2000000 | 0.00% | 0 | 0 | 43 | 2003906 | 0.20% |
| 2500000 | 0 | 1 | 85 | 2496094 | -0.16% | 0 | 0 | 85 | 2496094 | -0.16% |
| 3000000 | 0 | 1 | NA | 3000000 | 0.00% | 0 | 0 | NA | 3000000 | 0.00% |
| 4000000 | 0 | 0 | 43 | 4007812.5 | 0.20% | | | | | |
| 6000000 | 0 | 0 | NA | 6000000 | 0.00% | | | | | |

#### 18.3.1.2 Automatic Baud Rate Detection

**Note**

Communication baud rate detection is applicable to asynchronous serial port mode and LIN mode.

This function needs to be combined with a multi-function timing counter to realize automatic baud rate detection.

By setting USART_CR1.TOE=5 and USART_CR3.ABRE=1, the USART module enters the automatic baud rate detection mode. The hardware detects that there is an input data edge on RXD, the counter starts counting from 0, and the communication baud rate can be detected when the data frame of 0x55 is received. When the automatic baud rate detection is completed, stop counting, and the hardware automatically sets USART_SR.ABR to 1 and stores the baud rate detection value in the USART_BDR register. If USART_IER.ABR is set to 1, automatic baud rate detection can be generated interruption.

The communication baud rate can be calculated by the following formula:

$$Baud = \frac{f_{PCLK}}{4^{PSC} \times (BDR+1)}$$

Where:

- $f_{PCLK}$: PCLK frequency
- PSC: the value of USART_BRR.PSC
- BDR: the value of the USART_BDR register

When it is no longer necessary to detect the baud rate, set USART_CR3.ABRE to 0 to exit the automatic baud rate detection mode.

**Note**

- When the detected baud rate does not match the automatic baud rate detection range, the value of USART_BDR may overflow.
- If the counter overflows during the automatic baud rate detection process, the hardware automatically sets USART_SR.TO to 1, and if the corresponding interrupt is enabled, the multi-function timing counter interrupt can be generated.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

#### 18.3.1.3 DMA Continuous Communication

**Note**

- DMA continuous communication is applicable to asynchronous serial port mode, synchronous serial port mode, smart card mode, infrared mode and LIN mode.
- DMA mode is not supported when working in single-wire half-duplex.

The USART module supports the hardware handshaking logic of the DMAC.

- Write USART_CR2.TXDMAEN=1 to start the DMAC hardware handshake logic sent by USART. When the send buffer is empty, the USART will send a data transfer request to the DMAC. DMAC receives the sending request signal, it transfers the sending data of one frame from the DMAC source address to the USART_TDR. The above steps are repeated until all the data lengths configured in the DMAC are transferred.
- Write USART_CR2.RXDMAEN=1 to start the DMAC hardware handshake logic for USART reception. When a frame is received, the USART will send a data transfer request to the DMAC. the DMAC receives the receive request signal, it transfers the received data from the USART_RDR to the target address of the DMAC. The above steps are repeated until all the data lengths configured in the DMAC are transferred.

#### 18.3.1.4 Hardware Flow Control

**Note**

Hardware flow control is applicable to asynchronous serial port mode and synchronous serial port mode.

Hardware flow control refers to controlling the serial data flow between two USART devices through CTS and RTS signals, and the connection method is shown in the figure below.

The hardware flow control function is divided into two types, RTS flow control function and CTS flow control function.

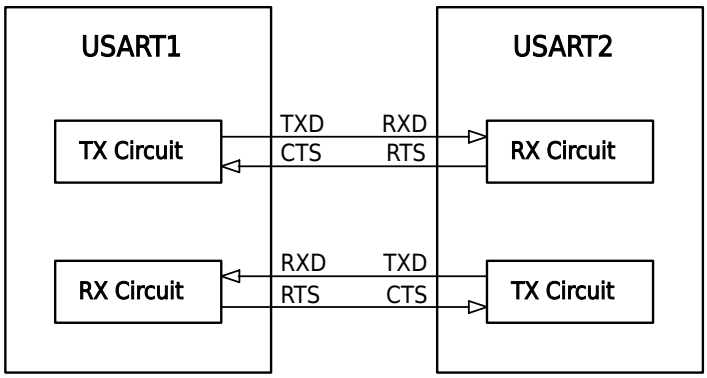



**Figure 18-1 Asynchronous Serial Port Mode Hardware Flow Control Example**

##### CTS Flow Control Function

When USART_CR2.CTSEN is set to 1, this module has CTS flow control function:

- When the CTS pin input is high level, the TXD pin will not send data out;
- When the CTS pin input is low level, the TXD pin sends out data.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

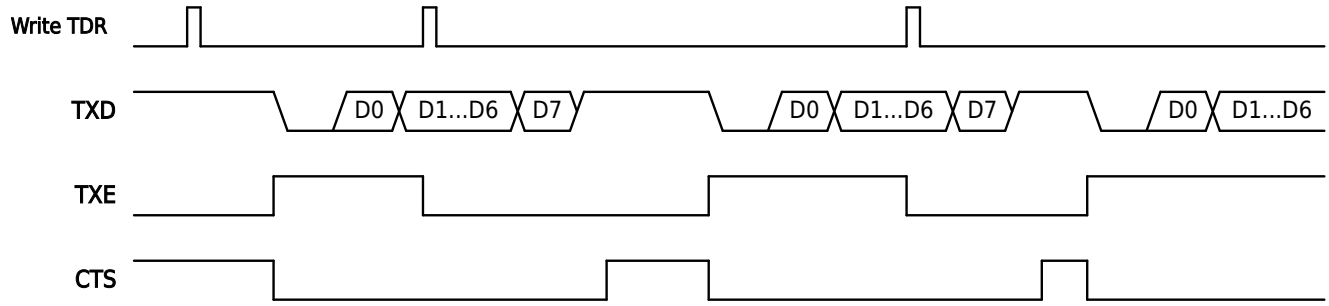



Figure 18-2 CTS Hardware Flow Control Signal

**Note**

The CTS pin becomes high level when data is being sent, the data frame being sent will not be affected, and after the current data frame is sent, the sending of new data will be suspended.

### RTS Flow Control Function

When USART_CR2.RTSEN is set to 1, this module has RTS flow control function:

- When the receive buffer is empty, the RTS pin outputs a low level;
- When the receive buffer is full, the RTS pin outputs a high level.

In asynchronous serial port mode, RTS pin will output low level only when all the following conditions are met:

1. USART_CR1.RE is 1, the receiving circuit is enabled.
2. USART_SR.RC is 0, the data in the USART_RDR register has been read.
3. USART_SR.PE, USART_SR.FE and USART_SR.OR are 0, no receive data check error, receive frame error and overflow error.

If all the above conditions cannot be met at the same time, the RTS pin will output a high level.

In synchronous serial port mode, RTS pin will output low level only when all the following conditions are met:

1. Receive enable (USART_CR1.RE=1) and not currently receiving data.
2. There is no unread data in the USART_RDR register (when USART_CR1.RE=1)
3. USART_TDR update completed (if USART_CR1.TE= 1).
4. USART_SR.OR is 0, no receive data overflow error.

If all the above conditions cannot be met at the same time, the RTS pin will output a high level.

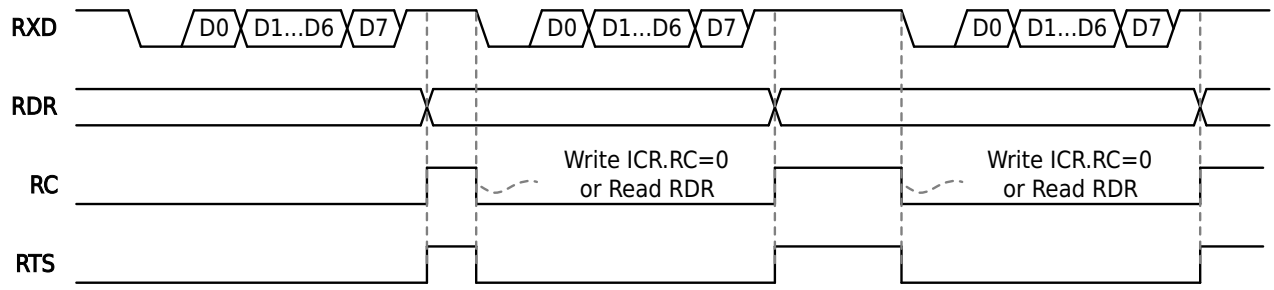



Figure 18-3 RTS Hardware Flow Control Signal

### 18.3.1.5 Multi-Function Timing Counter

This module is embedded with a multi-function timing counter, which can realize five timing and counting functions. Its timing counting cycle formula follows:


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

$t_{PCLK}$* $4^{PSC}$* (TOR + 1)

Envolves:

- $T_{PCLK}$: PCLK period
- PSC: the value of USART_BRR.PSC
- TOR: the value of the USART_TOR register

When the PCLK frequency is 48MHz, the minimum value of its timing counting range is 0.02μs, and the maximum value is about 1.4s.

**Note**

- Only when USART_CR1.TOE is 0, other values can be written to start the timing counter.
- When USART_SR.TO is 1 or the automatic baud rate detection is completed, the hardware automatically sets USART_CR1.TOE to 0.

Several functions are as shown in figure below:

**Table 18-8 Multi-Function Timing Counter Function**

| Function | Counter Start Condition | Application Scenario |
|---|---|---|
| Universal timing | Start immediately after writing 1 to USART_CR1.TOE | General timer |
| Receive timeout count | Start after writing 2 to USART_CR1.TOE and enabling reception | Wait for another master to reply with a message within a limited time |
| Receive interframe timeout count | Write 3 to USART_CR1.TOE and enabling reception, then start after receiving the first frame of data | Wait for variable-length packet transmission to complete |
| Synchronous interval measurements | Write 4 to USART_CR1.TOE, after USART_CR3.LBDE=1, the hardware detects that there is a falling edge on RXD | LIN mode |
| Automatic baud rate detection | Write 5 to USART_CR1.TOE, after USART_CR3.ABRE=1, the hardware detects that there is an input data edge on RXD | Baud rate detection (asynchronous serial port mode/LIN mode) |

### General Timer Function

When USART_CR1.TOE is 1, it works in general timing function. When 1 is written to USART_CR1.TOE, the counter starts immediately and starts counting from 0. If the count value is equal to USART_TOR, the hardware automatically sets USART_SR.TO to 1. If the corresponding interrupt is enabled, the multi-function timing counter interrupt can be generated.

### Receive Timeout Counting Function

When USART_CR1.TOE is 2, it works on receive timeout counting function. When 2 is written to USART_CR1.TOE and enabling reception, the counter starts immediately and starts counting from 0.
If the count value is equal to USART_TOR and RXD has not received new data, it stops counting and the hardware automatically sets USART_SR.TO to 1. If the corresponding interrupt can be interrupted, the multi-function timing counter interrupt can be generated.

If the count value is less than USART_TOR when RXD receives new data (USART_SR.RC=1), the counter is cleared and the counting function is started again, counting from 0.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### Receive Inter-Frame Timeout Counting Function

When USART_CR1.TOE is 3, it works on the timeout counting function between receiving frames. After writing 3 to USART_CR1.TOE and enabling reception, the counter starts counting from 0 after receiving the stop bit of the first byte. If the count value is equal to USART_TOR and RXD has not received new data, it stops counting and the hardware automatically set USART_SR.TO to 1, if the corresponding interrupt is enabled, the multi-function timing counter interrupt can be generated.

If the count value is less than USART_TOR when RXD receives new data (USART_SR.RC=1), the counter is cleared and the counting function is started again, counting from 0.

### Synchronous Interval Measurement Function

When USART_CR1.TOE is 4, it works in the synchronization interval measurement function. Write 4 to USART_CR1.TOE and write 1 to USART_CR2.LBDE, the hardware automatically recognizes the falling edge of the input signal RXD and counts the low level from 0, if the count value is equal to USART_TOR, then stop counting, and the hardware automatically sets USART_SR .LBD is 1, if the corresponding interrupt is enabled, the synchronization interval segment detection interrupt can be generated.

If the count value of the low level length count is less than USART_TOR, wait for the next falling edge of RXD to enter a new round of synchronization interval measurement.

### Automatic Baud Rate Detection Function

When USART_CR1.TOE is 5, it works in automatic baud rate detection function.

More information, see "Automatic Baud Rate Detection" section.

### 18.3.1.6 Interrupt

Refer to the correspondence between the interrupt source and the NVIC interrupt input described in the "Functional Description-Interrupt Source" section of the "NVIC". Each of the two USARTs in this section provides one type of interrupt: USARTx interrupt.

Table 18-9 Each Mode Interrupt Validity

| Function Name | Interrupt Enable Bit (USART_IER) | Asynchronous Serial Port | Synchronous Serial Port | Smart Card | Infrared | LIN |
|---|---|---|---|---|---|---|
| Transmit data empty interrupt | TXE | ✓ | ✓ | ✓ | ✓ | ✓ |
| Transmit complete interrupt | TC | ✓ | ✓(2) | ✓ | ✓ | ✓ |
| Receive complete interrupt | RC | ✓ | ✓ | ✓ | ✓ | ✓ |
| Receive data verification error interrupt | PE | ✓ | × | ✓ | ✓ | × |
| Receive frame error interrupt | FE | ✓ | × | ✓ | ✓ | ✓ |
| Receive data overflow error interrupt | OR | ✓ | ✓ | ✓ | ✓ | ✓ |
| Multi-function timing count interrupt | TO | ✓ | ✓ | ✓ | ✓ | ✓ |
| Address match interrupt in multi-device communication mode | AM | ✓ | × | × | × | × |
| CTS Interrupt | CTS | ✓ | ✓ | × | × | × |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Function Name | Interrupt Enable Bit (USART_IER) | Asynchronous Serial Port | Synchronous Serial Port | Smart Card | Infrared | LIN |
|---|---|---|---|---|---|---|
| Automatic baud rate detection interrupt | ABR | ✓ | × | × | × | ✓ |
| Sync interval detection interrupt | LBD | × | × | × | × | ✓ |
| Input data wake-up interrupt | DWK | ✓ | × | × | ✓ | ✓ |

**Note**

1. "✓" indicates valid; "×" indicates invalid.
2. In synchronous serial port mode, the transmit complete interrupt is not supported when using an external clock.

## 18.3.2 Asynchronous Serial Port Mode

### 18.3.2.1 Mode Overview

Asynchronous serial interface, which provides a flexible full-duplex data exchange asynchronous transmission, is usually also used as RS232 standard communication.

The interface is connected to other devices through 1 to 5 pins, and only two pins are usually used in applications: RXD and TXD.

- RXD: Receive data input, and distinguish data and noise through oversampling technology.
- TXD: Transmit data output, when not transmitting data, TXD pin keeps high level.

Asynchronous serial port mode is selected by setting USART_CR2.SYNC=0.

In asynchronous serial port mode, the following bits must be cleared:

- USART_CR3.SCEN, USART_CR3.LINEN and USART_CR3.IREN

### 18.3.2.2 Function Block Diagram

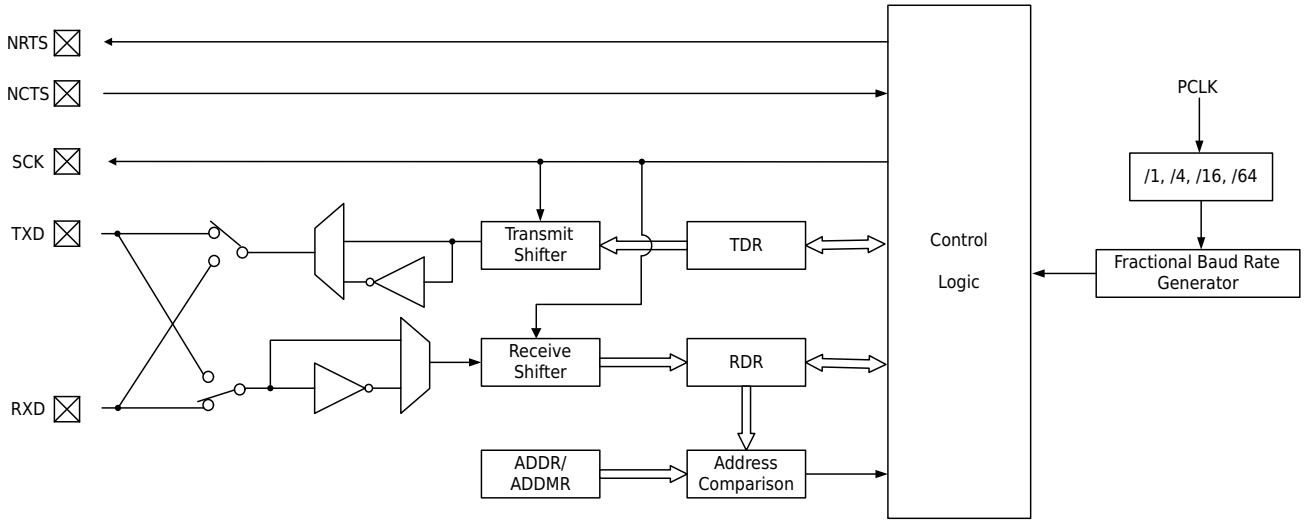



Figure 18-4 Asynchronous Serial Port Function Block Diagram

The function description of each pin is as follows:

- SCK: USART working clock output


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

- TXD: USART transmit data pins
- RXD: USART receive data pins
- NRTS: USART hardware flow control RTS output pin
- NCTS: USART hardware flow control CTS input pin

#### 18.3.2.3 Data Format

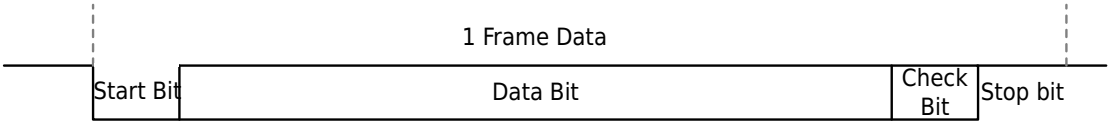



**Figure 18-5 Asynchronous Serial Port Mode Data Format**

The format of the USART serial data frame is: start bit, 8-9 data bits, parity bit (optional), 1-2 stop bits.

The polarity of the start bit is opposite to the data line idle state, and the stop bit is the same as the data line idle state.

Both start bit and stop bit are used for data synchronization during asynchronous data transfer.

The module supports two data bit orders: LSB first or MSB first.

When sending, the USART module reads the data in the sending buffer, and then outputs it from the TXD pin in the above data frame format.

When receiving, the USART module searches for the start bit on the RXD pin, and receives the data bit, parity bit, and stop bit according to the above data frame format after finding the start bit. If no stop bit is found, the USART module considers the transmission of the entire data frame to have failed.

This module supports two ways to find the start bit: the falling edge on the RXD pin or the low level on the RXD pin (when USART_CR2.RXINV=0).

#### 18.3.2.4 Transmitter

When setting USART_CR1.TE to 1, the TXD pin can send data externally. When there is data in the USART_TDR register, the hardware automatically loads the data into the transmit shift register, and shifts and outputs it from the TXD pin according to the data frame format configured by the user. The USART_SR.TXE flag is set to 1 by hardware whenever data in the USART_TDR register is loaded into the shift register. The user finds that the USART_SR.TXE flag is 1, the user can write the next data to be sent to the USART_TDR register. Whenever the TXD pin sends a frame of data, the USART_SR.TC flag will be set to 1 by hardware to indicate the current sending progress.

The figure below is an example of sending two bytes; the data length is eight bits, no parity, and one STOP bit.

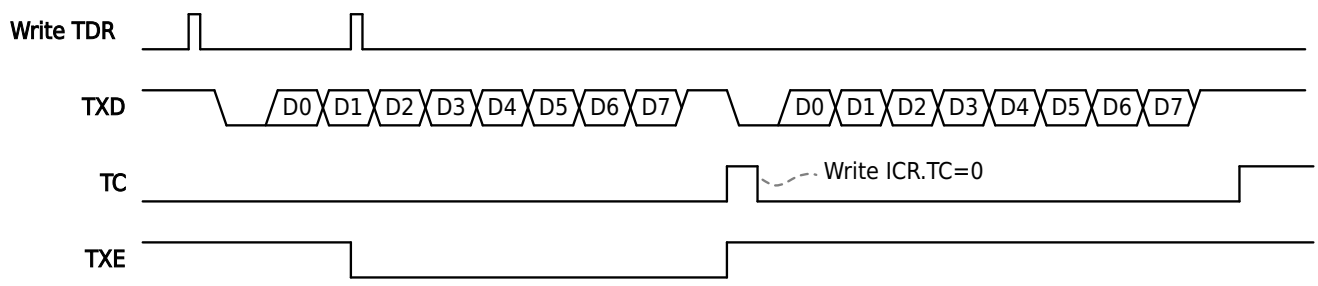



**Figure 18-6 Transmit Data Example**


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

**Note**

- Before writing data to be sent into USART_TDR, the USART_CR1.TE bit must be set to 1 first.
- Do not write USART_CR1.TE=0 during data transmission, otherwise the baud rate counter will be frozen, causing the TXD pin to output unexpected waveforms.

#### 18.3.2.5 Receiver

When setting USART_CR1.RE to 1, this module can receive the data received from the RXD pin. Whenever a frame of data is received, the USART_SR.RC flag will be set to 1 by hardware, and the received data can be read from the USART_RDR register.

This module supports two start bit detection methods: detecting the level on the RXD pin as the start bit, and detecting the edge on the RXD pin as the start bit, see USART_CR1.SBS description for details.

This module supports two sampling methods: 8 times sampling and 16 times sampling. Eight times sampling means that the signal on the RXD pin is sampled 8 times per baud rate cycle; 16 times sampling means that the signal on the RXD pin is sampled 16 times per baud rate cycle.

When the module receives data, it can detect three kinds of errors: overflow error, parity error, frame error. After any kind of error occurs, the module continues to receive data, and the user program should handle the error as soon as possible, and clear the error flag to restart data reception.

- Overflow Error: When USART_SR.RC is 1 (the data in USART_RDR is not read) and a new frame of data is received, an overflow error occurs. When this error occurs, USART_SR.OR will be set to 1 by hardware, and the newly received data will overwrite the previous frame data Write 0 to USART_ICR.OR to clear this flag. If the data in USART_RDR is not read after USART_SR.RC is 1, but USART_SR.RC is cleared to 0 by writing USART_ICR.RC = 0, then USART_SR.OR will not be set to 1.
- Parity Error: A parity error occurs when an incorrect parity of a data frame is detected. When this error occurs, USART_SR.PE will be set to 1 by hardware, and the received data will be sent to USART_RDR. Write 0 to USART_ICR.PE to clear this flag.
- Frame Error: A frame error occurs when a received data frame does not detect a stop bit at the expected time. When this frame error occurs, USART_SR.FE will be set to 1 by hardware, and the received data will be sent to USART_RDR. Write 0 to USART_ICR.FE to clear this flag.

The following figure is an example of receiving two bytes: the data length is eight-bit, no parity, and one STOP bit.

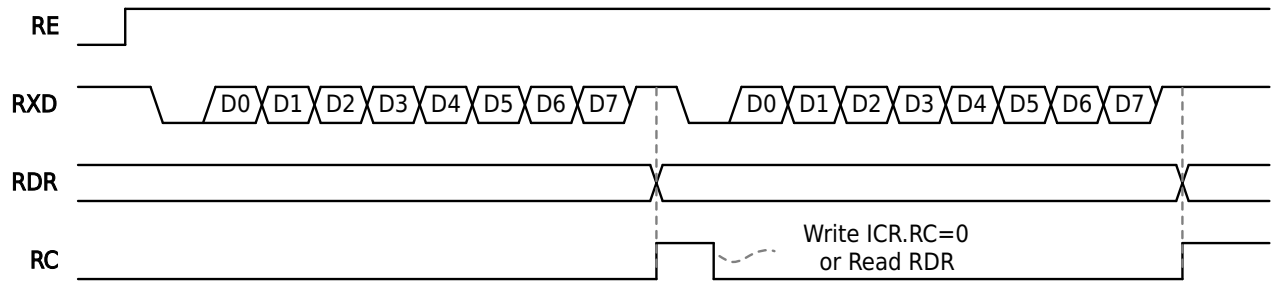



Figure 18-7 Receive Data Example

#### 18.3.2.6 Single-Wire Half-Duplex

When setting USART_CR1.COMB and USART_CR1.RE to 1 and USART_CR1.TE to 0, this module works in single-wire half-duplex mode, and only uses the TXD pin for data transmission and reception.

When no data is sent, the TXD pin is in the input state, waiting for other masters on the receiving bus to send data. Whenever the user writes data to the USART_TDR register, the TXD pin automatically


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

changes to the output state to send the data written by the user to the USART_TDR internal register to the bus.

To prevent collisions on a single bus, the user should take appropriate application-layer protection measures to ensure that multiple masters do not send data to the bus at the same time.

**Note**

DMA mode is not supported when working in single-wire half-duplex.

#### 18.3.2.7 Multiprocessor Communication

**Note**

- Multiprocessor communication is only applicable to asynchronous serial port mode.
- During multiprocessor communication, the USART module acting as a slave needs to configure TXD as open-drain output. After the slave finishes transmitting data, it needs to promptly set USART_CR1.TE to 0, and then communicate between the master and other slaves.

When setting USART_CR1.MPE to 1, USART works in multiprocessor communication mode. The master can send address frames and data frames, and the slave can receive data frames only after receiving matching address frames.

##### Data Format

**Figure 18-8 Multiprocessor Communication Data Format**

- **Start Bit (ST):** Composed of one bit at low level.
- **Data Bit (D):** Can be configured as eight bits or nine bits.
- **Multiprocessor Bit (MPB):**Can be configured to transmit communication data or send IDs.
- **Stop Bit (SP):** The stop bit is fixed at high level and can be configured as one bit or two bits.

##### Master Sends Address Frame and Data Frame

- If the word length is eight, so USART_CR1.CHLEN=0:
  - When USART_TDR.MPID is 1, the address frame is sent, and USART_TDR[7:0] is the address of the target slave.
  - When USART_TDR.MPID is 0, the data frame is sent, and USART_TDR[7:0] is the data to be sent.
- If the word length is nine, so USART_CR1.CHLEN=1:
  - When USART_TDR.MPID is 1, the address frame is sent, and USART_TDR[7:0] is the address of the target slave.
  - When USART_TDR.MPID is 0, the data frame is sent, and USART_TDR[8:0] is the data to be sent.

##### Slave Receives Address Frame and Data Frame

When USART_CR1.SLME is set to 1, the slave only receives address frames that match its own address, and ignores unmatched address frames and data frames. When the slave receives the address frame from RXD, the hardware automatically compares the address in the address frame with the local address; only when the two match, the received address will be stored in the USART_RDR register and USART_SR.RC and USART_SR.AM will be set..


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

#### Slave Address Configuration

The slave address consists of the user-configured address (USART_ADDR) and the address mask (USART_ADDMR).

Slave address: USART_ADDR & USART_ADDMR

When the address frame is received, it is detected that the address received from the bus matches the value of USART_ADDMR and the value is the same as the address configured by the user, then it is considered that the address frame matching the slave address has been received.

Example: USART_ADDR = 0x55, USART_ADDMR = 0xF0, then the address that the slave can match is 0x50-0x5F (user configuration address).

### 18.3.3 Synchronous Serial Port Mode

#### 18.3.3.1 Mode Overview

The synchronous serial port mode means that the sending and receiving of data is synchronized with the SCK Clock pin.

Synchronous serial port mode is selected by setting USART_CR2.SYNC=1.

In synchronous serial port mode, the following bits must be cleared:

- USART_CR1.COMB, USART_CR1.SLME, USART_CR1.MPE, USART_CR1.OVER8, USART_CR1.STOP, USART_CR1.CHLEN, USART_CR1.PCE
- USART_CR3.SCEN, USART_CR3.LINEN, USART_CR3.IREN

#### 18.3.3.2 Clock

There are two options for the clock source of the synchronous serial port mode, which can be set by USART_CR2.CLKC:

- Internal clock source: clock generated by internal baud rate generator

  The synchronous clock is output from the CK pin, and a frame of data outputs eight clock pulses. When neither sending data nor receiving data, the clock output is fixed at high level.
- External clock source: CK pin input clock

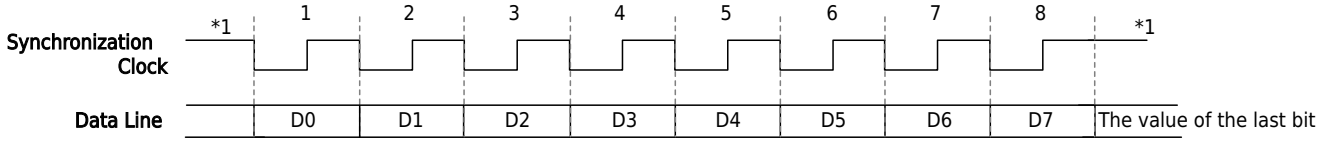
  The external clock source is the clock input from the CK pin as the communication clock.

#### 18.3.3.3 Data Format

In the synchronous serial port mode, a frame of data is fixedly composed of eight bits, and the sending and receiving of a frame of data requires eight synchronous clock pulses.

Data is sent at the falling edge of the synchronization clock when data is sent and sampled at the rising edge of the synchronization clock when data is received.

Except for the continuous transmission mode, the synchronous clock is fixed at high level when there is no data transmission; when there is data transmission, after the last bit of the sent data is sent, the data line keeps the value of the last bit.




*1: Unless in continuous transfer mode, the synchronous clock is fixed at high level

Figure 18-9 Synchronize Serial Port Mode Data Format


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

#### 18.3.3.4 Transmitter

When the transmit enable bit USART_CR1.TE=1, the data in the transmit shift register is output on the TXD pin, and the corresponding clock is output or input on the CK pin.

The transmit data register USART_TDR and the internal sending shift register form a double buffer structure, which can send data continuously.

The transmit data register USART_TDR can be completed through an interrupt program (CPU or DMA), and one interrupt can only write data once.

##### Transmit Data Setting Procedure (Use Internal Clock Source)

**Step1.** Set USART_CR1, USART_SR register to reset values;

**Step2.** Initialize the port. Configure TXD as a digital input with pull-up. Map TXD to the required pin; Map SCK to the required pin as the output clock;

**Step3.** Select internal clock source through the USART_CR2.CLKC bit;

**Step4.** Set USART_CR1 (except TE), USART_CR2, USART_CR3 registers;

**Step5.** Set USART_BRR.PSC to select the prescaler value, and set the USART_BRR communication baud rate;

**Step6.** Set USART_CR1.TE=1, USART_IER.TXE=1;

**Step7.** Write data to the USART_TDR through the interrupt routine. The data is transferred to the transmit shift register, and the transmission starts.

> **Note**
> When the CTS function is enabled, data is transferred to the transmit shift register and transmission starts when the CTS pin input is at low level.

**Step8.** If you need to send data continuously, repeat step7;

**Step9.** Write the last sent data through the interrupt program, write USART_IER.TXE=0, USART_IER.TC=1, after the last data is sent, a send completion interrupt occurs.

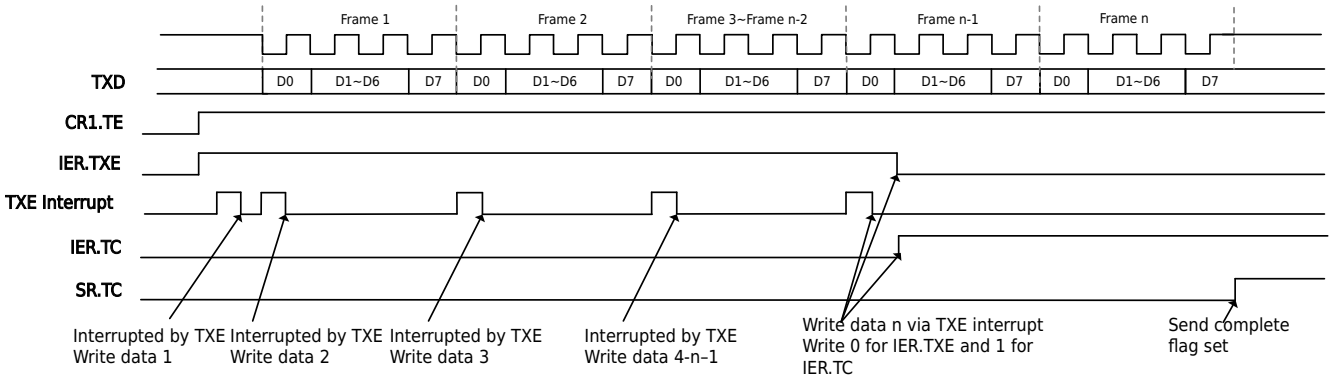



Figure 18-10 Synchronous Serial Port Mode Sending Data Legend 1 (LSB/No CTS Function)(Use Internal Clock Source)

##### Transmit Data Setting Procedure (Use External Clock Source)

**Step1.** Set USART_CR1, USART_SR register to reset values;

**Step2.** Initialize the port. Configure TXD as a digital input with pull-up. Map TXD to the required pin. Map SCK to the required pin as the input clock.

**Step3.** Select the external clock source through the USART_CR2.CLKC bit;

**Step4.** Set USART_CR1 (except TE), USART_CR2, USART_CR3 registers;

**Step5.** Set USART_CR1.TE=1, USART_IER.TXE=1;


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

**Step6.** Write data to the USART_TDR through the interrupt routine. The data is transferred to the transmit shift register, and the transmission starts.

> **Note**
> When the CTS function is enabled, data is transferred to the transmit shift register and transmission starts when the CTS pin input is at low level.

**Step7.** If you need to send data continuously, repeat step6;

**Step8.** Write the last transmit data through the interrupt routine, and then continue to wait for the TXE interrupt. In the next interrupt routine, write an arbitrary invalid data (if the most significant bit of the last data is 1, it is recommended to be 0xFF; if the most significant bit of the last data is 0, it is recommended to be 0x0). After the last data is transmitted, the occurrence of the TXE interrupt indicates the completion of transmission.

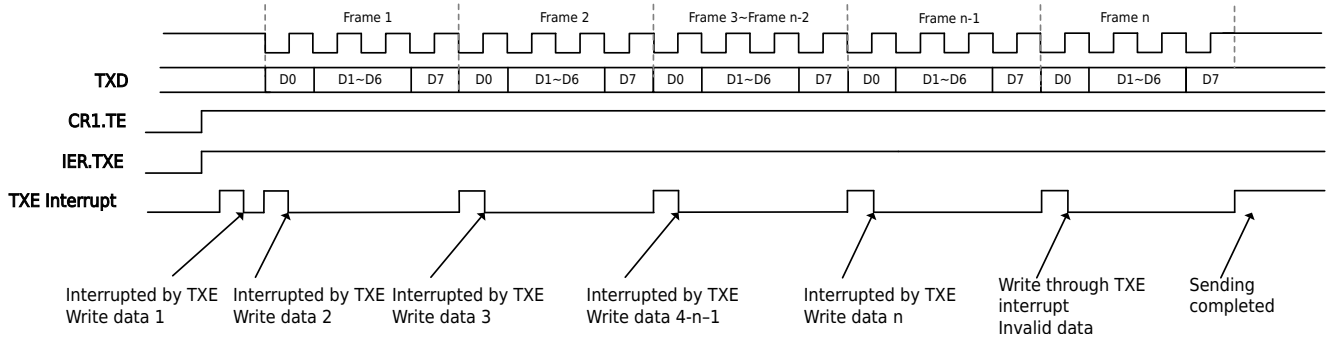



Figure 18-11 Synchronous Serial Port Mode Sending Data Legend 1 (LSB/No CTS Function) (Use External Clock Source)

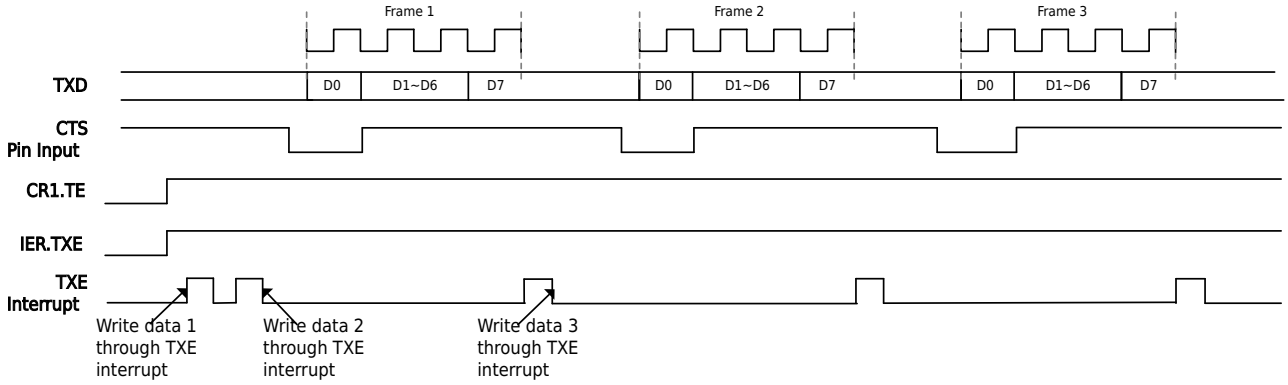



Figure 18-12 Synchronous Serial Port Mode to Send Data Legend 2 (LSB/CTS Function)

### Transmitter Interrupt

In the synchronous serial port mode, when using the internal clock source, transmitter supports two interrupts, the transmit data register empty interrupt and the transmit complete interrupt; When using the external clock source, the transmitter only supports the transmit data register empty interrupt.

USART_IER.TXE=1, an interrupt occurs when the value of the USART_TDR register is transferred to the transmit shift register.

When using the internal clock source, USART_IER.TC=1, when the last bit of data is sent, the USART_TDR register is not updated and an interrupt occurs.

## 18.3.3.5 Receiver

### Receive Data Setting Steps

**Step1.** Set USART_CR1, USART_SR register to reset values;


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

**Step2.** Initialize the port. Configure RXD as a digital input with pull-up. Map RXD to the required pin. Map SCK to the required pin;

**Step3.** Select the clock source through the USART_CR2.CLKC bit;

**Step4.** Set USART_CR1 (except RE), USART_CR2, USART_CR3 registers;

**Step5.** If the internal clock source is selected, it is necessary to set USART_BRR.PSC to select the prescaler value, and set the USART_BRR communication baud rate;

**Step6.** Set USART_IER.RC=1, USART_IER.OR=1, USART_CR1.RE=1;

**Note**

When using the RTS function, the RTS pin outputs a low level after the USART_CR1.RE=1

**Step7.** When the synchronous clock is input or the internally generated synchronous clock is working, data reception starts, and the data is received into the shift register.

- When an overflow error occurs, the data is overwritten and an overflow error interrupt occurs, and error handling is performed;
- When no overflow error occurs, the received data is transferred to the USART_RDR register, and a reception completion interrupt occurs. Through the reception completion interrupt program, the received data is read before the last bit of the next frame of data is received, and then step7 is repeated to realize the function of continuously receiving data.

**Note**

When using the RTS function, the RTS pin outputs a low level after the data is read

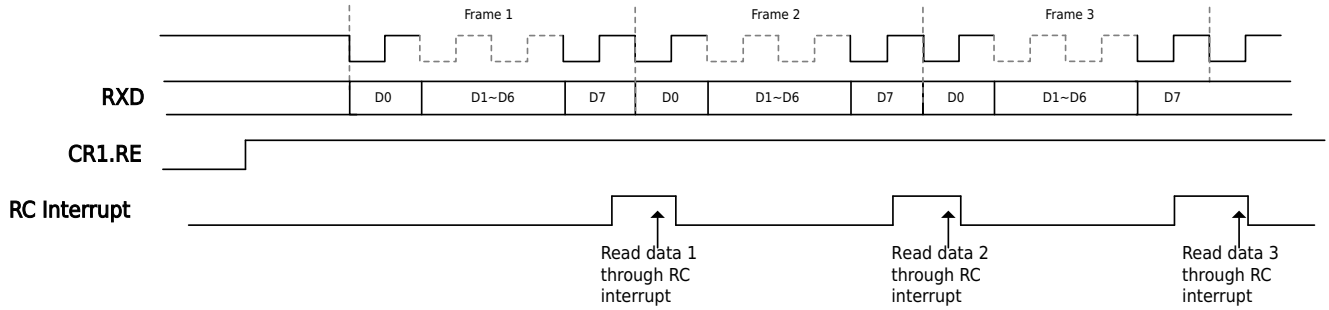



Figure 18-13 Synchronous Serial Port Mode Receiving Data Legend 1 (LSB/No RTS Function)

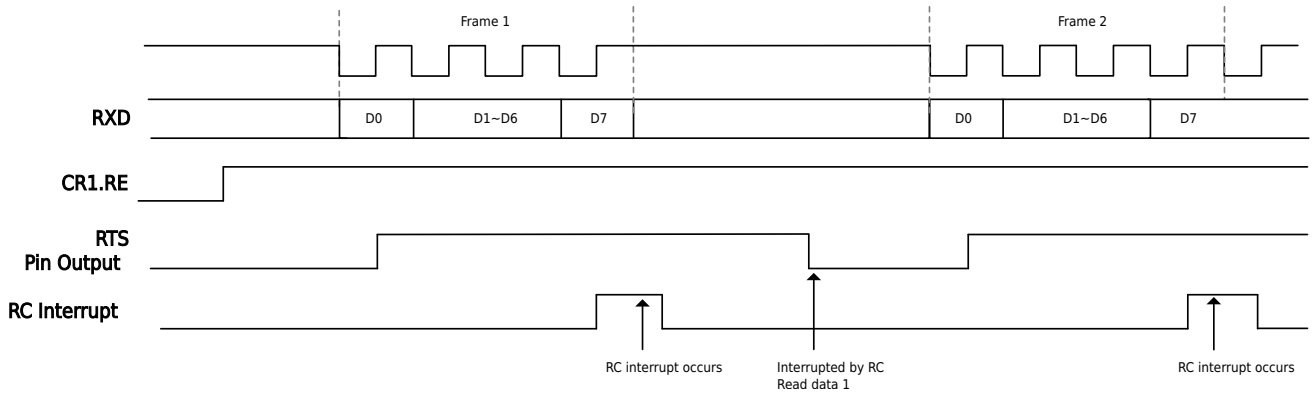



Figure 18-14 Synchronous Serial Port Mode Receiving Data Legend 2 (LSB/RTS Function)

## Error Handling

When receiving data in synchronous serial port mode, the receiving error is an overflow error (USART_SR.OR). The overflow error occurs when new data is received when the value of USART_RDR


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

has not been read. Data not read after an overflow error occurs is overwritten. The data transfer can be restarted by clearing the error flag to zero by writing the corresponding clear register bit.

In order to avoid overflow errors, the data received in the previous frame should be read before the last bit of the current frame is received.

#### Receiver Interrupt

In the synchronous serial port mode, the receiver supports two kinds of interrupts, receiving complete interrupt and overflow error interrupt.

USART_IER.RC=1, an interrupt occurs when data is transferred from the shift register to USART_RDR.

USART_IER.OR=1, the receive data overflow error interrupt occurs.

The above interrupts are all generated after the eighth CK pin rising edge.

### 18.3.3.6 Simultaneous Transmission and Reception

The USART synchronous serial port mode supports full-duplex operation, sending and receiving data at the same time.

When sending and receiving data at the same time, a command is required to write 1 to USART_CR1.RE and USART_CR1.TE at the same time, and set the interrupt enable bit. Other setting procedures are the same as those of the transmitter and receiver.

### 18.3.3.7 Precautions for Synchronous Serial Port Mode

1. When sending data using an external input clock, wait for more than six PCLKs before inputting the clock after USART_TDR is updated. Total period of the input clock must be greater than or equal to six PCLKs, and the periods of high and low levels must be greater than or equal to two PCLKs.
2. When data is sent consecutively, the next frame of data needs to be updated before the last bit of the current frame is sent.

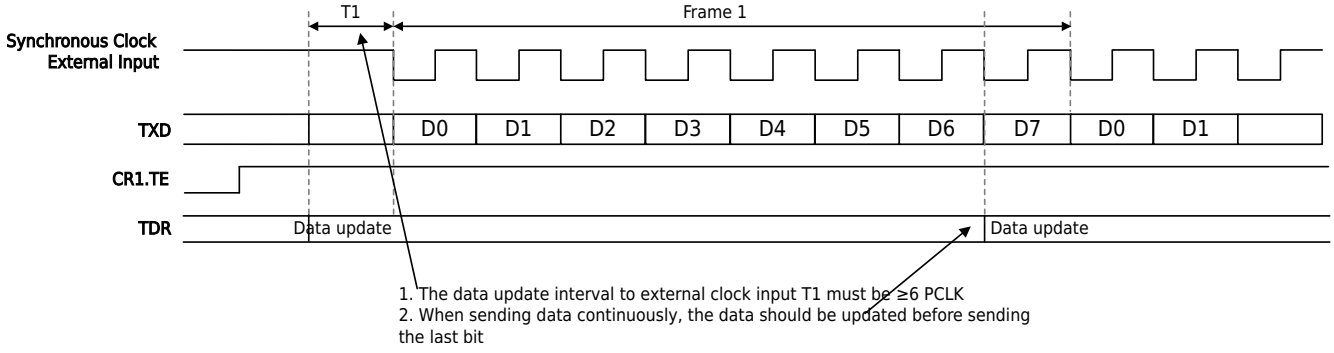



**Figure 18-15 Precautions for Synchronous Serial Port Mode**

## 18.3.4 Smart Card (7816) Mode

### 18.3.4.1 Mode Overview

ISO7816 is a contact card reading protocol. This module is only used as a smart card master interface, and the smart card mode can be selected by writing USART_CR3.SCEN=1. In smart card mode, the following bits must be cleared:

- USART_CR1.COMB, USART_CR1.SLME, USART_CR1.MPE, USART_CR1.SBS, USART_CR1.OVER8
- USART_CR2.SYNC
- USART_CR3.LINEN, USART_CR3.IREN

In addition, the USART_CR2.CLKC bit needs to be set to provide the clock for the smart card.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 18.3.4.2 Connection Schematic Diagram

The smart card interface supports the smart card communication protocol stipulated in ISO/IEC 7816-3, and supports T=0 (character mode) and T=1 (block mode) at the same time. The figure below is the connection diagram of the smart card mode:

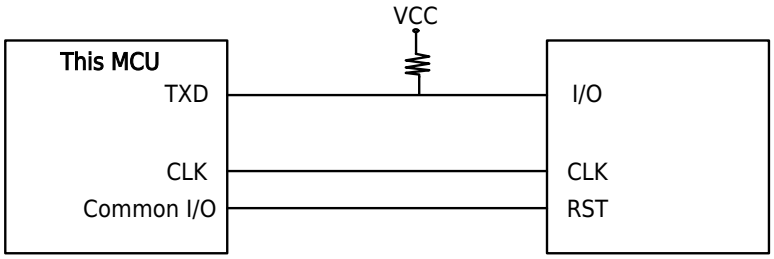



**Figure 18-16 Smart Card Connection Schematic Diagram**

### 18.3.4.3 Clock

#### Internal Clock Source

In smart card mode, only the clock generated by the internal baud rate generator can be used as the clock source.

The basic clock number for one-bit data transmission is the setting value of USART_CR3.BCN.

Register USART_CR2.CLKC bits and USART_BRR are used to set the clock output in smart card mode.

#### Sampling and Reception Tolerance

After detecting the falling edge of the receive pin (TXD), the USART clock synchronizes the received data based on the internal basic clock to start data reception.

Each bit of received data is sampled three times for a majority vote and then sent to the shift register.

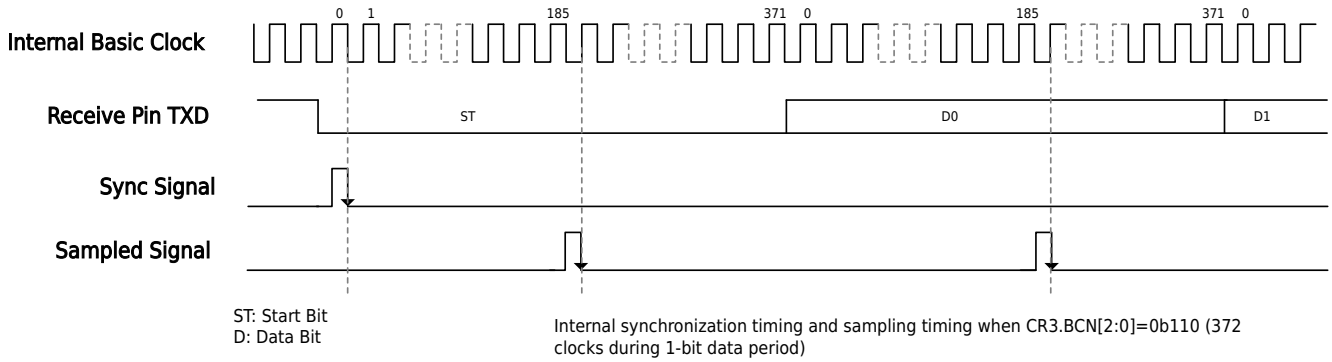



**Figure 18-17 Smart Card Mode Internal Synchronization Timing and Sample Timing Diagram**

Receive tolerance calculation formula:

$$M = \left| \left(0.5 - \frac{1}{2N}\right) - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100\%$$

Where:

- M: receive tolerance
- N: the number of clocks required for one-bit data transmission (BCN[2:0] setting value)
- L: the length of the frame (ten)
- D: clock duty cycle
- F: clock skew


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 18.3.4.4 Data Format

In smart card mode, a frame of data consists of a start bit, data bits, and a check bit.

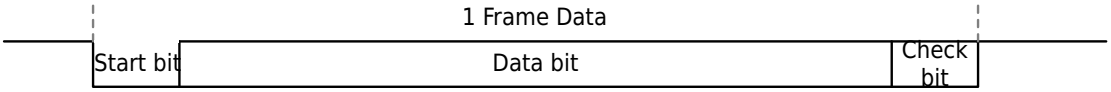



**Figure 18-18 Smart Card Mode Data Format**

- **Start Bit (ST):** Composed of one bit at low level.
- **Data Bit (D):** Bit is fixed as 8-bit data.
- **Prity Bit (P):** Need to be configured as 1-bit even parity.

The figure below shows the legend of data 0x3B data format.

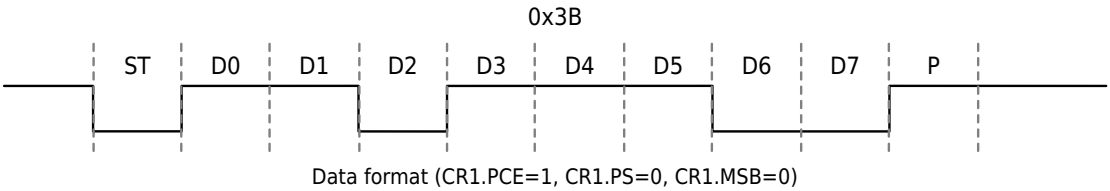



**Figure 18-19 Smart Card Mode Data Format Example**

### 18.3.4.5 Initialization Setup Procedure for Smart Cards

**Step1.** Configure the USART_CR1 register to its reset value;

**Step2.** Initialize ports, all set as digital input with pull-up; set the pins to be used;

**Step3.** The status register confirms that the USART_SR register is set to the reset value;

**Step4.** Set USART_CR1 (except TE, RE), USART_CR2, USART_CR3 registers;

**Step5.** Set USART_BRR.PSC to select the prescaler value, and set the USART_BRR communication baud rate;

**Step6.** Set the clock control of the USART_CR2.CLKC bit;

**Step7.** Set the USART_CR1 register (TE, RE bits), set TE and RE as required, and do not set TE and RE to 1 at the same time.

When switching transmit mode to receive mode, or receive mode to transmit mode, step7 through above must be reconfigured.

When using the smart card function, the word length USART_CR1.CHLEN, the check control bit USART_CR1.PCE , stop bit length USART_CR1.STOP can be any value, the parity check bit USART_CR1.PS must be set to odd or even as required, and the reverse control bit USART_CR2.TXINV=0. Both data reception and transmission are transmitted through TXD, regardless of the USART_CR1.COMB value.

### 18.3.4.6 Smart Card Mode Action Description

In smart card mode, the interrupt (send data empty interrupt) flag is USART_SR.TXE. An interrupt is generated when USART_SR.TXE=1 and USART_IER.TXE=1.

#### Functional Overview

During data transmission, there must be a protection time of at least 2etu (Elementary Time Unit) between two frames (from the end of the check bit to the start of the next frame's start bit).

If an error signal from the receiver is detected during data transmission, the data will be automatically retransmitted after 2etu.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

When a verification error occurs in the received data, a low level of 1etu is sent, which is an error signal. The timing of sending the error signal is 10.5etu after the start of reception.

The length of 1etu is determined by the number of SCK pin clocks specified by the USART_CR3.BCN value.

### Send Instructions

1. After a frame of data is sent, if an error signal sent by the receiver is detected, set USART_SR.FE=1 to generate a frame error interrupt (when USART_IER.FE=1), the USART_SR.TC flag is not set to 1, and the data is automatically resent. The USART_SR.FE bit must be cleared before receiving the next frame parity bit.
2. A frame of data is sent without error, set USART_SR.TC=1, and an interrupt occurs (USART_IER.TC=1). Data can be sent continuously by writing data through the interrupt program.

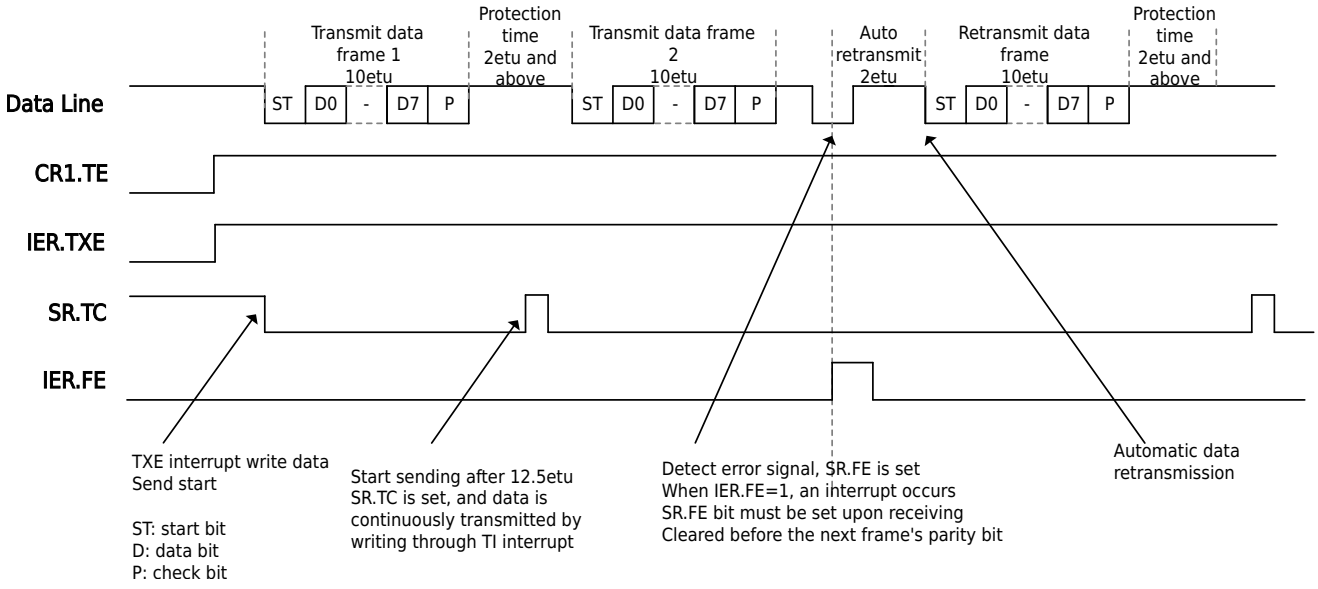



**Figure 18-20 Smart Card Mode Transmit Data Legend**

### Receive Instructions

1. When receiving data, if a parity error is detected, then USART_SR.PE=1, if USART_IER.PE=1, a parity error interrupt occurs. USART_SR.PE needs to be cleared before receiving the next frame check bit.
2. When a verification error occurs, a low level of 1etu will be sent, that is, an error signal, requiring the sender to resend the data.
3. To receive data normally, you can read the received data through the interrupt program and receive continuously.
4. When receiving data, an overflow error is detected.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

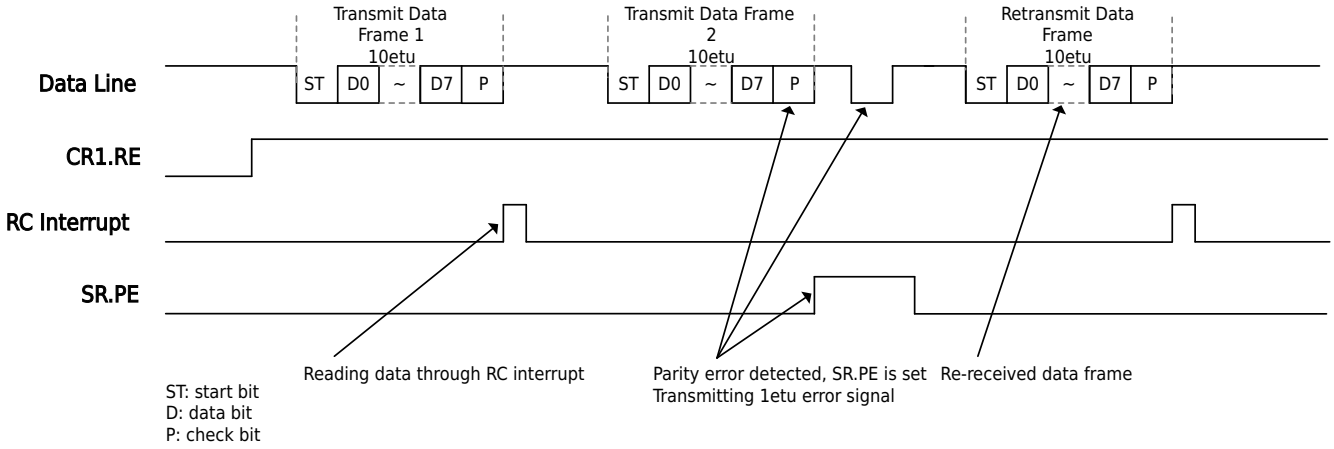



Figure 18-21 Smart Card Mode Receive Data Legend

## 18.3.5 Infrared Mode

### 18.3.5.1 Mode Overview

Select infrared mode by writing USART_CR3.IREN=1. In infrared mode, the following bits must be cleared:

- USART_CR1.COMB, USART_CR1.SLME, USART_CR1.MPE
- USART_CR3.SCEN, USART_CR3.LINEN

This module contains two infrared modes: infrared pulse and 38K modulation output with 3/16 baud rate period. Two modes of the same module cannot work at the same time.

Reasonably configure the word length USART_CR1.CHLEN, check control bits USART_CR1.PCE, USART_CR1.PS, stop bit length USART_CR1.STOP, inversion control bits USART_CR2.TXINV, USART_CR2.RXINV and other register bits to meet the needs of infrared applications.

### 18.3.5.2 Infrared Mode 1: Infrared Pulse Width for Baud Rate Period 3/16

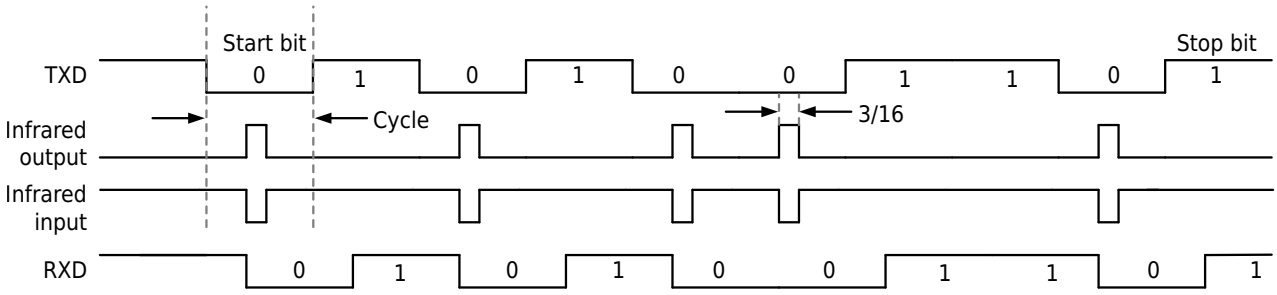



Figure 18-22 Infrared Data Modulation

Select the baud 3/16 rate period mode by writing USART_CR3.IREN=1 and USART_CR3.IRCW=0. In infrared mode, the USART_CR3.LINEN and USART_CR3.SCEN bits must be cleared.

The infrared protocol physical layer specifies the use of a return-to-zero-inverted (RZI) modulation scheme, which represents a logic 0 with pulses of infrared light (pictured above).

The USART's infrared output pulse stream is transmitted to external infrared LEDs on the chip. SIR ENDEC bit rate supported by USART is up to 115.2kbps. In normal mode, the transmitted pulse width is specified as 3/16 of one bit period.

The infrared receiver decoder outside the chip outputs the received non-return-to-zero (NRZ) serial bit stream to the USART. In the idle state, the decoder input is normally high (flag state). Send the encoder


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

output with the opposite polarity to the decoder input. A start bit is detected when the decoder input is low.

1. Infrared communication is a half-duplex communication protocol. To prevent data corruption, simultaneous sending and receiving should be avoided.
2. A "0" is sent as a high pulse, and a "1" is sent as a "0". In normal mode, the pulse width is specified for the selected bit period 3/16 (see figure above).
3. The specification requires that the pulse width value should be greater than 1.41μs.
4. In infrared mode, some bits in the USART_CR1 and USART_CR2 registers are masked as invalid bits. Whether the user writes 1 or 0, it has no impact on the normal operation of the infrared function. For details, refer to the bit function description of the register.
5. In the infrared mode, the input and output signals can be inverted by USART_CR2.RXINV and USART_CR2.TXINV and then input to the USART module or output from the USART module.

#### 18.3.5.3 Infrared Mode 2: 38K Modulation with Adjustable Duty Cycle

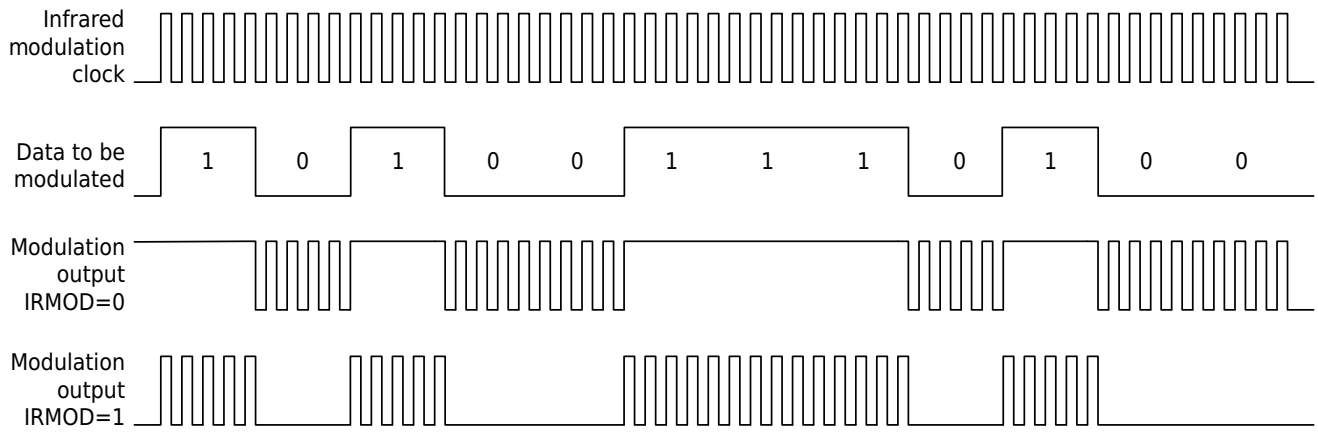



Figure 18-23 38K Infrared Modulation

The way of infrared modulation is: output infrared modulation when sending data 0, and not modulate when sending data 1. The 38K infrared modulation frequency is determined by the value of the register USART_IR and the current operating frequency of the module, and its calculation formula is:

F=PCLK/(USART_IR.DCNTH+USART_IR.DCNTL+2)

Envolves:

- PCLK: module working frequency
- USART_IR.DCNTH: the high level duration of the 38K modulation frequency
- USART_IR.DCNTL: the low level duration of the 38K modulation frequency

Considering the needs of PNP and NPN infrared tubes in application, USART_CR2.RXINV and USART_CR2.TXINV can be set to invert the received and sent data. USART_IR.DCNTH and USART_IR.DCNTL are the duty cycle settings when not inverting. If the sending data is inverting, the two data need to be exchanged.

### 18.3.6 LIN Mode

#### 18.3.6.1 Mode Overview

Select LIN mode by writing USART_CR3.LINEN=1. In LIN mode, the following bits must be cleared:

- USART_CR1.COMB, USART_CR1.STOP, USART_CR1.SLME, USART_CR1.MPE, USART_CR1.PCE
- USART_CR2.SYNC


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

- USART_CR3.SCEN, USART_CR3.IREN

LIN is the abbreviation of Local Interconnect Network, which is a low-speed (1-20kbps) serial communication protocol for reducing the cost of automotive networks.

LIN communication is a single-master communication, and a master device can connect up to 15 devices.

"Wire-AND" relationship between the devices on the bus, and the dominant level plays a leading role:

- When any node on the bus sends a dominant level (low level), the bus is at a dominant level;
- Only when all nodes send recessive level (high level) or do not send information (when no information is sent, the bus defaults to recessive level), the bus will show recessive level.

The LIN bus supports ISO9141; the LIN slave device is connected to the master control device through LIN, which is mainly used for the control of switches, transmissions and sensors.

The USART in this product integrates a multi-function timer to facilitate the realization of LIN bus functions, including data transmission, baud rate setting, synchronization interval detection, synchronization measurement and baud rate calculation, etc. RXD pin edge interrupt (USART_IER.DWK) and I/O port external interrupt input function can be used for wake-up signal detection.

According to the application, when USART is used as LIN, it is RXD/TXD two-wire mode, and the corresponding chip needs to be configured externally for boosting and bus conversion.

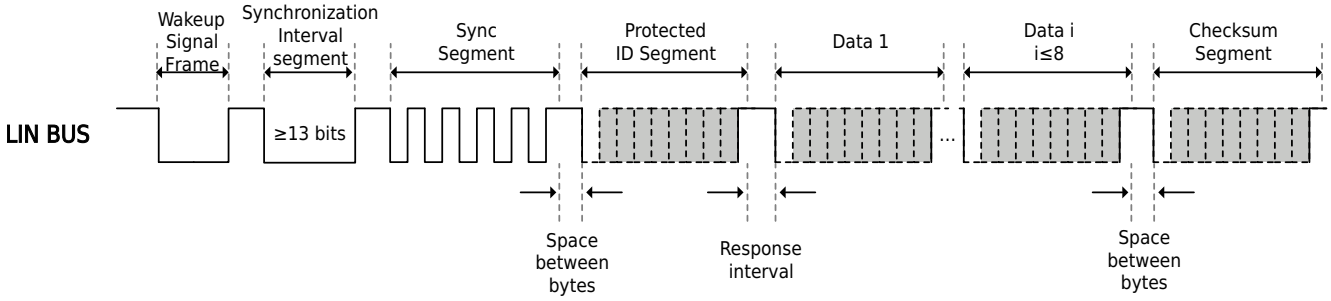



Figure 18-24 LIN Bus Data and Frame Structure

**Note**

Wake-up signal frame is only required when the device needs to wake up

### 18.3.6.2 LIN Data Format

One frame of LIN data consists of one start bit, eight data bits and one stop bit, and sends and receives data in LSB mode.

### 18.3.6.3 LIN Transmission

#### Transmission of Wake-Up Signal Frame

Set the appropriate baud rate, and realize the sending of the wake-up signal frame by sending data 0x80.

#### Synchronization Interval Segment Transmission

The synchronization interval segment indicates the beginning of a frame of data, and lasts at least 13 bits (based on the master node speed) of the dominant level (low voltage). This product uses USART_CR3.BREAK and USART_TDR as the synchronization interval time for sending dominant level, and the number of "0" s for sending continuous bits is set in the USART_TDR register.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### Synchronization Segment Transmission

Transmit data 0x55 to achieve the transmission of the synchronization segment.

### LIN Master Control Device Software Operation Steps

**Step1.** Set the USART_CR1 register to its reset value;

**Step2.** Refer to the relevant description of the port digital multiplexing function in the "General-Purpose I/O (GPIO)" chapter. Configure TXD and RXD as digital inputs with pull-up, and map TXD and RXD to the required pins.

**Step3.** Set USART_CR1 (except TE), USART_CR2, USART_CR3 (except LINEN) registers.

**Step4.** Set USART_BRR.PSC to select the prescaler value.

**Step5.** Set the USART_BRR register to set the baud rate of the signal frame.

**Step6.** (Optional) wake-up signal frame send:

a. Set USART_CR1.TE=1.

b. Write communication data 0x80 to USART_TDR, the data is transferred to the shift register, and the transmission starts.

c. Wait for delivery to complete.

**Step7.** Transmission of the synchronous interval segment:

a. Set USART_CR3.BREAK=1, USART_CR3.LINEN=1.

b. If step 6 has not been executed, then set USART_CR1.TE = 1; otherwise, skip this step of operation.

c. Write the length of the synchronization interval to USART_TDR, and send the low level of the synchronization interval to the bus.

d. Wait for delivery to complete.

**Step8.** The transmission of the synchronization segment has the same process as that of the asynchronous serial port transmission:

a. Write data 0x55 to USART_TDR.

b. Wait for delivery to complete.

**Step9.** Setup space between bytes:

a. Write the length of waiting time between bytes to USART_TOR.

b. Write USART_CR1.TOE=1 to start software timing.

c. Wait for USART_SR.TO=1.

**Step10.** Protected ID segment: write protected ID segment data(ID) to USART_TDR.

**Step11.** Wait for delivery to complete.

**Step12.** Setup response interval:

a. Write the length of the acknowledgment waiting time to USART_TOR.

b. Write USART_CR1.TOE=1 to start software timing.

c. Wait for USART_SR.TO=1.

**Step13.** Data segment 1: The data sender may be the master or the slave, depending on the application and the parameters of the protected ID segment. If the master sends data, write the data to USART_TDR and wait for the sending to complete (USART_SR.TC=1).

**Step14.** Inter-byte interval of data segment 1, refer to step9 to execute.

**Step15.** Data segment 2: refer to the data segment 1 of step13.

**Step16.** Inter-byte interval of data segment 2, refer to step14.

**Step17.** ……


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

**Step18.** Data segment n: refer to the data segment 1 of step13.

**Step19.** Interbyte interval of data segment n: refer to step14.

**Step20.** Checksum segment (checksum is implemented by software): if the data section is the data sent by the master, then refer to step13.

**Step21.** Wait for the transmission to complete, the LIN operation ends, and the control register is cleared.

**Note**

Check bit USART_CR1.PCE=0, stop bit USART_CR1.STOP=0, USART_CR1.COMB=0 must be set.

#### 18.3.6.4 LIN Receive

##### Detection of Wake-Up Signal Frame

Wake-up signal frame detection is achieved either through the falling edge detection of the I/O port's external interrupt input function or by setting USART_IER.DWK=1 to generate an interrupt; if LIN is in active state, this step is skipped.

##### Detection of Synchronous Interval Segments

The detection of the synchronization interval is realized by writing the synchronization interval detection enable bit USART_CR3.LBDE=1 and writing USART_CR1.TOE=4 to set the synchronization interval detection function.

Write the desired low-level width in USART_TOR (for example, 11 baud rate bit widths, PCLK=48M, LIN baud rate 20k, USART_BRR.PSC=1, then TOR=48M/20k/4*11=6600=0x19C8). If the falling edge of RXD is detected after USART_CR3.LBDE=1, the internal timer starts counting:

- If the counter value is equal to the USART_TOR value during RXD=0, the hardware sets USART_SR.LBD=1 and generates an interrupt (if USART_IER.LBD=1);
- RXD rising edge is detected when the counter value is less than the USART_TOR value, stop counting and return to the idle state to continue monitoring the arrival of the next synchronization interval.

##### Baud Rate Measurements for Synchronous Segments

Start the USART internal timer by writing the baud rate automatic detection enable bit USART_CR3.ABRE=1 (the count clock of the timer is the PCLK of the USART) to measure the baud rate:

The hardware continuously measures the period between the 5 falling edges of the RXD pin, accumulates the measured 4 period values and divides it by 8 to obtain the communication baud rate value. This process is completed by the hardware without software participation.

Baud rate = $1/(t_{PCLK} * 4^{PSC} * (BDR + 1))$

Envolves:

- $t_{PCLK:}$ module working clock cycle
- PSC: USART_BRR.PSC value
- BDR: USART_BDR value

After the measurement is completed, the hardware writes the baud rate value into USAR_BDR, and at the same time, the hardware sets the automatic baud rate detection flag USART_SR.ABR; when USART_SR.ABR=0, it is not recommended to read the value of USART_BDR, and the software can set USART_IER .ABR=1, a hardware interrupt is generated when USART_SR.ABR=1. The software can adjust the current USART baud rate according to the USART_BDR value for more precise synchronization. For the setting of the baud rate, see "Communication Baud Rate Programming".


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### LIN Slave Device Software Operation Steps

**Step1.** Set the USART_CR1 register to its reset value;

**Step2.** Refer to the relevant description of the port digital multiplexing function in the "General-Purpose I/O (GPIO)" chapter. Configure TXD and RXD as digital inputs with pull-up, and map TXD and RXD to the required pins.

**Step3.** Set USART_CR1 (except RE), USART_CR2, USART_CR3, USART_IER registers and USART_CR3.LINEN=1 (if there is step5, then USART_CR3.LINEN=0).

**Step4.** Set the prescaler and baud rate initial value of USART_BRR.

**Step5.** (Optional) Wait for the wake-up signal to arrive: Set the external interrupt of the I/O port to detect the falling edge or set USART_IER.DWK = 1, and wait for the CPU to be woken up.

**Step6.** Set the low-level width of the synchronization interval segment, for example, expect 13 baud rates, then calculate the TOR value according to the formula in the USART_TOR register and write it into the TOR register (if there is step5, set USART_CR3.LINEN=1).

**Step7.** Set USART_CR3.LBDE=1, USART_IER.LBD, USART_CR1.TOE=4 (synchronous interval measurement) and USART_CR1.RE=1. If USART_IER.LBD=0, then go to the step8; otherwise, after the interrupt occurs, go to the step8.

**Step8.** The software queries the USART_SR.LBD value, if it is 1, it means that the synchronization interval segment is detected, and then enters the synchronization segment measurement. Otherwise, continue to query the USART_SR.LBD value.

**Step9.** Synchronous segment baud rate measurement: Set USART_CR3.LBDE = 0, set the value of USART_IER.ABR and USART_CR3.ABRE = 1, USART_CR1.TOE = 5. If USART_IER.ABR=0, loop through the USART_SR.ABR value, if USART_IER.ABR=1, wait for the interrupt to occur and go to step10.

**Step10.** Set USART_CR3.ABRE = 0. According to the value of USART_BDR, if the baud rate needs to be adjusted, set USART_CR1.RE = 0, then reset USART_BRR. For the setting of the baud rate, see "Communication Baud Rate Programming". Finally, set USART_CR1.RE=1. If the baud rate does not need to be adjusted, skip to the step11.

**Step11.** Receive the protected ID segment, data, and checksum. The process is the same as that of receiving data via the asynchronous serial port:

a. Receive the protected ID segment: query the USART_SR.RC value, if it is 1, read the USART_RDR value and perform the operation specified by this value according to the agreement; if USART_SR.RC=0, continue to query. This step can also be done with an interrupt.

> **Note**
> If the slave is required to send data after parsing the protected ID segment, then ignore the following two steps, set USART_CR1.TE = 1 and refer to the step13-21 to operate in "LIN Transmission"."

b. Receive data: query USART_SR.RC value, if it is 1, read USART_RDR, otherwise continue to query.

c. Receive checksum: query USART_SR.RC value, if it is 1, read USART_RDR, otherwise continue to query. After receiving the checksum, perform a check on the previously received N data, and then compare it with the received checksum. Perform corresponding operations according to the comparison result.

**Step12.** After the reception is over, it enters the dormant state and waits for the next wake-up.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

## 18.4 Operation Example

### 18.4.1 Programming Example of Sending Data in Asynchronous Serial Port Mode

**Step1.** Refer to the relevant description of the port digital multiplexing function in the "General-Purpose I/O (GPIO)" chapter. When USART_CR2.TXINV = 0, configure TXD as a digital input with pull-up. When USART_CR2.TXINV=1, configure TXD as a digital input with pull-down. Map TXD to the required pin.

**Step2.** Set SYSCTRL module register PERI_CLKEN0.USARTn (n=0/1) to 1 to enable the USART configuration clock and operating clock.

**Step3.** Write 0x00 to USART_CR1 to make the USART module return to the initial state.

**Step4.** Configure USART_CR2.SYNC as 0 to make the USART module work in asynchronous serial port mode.

**Step5.** Configure the USART_CR1.OVER8 and USART_BRR registers to configure the communication baud rate.

**Step6.** Configure USART_CR1.CHLEN to configure the number of data bits in the frame.

**Step7.** Configure USART_CR1.STOP, USART_CR1.PCE and USART_CR1.PS, configure stop bit length and parity check mode.

**Step8.** Configure USART_CR1.MSB to configure the data sequence order when sending data.

**Step9.** If the USART needs to operate in the single-wire half-duplex mode, then set USART_CR1.COMB and USART_CR1.RE to 1, USART_CR1.TE to 0, and jump to the step11 to continue execution. If the USART is used as a slave in multi-processor communication, then configure TXD as an open-drain output.

**Step10.** Set USART_CR1.TE to 1 to enable TXD sending function.

**Step11.** Write 0x00 to USART_ICR.TC to clear the sending completion flag.

**Step12.** Write a frame of data to be sent to the USART_TDR register.

**Step13.** The query waits for USART_SR.TC to become 1, and the data frame written this time is sent.

**Step14.** If the data to be sent is not completed, then jump to the step11 to continue execution. If the data has been sent and the USART is used as a slave in multi-processor communication, set USART_CR1.TE to 0.

### 18.4.2 Programming Example of Receiving Data in Asynchronous Serial Port Mode

**Step1.** Refer to the relevant description of the port digital multiplexing function in the "General-Purpose I/O (GPIO)" chapter. Map RXD to the required pin. When USART_CR2.RXINV=0, configure RXD as a digital input with pull-up, when USART_CR2.RXINV=1, configure RXD as a digital input with pull-down to prevent false triggering caused by the RXD line floating when the transmitting end is not enabled while the receiving end is enabled first.

**Step2.** Set SYSCTRL module register PERI_CLKEN0.USARTn (n=0/1) to 1 to enable the USART configuration clock and operating clock.

**Step3.** Write 0x00 to USART_CR1 to make the USART module return to the initial state.

**Step4.** Configure USART_CR2.SYNC as 0 to make the USART module work in asynchronous serial port mode.

**Step5.** Configure the USART_CR1.OVER8 and USART_BRR registers to configure the communication baud rate.

**Step6.** Configure USART_CR1.CHLEN to configure the number of data bits in the frame.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

**Step7.** Configure USART_CR1.STOP, USART_CR1.PCE and USART_CR1.PS, configure stop bit length and parity check mode.

**Step8.** Configure USART_CR1.MSB to configure the data sequence order when receiving data.

**Step9.** Configure USART_CR1.SBS to select the judgment method of the start bit.

**Step10.** Write 0x00 to the USART_ICR register to clear all status flags.

**Step11.** Set USART_CR1.RE to 1.

**Step12.** The query waits for USART_SR.RC to become 1, and one frame of data is received.

**Step13.** Query USART_SR.PE, USART_SR.FE, USART_SR.OR to confirm whether the received data frame is valid. If it is invalid, perform error handling, if it is valid, read the data from the USART_RDR register and save it .

**Step14.** Write 0x00 to USART_ICR.RC to clear the receiving completion flag.

**Step15.** If the data to be received is not completed, jump to step12 to continue execution.

### 18.4.3 Infrared Mode 1: Transmission Data Setting

**Step1.** Set the USART_CR1 register to its reset value;

**Step2.** The initialization ports are all set as digital input with pull-up; set the pins that need to be used by infrared.

**Step3.** Set values other than TE in USART_CR1, set USART_CR2, USART_CR3 registers.

**Step4.** Set USART_BRR.PSC to select the prescaler value, and set the USART_BRR communication baud rate.

**Step5.** Set USART_CR3.IREN=1 (USART_CR2.TXINV is configured as needed), USART_CR1.TE=1, set USART_IER.TXE=1.

**Step6.** Write the communication data to USART_TDR by sending the interrupt program, and the sending starts.

**Step7.** To continuously transmit data, repeat step6.

**Step8.** Write the last sent data through the interrupt program, and write USART_IER.TXE=0, USART_IER.TC=1, after the last data is sent, a send completion interrupt occurs.

**Step9.** Write USART_CR1.TE=0.

### 18.4.4 Infrared Mode 1: Reception Data Setting

**Step1.** Set the USART_CR1 register to its reset value;

**Step2.** The initialization ports are all set as input with pull-up; set the pins that need to be used by infrared.

**Step3.** Set USART_IER.RC=1.

**Step4.** Set values other than RE in USART_CR1, set USART_CR2, USART_CR3 registers.

**Step5.** Set USART_BRR.PSC to select the prescaler value, and set the USART_BRR communication baud rate.

**Step6.** USART_CR3.IREN=1 (USART_CR2.RXINV is configured as needed), USART_CR1.RE=1.

**Step7.** When the start bit is detected, the data is received into the shift register after internal synchronization, and the check bit and stop bit are checked:

a. When the stop bit is low level, a frame error occurs, the received data is transferred to the USART_RDR register, and a receive frame error interrupt occurs, and the error interrupt handler is executed.

b. When a data overflow error occurs, the previous frame data is overwritten and a data overflow error interrupt occurs, and the error interrupt handler is executed.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

c. When an error occurs, a parity error occurs, the received data is transferred to USART_RDR, and a parity error interrupt occurs, and the error interrupt handler is executed.
d. When no error occurs, the received data is transferred to the USART_RDR register, and a reception completion interrupt occurs. After reading the received data through the interrupt program, repeat step7 to receive data continuously.

**Step8.** Write USART_CR1.RE=0.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

## 18.5 Register Description

Register bit access permission description:

1. USART_CR1.RE=0 and USART_CR1.TE=0, software can write all registers.
2. As long as one of USART_CR1.RE and USART_CR1.TE is 1, the only registers that can be written by software are USART_IER, USART_ICR, USART_TDR, USART_TOR, USART_BRR, USART_CR1.TOE, USART_CR1.SLME, USART_CR2.TXDMAEN, and USART_CR2.RXDMAEN.
3. As long as one of USART_CR1.RE and USART_CR1.TE is 1, the condition that USART_CR1 can be written into a new value by software is: USART_CR1.TE=0 and USART_CR1.RE=0 in the newly written value.
4. In LIN mode, the value of the USART_BRR register can be updated when the value of USART_CR1.TE is 1.
5. There are no restrictions on reading or writing the USART_CR3 register.

### 18.5.1 Register List

- USART0 base address: 0x4000A000
- USART1 base address: 0x4000A400

Table 18-10 Universal Synchronous Asynchronous Receiver/Transmitter (USART) Register Offset

| Offset | Register | Description | Width |
|---|---|---|---|
| 0x00 | USART_CR1 | Control Register 1 | 32 |
| 0x04 | USART_CR2 | Control Register 2 | 32 |
| 0x08 | USART_CR3 | Control Register 3 | 32 |
| 0x0C | USART_BRR | Baud Rate Register | 32 |
| 0x10 | USART_TOR | Timer Counter Register | 32 |
| 0x14 | USART_SR | Status Register | 32 |
| 0x18 | USART_TDR | Transmit Data Register | 32 |
| 0x1C | USART_IER | Interrupt Enable Register | 32 |
| 0x20 | USART_ICR | Interrupt Fag Clear Register | 32 |
| 0x24 | USART_RDR | Receive Data Register | 32 |
| 0x28 | USART_ADDR | Address Register | 32 |
| 0x2C | USART_ADDMR | Address Mask Register | 32 |
| 0x30 | USART_BDR | Baud Rate Detection Value Register | 32 |
| 0xF0 | USART_IR | Infrared Modulation Register | 32 |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 18.5.2 Control Register 1 (USART_CR1)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | SLME | MPE | COMB | SBS | TOE | | | OVER8 | MSB | STOP | CHLEN | PCE | PS | TE | RE |
| Access | | | | | | | | | | | | | | | | | | RW | RW | RW | RW | RW | | | RW | RW | RW | RW | RW | RW | RW | RW |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:15* | *Reserved* | *All bits read as 0, Write as 0* | - |
| 14 | SLME | When multiprocessor communication, silent mode enable bit<br>● 0b0: Normal mode<br>● 0b1: Silent mode<br>**Note**<br>● When SLME=1, data with MPB bit 0 will not be read from the receive shift register to the receive data register, and the error flags PE and FE bits will not be set. When receiving ID data with MPB=1 and address matching occurs, SLME is automatically cleared to initiate normal data reception.<br>● The SLME bit is only valid in multiprocessor communication in asynchronous serial port mode, and this bit must be set to 0 in other modes. | RW |
| 13 | MPE | Multiprocessor communication function enable bit (only applicable to asynchronous serial port mode)<br>● 0b0: Invalid<br>● 0b1: Valid<br>**Note**<br>● The non-asynchronous serial port mode must be set to 0.<br>● This bit can only be written when both USART_CR1.TE and USART_CR1.RE are 0. | RW |
| 12 | COMB | In asynchronous serial port mode, the composite enable bit of the transceiver port<br>● 0b0: TXD and RXD are used separately<br>● 0b1: Asynchronous serial port single-bus mode, TXD and RXD share TXD port<br>**Note**<br>● The non-asynchronous serial port mode must be set to 0.<br>● This bit can only be written when both USART_CR1.TE and USART_CR1.RE are 0. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 11 | SBS | When receiving data in asynchronous serial port mode, the start bit detection mode setting bit<br>● 0b0: The RXD pin edge for receiving data (falling edge when USART_CR2.RXINV=0, rising edge when USART_CR2.RXINV=1) serves as the start bit.<br>● 0b1: The RXD pin level for receiving data (low level when USART_CR2.RXINV=0, high level when USART_CR2.RXINV=1) serves as the start bit.<br>**Note**<br>● The non-asynchronous serial port mode must be set to 0.<br>● This bit can only be written when both USART_CR1.TE and USART_CR1.RE are 0. | RW |
| 10:8 | TOE | Multi-function timer working mode configuration bit<br>● 0b000: Stop timer counter<br>● 0b001: General timing function<br>● 0b010: Receive timing function<br>● 0b011: Timing function between data frames<br>● 0b100: In LIN mode, synchronous interval segment counting is enabled<br>● 0b101: In LIN mode or asynchronous serial port mode, baud rate automatic setection is enabled<br>● 0b110-0b111: Reserved, the same as 0b000, counting function disabled<br>**Note**<br>● TOE of all functions are cleared by hardware.<br>● If switching from one function (such as TOE = 0b010) to another function (such as TOE = 0b001), the software needs to write TOE = 0b000 first and then write TOE = 0b001. | RW |
| 7 | OVER8 | Asynchronous serial port mode/LIN mode, the sampling mode setting bit<br>● 0b0: 16 times sampling<br>● 0b1: 8 times sampling<br>**Note**<br>● Synchronous serial port mode and smart card mode must be set to 0, infrared mode setting is invalid.<br>● This bit can only be written when both USART_CR1.TE and USART_CR1.RE are 0. | RW |
| 6 | MSB | MSB/LSB mode select bit<br>● 0b0: LSB mode<br>● 0b1: MSB mode<br>**Note**<br>This bit can only be written when both USART_CR1.TE and USART_CR1.RE are 0. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 5 | STOP | Stop bit length setting bit<br>● 0b0: 1 stop bit<br>● 0b1: 2 stop bits<br>**Note**<br>● Synchronous serial port mode and LIN mode must be set to 0.<br>● This bit can only be written when both USART_CR1.TE and USART_CR1.RE are 0. | RW |
| 4 | CHLEN | Transmit/receive data length setting bit<br>● 0b0: 8-bit<br>● 0b1: 9-bit<br>**Note**<br>● Synchronous serial port mode and LIN mode must be set to 0.<br>● This bit can only be written when both USART_CR1.TE and USART_CR1.RE are 0. | RW |
| 3 | PCE | Parity valid bit<br>● 0b0: No parity<br>● 0b1: Verification<br>**Note**<br>● Synchronous serial port mode and LIN mode must be set to 0.<br>● This bit can only be written when both USART_CR1.TE and USART_CR1.RE are 0. | RW |
| 2 | PS | Parity select bit<br>● 0b0: Even parity<br>● 0b1: Odd parity<br>**Note**<br>● The PS bit is only valid when PCE=1.<br>● This bit can only be written when both USART_CR1.TE and USART_CR1.RE are 0. | RW |
| 1 | TE | Transmit enable bit<br>● 0b0: Transmit disable<br>● 0b1: Transmit enable<br>**Note**<br>● In synchronous serial port mode (USART_CR2.SYNC=1), the TE bit can only be written to 1 when TE=0 and RE=0 (transmit/receive disabled). If it is full-duplex in synchronous mode, TE and RE must be at the same time Write 1; when TE=1 or RE=1, this bit can only write 0 with the RE bit at the same time; when TE=1 and RE=1, this bit can only write 0 with the RE bit at the same time. Unrestricted when USART_CR2.SYNC=0.<br>● TE setting is invalid when single-wire half-duplex work. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 0 | RE | Receive enable bit<br>● 0b0: Receive disable<br>● 0b1: Receive enable<br>**Note**<br>In synchronous serial port mode (USART_CR2.SYNC=1), the RE bit can only be written to 1 when TE=0 and RE=0 (transmit/receive disabled). If it is full-duplex in synchronous mode, RE and TE must be at the same time Write 1; when TE=1 or RE=1, this bit can only write 0 with the TE bit at the same time; when TE=1 and RE=1, this bit can only write 0 with the TE bit at the same time. Unrestricted when USART_CR2.SYNC=0. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 18.5.3 Control Register 2 (USART_CR2)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x04 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | CLKC | | SWAP | SYNC | TXINV | RXINV | CTSEN | RTSEN | TXDMAEN | RXDMAEN |
| Access | | | | | | | | | | | | | | | | | | | | | | | RW | | RW | RW | RW | RW | RW | RW | RW | RW |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:10* | *Reserved* | *All bits read as 0, Write as 0* | - |
| 9:8 | CLKC | In asynchronous serial port mode/synchronous serial port mode/ smart card mode, the clock control bit<br>**Asynchronous Serial Port Mode**<br>● 0b00: The clock source is the clock generated by the internal baud rate generator, clock not output to CK pin, CK pin can be used as ordinary IO<br>● 0b01: The clock source is the clock generated by the internal baud rate generator, the clock output is routed to the CK pin, and the output clock frequency equals the baud rate.<br>● 0b10/0b11: Setting prohibited<br>**Synchronous Serial Port Mode**<br>● 0b00/0b01: The clock source is the clock generated by the internal baud rate generator and output to the CK pin.<br>● 0b10/0b11: Clock source is external input clock, input clock frequency equals baud rate<br>**Smart Card Mode**<br>● 0b00: The clock source is the clock generated by the internal baud rate generator, clock not output to CK pin, CK pin can be used as ordinary IO<br>● 0b01: The clock source is the clock generated by the internal baud rate generator, clock output to the CK pin<br>● 0b10/0b11: Setting prohibited<br>**Note**<br>● Infrared mode and LIN mode settings are invalid, neither CK can be output to CK pin.<br>● This bit can only be written when both USART_CR1.TE and USART_CR1.RE are 0. | RW |
| 7 | SWAP | TXD/RXD pin swap control bit<br>● 0b0: Do not exchange<br>● 0b1: Exchange<br>**Note**<br>This bit can only be written when both USART_CR1.TE and USART_CR1.RE are 0. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 6 | SYNC | Synchronous mode select bit<br>● 0b0: Asynchronous serial port mode<br>● 0b1: Synchronous serial port mode<br>**Note**<br>● Asynchronous serial port mode, smart card mode and LIN mode must write 0.<br>● Synchronous serial port mode must write 1.<br>● The infrared mode setting is invalid.<br>● This bit can only be written when both USART_CR1.TE and USART_CR1.RE are 0. | RW |
| 5 | TXINV | Output signal inversion control bit<br>● 0b0: Not inverted<br>● 0b1: TXD output pin signal inversion<br>**Note**<br>This bit can only be written when both USART_CR1.TE and USART_CR1.RE are 0. | RW |
| 4 | RXINV | Input signal inversion control bit<br>● 0b0: Not inverted<br>● 0b1: RXD input pin signal inversion<br>**Note**<br>This bit can only be written when both USART_CR1.TE and USART_CR1.RE are 0. | RW |
| 3 | CTSEN | Asynchronous serial port mode/synchronous serial port mode, CTS function enable bit<br>● 0b0: Disable<br>● 0b1: Enable<br>**Note**<br>● Non-asynchronous serial port mode and non-synchronous serial port mode must be set to 0.<br>● This bit can only be written when both USART_CR1.TE and USART_CR1.RE are 0. | RW |
| 2 | RTSEN | Asynchronous serial port mode/Synchronous serial port mode, RTS function enable bit<br>● 0b0: Disable<br>● 0b1: Enable<br>**Note**<br>● Non-asynchronous serial port mode and non-synchronous serial port mode must be set to 0.<br>● This bit can only be written when both USART_CR1.TE and USART_CR1.RE are 0. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 1 | TXDMAEN | TX DMAC hardware handshake signal enable bit<br>● 0b0: Turn off hardware handshake signal<br>● 0b1: Turn on the hardware handshake signal<br>**Note**<br>DMA mode is not supported when working in single-wire half-duplex. | RW |
| 0 | RXDMAEN | RX DMAC hardware handshake signal enable bit<br>● 0b0: Turn off hardware handshake signal<br>● 0b1: Turn on the hardware handshake signal<br>**Note**<br>DMA mode is not supported when working in single-wire half-duplex. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 18.5.4 Control Register 3 (USART_CR3)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x08 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | IRMOD | IRCW | IREN | Reserved | | | | | BCN | | | SCEN | BREAK | ABRE | LBDE | LINEN |
| Access | | | | | | | | | | | | | | | | | RW | RW | RW | | | | | | RW | | | RW | RW | RW | RW | RW |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:16* | *Reserved* | *All bits read as 0, Write as 0* | - |
| 15 | IRMOD | In infrared mode, 38K modulation mode<br>● 0b0: Output 38K carrier when data is 0<br>● 0b1: Output 38K carrier when data is 1<br>**Note**<br>Non-infrared mode setting is invalid. | RW |
| 14 | IRCW | In infrared mode, infrared carrier selection<br>● 0b0: 3/16 mode<br>● 0b1: 38K carrier mode<br>**Note**<br>Non-infrared mode setting is invalid. | RW |
| 13 | IREN | In infrared mode, the infrared mode enable bit<br>● 0b0: Infrared function module off<br>● 0b1: Infrared function module is turned on<br>**Note**<br>Non-infrared mode must be set to 0. | RW |
| *12:10* | *Reserved* | *All bits read as 0, Write as 0* | - |
| *9:8* | *Reserved* | - | - |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 7:5 | BCN | In smart card mode, sets the number of base clocks during a one-bit data transfer<br>● 0b000: 16<br>● 0b001: 32<br>● 0b010: 64<br>● 0b011: 128<br>● 0b100: 186<br>● 0b101: 256<br>● 0b110: 372<br>● 0b111: 512<br>**Note**<br>The non-smart card mode setting is invalid. | RW |
| 4 | SCEN | In smart card mode, the mode enable bit<br>● 0b0: Smart card mode is disabled<br>● 0b1: Enable smart card mode<br>**Note**<br>Non-smart card mode must be set to 0. | RW |
| 3 | BREAK | In LIN mode, send sync interval enable<br>● 0b0: The value written to the USART_TDR register represents the data sent normally<br>● 0b1: The value written into the USART_TDR register represents the number of dominant levels that need to be sent in the synchronization interval. The value should not be less than 13, and it will be automatically cleared by hardware after the transmission is completed.<br>**Note**<br>Non- LIN mode must be set to 0. | RW |
| 2 | ABRE | Asynchronous serial port mode/LIN mode, automatic baud rate detection is enabled<br>● 0b0: Stop detecting baud rate<br>● 0b1: Start detecting baud rate<br>**Note**<br>Non-asynchronous serial port mode and non- LIN mode must be set to 0. | RW |
| 1 | LBDE | LIN mode, the synchronization interval segment detection enable bit<br>● 0b0: Disable synchronization interval segment detection<br>● 0b1: Enable synchronization interval segment detection<br>**Note**<br>Non- LIN mode must be set to 0. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 0 | LINEN | LIN mode, the mode enable bit<br>● 0b0: Disable LIN Mode<br>● 0b1: Enable LIN mode<br>**Note**<br>Non- LIN mode must be set to 0. | RW |

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 18.5.5 Baud Rate Register (USART_BRR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x0C | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x0000FF7F | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | PSC | | DIVI | | | | | | | | FBME | DIVF | | | | | | |
| Access | | | | | | | | | | | | | | | RW | | RW | | | | | | | | RW | RW | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:18* | *Reserved* | - | - |
| 17:16 | PSC | When the internal clock source is used, the prescaler select bit<br>● 0b00: PCLK<br>● 0b01: PCLK/4<br>● 0b10: PCLK/16<br>● 0b11: PCLK/64 | RW |
| 15:8 | DIVI | Integer frequency divider register | RW |
| 7 | FBME | Fractional baud rate generation enable bit when internal baud rate generator is selected<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 6:0 | DIVF | Fractional frequency divider register<br>DIVF is only valid when FBME=1; it is recommended to enable the fractional frequency division when the module works at 8M and below. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 18.5.6 Timer Counter Register (USART_TOR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x10 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | TOR | | | | | | | | | | | |
| Access | | | | | | | | | | | | | | | | | | | | | RW | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:20* | *Reserved* | *All bits read as 0, Write as 0* | - |
| 19:0 | TOR | Timing counting time<br>Timing counting time = $t_{PCLK}$* $4^{PSC}$* (TOR + 1) | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 18.5.7 Status Register (USART_SR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x14 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000001 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | DWK | LBD | ABR | CTS | CTSV | AM | MPB | Reserved | TO | OR | FE | PE | RC | TC | TXE |
| Access | | | | | | | | | | | | | | | | | | RO | RO | RO | RO | RO | RO | RO | | RO | RO | RO | RO | RO | RO | RO |

| Bit | Name | Description | Access |
|---|---|---|---|
| *31:15* | *Reserved* | *All bits read as 0, Write as 0* | - |
| 14 | DWK | In asynchronous serial port mode/infrared mode/LIN mode, RXD line wake-up flag<br>● 0b0: No wake-up occurred.<br>● 0b0: Wake-up occurred.<br>Setting condition: Set to 1 by hardware.<br>Clearing condition: Write 0 to the USART_ICR.DWK bit. | RO |
| 13 | LBD | LIN mode, synchronous interval detection flag<br>● 0b0: No synchronize interval segment detected<br>● 0b1: Synchronize interval segment detected<br>Setting condition: Set to 1 by hardware.<br>Clearing condition: Write 0 to the USART_ICR.LBD bit. | RO |
| 12 | ABR | Automatic baud rate detection flag in asynchronous serial port mode/LIN mode<br>● 0b0: Detection not completed<br>● 0b1: Detection completed<br>Setting condition: Set to 1 by hardware.<br>Clearing condition: Write 0 to the USART_ICR.ABR bit. | RO |
| 11 | CTS | Asynchronous serial port mode/synchronous serial port mode, CTS interrupt flag bit<br>● 0b0: No CTS signal inversion<br>● 0b1: CTS signal inversion<br>Setting condition: Set to 1 by hardware.<br>Clearing condition: Write 0 to the USART_ICR.CTS bit. | RO |
| 10 | CTSV | Asynchronous serial port mode/synchronous serial port mode, the current level value of CTS pin<br>● 0b0: CTS pin is low level<br>● 0b1: CTS pin is high level | RO |
| 9 | AM | When multiprocessors communicate, the address match flag<br>● 0b0: Address mismatch<br>● 0b1: Address match | RO |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Description | Access |
|---|---|---|---|
| 8 | MPB | During multiprocessor communication, the multiprocessor flag bit<br>● 0b0: Received data is data<br>● 0b1: Received data is ID<br>**Note**<br>The MPB bit is only valid for multiprocessor communication. | RO |
| *7* | *Reserved* | *Read as 0, Write as 0* | - |
| 6 | TO | Timer match or overflow flag<br>● 0b0: Timer did not match or overflow<br>● 0b1: Timer has matched or overflowed<br>Setting condition: Set to 1 by hardware.<br>Clearing condition: Write 0 to the USART_ICR.TO bit. | RO |
| 5 | OR | Overflow error flag<br>● 0b0: No overflow<br>● 0b1: Overflow<br>Setting condition: When the received data register has a value, new data is received.<br>Clearing condition: Write 0 to the USART_ICR.OR bit.<br>**Note**<br>● RE=0 does not reset the OR bit.<br>● The data received after OR=1 will overwrite the previous data. | RO |
| 4 | FE | In asynchronous serial port mode/smart card mode/infrared mode/LIN mode, receive frame error flag<br>● 0b0: No receive frame error<br>● 0b1: Receive frame error occurred<br>Setting condition:<br>● Asynchronous serial port mode/infrared mode/LIN mode: the stop bit of the received data frame is 0.<br>● Smart card mode: sampled low level error signal flag.<br>Clearing condition: Write 0 to the USART_ICR.FE bit.<br>**Note**<br>● RE=0 does not reset the FE bit.<br>● The data received after FE=1 will overwrite the previous data. | RO |
| 3 | PE | In asynchronous serial port mode/smart card mode/infrared mode, receive data verification error flag<br>● 0b0: No parity error<br>● 0b1: Parity error occurred<br>Setting condition: a parity error occurs in the received data.<br>Clearing condition: Write 0 to the USART_ICR.PE bit.<br>**Note**<br>● RE=0 does not reset the PE bit.<br>● The data received after PE=1 will overwrite the previous data. | RO |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Description | Access |
|---|---|---|---|
| 2 | RC | Data receiving complete flag<br>● 0b0: Non-receive complete status<br>● 0b1: Receive completed<br>Setting condition: the stop bit is received,<br>Clearing condition:<br>● After the data in the USART_RDR register is read by software<br>● Write 0 to the USART_ICR.PE bit. | RO |
| 1 | TC | Send completion flag bit<br>● 0b0: During data transmission<br>● 0b1: Data transmission complete<br>Setting condition:<br>● Asynchronous mode and clock synchronous mode: When a frame of data transmission is completed (including stop bit), the transmit data register is empty (TXE=1).<br>● Smart card mode: After a certain time has elapsed after the last one byte of data is sent, FE=0 and the transmit data register is empty (TXE=1).<br>The specific timing for TC set is: 2.5 bit times after the check bit is transmitted.<br>Clearing condition: Write 0 to the USART_ICR.TC bit.<br>**Note**<br>In synchronous mode, this flag bit is not supported when using an external clock for communication | RO |
| 0 | TXE | Transmit data register empty flag<br>● 0b0: Data is not transferred to the shift register<br>● 0b1: Data is transferred to the shift register<br>Setting condition: After the data is transferred to the shift register, the hardware sets it to 1.<br>Clearing condition: After the data is written to the transmit data register, the hardware clears it to 0. | RO |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 18.5.8 Transmit Data Register (USART_TDR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x18 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x000001FF | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | MPID | TDR | | | | | | | | |
| Access | | | | | | | | | | | | | | | | | | | | | | | RW | RW | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:10* | *Reserved* | - | - |
| 9 | MPID | Multiprocessor ID bit<br>● 0b0: Send data<br>● 0b1: Transmit ID<br>**Note**<br>The MPID bit is only valid in multiprocessor communication, and must be set to 0 in other modes. | RW |
| 8:0 | TDR | Transmit Data<br>**Note**<br>The highest bit TDR[8] is valid only in the asynchronous serial port mode and the data length is set to nine bits. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 18.5.9 Interrupt Enable Register (USART_IER)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x1C | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | DWK | LBD | ABR | CTS | Reserved | AM | Reserved | | TO | OR | FE | PE | RC | TC | TXE |
| Access | | | | | | | | | | | | | | | | | | RW | RW | RW | RW | | RW | | | RW | RW | RW | RW | RW | RW | RW |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:15* | *Reserved* | *All bits read as 0, Write as 0* | - |
| 14 | DWK | Asynchronous serial port mode/infrared mode/LIN mode, RXD pin edge interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable<br>**Note**<br>USART_CR2.RXINV=0 corresponds to falling edge,<br>USART_CR2.RXINV=1 corresponds to rising edge. | RW |
| 13 | LBD | In LIN mode, synchronous interval segment interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 12 | ABR | Interrupt enable for auto baud rate detection in asynchronous serial mode/LIN mode<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 11 | CTS | Asynchronous serial port mode/synchronous serial port mode, CTS signal toggle interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| *10* | *Reserved* | *Read as 0, Write as 0* | - |
| 9 | AM | In the multi-machine communication mode, the interrupt enable after the address match<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| *8:7* | *Reserved* | *All bits read as 0, Write as 0* | - |
| 6 | TO | Timer/counter overflow interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 5 | OR | The error interrupt enable when the receive data register is not read and new data is received<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 4 | FE | Asynchronous serial port mode/smart card mode/infrared mode/LIN mode, the interrupt enable of receiving frame error<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 3 | PE | Asynchronous serial port mode/smart card mode/infrared mode, the check error interrupt enable after a frame of data is received<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 2 | RC | Interrupt request enable bit after a frame of data is received and written to the receive data register<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 1 | TC | Interrupt request enable bit when a frame of data transmission is completed<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 0 | TXE | Transmit buffer empty status interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 18.5.10 Interrupt Flag Clear Register (USART_ICR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x20 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00007A7E | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | DWK | LBD | ABR | CTS | Reserved | AM | Reserved | | TO | OR | FE | PE | RC | TC | Reserved |
| Access | | | | | | | | | | | | | | | | | | WO | WO | WO | WO | | WO | | | WO | WO | WO | WO | WO | WO | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:15* | *Reserved* | *All bits read as 0, Write as 0* | - |
| 14 | DWK | DWK flag clear<br>Write 0 to clear flag, write 1 has no effect; read always returns 1. | WO |
| 13 | LBD | LBD flag clear<br>Write 0 to clear flag, write 1 has no effect; read always returns 1. | WO |
| 12 | ABR | ABR flag clear<br>Write 0 to clear flag, write 1 has no effect; read always returns 1. | WO |
| 11 | CTS | CTS flag clear<br>Write 0 to clear flag, write 1 has no effect; read always returns 1. | WO |
| *10* | *Reserved* | *Read as 0, Write as 0* | - |
| 9 | AM | AM flag clear<br>Write 0 to clear flag, write 1 has no effect; read always returns 1. | WO |
| *8:7* | *Reserved* | *All bits read as 0, Write as 0* | - |
| 6 | TO | TO flag clear<br>Write 0 to clear flag, write 1 has no effect; read always returns 1. | WO |
| 5 | OR | OR flag clear<br>Write 0 to clear flag, write 1 has no effect; read always returns 1. | WO |
| 4 | FE | FE flag clear<br>Write 0 to clear flag, write 1 has no effect; read always returns 1. | WO |
| 3 | PE | PE flag clear<br>Write 0 to clear flag, write 1 has no effect; read always returns 1. | WO |
| 2 | RC | RC flag clear<br>Write 0 to clear flag, write 1 has no effect; read always returns 1. | WO |
| 1 | TC | TC flag clear<br>Write 0 to clear flag, write 1 has no effect; read always returns 1. | WO |
| *0* | *Reserved* | *Read as 0, Write as 0* | - |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 18.5.11 Receive Data Register (USART_RDR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x24 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | RDR | | | | | | | |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | RO | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:9* | *Reserved* | *All bits read as 0, Write as 0* | - |
| 8:0 | RDR | Receiving Data<br>**Note**<br>The highest bit RDR[8] is valid only in the asynchronous serial port mode and the data length is set to 9 bits. | RO |

### 18.5.12 Address Register (USART_ADDR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x28 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | ADDR | | | | | | | |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | RW | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:8* | *Reserved* | *All bits read as 0, Write as 0* | - |
| 7:0 | ADDR | Slave configuration address<br>This bit can only be written when both USART_CR1.TE and USART_CR1.RE are 0. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 18.5.13 Address Mask Register (USART_ADDMR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x2C | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | ADDM | | | | | | | |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | RW | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:8* | *Reserved* | *All bits read as 0, Write as 0* | - |
| 7:0 | ADDM | Slave address mask<br>● 0b0: Corresponding bit does not participate in the matching.<br>● 0b1: Corresponding bit participates in the matching.<br>**Note**<br>● Slave Address=USART_ADDR.ADDR & USART_ADDMR.ADDM.<br>● This bit can only be written when both USART_CR1.TE and USART_CR1.RE are 0. | RW |

### 18.5.14 Baud Rate Detection Value Register (USART_BDR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x30 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | BDR | | | | | | | | | | | |
| Access | | | | | | | | | | | | | | | | | | | | | RO | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:12* | *Reserved* | *All bits read as 0, Write as 0* | - |
| 11:0 | BDR | Baud rate detection value in asynchronous serial port mode/LIN mode<br>The communication baud rate can be calculated by the following formula:<br>$Baud = \frac{f_{PCLK}}{4^{PSC} \times (BDR+1)}$<br>**Note**<br>Only applicable to asynchronous serial port mode/LIN mode. | RO |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 18.5.15 Infrared Modulation Register (USART_IR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0xF0 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | DCNTH | | | | | | | | | | | Reserved | | | | | DCNTL | | | | | | | | | |
| Access | | | | | | RW | | | | | | | | | | | | | | | | RW | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:27* | *Reserved* | - | - |
| 26:16 | DCNTH | In infrared mode, high level count value<br>38K high level hold time t = $t_{PCLK}$* (DCNTH + 1) | RW |
| *15:11* | *Reserved* | - | - |
| 10:0 | DCNTL | In infrared mode, low level count value<br>38K low level hold time t = $t_{PCLK}$* (DCNTL + 1) | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

# 19 I2C

## 19.1 Introduction

Inter-Integrated Circuit Bus (I2C) is a two-wire bidirectional synchronous serial bus, which uses a clock line and a data line to transmit information between two devices connected to the bus, providing a simple and efficient method for data exchange between devices. Each device connected to the bus has a unique address, and any device can be used as both a master and a slave, but only one master is allowed at the same time. I2C standard is a real multi-master bus with a conflict detection mechanism and an arbitration mechanism. It can use the arbitration mechanism to avoid data conflicts and protect data when multiple masters request control of the bus at the same time.

I2C bus controller can meet various specifications of the I2C bus and support all transmission modes communicating with the I2C bus. The I2C logic can handle the transfer of bytes autonomously. It can keep track of the serial transfer, and there is a status register (I2C_STAT) that can reflect the status of the I2C bus controller and the I2C bus.

## 19.2 Features

I2C controller supports the following features:

- Support 4 working modes of master sending/receiving and slave sending/receiving
- Support 3 operating rates: Standard (100kbps)/Fast (400kbps)/Fast enhanced (1Mbps)
- Support 7-bit addressing function
- Support noise filtering function
- Support 3 slave addresses
- Support broadcast address
- Support interrupt status query function

## 19.3 Protocol Description

The I2C bus transfers information using the SCL (serial clock bus) and SDA (serial data bus) that connect devices. The master outputs a serial clock signal on the SCL line, and the data is transmitted on the SDA line, and each byte is transmitted (the highest bit MSB starts to be transmitted), followed by an acknowledge bit. One SCL clock pulse transfers one data bit.

### 19.3.1 Data Transmission on the I2C Bus

Usually the standard I2C transmission protocol consists of four parts: start (S) or repeated start signal (Sr), slave address and read and write bits, transmission data, and stop signal (P).

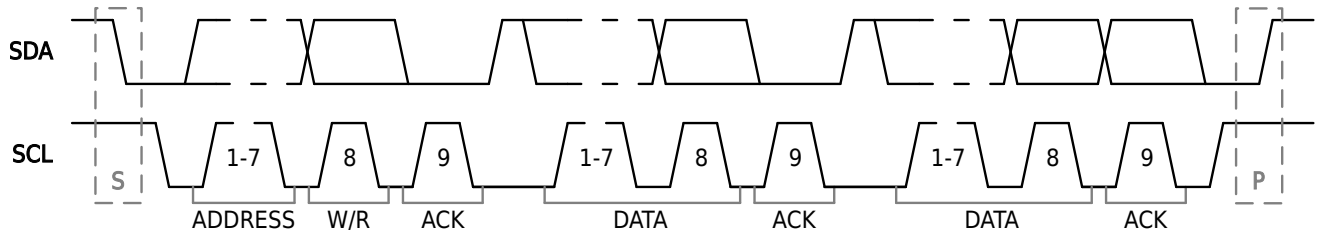



Figure 19-1 I2C Transmission Protocol


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

- Start signal, repeated start signal, stop signal

  Start signal: When the bus is in an idle state (the SCL and SDA lines are high at the same time), a signal from high to low appears on the SDA line, indicating that a start signal is generated.

  Repeated start signal: A repeated start signal occurs when there is no stop signal between two start signals. This method is used by a master to communicate with another slave or the same slave with a different transfer direction (for example: from writing to the device to reading from the device) without releasing the bus.

  Stop signal: When the SCL line is high, a low-to-high signal appears on the SDA line, which is generated a stop signal. The master sends a stop signal to the bus to end the data transfer.

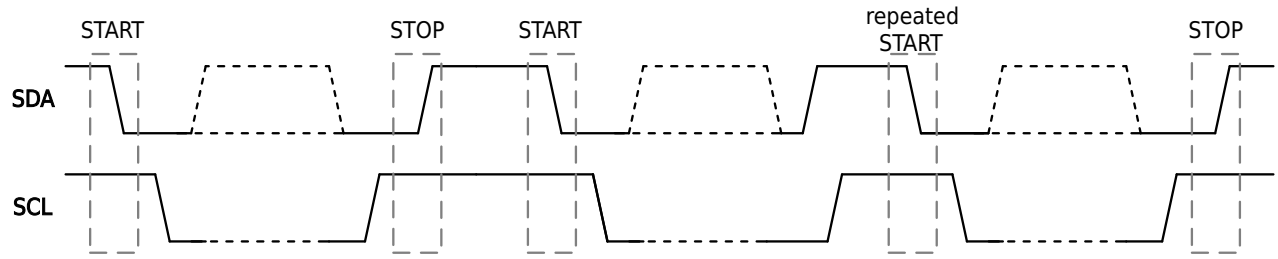



Figure 19-2 START, Repeated START and STOP

- Slave address and read/write bits

  When the start signal is generated, the master immediately transmits the first byte of data: 7-bit slave address + read/write bits, the read/write bits control the data transmission direction of the slave (0: write; 1: read). The addressed slave device will pull SDA low before the rising edge of the 9th SCL clock cycle as an acknowledgment (ACK).

- Transfer data

  During data transfer, one SCL clock pulse transfers one data bit, and the SDA line can only change when SCL is low.

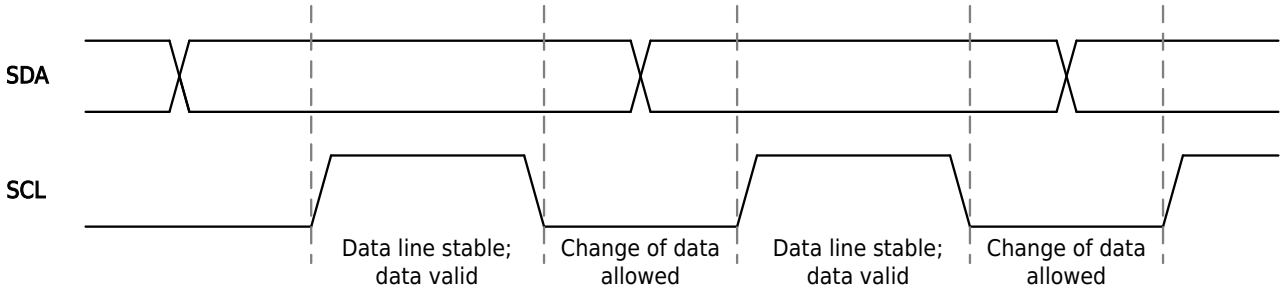



Figure 19-3 Bit Transfer on I2C Bus

## 19.3.2 Acknowledgment Signal on the I2C Bus

After transmitting each byte, the sending device releases control of SDA at the start of the 9th SCL clock cycle, while the receiving device generates an acknowledgment signal before the rising edge of that same SCL cycle. There are two types of acknowledgment bits:

- ACK: A low level signal. The receiving device pulls SDA low before the rising edge of the 9th SCL cycle and maintains it stable low throughout the high phase of this cycle. This signal indicates successful data reception by the receiver.
- NACK: A high level signal. The receiving device refrains from driving SDA during the 9th SCL cycle, leaving SDA persistently high. When the master acts as a transmitter and the slave generates a negative acknowledgment signal (NACK), the master may either generate a stop signal to terminate


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

transmission or issue a repeated start signal to initiate a new transfer. When the master acts as a receiver and generates NACK, the slave releases control of SDA. The master then generates either a stop or repeated start signal.

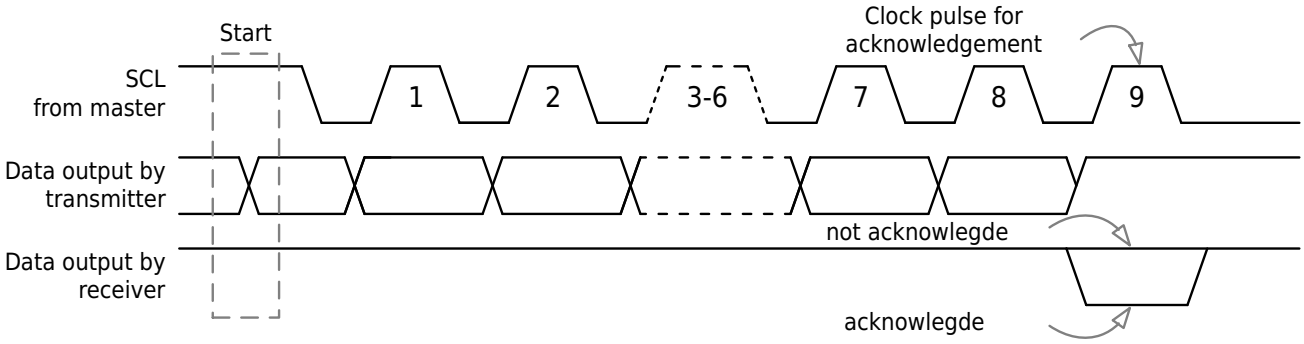



**Figure 19-4 Acknowledgment Signal on I2C Bus**

### 19.3.3 Arbitration on the I2C Bus

Arbitration on the I2C bus is divided into two parts:

- Synchronization on SCL line (clock synchronization)

  Since the I2C bus has the logic function of "AND", as long as one node on the SCL line sends a low level, the bus will show a low level. The bus can only behave as a high level when all nodes are sending a high level. Therefore, the clock low time is determined by the device with the longest clock low level period, and the clock high time is determined by the device with the shortest clock high level period. Due to the characteristics of I2C, when multiple nodes send clock signals at the same time, a unified clock signal is represented on the bus.

  When a slave can receive a byte of data at a higher rate but requires additional time to process it, the slave may suspend bus transmission by pulling SCL low after receiving the data and returning an ACK. The master then enters a wait state until the slave completes processing and releases SCL, after which the master proceeds with the next byte transfer. This slave clock stretching mechanism is implemented by controlling the I2C_CR.SI bit. Upon receiving data, I2C_CR.SI is automatically set to 1 by hardware, causing SCL to be held low by the slave. After processing the data, the slave clears I2C_CR.SI to release SCL.

- Arbitration on SDA line

  The arbitration on the SDA line is also due to the logical function of the "AND" of the I2C bus. After the master sends data, it decides whether to exit the competition by comparing the data on the bus. The master that loses the arbitration immediately switches to the unaddressed slave state to ensure that it can be addressed by the master that wins the arbitration. A master that loses arbitration continues to output clock pulses (on SCL) until the current serial byte has been sent. This principle can ensure that the I2C bus will not lose data when multiple masters attempt to control the bus.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

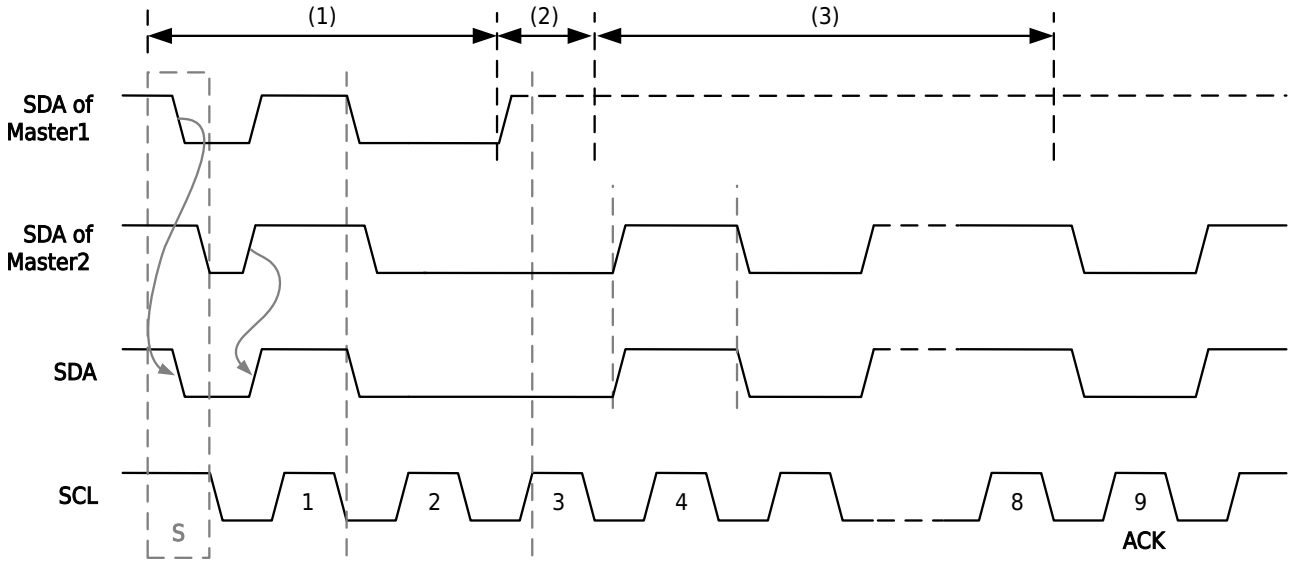



Figure 19-5 Arbitration on the I2C Bus

In the diagram above, the arbitration-triggering process is handled as follows:

- First, after transmitting the start signal, both Master 1 and Master 2 simultaneously transmit serial data.
- Second, during the low phase of the third SCL cycle, Master 1 transmits logic '1' while Master 2 transmits logic '0'. The logic '0' from Master 2 pulls SDA low, overriding the logic '1' of Master 1, thereby causing arbitration loss in Master 1 and transitioning it to an unaddressed slave mode.
- Third, after losing arbitration and entering the unaddressed slave mode, Master 1 continues clock pulse generation until completion of the current byte transmission, but ceases clock pulse generation for subsequent bytes. After Master 2 wins arbitratoin, subsequent data transfer on SDA is initiated by Master 2.

## 19.4 Functional Description

The I2C bus uses two wires to transfer information between devices connected to the bus SCL (serial clock line) and SDA (serial data line). Filtering logic can filter glitches on the data bus to protect data integrity. Since there are only non-directional ports, I2C requires that the pins used be configured as open-drain outputs. Each device connected to the bus can be addressed by a specific address. The I2C standard is a true multi-master bus with a collision detection mechanism and an arbitration mechanism. It prevents data collisions when two or more masters start transmitting data at the same time. I2C bus can be queried in the status register.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 19.4.1 Function Block Diagram

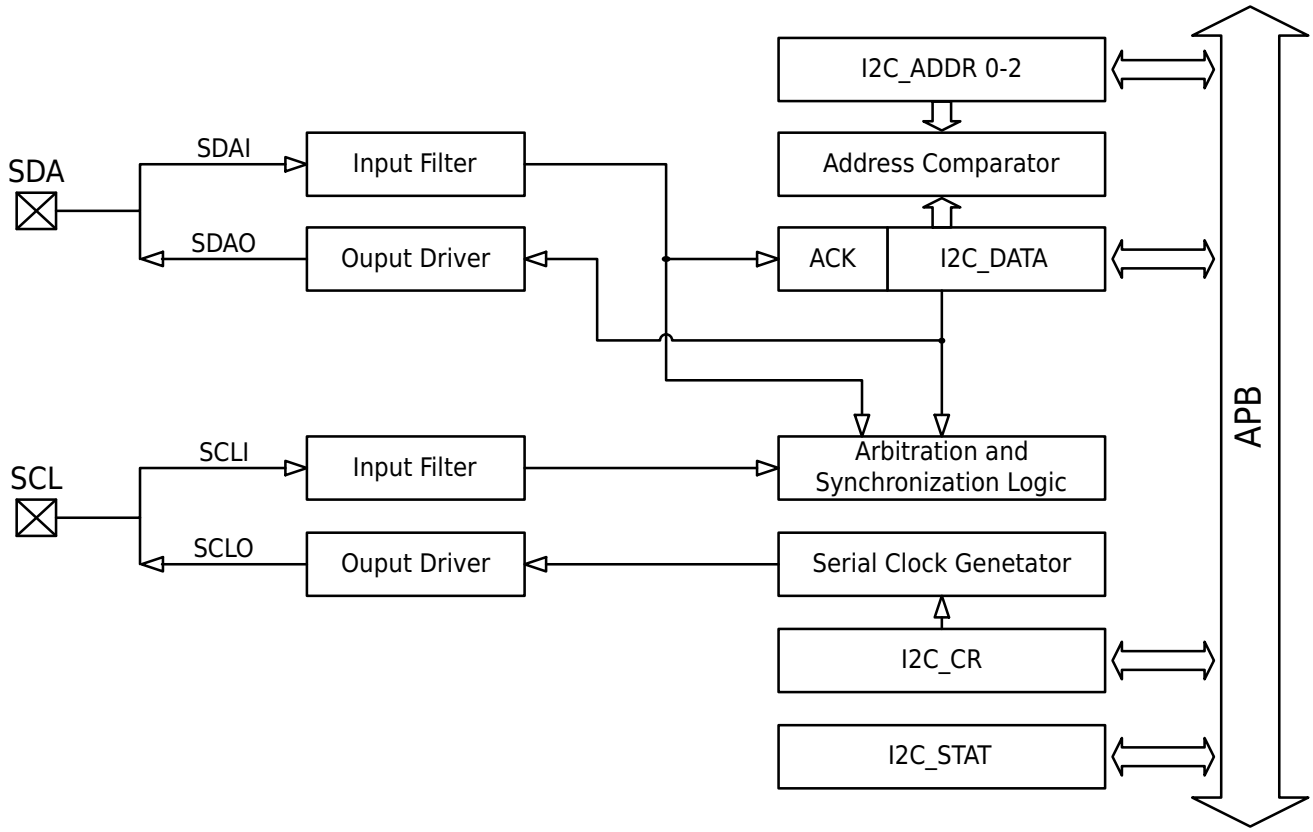



Figure 19-6 I2C Function Block Diagram

### 19.4.2 Serial Clock Generator

The serial clock generator uses an 8-bit counter as the baud rate generator. The frequency relationship between the SCL signal and the PCLK signal is $f_{SCL}= f_{PCLK}/8/(I2C\_TM+1)$, where I2C_TM should be greater than 0.

The following table lists the output frequency value of SCL signal when PCLK frequency is combined with I2C_TM.

Table 19-1 I2C Clock Signal Baud Rate

| PCLK (kHz) | I2C_TM | | | | | | |
|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1000 | 62 | 41 | 31 | 25 | 20 | 17 | 15 |
| 2000 | 125 | 83 | 62 | 50 | 41 | 35 | 31 |
| 4000 | 250 | 166 | 125 | 100 | 83 | 71 | 62 |
| 6000 | 375 | 250 | 187 | 150 | 125 | 107 | 93 |
| 8000 | 500 | 333 | 250 | 200 | 166 | 142 | 125 |
| 10000 | 625 | 416 | 312 | 250 | 208 | 178 | 156 |
| 12000 | 750 | 500 | 375 | 300 | 250 | 214 | 187 |
| 14000 | 875 | 583 | 437 | 350 | 291 | 250 | 218 |
| 16000 | 1000 | 666 | 500 | 400 | 333 | 285 | 250 |

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 19.4.3 Input Filter

The input signal is synchronous with PCLK, and the spike pulse signal lower than the period of PCLK will be filtered out. By adjusting the value of I2C_CR.H1M, the intensity of filtering can be changed.

- When used as the master, if the value of I2C_TM is less than or equal to 9, I2C_CR.H1M should be set to 1 to make simple filtering; if the value of I2C_TM is greater than 9, I2C_CR.H1M should be set to 0 to make advanced filtering.
- When used as a slave, if the ratio of PCLK to SCL frequency is less than or equal to 30, I2C_CR.H1M should be set to 1 to make simple filtering; if the ratio of PCLK to SCL frequency is greater than 30, I2C_CR.H1M should be set to 0 to make advanced filtering.

### 19.4.4 Address Comparator

The I2C address comparator compares the received 7-bit address with the three slave addresses. According to the configuration of I2C_ADDR0.GC register, when the GC configuration is 0, the high 7 bits of the first received 8-bit byte are compared with the 7-bit Addr value of the I2C_ADDRx register. When the GC configuration is 1, the high 7 bits of the first received 8-bit byte are compared with the broadcast address (0x00). If any one is the same, the I2C_CR.SI bit will be set to 1 and an interrupt request will be generated. Query the I2C_MATCH register to obtain the matching address as the serial number.

### 19.4.5 Acknowledgement Flag Bit

I2C_CR.AA is the acknowledgement control bit. When the AA bit is 1, the I2C module responds with an acknowledgement signal (ACK) after receiving the data; when the AA bit is 0, the I2C module responds with a non-acknowledgement signal (NACK) after receiving the data.

### 19.4.6 Interrupt Generator

I2C_CR.SI is an interrupt flag. Whenever the value of the status register (I2C_STAT) changes (except for 0xF8), the SI flag will be set to 1. When an interrupt is generated, the status of the I2C bus can be obtained by querying the status register (I2C_STAT) to determine the actual source of the interrupt. In order to proceed to the next step, the SI flag must be cleared by software.

### 19.4.7 Operating Mode

The I2C component supports 8-bit bidirectional data transmission. The transmission rate can reach 100kbps in standard mode, 400kbps in fast mode, and 1Mbps in fast enhanced mode. It can operate in four modes: master sending mode, master receiving mode, slave receiving mode, and slave sending mode. There is also a special mode, broadcast call mode, which operates in a similar way to slave receive mode.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

- Master sending mode

  The master sends multiple bytes to the slave, and the master generates a clock, so it is necessary to fill in the set value in I2C_TM. I2C_CR.STA needs to be set to 1 in master sending mode. When the bus is free, the master initiates a start bit START. I2C_CR.SI is set to 1 if successful. Next, write the slave address and write bit (SLA+W) into I2C_DATA, and after clearing the SI bit, SLA+W is sent on the bus. After the master sends the SLA+W and receives the acknowledge signal (ACK) from the slave machine, the SI is set to 1. The data is then sent according to the user-defined format. After all the data is sent, set I2C_CR.STO to 1, clear the SI bit and send a STOP signal, or send a repeated start signal for a new round of data transmission.

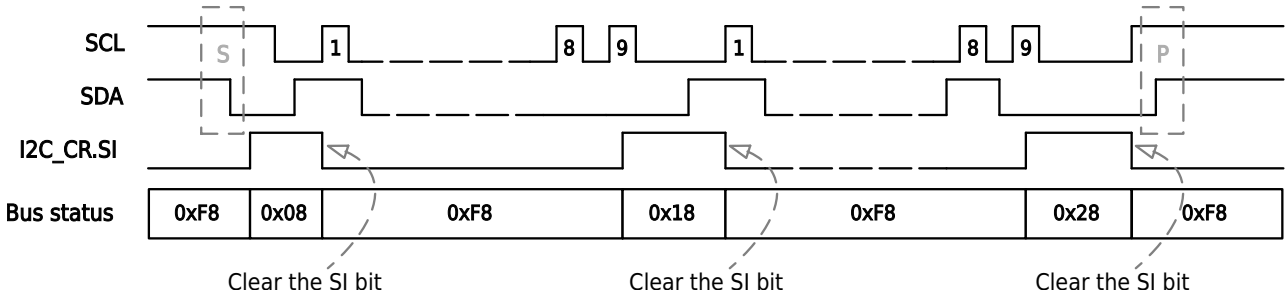



Figure 19-7 Data Synchronization Diagram of Master Sending Mode

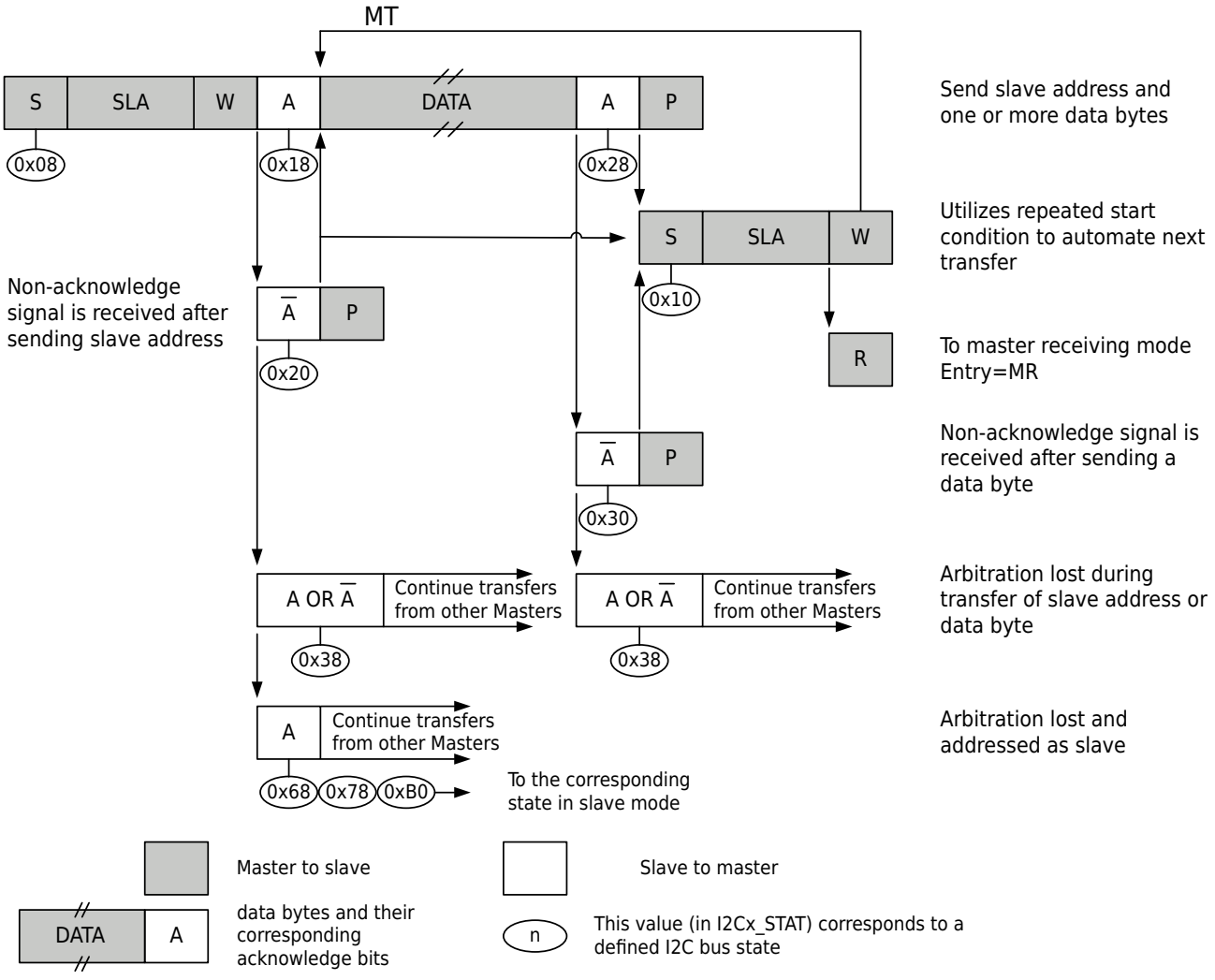



Figure 19-8 I2C Master Sending State Diagram


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

- Master receiving mode

  In master receiving mode, the data is transferred from the slave. The initialization setting is the same as the master sending mode, after the master sends the start bit, I2C_DATA should be written into the slave address and " read bit " (SLA+R). After receiving the acknowledge signal (ACK) from the slave device, I2C_CR.SI is set to 1. After SI is cleared to 0, the master starts to receive data from the slave. If I2C_CR.AA is 1, the master responds with an acknowledge signal (ACK) after receiving the data; if it is 0, the master responds with a non-acknowledge signal (NACK). After that, the master can send a stop signal or repeat the start signal to begin the next round of data transmission.

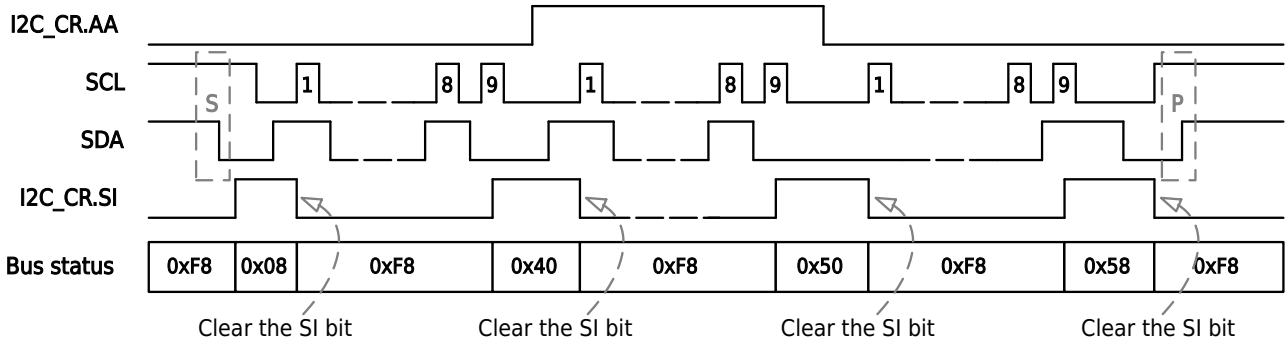



Figure 19-9 Data Synchronization Diagram of Master Receiving Mode

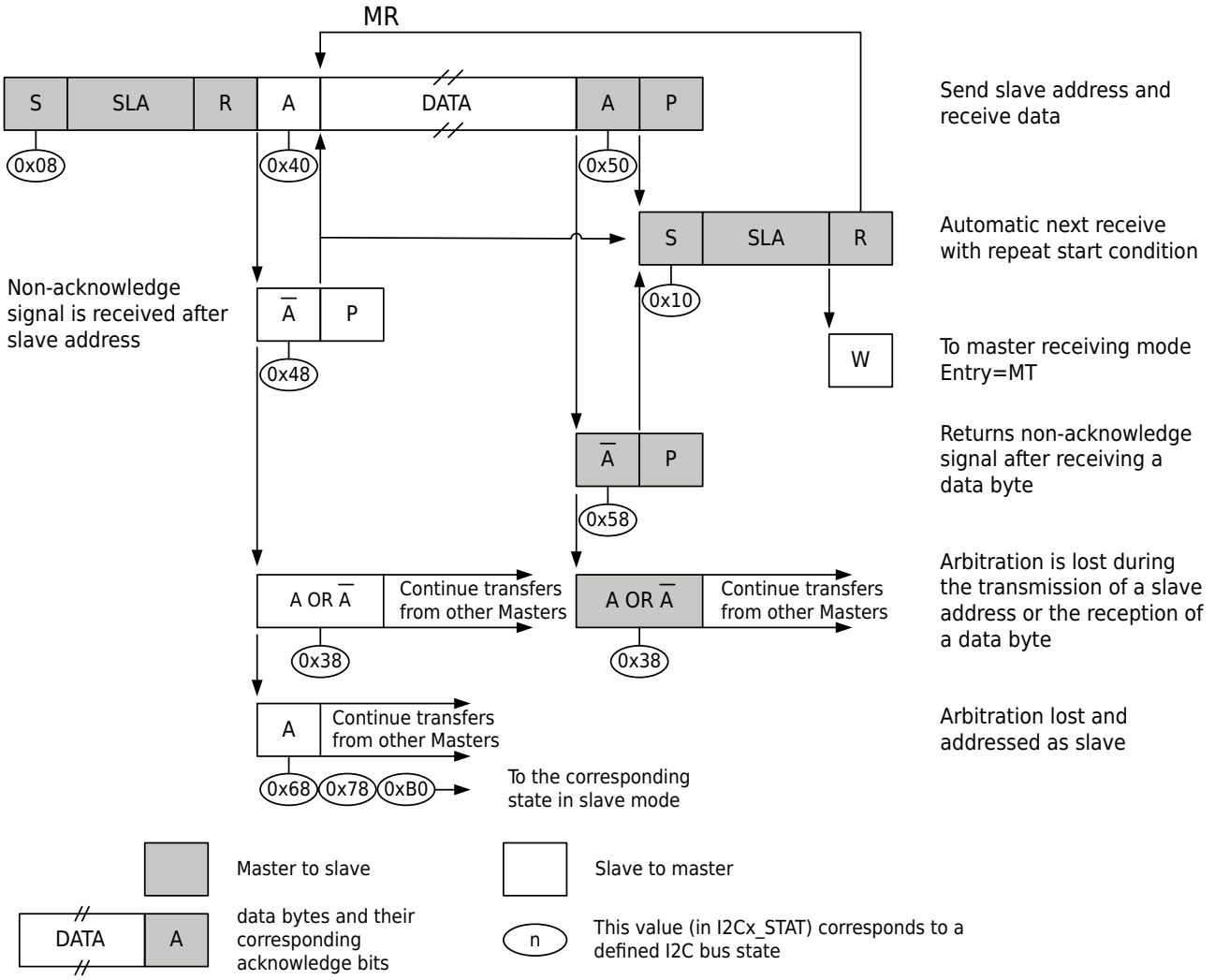



Figure 19-10 I2C Master Receiving State Diagram


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

- Slave receiving mode

  In slave receiving mode, the slave receives data from the master. Before the transmission starts, I2C_ADDRx should be written to the slave address, and I2C_CR.AA is set to 1 to respond to the addressing of the master. After the above initialization, the slave enters idle mode and waits for a "write" signal (SLA+W). If the master arbitration fails, it will directly enter the mode where the unaddressed slave receives data.

  When the slave is addressed by the "write" signal SLA+W, the SI bit needs to be cleared to receive data from the master. If during the transmission process I2C_CR.AA = 0, the slave device will send a non-acknowledge signal (NACK) after receiving the next byte. The slave device will also switch to an unaddressed slave mode, terminate the connection with the master, stop receiving data, and keep the I2C_DATA as the data it previously received. Slave address recognition can be restored by setting AA, which means that the AA bit temporarily detaches the I2C module from the I2C bus.

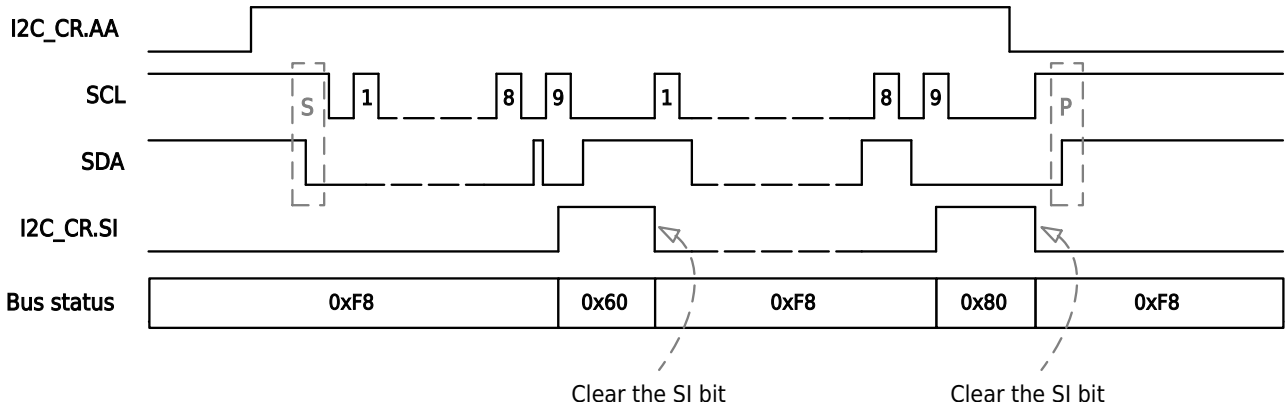



Figure 19-11 Slave Receiving Mode Data Synchronization Diagram

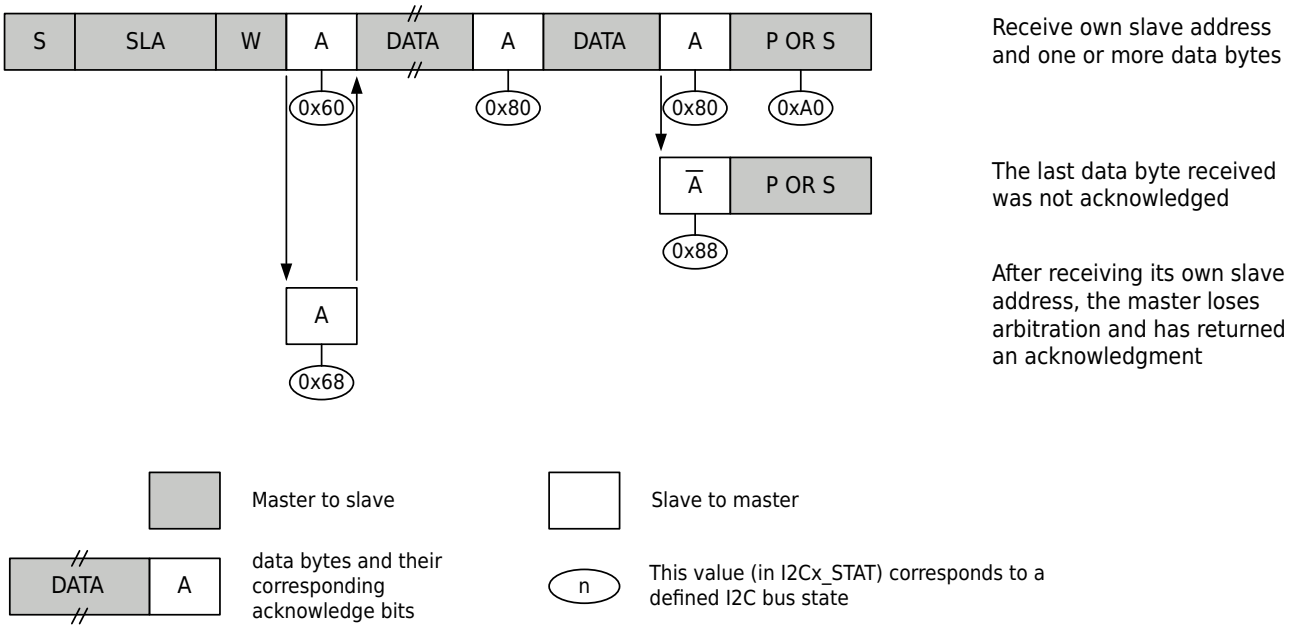



Figure 19-12 Slave Receiving State Diagram


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

- Slave sending mode

  In the slave sending mode, the data is sent from the slave to the master. After initializing the I2C_ADDRx and I2C_CR.AA values, the device waits until its own address is addressed by the "read" signal (SLA+R). If the master arbitration fails, it can also enter the mode where the unaddressed slave devices send data.

  When the slave is addressed by the "read" signal SLA+R, it needs to clear SI to send data to the master. Usually, the master will send an acknowledgement signal (ACK) after receiving each byte of data.

  If the I2C_CR.AA register of the master is cleared during the transmission process, the slave will stop sending data and release SDA, and will change itself to an unaddressed slave.

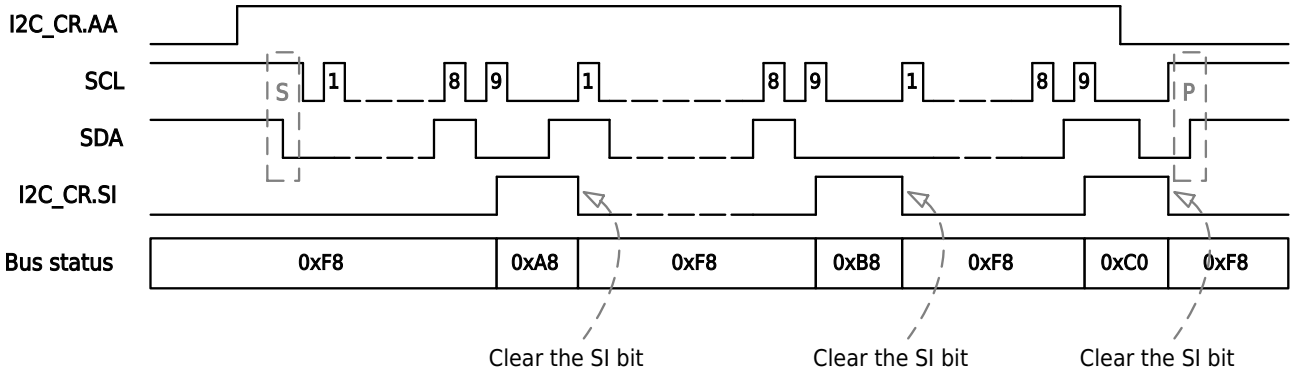



Figure 19-13 Slave Sending Mode Data Synchronization Diagram

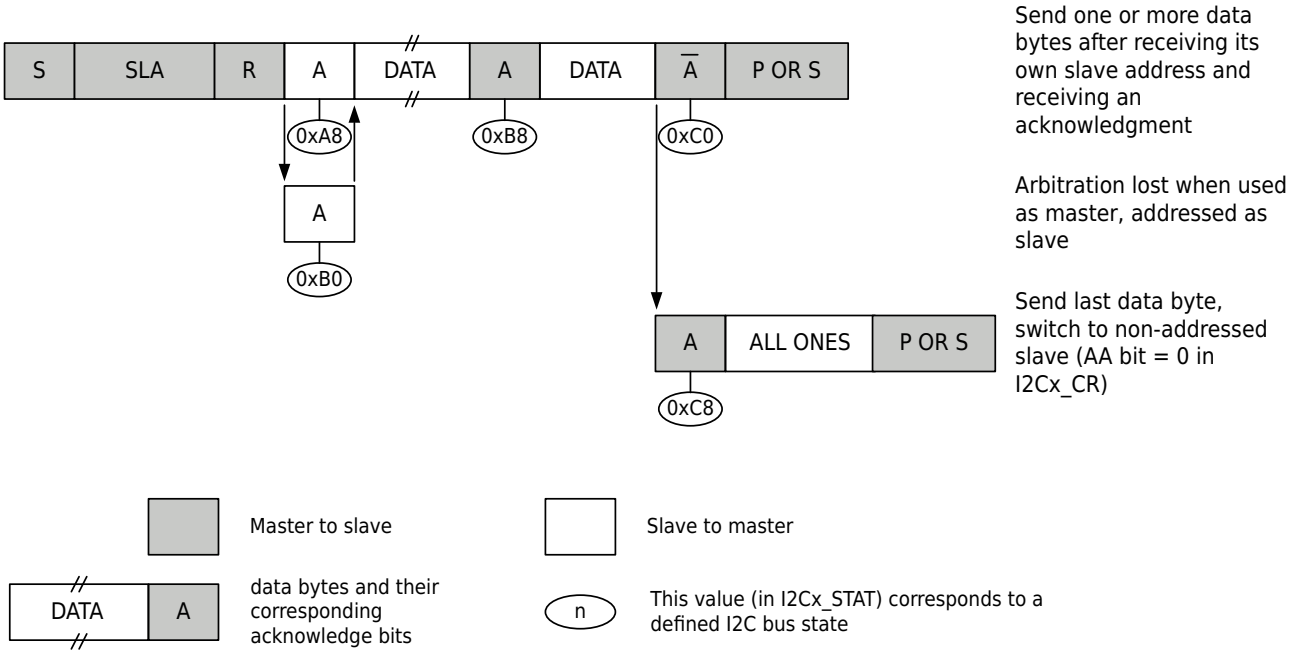



Figure 19-14 I2C Slave Sending State Diagram


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

- Broadcast call mode

  Broadcast call mode is a special slave receiving mode, The addressing mode is 0x00, and the slave address and reading and writing are both 0. When both I2C_ADDR0.GC and I2C_CR.AA are set to 1, the broadcast call receiving mode is enabled. In this mode, the I2C_STAT value is different from the normal slave receiving mode I2C_STAT value. Arbitration failure may also enter broadcast call mode.

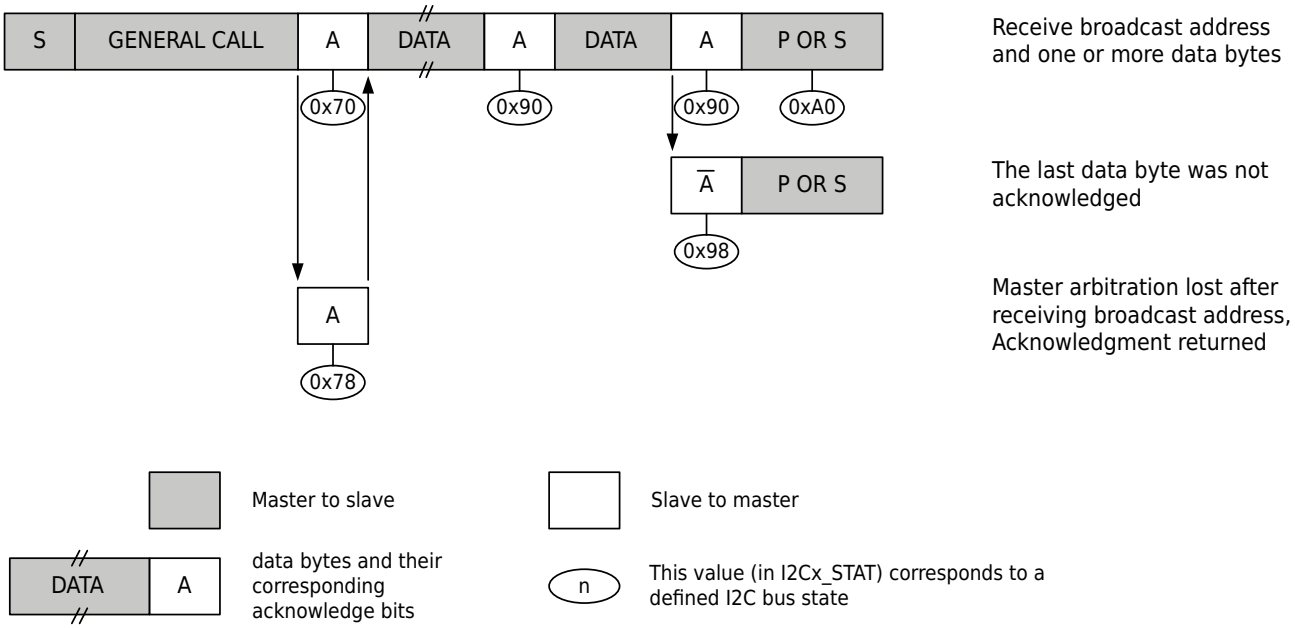



Figure 19-15 I2C Broadcast Call State Diagram

## 19.4.8 Status Code Expression

There are two special states in the I2C status register:

- 0xF8: This status code indicates that no relevant information is available, because the serial interrupt flag"SI" has not been set. This condition occurs between other states and before the I2C module has started to perform a serial transfer.
- 0x00: This status code indicates that a bus error occurred during I2C serial transmission. A bus error occurs when a START or STOP condition occurs at an illegal position in a format frame. These illegal locations refer to address bytes, data bytes, or acknowledge bits during serial transfers. When external disturbances affect the internal I2C block signals, bus errors can also occur. SI is set when a bus error occurs.

Table 19-2 I2C Status Code Expression

| Status Code | Description |
|---|---|
| **Master Sending Mode** | |
| 0x08 | Start condition sent |
| 0x10 | Repeated start condition sent |
| 0x18 | SLA+W sent, ACK received |
| 0x20 | SLA+W sent, NACK received |
| 0x28 | Data in I2C_DATA has been sent, ACK has been received |
| 0x30 | Data in I2C_DATA has been sent, NACK has been received |
| 0x38 | Loss of Arbitration when SLA+ reads or writes data bytes |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Status Code | Description |
|---|---|
| **Master Receiving Mode** | |
| 0x08 | Start condition sent |
| 0x10 | Repeated start condition sent |
| 0x38 | Loss of arbitration in NACK (that is, the master detects that SDA has been pulled low by another device when responding with ACK) |
| 0x40 | SLA+R sent, ACK received |
| 0x48 | SLA+R sent, NACK received |
| 0x50 | Data byte has been received, ACK has been returned |
| 0x58 | Data byte has been received, NACK has been returned |
| **Slave Receiving Mode** | |
| 0x60 | Has received its own SLA+W and returned ACK |
| 0x68 | When the master is in SLA+ read and write lost arbitration, has received its own SLA+W, and has returned ACK |
| 0x80 | The previous addressing used its own slave address, received data bytes, and returned ACK |
| 0x88 | The previous addressing used its own slave address, received data bytes, and returned NACK |
| 0xA0 | During static addressing, when the stop condition or the repeated start condition is received. |
| **Slave Sending Mode** | |
| 0xA8 | Received its own SLA+R and returned ACK |
| 0xB0 | When the master loses arbitration, has received its own SLA+R, and has returned ACK |
| 0xB8 | Data sent, ACK received |
| 0xC0 | Data bytes sent, NACK received |
| 0xC8 | Loaded data bytes have been sent, ACK has been received |
| **Broadcast Call Mode** | |
| 0x70 | Received broadcast address (0x00), returned ACK |
| 0x78 | When the master is in SLA+ read and write lost arbitration, the broadcast address has been received, and ACK has been returned |
| 0x90 | The previous addressing used the broadcast address, data bytes have been received, and ACK has been returned |
| 0x98 | The previous addressing used the broadcast address, data bytes have been received, and NACK has been returned |
| 0xA0 | During static addressing, when the stop condition or the repeated start condition is received. |
| **Other Miscellaneous Status** | |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Status Code | Description |
|---|---|
| 0xF8 | No relevant status information available, SI=0 |
| 0x00 | A bus error occurs during transmission, or external interference causes I2C to enter an undefined state |

## 19.5 Operation Example

### 19.5.1 Master Sending Example

#### Procedure

**Step1.** Refer to the relevant description of the port digital multiplexing function in the "General-Purpose I/O (GPIO)" chapter. Map SCL and SDA to the required pins; and configure the SCL and SDA pins to the open-drain output mode.

**Step2.** Set SYSCTRL module register PERI_CLKEN0.I2Cx to 1 to enable the I2C working clock.

**Step3.** Write 0 and 1 to SYSCTRL module register PERI_RESET0.I2Cx in sequence to reset the I2C module.

**Step4.** Configure I2C_TM, so that the clock rate of SCL meets the application requirements.

**Step5.** Set I2C_TMRUN to 1 to enable the SCL clock generator.

**Step6.** Set I2C_CR.ENS to 1 to enable the I2C module.

**Step7.** Set I2C_CR.STA to 1, the bus tries to send the Start signal.

**Step8.** Wait for I2C_CR.SI to become 1 the Start signal has been sent to the bus, set I2C_CR.STA to 0.

**Step9.** Query I2C_STAT, if the register value is 0x08 or 0x10, continue to the next step, otherwise, perform error handling.

**Step10.** Write SLA+W to I2C_DATA, set I2C_CR.SI to 0 and send SLA+W.

**Step11.** Wait for I2C_CR.SI to become 1, SLA+W has been sent to the bus.

**Step12.** Query I2C_STAT, if the register value is 0x18, proceed to the next step. Otherwise, perform error handling.

**Step13.** Write the data to be sent to I2C_DATA, set I2C_CR.SI to 0, and send the data.

**Step14.** Wait for I2C_CR.SI to become 1, the data has been sent to the bus.

**Step15.** Query I2C_STAT, if the register value is 0x28, proceed to the next step. Otherwise, perform error handling.

**Step16.** If the data to be sent is not completed, then jump to step13 to continue execution.

**Step17.** Set I2C_CR.STO to 1, set I2C_CR.SI to 0, and the bus tries to send a Stop signal.

### 19.5.2 Master Receiving Example

#### Procedure

**Step1.** Refer to the relevant description of the port digital multiplexing function in the "General-Purpose I/O (GPIO)" chapter. Map SCL and SDA to the required pins; and configure the SCL and SDA pins to the open-drain output mode.

**Step2.** Set SYSCTRL module register PERI_CLKEN0.I2Cx to 1 to enable the I2C working clock.

**Step3.** Write 0 and 1 to SYSCTRL module register PERI_RESET0.I2Cx in sequence to reset the I2C module.

**Step4.** Configure I2C_TM, so that the clock rate of SCL meets the application requirements.

**Step5.** Set I2C_TMRUN to 1 to enable the SCL clock generator.

**Step6.** Set I2C_CR.ENS to 1 to enable the I2C module.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

**Step7.** Set I2C_CR.STA to 1, the bus tries to send the Start signal.

**Step8.** Wait for I2C_CR.SI to become 1, the Start signal has been sent to the bus, set I2C_CR.STA to 0.

**Step9.** Query I2C_STAT. If the register value is 0x08 or 0x10, continue to the next step, otherwise perform error handling.

**Step10.** Write SLA+R to I2C_DATA, set I2C_CR.SI to 0, and send SLA+R.

**Step11.** Wait for I2C_CR.SI to become 1, SLA+R has been sent to the bus.

**Step12.** Query I2C_STAT, if the register value is 0x40, continue to the next step, otherwise, perform error handling.

**Step13.** Set I2C_CR.AA to 1 to enable the acknowledge flag.

**Step14.** Set I2C_CR.SI to 0, the slave sends data, and the master sends ACK or NACK according to I2C_CR.AA.

**Step15.** Wait for I2C_CR.SI to become 1, read the received data from I2C_DATA.

**Step16.** Query I2C_STAT, if the register value is 0x50 or 0x58, continue to the next step, otherwise, perform error handling.

**Step17.** If the data to be received is only the last byte, set I2C_CR.AA to 0 and enable the non-acknowlege flag.

**Step18.** If the data to be received is not completed, jump to step14 to continue execution.

**Step19.** Set I2C_CR.STO to 1, set I2C_CR.SI to 0, and the bus tries to send a Stop signal.

### 19.5.3 Slave Sending Example

#### Procedure

**Step1.** Refer to the relevant description of the port digital multiplexing function in the "General-Purpose I/O (GPIO)" chapter. Map SCL and SDA to the required pins; and configure the SCL and SDA pins to the open-drain output mode.

**Step2.** Set SYSCTRL module register PERI_CLKEN0.I2Cx to 1 to enable the I2C working clock.

**Step3.** Write 0 and 1 to SYSCTRL module register PERI_RESET0.I2Cx in sequence to reset the I2C module.

**Step4.** Set I2C_CR.ENS to 1 to enable the I2C module.

**Step5.** Configure I2C_ADDRx as the slave address.

**Step6.** Set I2C_CR.AA to 1 to enable the acknowledge flag.

**Step7.** Wait for I2C_CR.SI to become 1 and be addressed by SLA+R;

**Step8.** Query I2C_STAT, if the value of this register is 0xA8, continue to the next step, otherwise, perform error handling.

**Step9.** Write the data to be sent to I2C_DATA, set I2C_CR.SI to 0, and send the data.

**Step10.** Wait for I2C_CR.SI to become 1, the data has been sent to the bus.

**Step11.** Query I2C_STAT, if the value of this register is 0xB8 or 0xC0, continue to the next step, otherwise, perform error handling.

**Step12.** If the data to be sent is not completed, then jump to step9 to continue execution.

**Step13.** Set I2C_CR.AA to 0 and set I2C_CR.SI to 0.

### 19.5.4 Slave Receiving Example

#### Procedure

**Step1.** Refer to the relevant description of the port digital multiplexing function in the "General-Purpose I/O (GPIO)" chapter. Map SCL and SDA to the required pins; and configure the SCL and SDA pins to the open-drain output mode.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

**Step2.** Set SYSCTRL module register PERI_CLKEN0.I2Cx to 1 to enable the I2C working clock.

**Step3.** Write 0 and 1 to SYSCTRL module register PERI_RESET0.I2Cx in sequence to reset the I2C module.

**Step4.** Set I2C_CR.ENS to 1 to enable the I2C module.

**Step5.** Configure I2C_ADDRx as the slave address.

**Step6.** Set I2C_CR.AA to 1 to enable the acknowledge flag.

**Step7.** Wait for I2C_CR.SI to become 1 and be addressed by SLA+W.

**Step8.** Query I2C_STAT, if the register value is 0x60, continue to the next step, otherwise, perform error handling.

**Step9.** Set I2C_CR.SI to 0, the master sends data, and the slave returns ACK or NACK according to I2C_CR.AA.

**Step10.** Wait for I2C_CR.SI to become 1, read the received data from I2C_DATA.

**Step11.** Query I2C_STAT, if the register value is 0x80, continue to the next step, otherwise, perform error handling.

**Step12.** If the data to be received is not completed, jump to step9 to continue execution.

**Step13.** Set I2C_CR.AA to 0 and set I2C_CR.SI to 0.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

## 19.6 Register Description

### 19.6.1 Register List

- I2C0 base address: 0x40000400
- I2C1 base address: 0x40008400

Table 19-3 I2C Bus (I2C) Register Offset

| Offset | Register | Description | Width |
|---|---|---|---|
| 0x00 | I2C_TMRUN | I2C Baud Rate Generator Enable Register | 32 |
| 0x04 | I2C_TM | I2C Baud Rate Counter Configuration Register | 32 |
| 0x08 | I2C_CR | I2C Configuration Register | 32 |
| 0x0C | I2C_DATA | I2C Data Register | 32 |
| 0x14 | I2C_STAT | I2C Status Register | 32 |
| 0x10 | I2C_ADDR0 | I2C Slave Address Register 0 | 32 |
| 0x20 | I2C_ADDR1 | I2C Slave Address Register 1 | 32 |
| 0x24 | I2C_ADDR2 | I2C Slave Address Register 2 | 32 |
| 0x28 | I2C_MATCH | I2C Slave Address Match Register | 32 |

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 19.6.2 Baud Rate Generator Enable Register (I2C_TMRUN)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | TME | |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | RW | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:1* | *Reserved* | - | - |
| 0 | TME | Baud rate generator enable control<br>● 0b0: Disable<br>● 0b1: Enable | RW |

### 19.6.3 Baud Rate Counter Configuration Register (I2C_TM)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x04 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | TM | | | | | | | |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | RW | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:8* | *Reserved* | - | - |
| 7:0 | TM | I2C bus SCL baud rate configuration<br>$f_{SCL}= f_{PCLK}/8/(TM+1)$, and TM>0 | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 19.6.4 Configuration Register (I2C_CR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x08 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | ENS | STA | STO | SI | AA | Reserved | H1M |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | RW | RW | RW | RW | RW | | RW |

| Bit | Name | Description | Access |
|---|---|---|---|
| *31:7* | *Reserved* | - | - |
| 6 | ENS | Module enable control<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 5 | STA | Bus state control<br>● 0b0: No function<br>● 0b1: Send START to the bus<br>**Note**<br>User program should set this bit to 0 after START transmission is completed. | RW |
| 4 | STO | Bus state control<br>● 0b0: No function<br>● 0b1: Send STOP to the bus<br>**Note**<br>The hardware automatically sets this bit to 0 after the STOP transmission is completed. | RW |
| 3 | SI | I2C interrupt flag<br>Writing 0 clears the I2C interrupt flag and makes the state machine to perform the next action; writing 1 has no effect on the value of this bit.<br>Read:<br>● 0b0: No I2C interrupt occurred<br>● 0b1: I2C interrupt has occurred | RW |
| 2 | AA | Acknowledge control<br>● 0b0: Send NAK during the acknowledge phase<br>● 0b1: Send ACK during the acknowledge phase | RW |
| *1* | *Reserved* | - | - |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Description | Access |
|---|---|---|---|
| 0 | H1M | I2C filter parameter configuration<br>● 0b0: advanced filtering, higher anti-interference performance<br>● 0b1: simple filtering, faster communication rate<br>**Note**<br>For related functions, please refer to the input filter. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 19.6.5 Data Register (I2C_DATA)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x0C | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | DAT | | | | | | | |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | RW | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:8* | *Reserved* | - | - |
| 7:0 | DAT | Data register<br>● In send mode, write the data to be sent<br>● In receive mode, read out received data | RW |

### 19.6.6 Status Register (I2C_STAT)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x14 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x000000F8 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | STAT | | | | | | | |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | RO | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:8* | *Reserved* | - | - |
| 7:0 | STAT | I2C state register<br>For the specific definitions of status codes, please refer to the Status Code Expression. | RO |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 19.6.7 Slave Address Register 0 (I2C_ADDR0)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x10 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | ADDR0 | | | | | | | GC |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | RW | | | | | | | RW |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:8* | *Reserved* | - | - |
| 7:1 | ADDR0 | I2C slave mode address 0 | RW |
| 0 | GC | Broadcast address acknowledgment enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |

### 19.6.8 Slave Address Register 1 (I2C_ADDR1)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x20 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | ADDR1 | | | | | | | Reserved |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | RW | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:8* | *Reserved* | - | - |
| 7:1 | ADDR1 | I2C slave mode address 1 | RW |
| *0* | *Reserved* | - | - |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 19.6.9 Slave Address Register 2 (I2C_ADDR2)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x24 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | ADDR2 | | | | | | | Reserved |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | RW | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:8* | *Reserved* | - | - |
| 7:1 | ADDR2 | I2C slave mode address 2 | RW |
| *0* | *Reserved* | - | - |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 19.6.10 Slave Address Match Register (I2C_MATCH)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x28 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | AD2F | AD1F | AD0F |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | RO | RO | RO |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:3* | *Reserved* | - | - |
| 2 | AD2F | I2C slave mode address 2 match flag bit<br>● 0b0: The device address received from the bus is not the same as ADDR2<br>● 0b1: The device address received from the bus is the same as ADDR2 | RO |
| 1 | AD1F | I2C slave mode address 1 match flag bit<br>● 0b0: The device address received from the bus is not the same as ADDR1<br>● 0b1: The device address received from the bus is the same as ADDR1 | RO |
| 0 | AD0F | I2C slave mode address 0 match flag bit<br>● 0b0: The device address received from the bus is not the same as ADDR0<br>● 0b1: The device address received from the bus is the same as ADDR0 | RO |

**Note**

The address match flag bit will be cleared under the following conditions:

- When the module is reset
- When the START/STOP is sent
- When the STOP is generated internally by the slave.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

# 20 HSI2C

## 20.1 Introduction

I2C is a two-wire bidirectional synchronous serial bus, which uses a clock line and a data line to transmit information between two devices connected to the bus, providing a simple and efficient method for data exchange between devices. Each device connected to the bus has a unique address, and any device can be used as both a master and a slave, but only one master is allowed at the same time. I2C standard is a real multi-master bus with a conflict detection mechanism and an arbitration mechanism. It can use the arbitration mechanism to avoid data conflicts and protect data when multiple masters request control of the bus at the same time.

This I2C module complies with the System Management Bus specification (Version 3). SMBus is a single-ended, simple two-wire bus typically used for low-bandwidth communications.

## 20.2 Features

HSI2C supports:

- Supports standard mode, fast mode, fast enhanced mode, and high-speed mode.
- Hs-mode in slave mode.
- Supports multi-master, including synchronization and arbitration. Multi-master means any number of masters can be present. In addition, the master and slave roles can be changed during the transmission gap (after a STOP is sent).
- Supports 7-bit and 10-bit addressing.
- Supports software reset, START byte, and device ID (software support is also required).
- Supports DMA transmission and reception.

HSI2C master supports:

- One word of command/transmit buffer (8-bit transmit data + 3-bit command), one word of receive buffer (8-bit receive data).
- The command buffer will wait for the I2C bus to be idle before starting the transmission.
- The command buffer can initialize (repeated) START and STOP conditions, as well as one or more master receive transmissions.
- The STOP condition can be generated from the command buffer or automatically when the transmit buffer is empty.
- Flexible receive data matching function can generate interrupts when data matching and/or discard unwanted data.
- Flags and optional interrupts to indicate Repeated START conditions, STOP conditions, arbitration loss, NACK, and command word errors.
- Supports configurable bus idle timeout and pin-stuck-low timeout.

HSI2C slave supports:

- Separate I2C slave registers to minimize software overhead for switching between master/slave.
- Supports SMBus alert address and broadcast address.
- Supports clock stretching.
- Supports transmit data register interrupt and receive data register interrupt.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

- Software-controllable ACK or NACK, configurable ACK/NACK bit clock stretching. This is to avoid transmit buffer underrun and receive buffer overflow errors.
- Supports flags and interrupts for end of packet, STOP condition, or bit error detection.
- Support Ultra low power wake-up mode.

## 20.3 Functional Description

### 20.3.1 System Block Diagram

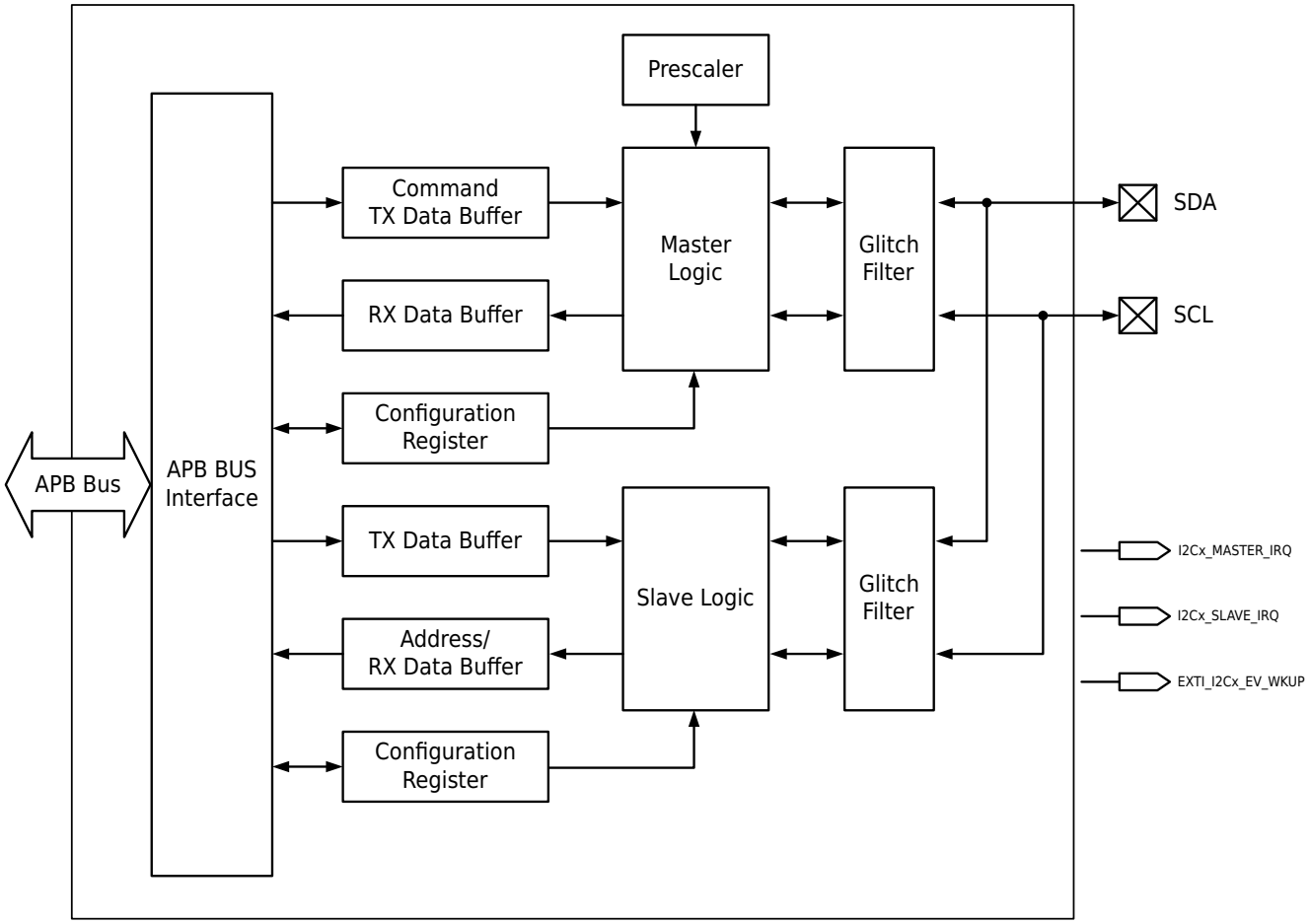



**Figure 20-1 I2C System Block Diagram**

### 20.3.2 Protocol Description

The I2C bus consists of a clock line (SCL) and a data line (SDA). All connection devices must be drain open output. The SCL and SDA lines are connected to external pull-up resistors, and the resistance value depends on the system application.

In general, a complete communication process consists of the following four parts:

1. START condition
2. Address transmission
3. Data transmission
4. STOP condition


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

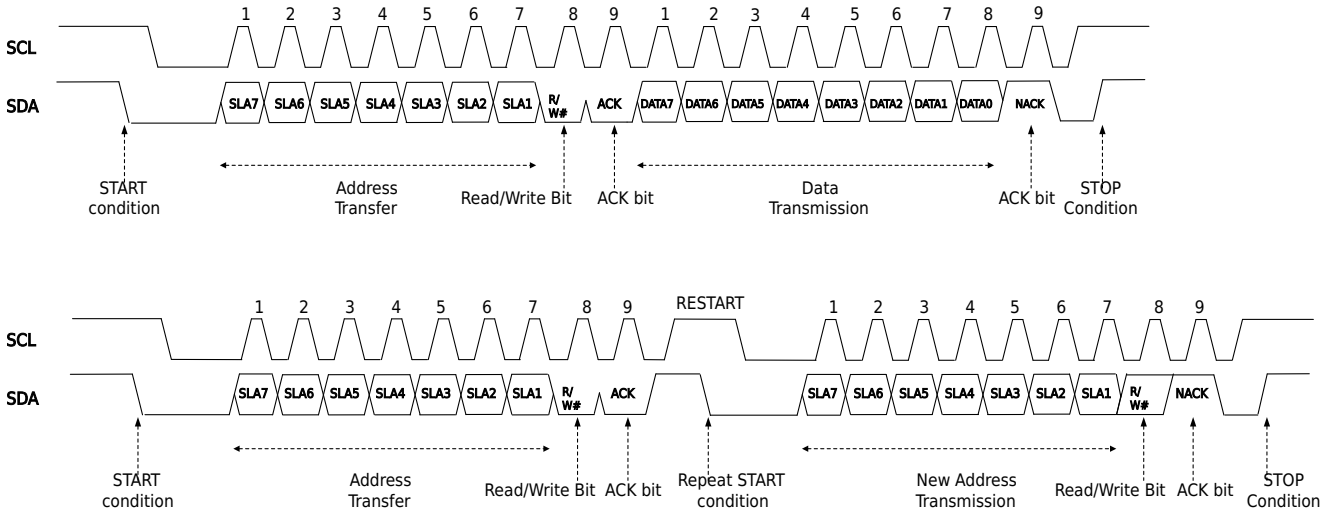



Figure 20-2 I2C Bus Timing Diagram

## 20.3.3 Master Mode

### 20.3.3.1 Transmit and Command Buffer Commands

The transmit buffer stores command data to initiate various I2C operations. As long as I2C is enabled (MCR.MEN bit is set), the following operations can be initiated by commands in the transmit buffer:

- Address byte and START or STOP condition expecting ACK or NACK.
- Transmit data (this is the default behavior for zero-extended byte writes to the transmit buffer).
- Receive a byte of data (can also be configured to discard the receive data and not store it in the receive buffer).
- STOP condition (can also be configured to send a STOP condition when the transmit buffer is empty).

Multiple transmit and receive commands can be inserted between start and stop conditions; transmit and receive commands must not be interleaved. Receive data commands, receive data, and discard commands can be interleaved to ensure that only the desired receive data is stored in the receive buffer.

The I2C master automatically sends a NACK on the last byte of a receive data command unless the next command in the buffer is also a receive data command. A NACK is also automatically sent if the transmit buffer is empty when a receive data command completes.

The I2C master enable (MCR.MEN) should be turned off after all transmits are complete.

### 20.3.3.2 Master Operation

Whenever I2C is started, it monitors the I2C bus to detect when the I2C bus is idle (MSR.BBF). The I2C bus is no longer considered idle if SCL or SDA is low; the I2C bus becomes idle if a STOP condition is detected or the bus idle timeout (MCFGR2.BUSIDLE configuration) occurs. After the I2C bus is idle, the transmit buffer is not empty, and then the I2C master initiates a transfer on the I2C bus. Refer to the following steps:

- The bus idle wait time is equal to (MCCR0.CLKLO + 1) multiplied by the prescaler (MCFGR1.PRESCALE).
- The START condition and address byte are sent using the timing configuration in the Master Clock Configuration 0 (MCCR0); if configured for high-speed mode, the timing configuration of the Master Clock Configuration 1 (MCCR1) is used.

According to the configuration of the transmit buffer, either a master transmit transfer or a master receive transfer is performed.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

- A NACK is sent on the last byte of a master receive transfer, unless the next command in the transmit buffer is also a receive data command and the transmit buffer is not empty.
- The sending of a repeated START or STOP condition is configured by the transmit buffer and/or MCFGR1.AUTOSTOP. The repeated START condition can change the timing configuration register used.

When the I2C master is disabled (MCR.MEN is cleared or mode is entered), the I2C continues to empty the transmit buffer until a STOP condition is sent. However, the I2C will no longer pause the I2C bus waiting for the transmit or receive buffer, and the I2C automatically generates a STOP condition after the transmit buffer is empty.

The I2C master can stop the I2C bus under the following specific conditions; this causes SCL to be pulled low continuously on the first bit of the byte until the condition is removed:

- The I2C master is enabled and busy, the transmit buffer is empty, and MCFGR1.AUTOSTOP is cleared.
- The I2C master is enabled and receiving data, the receive data has not been discarded (due to a command or receive data match), and the receive buffer is full.

#### 20.3.3.3 Receive Buffer and Data Matching

The receive buffer is used to store receive data during the master receive transmission. The receive data can also be configured to discard the receive data instead of storing it in the receive buffer; this is configured by the command word in the transmit buffer.

The receive data supports the receive data matching function, which can match the received data with one of the two bytes or the mask data byte. The data match function can also be configured to compare only the previous one or two received data words since the last (repeated) START condition. Received data that is discarded due to a command word will not cause a data match, and the match on the first received data word is delayed until after the discarded data is received. The receiver match function can also be configured to discard all received data until a data match is detected, using the MCFGR1.RDMO control bit. Clear the MCFGR1.RDMO control bit after data match and clear MCFGR1.RDMO before clearing MSR.DMF to allow all subsequent data to be received.

#### 20.3.3.4 Timing Parameters

The I2C master can configure the following timing parameters. The parameters for high-speed mode (Master Clock Configuration 1 (MCCR1)) and other modes (Master Clock Configuration 0 (MCCR0)) can be configured separately. This allows high-speed mode master code to be transmitted using standard timing parameters, then switching to high-speed mode timing (after the repeated START), until the next STOP condition occurs.

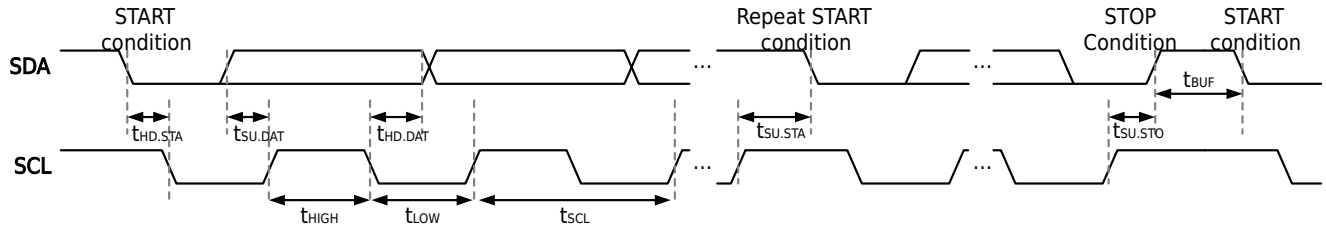



Figure 20-3 I2C Bus Timing Parameters Diagram

The timing parameters of the I2C master are usually related to the I2C function clock cycle, and the parameter configuration needs to meet the specifications required by the corresponding mode.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Table 20-1 Host Timing Parameters

| Parameters | Symbol | Timing Parameters (I2C function clock cycle) |
|---|---|---|
| SCL clock cycle | $t_{SCL}$ | (CLKHI + CLKLO + 2 + SCL_LATENCY) x $2^{PRESCALE}$ |
| SCL clock low time | $t_{LOW}$ | (CLKLO + 1) x $2^{PRESCALE}$ |
| SCL clock high time | $t_{HIGH}$ | (CLKHI + 1 + SCL_LATENCY) x $2^{PRESCALE}$ |
| (Repeated) START condition hold time | $t_{HD:STA}$ | (SETHOLD + 1) x $2^{PRESCALE}$ |
| Data setup time | $t_{SU;DAT}$ | (SDA_LATENCY + 1) x $2^{PRESCALE}$ |
| Data hold time | $t_{HD;DAT}$ | (DATAVD + 1) x $2^{PRESCALE}$ |
| Repeated START condition setup time | $t_{SU:STA}$ | (SETHOLD + 1 + SCL_LATENCY) x $2^{PRESCALE}$ |
| STOP condition setup time | $t_{SU:STO}$ | (SETHOLD + 1 + SCL_LATENCY) x $2^{PRESCALE}$ |
| Bus Idle (STOP condition to START condition) | $t_{BUF}$ | (CLKLO + 1 + SDA_LATENCY) x $2^{PRESCALE}$ |
| Data validity period | $t_{VD:DAT}$ | (DATAVD + 1) x $2^{PRESCALE}$ |
| Data response valid time | $t_{VD:ACK}$ | (DATAVD + 1) x $2^{PRESCALE}$ |

The following table defines the delay parameters, which assume that the rise time is less than one I2C function clock cycle. The rise time depends on several factors, including I/O propagation delay, I2C bus loading, and the value of the external pull-up resistor. A larger rise time increases the number of clock cycles required for the signal to pass through the synchronizer (and glitch filter), thus increasing latency.

Table 20-2 Synchronization Delay

| Parameters | Definition |
|---|---|
| SCL_LATENCY | ROUNDDOWN((2 + FILTSCL + SCL_RISETIME)/$2^{PRESCALE}$) |
| SDA_LATENCY | ROUNDDOWN((2 + FILTSDA + SDA_RISETIME)/$2^{PRESCALE}$) |

The I2C configuration needs to follow the following timing restrictions to avoid unexpected START or STOP conditions on the I2C bus and to prevent the I2C master from detecting unexpected START or STOP conditions. SDA cannot change while SCL is high outside of sending a (repeated) START or STOP condition.

Table 20-3 I2C Timing Parameter Limitations

| Parameters | Minimum | Maximum | Description |
|---|---|---|---|
| CLKLO | 0x03 | - | Satisfy CLKLO>SCL_LATENCY at the same time |
| CLKHI | 0x01 | - | Configure CLKHI to meet the duty cycle requirements in the I2C specification |
| SETHOLD | 0x02 | - | Satisfy both SETHOLD>SDA_LATENCY |
| DATAVD | 0x01 | CLKLO-SDA_LATENCY-1 | Configure DATAVD to meet the data retention requirements in the I2C specification |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Parameters | Minimum | Maximum | Description |
|---|---|---|---|
| FILTSCL | 0x00 | [CLKLOx2$^{PRESCALE}$] - 3 | FILTCSL and FILTSDA are the only parameters that are not multiplied by 2$^{PRESCALE}$ |
| FILTSDA | FILTSCL | [CLKLOx2$^{PRESCALE}$] - 3 | Configuring FILTSDA greater than FILTSL can delay the SDA input to compensate for board-level offsets |
| BUSIDLE | (CLKLO + SETHOLD + 2) x 2 | - | Satisfy BUSIDLE>CLKHI + 1 at the same time |

The timing parameters must be configured to meet the requirements of the I2C specification; this depends on the supported modes and the I2C function clock frequency. When switching between two modes (for example, fast and high-speed modes) using different clock configuration registers, the PRESCALE factor must remain constant between the modes. Some example timing configurations are provided below.

Table 20-4 I2C Example Timing Configuration

| I2C Mode | Clock Frequency | Baud Rate | PRESCALE | FILTSCL/FILTSDA | SETHOLD | CLKO | CLKHI | DATAVD |
|---|---|---|---|---|---|---|---|---|
| Standard | 8MHz | 100kbps | 0x0 | 0xD/0xF | 0x37 | 0x29 | 0x16 | 0x0D |
| Fast | 8MHz | 400kbps | 0x0 | 0x0/0x0 | 0x04 | 0x0B | 0x05 | 0x02 |
| Fast+ | 8MHz | 1Mbps | 0x0 | 0x0/0x0 | 0x02 | 0x03 | 0x01 | 0x01 |
| Standard | 48MHz | 100kbps | 0x2 | 0x2/0xE | 0x3A | 0x3B | 0x39 | 0x0C |
| Fast | 48MHz | 400kbps | 0x0 | 0x1/0x1 | 0x1D | 0x3E | 0x35 | 0x0F |
| Fast | 48MHz | 400kbps | 0x2 | 0x1/0x1 | 0x07 | 0x11 | 0x0B | 0x03 |
| Fast+ | 48MHz | 1Mbps | 0x2 | 0x1/0x1 | 0x03 | 0x06 | 0x04 | 0x04 |
| HS Mode | 48MHz | 3.2Mbps | 0x0 | 0x0/0x0 | 0x07 | 0x08 | 0x03 | 0x01 |

#### 20.3.3.5 Error Condition Description

The I2C master monitors for errors while active. The following conditions will generate error flags and block the issuance of a new START condition until the flags are cleared by software:

- A START or STOP condition is detected that was not generated by the I2C master (MSR.ALF is set). Software must respond to the MSR.ALF flag by terminating the current command (by setting MCR.RST).
- A different value is received on SDA than the data being transmitted (MSR.ALF is set).
- A NACK is detected during data transmission and MCFGR1.IGNACK is cleared (MSR.NDF is set).
- A NACK is detected when an ACK was expected for the address byte, and MCFGR1.IGNACK is cleared (MSR.NDF is set).
- An ACK is detected when a NACK was expected for the address byte, and MCFGR1.IGNACK is cleared (MSR.NDF is set).
- A data transmission or reception is attempted by the transmit buffer without a preceding START condition (MSR.FEF is set).


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

- SCL (or SDA if MCFGR1.TIMECFG is set) remains low for (MCFGR3.PINLOW × 256) prescaler cycles without a pin state change (MSR.PLTF is set).

  Software must respond to the MSR.PLTF flag by terminating the current command (by setting MCR.RST).

The MCFGR2.BUSIDLE field can be configured to treat the I2C bus as idle only if both SCL and SDA remain high for (MCFGR2.BUSIDLE + 1) prescaler cycles. Normally, the I2C bus is considered idle when the master is first enabled. However, when MCFGR2.BUSIDLE is configured to a non-zero value, the bus will only be considered idle after both SCL and SDA remain high for (MCFGR2.BUSIDLE + 1) prescaler cycles.

#### 20.3.3.6 Interrupts and DMA Requests

The table below lists the master mode flags that can generate I2C master interrupts and trigger DMA requests for master transmit/receive operations.

| Flag | Description | Interrupt | Wakeup | DMA Request |
|---|---|---|---|---|
| TDF | Data can be written to the transmit buffer. | ✓ | - | ✓ |
| RDF | Data can be read from the receive buffer. | ✓ | - | ✓ |
| EPF | The master has issued a repeated START or STOP condition. | ✓ | - | - |
| SDF | The master has issued a STOP condition. | ✓ | - | - |
| NDF | • A NACK was detected on the address byte when the master expected an ACK.<br>• An ACK was detected on the address byte when the master expected a NACK.<br>• A NACK was detected by the master during transmission of a data byte. | ✓ | - | - |
| ALF | • A START or STOP condition was detected at an invalid time, and the master lost arbitration.<br>• The master was transmitting data, but the received data did not match the transmitted data. | ✓ | - | - |
| FEF | The master expected a START condition in the command buffer, but the next entry in the command buffer was not a START condition | ✓ | - | - |
| PLTF | Pin low timeout is enabled and SCL (or SDA if configured) is held low for longer than the configured timeout period | ✓ | - | - |
| DMF | The received data matches the configured data, and the received and discarded data configured by MTDR.CMD will not set DMF | ✓ | - | - |
| MBF | The I2C master is busy sending/receiving data | - | - | - |
| BBF | The I2C master is enabled and activity is detected on the I2C bus, but no STOP condition is detected and no bus idle timeout occurs (if enabled) | - | - | - |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 20.3.4 Slave Mode

To perform all slave mode transfers on the I2C bus, the I2C slave logic runs independently of the I2C master logic.

#### 20.3.4.1 Address Matching

The I2C slave can be configured:

- Matches one of two addresses in 7-bit or 10-bit addressing mode.
- Matches a range of addresses in 7-bit or 10-bit addressing mode.
- Matches a broadcast address and generates the corresponding flag.
- Matches an SMBus alert address and generates the corresponding flag.
- Detects high-speed mode master code, disables digital filters and output valid delay time until the next STOP condition is detected.

After matching a valid address, the I2C slave automatically performs a slave transmit transfer or a slave receive transfer until:

- A NACK is detected (unless SCR1.IGNACK is set)
- A bit error is detected (the I2C slave is driving SDA, but a different value is sampled)
- A (repeated) START or STOP condition is detected

#### 20.3.4.2 Transmit and Receive Data

- The transmit and receive data registers are double buffered and updated only during slave transmit transfers and slave receive transfers respectively.
- The received slave address can be configured to be read from the Slave Receive Data Register (SRDR) or the Slave Address Status Register (SASR).
- The Transmit Data Register can be configured to request data only after a slave transmit transfer is detected, or to request new data when the Transmit Data Register is empty.
- The Slave Transmit Data Register (STDR) can be written only when the Transmit Data Flag (SSR.TDF) is set high.
- The Slave Receive Data Register (SRDR) can be read only when the Receive Data Flag (SSR.RDF) is set high (or the Address Valid Flag is set high and SCR1.RxCFG=1).
- The Slave Address Status Register (SASR) can be read only when the Address Valid Flag (SSR.AVF) is set high.

#### 20.3.4.3 Clock Stretching

The I2C slave supports clock stretching. The following conditions can be configured to perform clock stretching:

- The Address Valid Flag (SSR.AVF) is set high during the 9th clock pulse of the address byte.
- The Transmit Data Flag (SSR.TDF) is set high during the 9th clock pulse of the slave transmit transfer.
- The Receive Data Flag (SSR.RDF) is set high during the 9th clock pulse of the slave receive transfer.
- The Transmit ACK Flag (SSR.TAF) is set high during the 8th clock pulse of the address byte or slave receive transfer. In high-speed mode, this configuration is disabled.
- Clock stretching can also be extended to the SCR2.CLKHOLD period to allow additional setup time to support external sampling of the SDA pin. In high-speed mode, this configuration is disabled.

Unless extended by the SCR2.CLKHOLD configuration, clock stretching will extend one peripheral bus clock cycle after SDA is updated when clock stretching is enabled.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

#### 20.3.4.4 Timing Parameters

The I2C slave can configure the following timing parameters. These parameters are disabled when SCR2.FILTEN is cleared or when the I2C slave detects high-speed mode. When disabled, the I2C slave is clocked directly from the I2C bus and may not meet all timing requirements of the I2C specification (such as SDA minimum hold time in standard/fast mode).

- SDA data valid time from SCL pull low to SDA update
- SCL hold time when clock stretching is enabled to increase setup time when sampling external SDA
- SCL glitch filter time
- SDA glitch filter time

The I2C slave has the following restrictions on timing parameters:

- SCR2.FILTSDA must be configured to be greater than or equal to SCR2.FILTSCL (unless compensating for board level offsets between SDA and SCL).
- The configuration of SCR2.DATAVD must be less than the minimum SCL low period.

#### 20.3.4.5 Error Condition Description

An I2C slave can detect the following error conditions:

- When the I2C slave drives SDA and the sampled value is different from the expected value, the bit error flag (SSR.BEF) will be set high.
- When the transmit data is underrun or the receive data is overflowed, the buffer error flag (SSR.FEF) will be set high. To avoid underrun and overflow, turn on the clock stretching function.
- When SCR1.RxCFG is set high, if the address overflows, the buffer error flag (SSR.FEF) will be set high, otherwise the address overflow will not be marked. To avoid overflow, the clock stretching function can be turned on.

The I2C slave will not implement time-out detection due to SCL and/or SDA being pulled low. If detection of this condition is required, the I2C master logic should be used so that software can reset the I2C slave if such a condition is detected.

#### 20.3.4.6 Interrupts and DMA Requests

The following table lists the slave mode flags that can generate I2C slave interrupts and slave transmit/receive DMA requests.

| Flag | Description | Interrupt | Wakeup | DMA Request |
|---|---|---|---|---|
| TDF | Data can be written to the Slave Transmit Data Register (STDR) | ✓ | - | ✓ |
| RDF | Data can be read from the Slave Receive Data Register (SRDR) | ✓ | - | ✓ |
| AVF | The address can be read from the Slave Address Status Register (SASR) | ✓ | ✓ | - |
| TAF | ACK/NACK can be written to the Slave Transmit ACK Register (STAR) | ✓ | - | - |
| RSF | The slave detects an address match followed by a repeated START condition | ✓ | - | - |
| SDF | The slave detects an address match and a STOP condition follows | ✓ | - | - |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Flag | Description | Interrupt | Wakeup | DMA Request |
|---|---|---|---|---|
| BEF | The slave is sending data, but the data received is different from the data sent | ✓ | - | - |
| FEF | ● Sending data underload<br>● Receive data overflow<br>● Address status overflow (when receiving data configuration SCR1.RxCFG = 1)<br>The FEF flag can only be set when clock stretching is disabled. | ✓ | - | - |
| AM0F | The slave detects an address match with SADDR.ADDR0 | ✓ | - | - |
| AM1F | The slave detects an address match with SADDR.ADDR1 or uses an address range | ✓ | - | - |
| GCF | The slave detects an address that matches the broadcast address | ✓ | - | - |
| SARF | The slave detects an address that matches the SMBus alert address | ✓ | - | - |
| SBF | The I2C slave is busy receiving an address byte or sending/receiving data. | - | - | - |
| BBF | The I2C slave is enabled and a START condition is detected on the I2C bus, but a STOP condition is not detected | - | - | - |

### 20.3.5 Communication Level Selection Function

The HSI2C interface supports signal communication at different voltage levels For functional details, refer to the corresponding descriptions in "HSI2C Communication Level Selection Function" of "General-Purpose I/O (GPIO)" chapter.

## 20.4 Operation Example

### 20.4.1 Master Transmission Mode

The operation flow in the master transmission mode is as follows:

**Step1.** Enable the I2C module clock and reset the I2C module.

**Step2.** Configure the I2C module SCL and SDA pins.

**Step3.** Configure the I2C register according to the transmission baud rate so that the frequency of the sent SCL is consistent with the transmission baud rate:

- MCFGR1.PRESCALE
- MCCR0.CLKLO
- MCCR0.CLKHI
- MCCR0.DATAVD
- MCCR0.SETHOLD

**Step4.** Set MCR.MEN to 1 to enable the I2C module.

**Step5.** Check whether MSR.BBF is 0. If it is 0, proceed to the next step.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

**Step6.** Write a START instruction and address + write operation to the MTDR register, for example 0x4AA.

**Step7.** After waiting for MSR.TDF to be 1, write the data to be sent into MTDR, continue to query MSR.TDF, and wait for MSR.TDF to be 1 before writing the next data to be sent into MTDR. This step is repeated until all the data to be sent are written into the MTDR.

**Step8.** Continue to query MSR.TDF, wait for MSR.TDF to be 1, and then write a STOP command to MTDR.CMD, for example, 0x2. Check whether the I2C master MSR.SDF is 1. If it is 1, it means that the STOP condition has been issued and the sending operation is completed.

### 20.4.2 Master Receiving Mode

The operation flow in the master receiving mode is as follows(7-bit address matching is taken as an example, and for other methods, please refer to the protocol content):

**Step1.** Enable the I2C module clock and reset the I2C module.

**Step2.** Configure the I2C module SCL and SDA pins.

**Step3.** Configure the I2C register according to the transmission baud rate so that the frequency of the sent SCL is consistent with the transmission baud rate:

- MCFGR1.PRESCALE
- MCCR0.CLKLO
- MCCR0.CLKHI
- MCCR0.DATAVD
- MCCR0.SETHOLD

**Step4.** Set MCR.MEN to 1 to enable the I2C module.

**Step5.** Check whether MSR.BBF is 0. If it is 0, proceed to the next step.

**Step6.** Write the START instruction and address + read operation to the MTDR register, for example, 0x4AB.

**Step7.** Wait for MSR.TDF to be 1 and then write the receive data instruction + data quantity (the data quantity is the actual received data quantity minus 1) to MTDR, for example, 0x103, indicating that 4 data are received.

**Step8.** Wait for MSR.RDF to be 1 and then read the received data from MRDR, continue to query MSR.RDF, and wait for MSR.RDF to be 1 and then read the next received data from MRDR. Repeat this step until all received data are read from MRDR.

**Step9.** Continue to query MSR.TDF, wait for MSR.TDF to be 1, and then write a STOP command to MTDR.CMD, for example, 0x2. Check whether the I2C master MSR.SDF is 1. If it is 1, it means that the STOP condition has been issued and the receiving operation is completed.

### 20.4.3 Slave Transmission Mode

The operation flow in slave transmission mode is as follows:

**Step1.** Enable the I2C module clock and reset the I2C module.

**Step2.** Configure the I2C module SCL and SDA pins.

**Step3.** Configure I2C slave control 1 register and slave address register:

- SCR1.ADDRMODE
- SCR1.TXCFG
- SADDR.ADDR0

**Step4.** Set SCR0.SEN.EN to 1 to enable the I2C module.

**Step5.** Query whether SSR.AVF is 1. If it is 1, proceed to the next step.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

**Step6.** Read SASR.RADDR[0]:
- 0 indicates master write operation and the slave is in receive mode;
- 1 indicates master read operation and the slave is in transmit mode.

**Step7.** Wait for SSR.TDF to be 1 before writing the data to be sent to STDR. Continue to query SSR.TDF. Wait for SSR.TDF to be 1 before writing the next data to be sent to STDR. Repeat this step until all data to be sent are written to STDR.

**Step8.** Query whether SSR.AVF is 0. If it is 0, it means that the I2C address status register SASR has been read. Query the I2C slave SSR.SDF to see if it is 1. If it is 1, it means that the STOP condition has been detected and the sending operation is finished.

### 20.4.4 Slave Receiving Mode

The operation flow in slave receiving mode is as follows:

**Step1.** Enable the I2C module clock and reset the I2C module.

**Step2.** Configure the I2C module SCL and SDA pins.

**Step3.** Configure I2C slave control 1 register and slave address register:
- SCR1.ADDRMODE
- SADDR.ADDR0

**Step4.** Set SCR0.SEN.EN to 1 to enable the I2C module.

**Step5.** Query whether SSR.AVF is 1. If it is 1, proceed to the next step.

**Step6.** Read SASR.RADDR[0]:
- 0 indicates master write operation and the slave is in receive mode;
- 1 indicates master read operation and the slave is in transmit mode.

**Step7.** Wait for SSR.RDF to be 1 before reading the received data from SRDR, continue to query SSR.RDF, wait for SSR.RDF to be 1 before reading the next received data from SRDR. Loop this step until all received data is read from SRDR.

**Step8.** Query whether SSR.AVF is 0, which means that the I2C address status register SASR has been read, query whether the I2C slave SSR.SDF is 1, which means that the STOP condition has been detected and the receiving operation is over.

### 20.4.5 Master High-Speed Mode Switching

After a reset, the master operates in fast mode. To switch to high-speed mode, the master must complete the following sequence of operations:

**Step1.** Enable the I2C module clock and reset the I2C module.

**Step2.** Configure the I2C module SCL and SDA pins.

**Step3.** Configure the I2C register according to the transmission baud rate so that the frequency of the sent SCL is consistent with the transmission baud rate:
- MCFGR1.PRESCALE
- MCCR0.CLKLO
- MCCR0.CLKHI
- MCCR0.DATAVD
- MCCR0.SETHOLD

**Step4.** Configure the I2C registers according to the baud rate of high-speed mode to ensure that the frequency of the SCL pulses transmitted after switching matches the baud rate of high-speed mode.
- MCCR1.CLKLO


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

- MCCR1.CLKHI
- MCCR1.DATAVD
- MCCR1.SETHOLD

**Step5.** Set MCR.MEN to 1 to enable the I2C module.

**Step6.** Check whether MSR.BBF is 0. If it is 0, proceed to the next step.

**Step7.** Write the START command and control code (0x00001xxx) to the MTDR register, for example, 0x509. After the master expects a NACK response, switch to high-speed mode.

**Step8.** Wait until MSR.TDF=1, then write the START command and address + read/write operation to MTDR register, for example: 0x6AB/0x6AA.

**Step9.** Subsequent operation:

- Read operations: refer to the Host Receive Mode steps7-9.
- Write operations: refer to the Host Transmit Mode steps7-8.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

## 20.5 Register Description

### 20.5.1 Register List

HSI2C base address: 0x4000BC00

Table 20-7 High Speed Integrated Circuit Bus (HSI2C) Register Offset

| Offset | Register | Description | Width |
|---|---|---|---|
| 0x010 | I2C_MCR | Master control register | 32 |
| 0x014 | I2C_MSR | Master status register | 32 |
| 0x018 | I2C_MIER | Master interrupt enable register | 32 |
| 0x01C | I2C_MSCR | Master status clear register | 32 |
| 0x024 | I2C_MCFGR1 | Master configuration 1 register | 32 |
| 0x028 | I2C_MCFGR2 | Master configuration 2 register | 32 |
| 0x02C | I2C_MCFGR3 | Master configuration 3 register | 32 |
| 0x030 | I2C_MCCR0 | Master clock configuration 0 register | 32 |
| 0x034 | I2C_MCCR1 | Master clock configuration 1 register | 32 |
| 0x038 | I2C_MTDR | Master transmit data register | 32 |
| 0x03C | I2C_MRDR | Master receive data register | 32 |
| 0x040 | I2C_MDMR | Master data configuration register | 32 |
| 0x04C | I2C_MDER | Master DMA enable register | 32 |
| 0x110 | I2C_SCR0 | Slave control 0 register | 32 |
| 0x114 | I2C_SSR | Slave status register | 32 |
| 0x118 | I2C_SIER | Slave interrupt enable register | 32 |
| 0x11C | I2C_SSCR | Slave status clear register | 32 |
| 0x120 | I2C_SCR1 | Slave control 1 register | 32 |
| 0x124 | I2C_SCR2 | Slave control 2 register | 32 |
| 0x128 | I2C_STAR | Slave transmit ACK register | 32 |
| 0x12C | I2C_STDR | Slave transmit data register | 32 |
| 0x130 | I2C_SRDR | Slave receive data register | 32 |
| 0x13C | I2C_SDER | Slave DMA enable register | 32 |
| 0x140 | I2C_SADDR | Slave address register | 32 |
| 0x150 | I2C_SASR | Slave address status register | 32 |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 20.5.2 Master Control Register (I2C_MCR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x010 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | DBGEN | Reserved | RST | MEN |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | | | | RW | | RW | RW |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:4* | *Reserved* | - | - |
| 3 | DBGEN | Debug Enable<br>Enable the master in debug mode.<br>● 0b0: Disable the master in debug mode<br>● 0b1: Enable the master in debug mode | RW |
| *2* | *Reserved* | - | - |
| 1 | RST | Software Reset<br>● 0b0: Master logic is not reset<br>● 0b1: Master logic reset<br>Resets all internal master logic and registers, except the Master Control Register (MCR). RST remains asserted until cleared by software. The reset takes effect immediately and remains set until cleared by software. No minimum delay is required before clearing the software reset. | RW |
| 0 | MEN | Master Enable<br>● 0b0: Disable master mode<br>● 0b1: Enable master mode | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 20.5.3 Master Status Register (I2C_MSR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x014 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000001 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | BBF | MBF | Reserved | | | | | | | | | DMF | PLTF | FEF | ALF | NDF | SDF | EPF | Reserved | | | | | | RDF | TDF |
| Access | | | | | | | RO | RO | | | | | | | | | | RO | RO | RO | RO | RO | RO | RO | | | | | | | RO | RO |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:26* | *Reserved* | - | - |
| 25 | BBF | Bus busy flag<br>● 0b0: I2C bus idle<br>● 0b1: I2C bus busy | RO |
| 24 | MBF | Master busy flag<br>● 0b0: I2C master idle<br>● 0b1: I2C master busy | RO |
| *23:15* | *Reserved* | - | - |
| 14 | DMF | Data match flag<br>● 0b0: Matching data has not been received<br>● 0b1: Matching data has been received<br>Indicates that the received data has matched the MDMR.MATCH0 and/or MDMR.MATCH1 fields (configured by MCFGR1.MATCFG). Received data that is discarded due to the MTDR.CMD field does not cause the DMF to be set. | RO |
| 13 | PLTF | Pin low timeout flag<br>● 0b0: Pin Low Timeout has not occurred or is disabled<br>● 0b1: Pin Low Timeout has occurred<br>Set when the SCL and/or SDA inputs are low for longer than the pin low period (pin low timeout, MCFGR3.PINLOW), even when the I2C master is idle.<br>1. Software is responsible for resolving pin low conditions<br>2. PLTF must not be cleared as long as the pin low timeout persists<br>3. PLTF must be cleared before the I2C can initiate a START condition. | RO |
| 12 | FEF | Buffer error flag<br>● 0b0: No error.<br>● 0b1: Master sending or receiving data without a START condition.<br>Detects an attempt to send or receive data without first generating a (repeated) START condition. This can occur if the transmit buffer underflows when MCFGR1.AUTOSTOP=1. When FEF is set, the I2C master sends a STOP condition (if busy) and will not initiate a new START condition until FEF is cleared. | RO |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 11 | ALF | Arbitration lost flag<br>● 0b0: Master did not lose arbitration<br>● 0b1: Master lost arbitration<br>Set if any of the following conditions exist:<br>1. The I2C master sends a logic 1 and detects a logic 0 on the I2C bus<br>2. The I2C master detects a START or STOP condition while sending data<br>When ALF is set, the I2C master releases the I2C bus (becomes idle), and the I2C master will not initiate a new START condition until ALF is cleared. | RO |
| 10 | NDF | NACK detection flag<br>● 0b0: NACK not expected does not detected<br>● 0b1: NACK not expected detected<br>Set if the I2C master detects a NACK that it did not expect when sending an address or data. When asserted, the master will not initiate a new START condition until NDF is cleared. If a NACK is required for a given address (configured by the command word), NDF is set if a NACK is not generated.<br>When NDF is set, the I2C master automatically sends a STOP condition if MCFGR1.AUTOSTOP = 1 or the transmit buffer is not empty. | RO |
| 9 | SDF | Stop detection sign<br>● 0b0: The master did not generate a STOP condition<br>● 0b1: The master has generated a STOP condition<br>Set when the I2C master generates a STOP condition. | RO |
| 8 | EPF | End of packet flag<br>● 0b0: The master did not generate a STOP or repeated START condition<br>● 0b1: The master has generated a STOP and a repeated START condition<br>Asserted when the I2C master generates a repeated START condition or a STOP condition. Not asserted the first time the master generates a START condition. | RO |
| *7:2* | *Reserved* | - | - |
| 1 | RDF | Receive data flag<br>● 0b0: Not ready to receive data<br>● 0b1: Ready to receive data<br>Whenever the receive buffer is full, the receive data flag RDF is set. | RO |
| 0 | TDF | Send data flag<br>● 0b0: Send data request not generated<br>● 0b1: Send data request has been generated<br>TDF is set when the transmit buffer is empty. | RO |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 20.5.4 Master Interrupt Enable Register (I2C_MIER)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x018 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | DMIE | PLTIE | FEIE | ALIE | NDIE | SDIE | EPIE | Reserved | | | | | | RDIE | TDIE |
| Access | | | | | | | | | | | | | | | | | | RW | RW | RW | RW | RW | RW | RW | | | | | | | RW | RW |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:15* | *Reserved* | - | - |
| 14 | DMIE | Data match interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 13 | PLTIE | Pin low timeout interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 12 | FEIE | Buffer error interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 11 | ALIE | Arbitration lost interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 10 | NDIE | NACK detection interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 9 | SDIE | STOP detection interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 8 | EPIE | End of packet interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| *7:2* | *Reserved* | - | - |
| 1 | RDIE | Receive data interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 0 | TDIE | Transmit data interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 20.5.5 Master Status Clear Register (I2C_MSCR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x01C | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | DMC | PLTC | FEC | ALC | NDC | SDC | EPC | Reserved | | | | | | | |
| Access | | | | | | | | | | | | | | | | | | WO | WO | WO | WO | WO | WO | WO | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:15* | *Reserved* | - | - |
| 14 | DMC | Data match flag cleared<br>Write 1 to clear flag, write 0 has no effect. | WO |
| 13 | PLTC | Pin low timeout flag cleared<br>Write 1 to clear flag, write 0 has no effect. | WO |
| 12 | FEC | Buffer error flag cleared<br>Write 1 to clear flag, write 0 has no effect. | WO |
| 11 | ALC | Arbitration lost flag cleared<br>Write 1 to clear flag, write 0 has no effect. | WO |
| 10 | NDC | NACK detection flag clear<br>Write 1 to clear flag, write 0 has no effect. | WO |
| 9 | SDC | STOP detection flag clear<br>Write 1 to clear flag, write 0 has no effect. | WO |
| 8 | EPC | End packet flag cleared<br>Write 1 to clear flag, write 0 has no effect. | WO |
| *7:0* | *Reserved* | - | - |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 20.5.6 Master Configuration 1 Register (I2C_MCFGR1)

**Note**

This register can be modified only when the I2C master is disabled.

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x024 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | MATCFG | | | RDMO | Reserved | TIMECFG | | IGNACK | AUTOSTOP | Reserved | | | | | PRESCALE | | |
| Access | | | | | | | | | | | | | | | | | RW | | | RW | | RW | | RW | RW | | | | | RW | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:16* | *Reserved* | - | - |
| 15:13 | MATCFG | Match configuration<br>Configuring the following values will cause MSR.DMF to be set. Refer to the Master Data Configuration Register (MDMR).<br>● 0b000: Match disabled<br>● 0b001: Setting prohibited<br>● 0b010: Match enabled (first data word equals MDMR.MATCH0 or MDMR.MATCH1)<br>● 0b011: Match enabled (any data word equals MDMR.MATCH0 or MDMR.MATCH1)<br>● 0b100: Match enabled (first data word equals MDMR.MATCH0 and second data word equals MDMR.MATCH1)<br>● 0b101: Match enabled (any data word equals MDMR.MATCH0 and next data word equals MDMR.MATCH1)<br>● 0b110: Match enabled (first data word logical and MDMR.MATCH1 equals MDMR.MATCH0 logical and MDMR.MATCH1)<br>● 0b111: Match enabled (any data word logical and MDMR.MATCH1 equals MDMR.MATCH0 logical and MDMR.MATCH1) | RW |
| 12 | RDMO | Receive data match only<br>● 0b0: Received data is stored in the receive buffer<br>● 0b1: Received data is discarded unless the data match flag (MSR.DMF) is set<br>When enabled, all received data that does not cause the data match flag (MSR.DMF) to be set is discarded. When MSR.DMF is set, the RDMO configuration is ignored. When RDMO is disabled, clear RDMO before clearing MSR.DMF to ensure that no received data is lost. | RW |
| *11* | *Reserved* | - | - |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 10 | TIMECFG | Timeout configuration<br>● 0b0: MSR.PLTF set if SCL low level duration exceeds configured timeout<br>● 0b1: MSR.PLTF set if SCL or SDA low level duration exceeds configured timeout<br>TIMECFG configures the pin low-level timeout flag field (MSR.PLTF) timeout. | RW |
| 9 | IGNACK | IGNACK configuration<br>● 0b0: I2C master receives ACK/NACK normally<br>● 0b1: The I2C master treats the received NACK as an ACK<br>When set, a received NACK field is ignored and assumed to be an ACK. The IGNACK bit needs to be set in fast-mode plus. | RW |
| 8 | AUTOSTOP | Automatic STOP generation<br>● 0b0: No effect<br>● 0b1: Automatically generate a STOP condition whenever the transmit buffer is empty and the I2C master is busy<br>When enabled, a STOP condition is generated whenever the I2C master is busy and the transmit buffer is empty; a STOP condition can also be generated using the transmit buffer command. | RW |
| *7:3* | *Reserved* | - | - |
| 2:0 | PRESCALE | Prescaler configuration<br>Configures the clock prescaler used for all I2C masters, except the digital glitch filter.<br>● 0b000: PCLK clock<br>● 0b001: PCLK clock divided by 2<br>● 0b010: PCLK clock divided by 4<br>● 0b011: PCLK clock divided by 8<br>● 0b100: PCLK clock divided by 16<br>● 0b101: PCLK clock divided by 32<br>● 0b110: PCLK clock divided by 64<br>● 0b111: PCLK clock divided by 128 | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 20.5.7 Master Configuration 2 Register (I2C_MCFGR2)

**Note**

This register can be modified only when the I2C master is disabled.

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x028 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x80800000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | FILTBPSDA | Reserved | | | FILTSDA | | | | FILTBPSCL | Reserved | | | FILTSCL | | | | Reserved | | | | BUSIDLE | | | | | | | | | |
| Access | RW | | | | RW | | | | RW | | | | RW | | | | | | | | RW | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 31 | FILTBPSDA | Analog filter SDA bypass<br>● 0b0: SDA passes through the analog filter<br>● 0b1: SDA bypasses the analog filter | RW |
| *30:28* | *Reserved* | - | - |
| 27:24 | FILTSDA | Glitch filter SDA<br>Configure the I2C master digital glitch filter for the SDA input.<br>1. Configuring to 0 disables the glitch filter.<br>2. Glitches equal to or shorter than the FILTSDA period are filtered out and ignored.<br>3. The delay through the glitch filter is equal to the FILTSDA period, and it must be configured to be shorter than the minimum SCL low or high period.<br>4. The glitch filter cycle count is not affected by the PRESCALE configuration and is automatically bypassed in high-speed mode. | RW |
| 23 | FILTBPSCL | Analog filter SCL bypass<br>● 0b0: SCL passes through the analog filter<br>● 0b1: SCL bypasses the analog filter | RW |
| *22:20* | *Reserved* | - | - |
| 19:16 | FILTSCL | Glitch filter SCL<br>Configure theI2C master digital glitch filter for the SCL input.<br>1. Configuring to 0 disables the glitch filter.<br>2. Glitches equal to or shorter than the FILTSCL period are filtered out and ignored. The FILTSCL period is based on the function clock.<br>3. The delay through the glitch filter is equal to the FILTSCL period and must be configured to be shorter than the minimum SCL low or high period.<br>4. The glitch filter cycle count is not affected by the PRESCALE configuration and is automatically bypassed in high-speed mode. | RW |
| *15:12* | *Reserved* | - | - |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 11:0 | BUSIDLE | Bus idle timeout<br>Configure the bus idle timeout period in clock cycles.<br>1. If the high level of both SCL and SDA persists for longer than the BUSIDLE period, the I2C bus is considered idle, and the master can generate a START condition.<br>2. When the bus idle timeout is set to zero, the bus idle timeout is disabled. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 20.5.8 Master Configuration 3 Register (I2C_MCFGR3)

**Note**

This register can be modified only when the I2C master is disabled.

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x02C | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | PINLOW | | | | | | | | Reserved | | | | | | | |
| Access | | | | | | | | | | | | | | | | | RW | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:20* | *Reserved* | - | - |
| 19:8 | PINLOW | Pin low timeout<br>Configure the pin low timeout flag in clock cycles.<br>1. If the low level of either SCL or SCL/SDA persists for longer than (PINLOW * 256) cycles, the MSR.PLTF flag is set.<br>2. When the pin low timeout is set to zero, the pin low timeout function is disabled. | RW |
| *7:0* | *Reserved* | - | - |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 20.5.9 Master Clock Configuration Register 0 (I2C_MCCR0)

**Note**

This register cannot be modified when the I2C master is enabled and is used for standard, fast, fast mode plus, and ultra-fast transmission.

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x030 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | DATAVD | | | | | | Reserved | | SETHOLD | | | | | | Reserved | | CLKHI | | | | | | Reserved | | CLKLO | | | | | |
| Access | | | RW | | | | | | | | RW | | | | | | | | RW | | | | | | | | RW | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:30* | *Reserved* | - | - |
| 29:24 | DATAVD | Data valid delay<br>Used as the minimum number of cycles (minus 1) for SDA data hold time. The configured time must be shorter than the minimum SCL low period. | RW |
| *23:22* | *Reserved* | - | - |
| 21:16 | SETHOLD | Setup hold delay<br>The minimum number of cycles (minus 1) used by the master under the following conditions:<br>1. Hold time for the START condition<br>2. Setup and hold time for the repeated START condition<br>3. Setup time for the STOP condition<br>The setup time will be extended by the time required to detect the rising edge on the external SCL pin. Any additional board delay caused by external load is ignored, and this time is equal to (2 + FILTSCL)/$2^{PRESCALE}$ cycles. | RW |
| *15:14* | *Reserved* | - | - |
| 13:8 | CLKHI | Clock high period<br>The minimum number of cycles (minus 1) for which the SCL clock is driven high by the master.<br>The SCL high time is extended by the time required to detect the rising edge on the external SCL pin. Any additional board delay caused by external load is ignored, and this time is equal to (2 + FILTSCL)/$2^{PRESCALE}$ cycles. | RW |
| *7:6* | *Reserved* | - | - |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 5:0 | CLKLO | Clock low period<br>The minimum number of cycles (minus 1) for which the SCL clock is driven low by the master.<br>The clock low period value is also used for the minimum bus idle time between the STOP and START conditions; this is extended by the time required to detect the rising edge on the external SCL pin. Any additional board delay caused by external load is ignored, and this time is equal to (2 + FILTSCL)/$2^{PRESCALE}$ cycles. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 20.5.10 Master Clock Configuration Register 1 (I2C_MCCR1)

**Note**

When the I2C master is enabled and used for high-speed mode transmission, this register cannot be modified. The separate clock configuration for high-speed mode allows arbitration to occur in fast mode (timing configured by the Master Clock Configuration 0 register, MCCR0) before switching to high-speed mode (timing configured by MCCR1).

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x034 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | DATAVD | | | | | | Reserved | | SETHOLD | | | | | | Reserved | | CLKHI | | | | | | Reserved | | CLKLO | | | | | |
| Access | | | RW | | | | | | | | RW | | | | | | | | RW | | | | | | | | RW | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:30* | *Reserved* | - | - |
| 29:24 | DATAVD | Data valid delay<br>Used as the minimum number of cycles (minus 1) for SDA data hold time. The configured time must be shorter than the minimum SCL low period. | RW |
| *23:22* | *Reserved* | - | - |
| 21:16 | SETHOLD | Setup hold delay<br>The minimum number of cycles (minus 1) used by the master under the following conditions:<br>1. Hold time for the START condition<br>2. Setup and hold time for the repeated START condition<br>3. Setup time for the STOP condition<br>The setup time will be extended by the time required to detect the rising edge on the external SCL pin. Any additional board delay caused by external load is ignored, and this time is equal to (2 + FILTSCL)/$2^{PRESCALE}$ cycles. | RW |
| *15:14* | *Reserved* | - | - |
| 13:8 | CLKHI | Clock high period<br>The minimum number of cycles (minus 1) for which the SCL clock is driven high by the master.<br>The SCL high time is extended by the time required to detect the rising edge on the external SCL pin. Any additional board delay caused by external load is ignored, and this time is equal to (2 + FILTSCL)/$2^{PRESCALE}$ cycles. | RW |
| *7:6* | *Reserved* | - | - |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 5:0 | CLKLO | Clock low period<br>The minimum number of cycles (minus 1) for which the SCL clock is driven low by the master.<br>The clock low period value is also used for the minimum bus idle time between the STOP and START conditions; this is extended by the time required to detect the rising edge on the external SCL pin. Any additional board delay caused by external load is ignored, and this time is equal to (2 + FILTSCL)/$2^{PRESCALE}$ cycles. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 20.5.11 Master Transmit Data Register (I2C_MTDR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x038 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | CMD | | | DATA | | | | | | | |
| Access | | | | | | | | | | | | | | | | | | | | | | WO | | | WO | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:11* | *Reserved* | - | - |
| 10:8 | CMD | Command data<br>● 0b000: Send DATA[7:0]<br>● 0b001: Receive (DATA[7:0] + 1) bytes<br>● 0b010: Generate STOP condition<br>● 0b011: Receive and discard (DATA[7:0] + 1) bytes<br>● 0b100: Generate (repeated) START and send address in DATA[7:0]<br>● 0b101: Generate (repeated) START and send address in DATA[7:0]. This transmission expects to return NACK<br>● 0b110: Generate (repeated) START and send address in DATA[7:0] in high-speed mode<br>● 0b111: Generate (repeated) START and send address in DATA[7:0] in high-speed mode. This transmission expects to return NACK | WO |
| 7:0 | DATA | Transmit data<br>Perform 8-bit write to data with zero extension of the CMD field | WO |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 20.5.12 Master Receive Data Register (I2C_MRDR)

**Note**

Reading the Receive Data Register returns the data that the I2C master has received but has not yet discarded.

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x03C | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00004000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | RXEMPTY | Reserved | | | | | | | | | | | DATA | | |
| Access | | | | | | | | | | | | | | | | | | RO | | | | | | | | | | | | RO | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:15* | *Reserved* | - | - |
| 14 | RXEMPTY | RX empty status<br>RXEMPTY indicates the empty status of the master receive data buffer<br>● 0b0: The receive buffer is not empty<br>● 0b1: The receive buffer is empty | RO |
| *13:8* | *Reserved* | - | - |
| 7:0 | DATA | Receiving data<br>Received data can be discarded due to the CMD field, or the master can be configured to discard unmatched data. | RO |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 20.5.13 Master Data Configuration Register (I2C_MDMR)

**Note**

This register can only be modified when the I2C master is disabled or idle.

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x040 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | MATCH1 | | | | | | | | Reserved | | | | | | | MATCH0 | | | | | | | |
| Access | | | | | | | | | | | | | | | | | RW | | | | | | | | | | | | | | | RW | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:24* | *Reserved* | - | - |
| 23:16 | MATCH1 | Match value 1<br>When enabled, the received data is compared with the match value. | RW |
| *15:8* | *Reserved* | - | - |
| 7:0 | MATCH0 | Match value 0<br>When enabled, the received data is compared with the match value. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 20.5.14 Master DMA Enable Register (I2C_MDER)

**Note**

This register can only be modified when the I2C master is disabled or idle.

| Offset | Bit position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x04C | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | RDDE | TDDE |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | RW | RW |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:2* | *Reserved* | - | - |
| 1 | RDDE | Receive DMA enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 0 | TDDE | Transmit DMA enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 20.5.15 Slave Control Register 0 (I2C_SCR0)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x110 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | RST | SEN |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | RW | RW |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:2* | *Reserved* | - | - |
| 1 | RST | Software reset<br>● 0b0: Slave mode logic not reset<br>● 0b1: Slave mode logic reset<br>The reset takes effect immediately and remains set until cleared by software. No minimum delay is required before clearing the software reset. | RW |
| 0 | SEN | Enable I2C slave mode<br>● 0b0: Disable slave mode<br>● 0b1: Enable slave mode | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 20.5.16 Slave Status Register (I2C_SSR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x114 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | BBF | SBF | Reserved | | | | | | | | SARF | GCF | AM1F | AM0F | FEF | BEF | SDF | RSF | Reserved | | | | TAF | AVF | RDF | TDF |
| Access | | | | | | | RO | RO | | | | | | | | | RO | RO | RO | RO | RO | RO | RO | RO | | | | | RO | RO | RO | RO |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:26* | *Reserved* | - | - |
| 25 | BBF | Bus busy flag<br>● 0b0: I2C bus idle<br>● 0b1: I2C bus busy | RO |
| 24 | SBF | Slave busy flag<br>● 0b0: I2C slave is idle<br>● 0b1: I2C slave is busy | RO |
| *23:16* | *Reserved* | - | - |
| 15 | SARF | SMBus alarm response flag<br>● 0b0: SMBus alarm response is disabled or not detected<br>● 0b1: SMBus alarm response is enabled and detected<br>**Note**<br>1. The SARF (SMBus Alarm Response Flag) can be cleared by reading the Slave Address Status Register (SASR).<br>2. SARF cannot generate an asynchronous wake-up. | RO |
| 14 | GCF | Broadcast flag<br>Indicates whether the slave has detected the broadcast address.<br>● 0b0: The slave has not detected the broadcast address or the broadcast address is disabled<br>● 0b1: The slave has detected the broadcast address<br>**Note**<br>1. The broadcast flag can be cleared by reading the Slave Address Status Register (SASR).<br>2. The broadcast flag cannot generate an asynchronous wake-up. | RO |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 13 | AM1F | Address match 1 flag<br>Indicates that the received address matches the ADDR1 field or the ADDR0 to ADDR1 range configured in the Slave Address Register (SADDR).<br>● 0b0: No address matching ADDR1 or ADDR0/ADDR1 range has been received<br>● 0b1: Received an address that matches ADDR1.ADDR0/ADDR1 range<br>**Note**<br>1. The address match 1 flag can be cleared by reading the Slave Address Status Register (SASR).<br>2. The address match 1 flag cannot generate an asynchronous wake-up.<br>When SCR1.ADDRMODE is configured to 0x6 or 0x7, the address matching uses the ADDR0/ADDR1 range. If ADDR0 > ADDR1, AM1F will be 0. | RO |
| 12 | AM0F | Address match 0 flag<br>Indicates that the received address matches the ADDR0 field configured in SCR1.ADDRMODE.<br>● 0b0: No ADDR0 matching address has been received<br>● 0b1: Received an address that matches ADDR0<br>**Note**<br>1. The AM0F flag can be cleared by reading the Slave Address Status Register (SASR).<br>2. The AM0F flag cannot generate an asynchronous wake-up. | RO |
| 11 | FEF | Buffer error flag<br>FEF can only be set when clock stretching is disabled.<br>● 0b0: No buffer underflow or overflow has been detected<br>● 0b1: Buffer underflow or overflow has been detected | RO |
| 10 | BEF | Bit error flag<br>● 0b0: Slave has not detected a bit error<br>● 0b1: Slave has detected a bit error<br>**Note**<br>If the I2C slave sends a logical 1 but detects a logical 0 on the I2C bus, the Bit Error Flag (BEF) is set. The slave will ignore the rest of the transmission until the next (repeated) START condition. | RO |
| 9 | SDF | STOP detection flag<br>● 0b0: Slave has not detected STOP<br>● 0b1: Slave has detected STOP<br>The SDF is set when the I2C slave detects a STOP condition and the I2C slave matches the last address byte. | RO |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 8 | RSF | Repeated START flag (RSF)<br>● 0b0: Slave has not detected a repeated START condition<br>● 0b1: Slave has detected a repeated START condition<br>The RSF (Repeated START Flag) is set when the I2C slave detects a repeated START condition and the I2C slave matches the last address byte. The RSF is not set when the slave detects the first START condition. | RO |
| *7:4* | *Reserved* | - | - |
| 3 | TAF | Send ACK flag<br>● 0b0: No ACK/NACK is required to be sent<br>● 0b1: ACK/NACK is required to be sent<br>The TAF (Transmit ACK Flag) is cleared by writing to the Slave Transmit ACK Register (STAR). | RO |
| 2 | AVF | Address valid flag (AVF)<br>● 0b0: Address status register disabled<br>● 0b1: Address status register enabled<br>The address valid flag AVF is cleared by reading the Slave Address Status Register (SASR). When SCR1.RxCFG = 1, SSR.AVF is also cleared by reading the Slave Receive Data Register (SRDR). | RO |
| 1 | RDF | Receive data flag<br>● 0b0: Receive data not ready<br>● 0b1: Ready to receive data<br>The receive data flag RDF is cleared by reading the Slave Receive Data Register (SRDR). When SCR1.RxCFG = 1, if SSR.AVF is set, RDF will not be cleared when reading the Slave Receive Data Register (SRDR). | RO |
| 0 | TDF | Send data flag<br>● 0b0: No request to send data<br>● 0b1: Request to send data<br>The transmit data flag TDF is cleared by writing to the Slave Transmit Data Register (STDR). When SCR1.TxCFG = 0, TDF is also cleared if a NACK or repeated START or STOP condition is detected. | RO |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 20.5.17 Slave Interrupt Enable Register (I2C_SIER)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x118 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | SARIE | GCIE | AM1IE | AM0IE | FEIE | BEIE | SDIE | RSIE | Reserved | | | | TAIE | AVIE | RDIE | TDIE |
| Access | | | | | | | | | | | | | | | | | RW | RW | RW | RW | RW | RW | RW | RW | | | | | RW | RW | RW | RW |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:16* | *Reserved* | - | - |
| 15 | SARIE | SMBus alarm response interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 14 | GCIE | Broadcast interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 13 | AM1IE | Address match 1 interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 12 | AM0IE | Address match 0 interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 11 | FEIE | Buffer error interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 10 | BEIE | Bit error interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 9 | SDIE | STOP detection interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 8 | RSIE | Repeat START interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| *7:4* | *Reserved* | - | - |
| 3 | TAIE | Send ACK interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 2 | AVIE | Address valid interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 1 | RDIE | Receive data interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 0 | TDIE | Transmit data interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 20.5.18 Slave Status Clear Register (I2C_SSCR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x11C | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | FEC | BEC | SDC | RSC | Reserved | | | | | | | |
| Access | | | | | | | | | | | | | | | | | | | | | WO | WO | WO | WO | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:12* | *Reserved* | - | - |
| 11 | FEC | Buffer error flag cleared<br>Write 1 to clear flag, write 0 has no effect. | WO |
| 10 | BEC | Bit error flag clear<br>Write 1 to clear flag, write 0 has no effect. | WO |
| 9 | SDC | STOP detection flag clear<br>Write 1 to clear flag, write 0 has no effect. | WO |
| 8 | RSC | Repeat START flag clear<br>Write 1 to clear flag, write 0 has no effect. | WO |
| *7:0* | *Reserved* | - | - |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

## 20.5.19 Slave Control 1 Register (I2C_SCR1)

**Note**

This register can be modified only when the I2C slave is disabled.

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x120 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | ADDRMODE | | Reserved | | HSMEN | IGNACK | RxCFG | TxCFG | SAEN | GCEN | Reserved | | | | ACKSTALL | TxDSTALL | RXSTALL | ADRSTALL |
| Access | | | | | | | | | | | | | | | | RW | | | RW | RW | RW | RW | RW | RW | | | | | RW | RW | RW | RW |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:19* | *Reserved* | - | - |
| 18:16 | ADDRMODE | Address mode<br>Select the condition that causes an address match<br>● 0b000: Address match 0 (7-bit)<br>● 0b001: Address match 0 (10-bit)<br>● 0b010: Address match 0 (7-bit) or Address match 1 (7-bit)<br>● 0b011: Address match 0 (10-bit) or Address match 1 (10-bit)<br>● 0b100: Address match 0 (7-bit) or Address match 1 (10-bit)<br>● 0b101: Address match 0 (10-bit) or Address match 1 (7-bit)<br>● 0b110: From address match 0 (7-bit) to address match 1 (7-bit)<br>● 0b111: From address match 0 (10-bit) to address match 1 (10-bit) | RW |
| *15:14* | *Reserved* | - | - |
| 13 | HSMEN | High-speed mode enable<br>● 0b0: Disable detection of HS-mode master code<br>● 0b1: Enable detection of HS-mode master code<br>Detection of a high-speed mode master code at slave address 0000_1XX, but does not result in an address match on that code. When set and any HS-mode master code is detected, the SCR2.FILTEN and SCR1.ACKSTALL bits are ignored until the next STOP condition is detected. | RW |
| 12 | IGNACK | Ignore NACK<br>● 0b0: The slave ends the transmission when NACK is detected<br>● 0b1: The slave does not end the transmission when NACK is detected<br>When Ignore NACK is set, the I2C slave continues to transmit after detecting NACK. The Ignore NACK bit needs to be set in fast plus mode. | RW |
| 11 | RXCFG | Receive data configuration<br>● 0b0: Read the slave receive data register to return the received data and clear the receive data flag (SSR.RDF) | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 11 | RXCFG | ● 0b1: Read the slave receive data register when the address valid flag (SSR.AVF) is set, return the slave address status register and clear the address valid flag. Read the slave receive data register when the address valid flag is cleared, return the received data and clear the receive data flag (SSR.RDF) | RW |
| 10 | TXCFG | Transmit flag configuration<br>● 0b0: The Transmit Data Flag is set only when the Slave Transmit Data Register is empty during a slave transmit<br>● 0b1: The Transmit Data Flag is set whenever the Slave Transmit Data Register is empty<br>The Transmit Data Flag is always set before a NACK is detected at the end of a slave transmit. This may cause an extra word to be written to the Transmit Data Buffer.<br>1. When TxCFG = 0, the Slave Transmit Data Register is automatically cleared when a slave transmit is detected. This causes the Transmit Data Flag to be set whenever a slave transmit is detected and causes the Transmit Data Flag to flip at the end of a slave transmit.<br>2. When TxCFG = 1, the Transmit Data Flag is set whenever the Slave Transmit Data Register is empty and flips when the Transmit Data Register is full. This allows the Transmit Data Register to be filled before a slave transmit transfer is detected, but can cause the Transmit Data Register to be written before a NACK is detected on the last byte of a slave transmit transfer. | RW |
| 9 | SAEN | SMBus alert enable<br>● 0b0: Disable SMBus alert match<br>● 0b1: Enable SMBus alert match | RW |
| 8 | GCEN | Broadcast enabled<br>● 0b0: Broadcast address disabled<br>● 0b1: Broadcast address enabled | RW |
| *7:4* | *Reserved* | - | - |
| 3 | ACKSTALL | ACK SCL pause<br>● 0b0: Clock stretching disabled<br>● 0b1: Clock stretching enabled<br>Enable SCL clock stretching during the slave transmit address byte and the slave receive address and data bytes so that software can write to the Slave Transmit ACK Register (STAR) before sending an ACK or NACK. Clock stretching occurs when the 9th bit is transmitted, so it is not compatible with high-speed mode. When ACKSTALL is enabled, it is not necessary to set SCR1.RxSTALL or SCR1.ADRSTALL. | RW |
| 2 | TxDSTALL | TX Data SCL pause<br>● 0b0: Clock stretching disabled<br>● 0b1: Clock stretching enabled<br>When SSR.TDF = 1, SCL clock stretching is enabled during slave transmit operations. Clock stretching occurs after the 9th bit, making it compatible with high-speed mode. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 1 | RXSTALL | RX SCL pause<br>● 0b0: Clock stretching disabled<br>● 0b1: Clock stretching enabled<br>When SSR.RDF = 1, SCL clock stretching is enabled during slave receive operations. Clock stretching occurs after the 9th bit, making it compatible with High-Speed mode. | RW |
| 0 | ADRSTALL | Address SCL pause<br>● 0b0: Clock stretching disabled<br>● 0b1: Clock stretching enabled<br>When SSR.AVF = 1, SCL clock stretching is enabled. Clock stretching occurs only after the 9th bit, making it compatible with High-Speed mode. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

## 20.5.20 Slave Control 2 Register (I2C_SCR2)

**Note**

This register can be modified only when the I2C slave is disabled.

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x124 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | FILTSDA | | | | FILTSCL | | | | Reserved | FILTEN | DATAVD | | | | | | | | Reserved | | | | CLKHOLD | | |
| Access | | | | | | | | | RW | | | | RW | | | | | RW | RW | | | | | | | | | | | | RW | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:24* | *Reserved* | - | - |
| 23:20 | FILTSDA | Glitch filter SDA<br>Configures the I2C slave digital glitch filter for the SDA input.<br>1. Configuring to 0 disables the glitch filter.<br>2. Glitches equal to or shorter than the FILTSDA period are filtered out and ignored.<br>3. The delay introduced by the glitch filter is equal to FILTSDA+2 cycles and must be configured to be less than the minimum SCL low or high period.<br>4. The glitch filter cycle count is not affected by the PRESCALE configuration and is disabled in high-speed mode. | RW |
| 19:16 | FILTSCL | Glitch filter SCL<br>Configures the I2C slave digital glitch filter for the SCL input.<br>1. Configuring to 0 disables the glitch filter.<br>2. Glitches equal to or shorter than the FILTSCL period are filtered out and ignored.<br>3. The delay introduced by the glitch filter is equal to FILTSCL+2 cycles and must be configured to be less than the minimum SCL low or high period.<br>4. The glitch filter cycle count is not affected by the PRESCALE configuration and is disabled in high-speed mode. | RW |
| *15* | *Reserved* | - | - |
| 14 | FILTEN | Filter enable<br>● 0b0: Disables the digital filter and output delay counter in slave mode<br>● 0b1: Enables the digital filter and output delay counter in slave mode<br>FILTEN should only be updated when the I2C slave is disabled. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 13:8 | DATAVD | Data valid delay<br>Configures the SDA data valid delay time for the I2C slave, which equals FILTSCL + DATAVD + 2 cycles.<br>1. The data valid delay must be configured to be less than the minimum SCL low period.<br>2. The I2C slave data valid delay is not affected by the PRESCALE configuration and is disabled in high-speed mode. | RW |
| *7:4* | *Reserved* | - | - |
| 3:0 | CLKHOLD | Clock hold time<br>Configures the minimum clock hold time for the I2C slave when clock stretching is enabled.<br>1. The minimum hold time is equal to CLKHOLD + 3 cycles.<br>2. The I2C slave clock hold time is not affected by the PRESCALE configuration and is disabled in high-speed mode. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 20.5.21 Slave Transmit ACK Register (I2C_STAR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x128 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | TXNACK |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | RW |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:1* | *Reserved* | - | - |
| 0 | TXNACK | Transmit NACK<br>● 0b0: Write ACK for each received word<br>● 0b1: Write NACK for each received word<br>After receiving each word, software can send ACK (logic 0) or NACK (logic 1); sending NACK chooses whether to use ACK or NACK.<br>1. When SCR1.ACKSTALL is set, a NACK must be written for each matching address byte and each received word. ACKSTALL must be set because it pauses data transmission until software reads the received word (and decides whether to respond with ACK or NACK).<br>2. To configure the default ACK/NACK, NACK can also be written when the I2C slave is disabled or idle. | RW |

**Note**

The STAR register can only be written when the ACK SCL pause position bit (SCR1.ACKSTALL) is set. The SCR1.ACKSTALL bit enables clock stretching during the ACK/NACK period, during which software can write to the STAR register. This logic ensures that the clock stretching lasts at least 1 bus clock cycle after the STAR register is updated. The clock hold time field (SCR2.CLKHOLD) can be used to further extend the clock stretching duration.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

## 20.5.22 Slave Transmit Data Register (I2C_STDR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x12C | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | DATA | | | | | | | |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | WO | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:8* | *Reserved* | - | - |
| 7:0 | DATA | Transmit data<br>Writing to the Slave Transmit Data Register (STDR) stores the I2C slave transmit data in the slave transmit data register. | WO |

**Note**

Clock stretching (enabled or disabled) affects the transmission timing of data. The SCR1.TXDSTALL bit enables clock stretching during the first data bit of the slave transmit transfer.

1. If clock stretching is enabled (SCR1.TXDSTALL=1), the data transmission is paused until the STDR is updated. After the STDR is updated, the clock stretching is extended for at least 1 bus clock cycle, and the clock hold time field (SCR2.CLKHOLD) can be used to extend the duration.
2. If clock stretching is disabled (SCR1.TXDSTALL=0), data should be written to the slave transmit register before the slave transmission begins; otherwise, if data is not written before the transmission starts, the buffer error flag (SSR.FEF) will be set.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 20.5.23 Slave Receive Data Register (I2C_SRDR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x130 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00004000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | SOF | REMP | Reserved | | | | | | DATA | | | | | | | |
| Access | | | | | | | | | | | | | | | | | RO | RO | | | | | | | RO | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:16* | *Reserved* | - | - |
| 15 | SOF | Frame start<br>Indicates whether this is the first data word.<br>● 0b0: This is not the first data word since a (repeated) START or STOP condition<br>● 0b1: This is the first data word since a (repeated) START or STOP condition | RO |
| 14 | REMP | RX empty<br>Indicates the empty state of the slave receive data register.<br>● 0b0: The receive data register is not empty<br>● 0b1: The receive data register is empty | RO |
| *13:8* | *Reserved* | - | - |
| 7:0 | DATA | Receiving data<br>I2C slave received data | RO |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 20.5.24 Slave DMA Enable Register (I2C_SDER)

**Note**

This register can be modified only when the I2C slave is disabled.

| Offset | Bit position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x13C | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | AVDE | RDDE | TDDE |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | RW | RW | RW |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:3* | *Reserved* | - | - |
| 2 | AVDE | Address DMA enable<br>Address DMA request and receive DMA request are shared. If both address DMA request and receive DMA request are enabled, configure SCR1.RXCFG to 1 to allow DMA to read the address from the SRDR register. | RW |
| 1 | RDDE | Receive DMA enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 0 | TDDE | Transmit DMA enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 20.5.25 Slave Address Register (I2C_SADDR)

**Note**

This register can be modified only when the I2C slave is disabled.

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x140 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | ADDR1 | | | | | | | | | | Reserved | | | | | | ADDR0 | | | | | | | | | Reserved |
| Access | | | | | | RW | | | | | | | | | | | | | | | | RW | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:27* | *Reserved* | - | - |
| 26:17 | ADDR1 | Address value 1<br>Compares with the received address to detect the slave address.<br>● In 10-bit mode, the first address byte is compared with {11110, ADDR1[26:25]}, and the second address byte is compared with ADDR1[24:17].<br>● In 7-bit mode, the address is compared with ADDR1[23:17]. | RW |
| *16:11* | *Reserved* | - | - |
| 10:1 | ADDR0 | Address value 0<br>Compares with the received address to detect the slave address.<br>● In 10-bit mode, the first address byte is compared with {11110, ADDR0[10:9]}, and the second address byte is compared with ADDR0[8:1].<br>● In 7-bit mode, the address is compared with ADDR0[7:1]. | RW |
| *0* | *Reserved* | - | - |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 20.5.26 Slave Address Status Register (I2C_SASR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x150 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00004000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | ANV | Reserved | | | RADDR | | | | | | | | | | |
| Access | | | | | | | | | | | | | | | | | | RO | | | | RO | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:15* | *Reserved* | - | - |
| 14 | ANV | Address disabled<br>● 0b0: Receive address (RADDR) enabled<br>● 0b0: Receive address (RADDR) disabled | RO |
| *13:11* | *Reserved* | - | - |
| 10:0 | RADDR | Received address<br>Each time the AMF is set, the received address is updated; the AMF can be cleared by reading the slave address status register.<br>● In 7-bit mode, the address byte is stored in RADDR[7:0].<br>● In 10-bit mode, the first address byte is {11110, RADDR[10:9], RADDR[0]}, and the second address byte is RADDR[8:1]. Thus, the R/W bit is always stored in RADDR[0]. | RO |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

# 21 SPI

## 21.1 Introduction

The serial peripheral interface (SPI) provides data transmission function conforming to the SPI communication protocol in the master-slave mode, which can be used for synchronous serial communication between the MCU and external devices. This module supports full-duplex and single-wire half-duplex and simplex communication.

## 21.2 Features

- Can be configured as master or slave, support multi-machine mode
- The maximum division factor in master mode is PCLK/2
- The maximum division factor in slave mode is PCLK/4
- Multiple communication modes: full-duplex, single-wire half-duplex, simplex
- 2 transmission orders: MSB first or LSB first
- Various data frame lengths: 4bits-16bits
- 2 NSS methods: hardware control, software control
- Configurable serial clock polarity and phase
- Support DMA
- Support master mode delayed sampling

## 21.3 Functional Description

### 21.3.1 Function Block Diagram

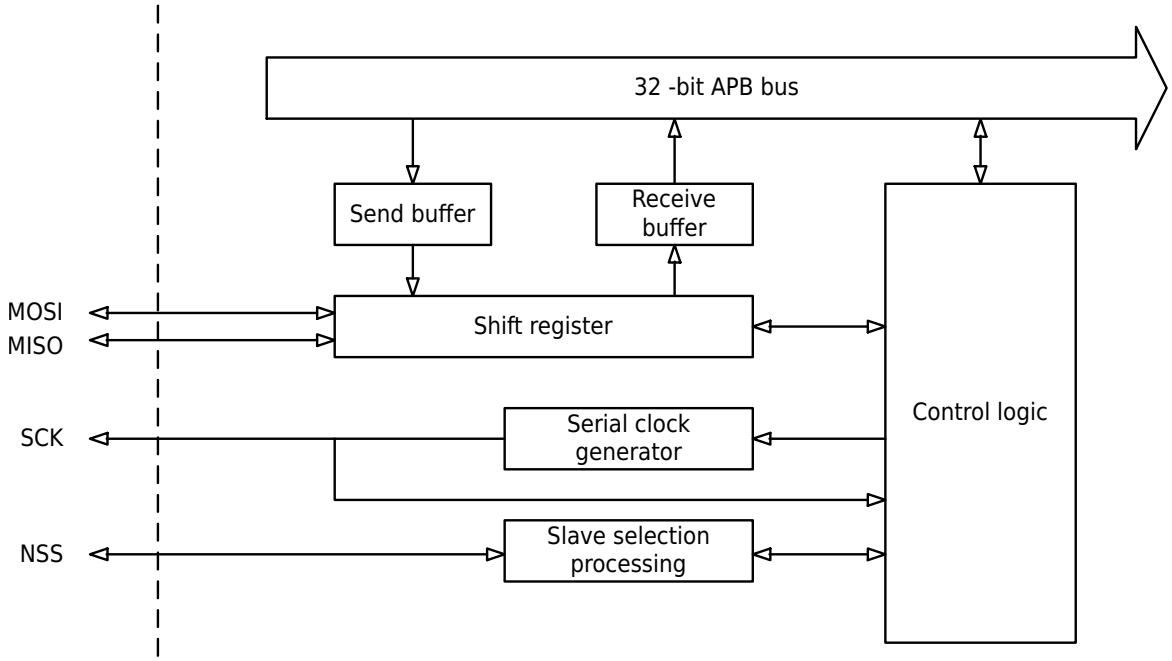



**Figure 21-1 SPI Block Diagram**


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

**Note**

The NSS pin mentioned in this chapter corresponds to SPIx_CS in the "Port Digital Multiplexing Function" section of the chapter "General-Purpose I/O (GPIO)".

### 21.3.2 Communication Format and Timing

SPI serial frame format depends on serial clock polarity CPOL, serial clock phase CPHA and data frame format. The normal communication of the SPI interface requires that all masters and slaves must be configured to the same communication format.

The clock polarity CPOL refers to the level state of the serial clock in the idle state where the SPI does not transmit data. When SPI_CR0.CPOL is 0, the serial clock SCK is low when idle; when SPI_CR0.CPOL is 1, the serial clock SCK is high when idle.

The clock phase CPHA refers to the sequence in which the master and slave of the SPI sample and shift according to the serial clock SCK. When SPI_CR0.CPHA is 0, the master and slave sample data on the front edge and shift data on the back edge; When SPI_CR0.CPHA is 1, the master and slave shift data on the front edge, and sample data on the back edge.

By setting SPI_CR0.WIDTH, the width of each frame of data can be configured from 4bits to 16bits.

By setting SPI_CR0.LSBF, the transmission sequence of each frame of data can be configured as LSB first or MSB first.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

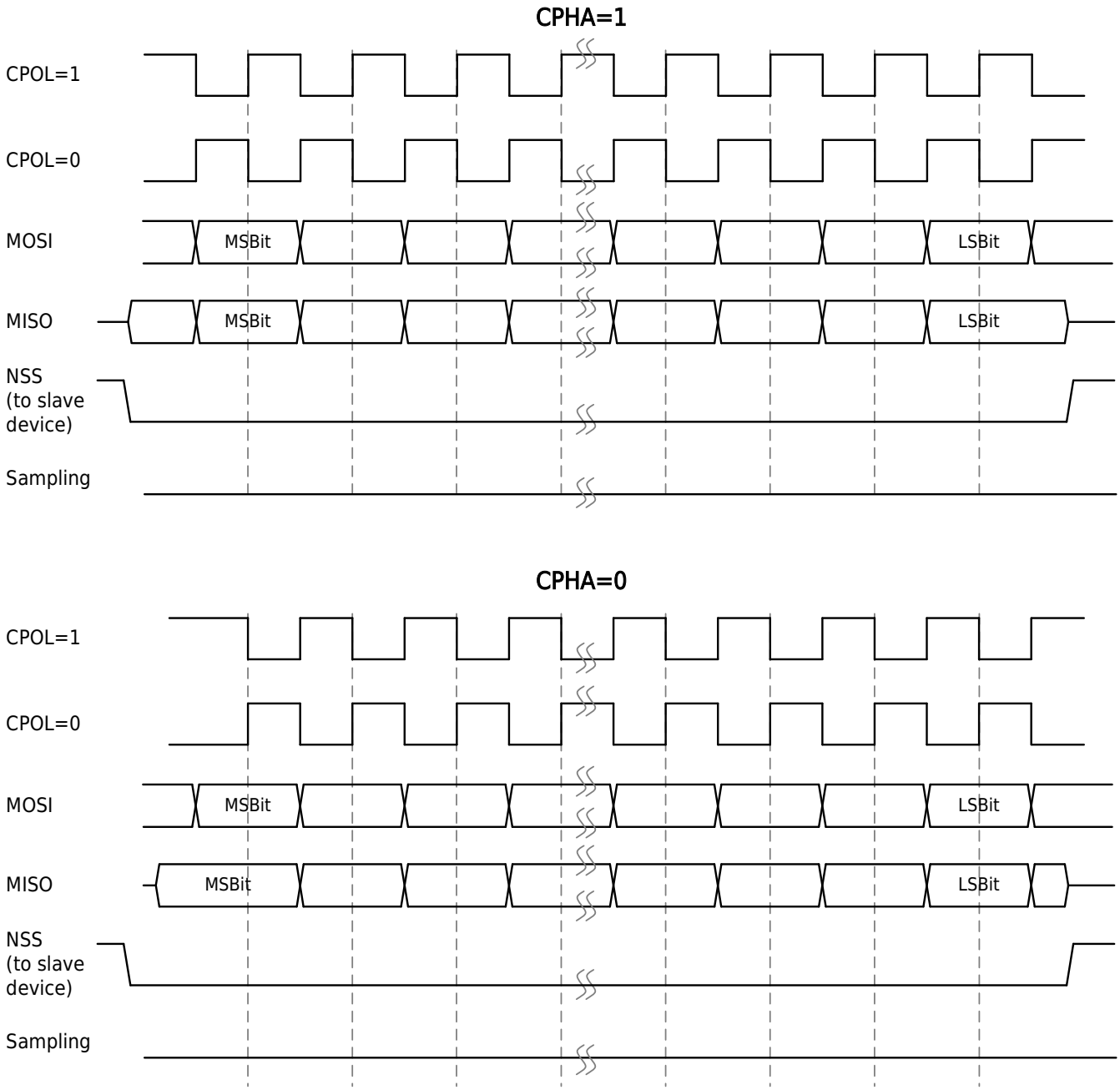



Figure 21-2 SPI Communication Format Timing Diagram

### 21.3.3 Slave Select Pin Configuration

This module provides a variety of slave selection methods to adapt to different application scenarios.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

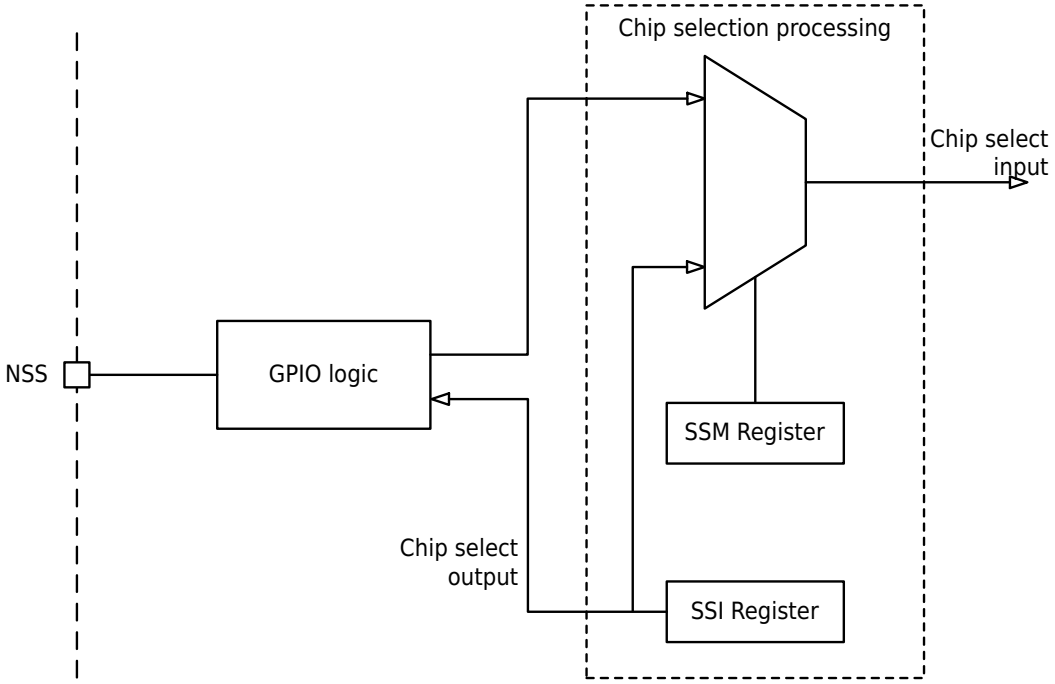



Figure 21-3 Slave Select NSS Pin Configuration

- When the SPI works in master mode and SPI_CR0.SSM is 0, the slave selects NSS as an input, and its state is determined by the level of the NSS pin.
- When the SPI works in master mode and SPI_CR0.SSM is 1, the slave selects NSS as an output, and the output level of the NSS pin is determined by the internal register SPI_SSI.SSI. When SPI_SSI.SSI is 0, the NSS pin outputs a low level; when SPI_SSI.SSI is 1, the NSS pin outputs a high level.
- When SPI works in slave mode and SPI_CR0.SSM is 0, the low level of NSS pin can select this module.
- When SPI operates in slave mode and SPI_CR0.SSM is set to 1, the NSS pin no longer functions as a chip select. The module can be selected when the SPI_SSI.SSI register is set to 0.

### 21.3.4 Full-duplex Communication

The full-duplex communication method allows the master and slave to perform two-way communication of sending and receiving at the same time relying on two unidirectional data lines. In this mode, the master provides a clock signal through the SCK pin to synchronize data input and output. The master and slave MOSI pins are connected to each other, and the MISO pins are connected to each other. The master sends data to the slave through the MOSI pin, and the slave returns the data through the MISO pin. Whenever the user writes data to the SPI_DR register, the SPI master will send data to the SPI slave through SCK and MOSI; and receive data from the slave through SCK and MISO.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

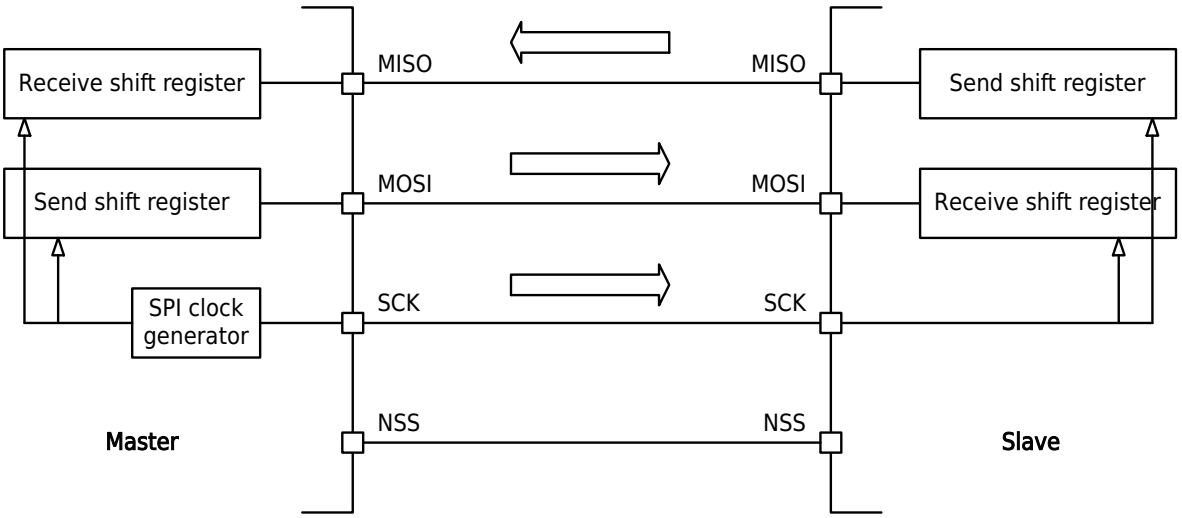



Figure 21-4 SPI Full-duplex Communication

## 21.3.5 Single-wire Half-duplex Communication

The single-wire half-duplex communication method allows the master and slave to perform time-sharing two-way communication based on a two-way data line. The pin used as a master is MOSI, and the pin used as a slave is MISO. The data transfer direction is determined by SPI_HDOE.HDOE. When SPI_HDOE.HDOE is set to 1, the data line transmits data. Unused pins do not participate in SPI communication and can be used by other functions. When the master operates in receive mode, arbitrary data must be written to the SPI_DR register to trigger the SPI master to send SCK for data transmission.

When the master-slave performs single-wire half-duplex communication, since the switching of sending and receiving may not be synchronized, they may have a driving conflict for the bidirectional shared data line. In order to avoid possible damage to the device due to such situations, it is recommended to insert a current-limiting resistor in series on the data line to protect chip pins.

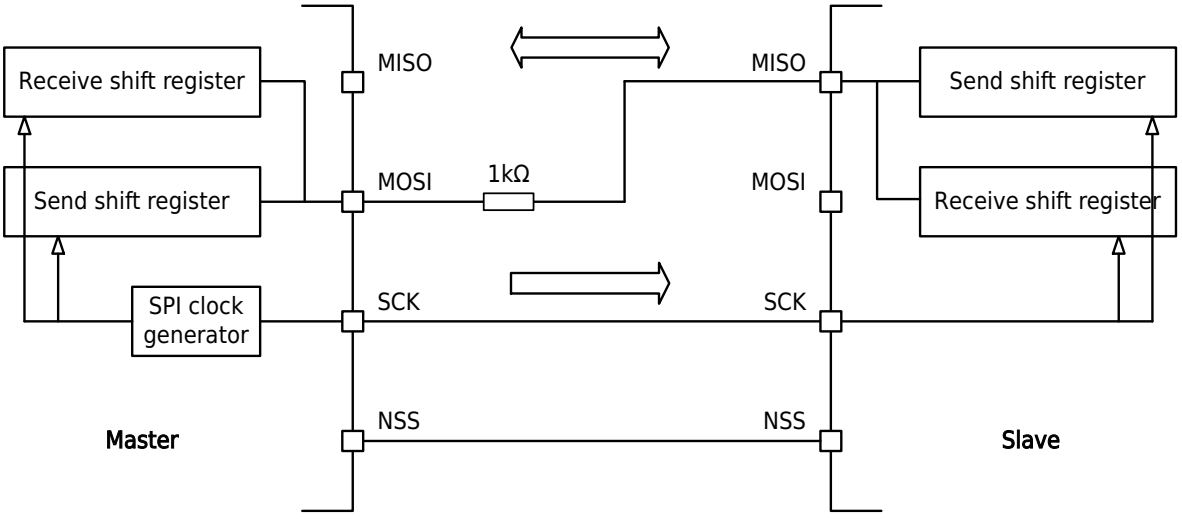



Figure 21-5 SPI Single-wire Half-duplex Communication

## 21.3.6 Simplex Communication

The simplex communication method allows the master and slave to rely on a one-way data line to perform one-way communication of fixed reception or fixed transmission. By setting SPI_CR0.CM, select the simplex only send or simplex only receive mode. Unused pins do not participate in SPI communication and can be used by other functions.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

When the master works in the sending state and the slave works in the receiving state, the master writes data to the SPI_DR register to trigger the SPI master to send SCK for data transmission.

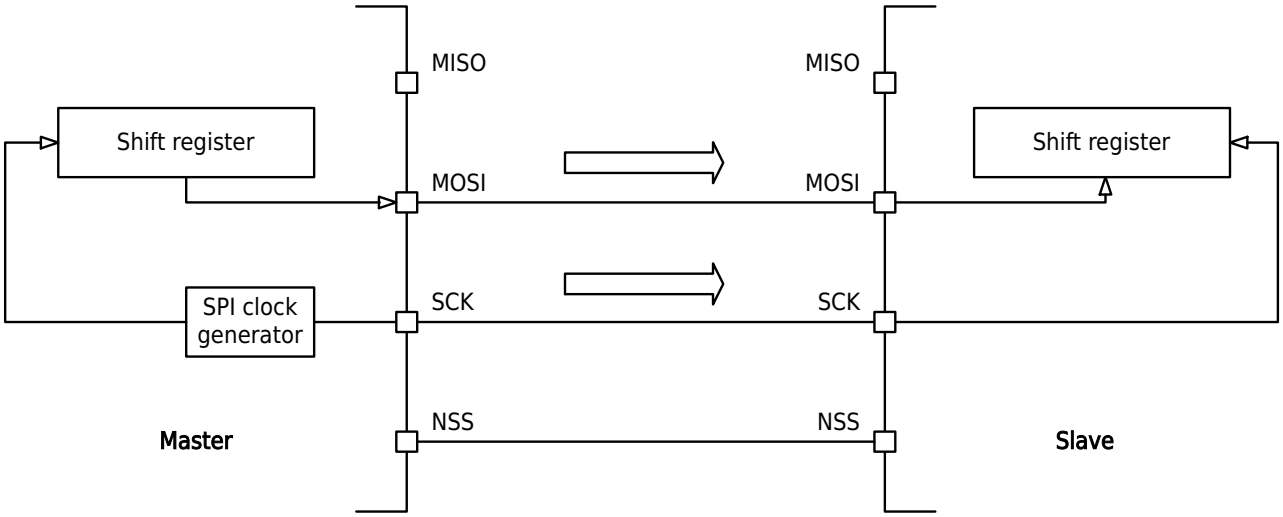



Figure 21-6 SPI Simplex Transmit Communication Diagram

When the master works in the receiving state and the slave works in the sending state, the master also needs to write arbitrary data to the SPI_DR register to trigger the SPI master to send SCK for data transmission.

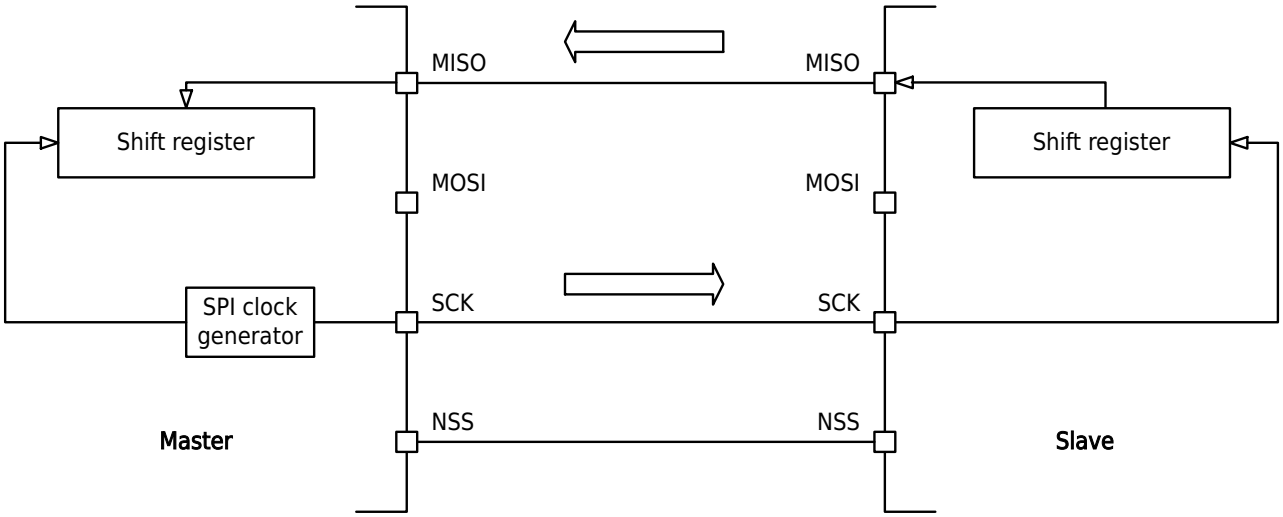



Figure 21-7 SPI Simplex Receive Communication Diagram

**Note**

- NSS pin can be used to provide hardware control flow between master and slave devices. Peripherals can also choose not to use these pins. After that, the hardware control flow must be handled internally for the master and slave.
- In this configuration, two idle pins (MOSI or MISO) can be used as GPIO.

## 21.3.7 Multi-machine Communication

In a multi-machine communication system, the slave selection NSS pin of the master is configured as a hardware input (SPI_CR0.SSM is 0), and the GPIO is responsible for outputting the slave selection signal, and the NSS input of the master is driven by the GPIO of another master. Once the slave selection NSS pin of the master is pulled low by the GPIO of other masters, a mode error MODF will be generated, which will cause the SPI of this module to be forced to return to the disabled state (that is, SPI_CR0.EN


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

is cleared), and the master-slave mode selection register (SPI_CR0.MSTR) will also be cleared (that is, switched to slave mode). When the system needs to use SPI communication again, it needs to reset and initialize the module again. Therefore, when the master tries to obtain the SPI bus control right again, it needs to detect whether the NSS input is high. If it is high and there is no MODF, the bus control right can be successfully obtained. In a system with multiple slaves, each slave shares the serial clock and data lines, but requires its own independent slave selection signal, which requires the master to output the corresponding slave selection signal to each slave through different GPIOs.

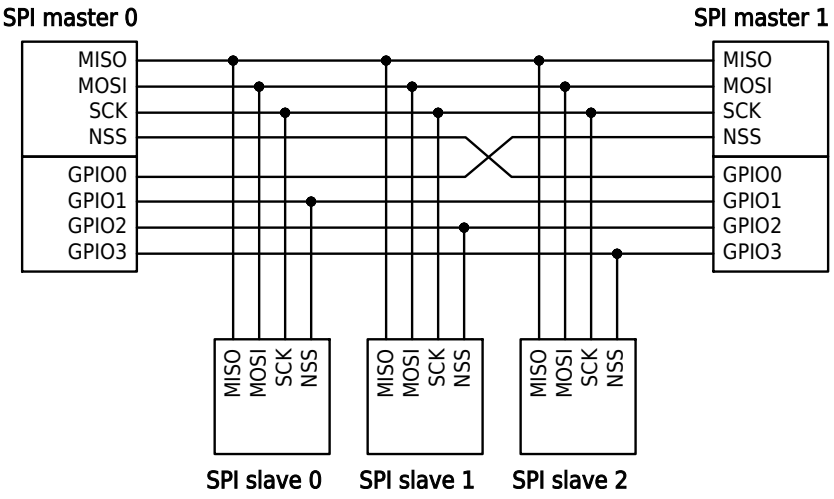



Figure 21-8 SPI Multi-machine System

## 21.3.8 Master Mode Delayed Sampling

In SPI communication, due to the following factors:

- The time required for the slave to send data from the shift edge of the serial clock SCK to MISO;
- Propagation time for data from slave data output MISO to master data input MISO.

When the master receives valid data, there is a certain delay compared to the shift edge of the serial clock. In the traditional SPI protocol, the master needs to sample the MISO data at the sampling edge of half an SCK cycle after the shift edge, which requires that the above-mentioned delay must be less than half an SCK cycle, and the difference between half an SCK cycle and this delay must meet the requirement of MISO setup time. Therefore, when the above-mentioned delay is obvious, the rate of SPI communication will be restricted, and data errors will occur when the rate is high.

This module provides the function of delayed sampling in the master mode, which can effectively alleviate the communication speed bottleneck problem caused by the above problems. By setting the register SPI_CR0.SMP to 1, enable the master mode to delay sampling. At this time, as the SPI master, this module will take the sampling edge of the traditional SPI standard as the reference, and delay half the SCK cycle for sampling. This not only provides a higher tolerance to delay for the entire system, but also does not affect the data transmission of the next bit, thereby achieving a higher communication rate.

**Note**

1. This function is only valid in master mode, not in slave mode.
2. SPI_CR0.SMP does not affect the configuration of the communication format of the master and slave — whether this function is enabled or not, the master and slave must be configured with the same CPOL and CPHA according to the requirements of conventional SPI.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 21.3.9 Status Flags and Error Flags

SPI provides a variety of status flags to indicate the current working status, including general status flags and error flags.

#### Slave Select Input (NSS) State

The slave selection input status indicates the level of the currently input slave selection signal, which corresponds to the register SPI_SR.SSLVL. When SPI_SR.SSLVL is 0, the slave selection input is low; when SPI_SR.SSLVL is 1, the slave selection input is high.

#### Bus Busy Flag Bit

The bus busy flag indicates whether the current SPI interface is busy or idle, corresponding to the register SPI_SR.BUSY. When SPI_SR.BUSY is 0, the SPI interface is in an idle state, and there is no data transmission currently; when SPI_SR.BUSY is 1, the SPI interface is in a busy state, and data transmission is currently being prepared or in progress.

#### Slave Select Input Rising Edge Flag Bit

The slave selection input rising edge flag indicates whether the current input slave selection signal has a rising edge, corresponding to the register SPI_SR.SSR. When SPI_SR.SSR is 0, there is no rising edge on the slave selection input; when SPI_SR.SSR is 1, there is a rising edge on the slave selection input.

#### Slave Select Input Falling Edge Flag Bit

The slave selection input falling edge flag indicates whether the current input slave selection signal has a falling edge, corresponding to the register SPI_SR.SSF. When SPI_SR.SSF is 0, there is no falling edge on the slave select input; when SPI_SR.SSF is 1, there is a falling edge on the slave select input.

#### Transmit Buffer Empty Flag Bit

The send buffer empty flag indicates whether the current SPI send buffer is loaded with data to be sent, corresponding to the register SPI_SR.TXE. When SPI_SR.TXE is 0, the SPI send buffer has been loaded with data to be sent; when SPI_SR.TXE is 1, the SPI send buffer is empty, and the next data to be sent can be written.

#### Receive Buffer Not Empty Flag Bit

The receiving buffer non-empty flag indicates whether there is received data in the receiving buffer of the current SPI, corresponding to the register SPI_SR.RXNE. When SPI_SR.RXNE is 0, the SPI receive buffer has not been loaded with valid received data; when SPI_SR.RXNE is 1, the SPI receive buffer is loaded with received data, which is available for reading.

#### Slave Selection Error Flag Bit in Slave Mode

When the SPI is working in slave mode and valid data transmission is in progress, the input slave selection signal should remain selected (low level) to ensure normal communication. If the slave selection is pulled high during this period, a slave selection error occurs, the corresponding register SPI_SR.SSERR will be set.

If a slave selection error occurs, the corresponding data transmission error will occur, and the SPI interface will also enter the unselected state due to the slave selection being pulled high. After a slave selection error occurs, the corresponding error flag is set and does not prevent subsequent normal data transmission, but the error flag will remain set until the user program clears it or the SPI is reset by the system control module.

It should be noted that the slave selection error SSERR is an error that only occurs in the slave mode, and it will only be triggered when the slave selection is pulled high when valid data is still being


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

transmitted. When there is no valid data transmission between frames, a normal slave select pulling high does not trigger this error.

#### Receive Buffer Overflow Error Flag Bit

The size of the SPI receive buffer is one frame. When the data in the receive buffer has not been read out and the data of the new frame has been received, a receive buffer overflow error occurs, and the corresponding register SPI_SR.OVF will be set.

When a receive buffer overflow error occurs, the original data in the buffer will be overwritten by the newly received data, resulting in data loss. After a receive buffer overflow error occurs, the corresponding error flag bit does not prevent subsequent normal data transmission, but the error flag bit will remain set until the user program clears it or the SPI is reset by the system control module.

#### Transmit Buffer Underflow Error Flag Bit in Slave Mode

SPI send buffer is one frame. When the send buffer is empty in slave mode and the transmission of a new frame starts, a send buffer underflow error occurs, and the corresponding register SPI_SR.UDF will be set.

The occurrence of a send buffer underflow error in slave mode means that the send buffer is not filled with data in time, and the data sent to the master is missed. After a buffer underflow error occurs in the slave mode, the corresponding error flag bit does not prevent subsequent normal data transmission, but the error flag bit will remain set until the user program clears it or the SPI is reset by the system control module.

#### Mode Error Flag Bit in Master mode

When the SPI works in master mode and the slave selection pin is configured as an input, the input slave selection signal must remain unselected (high level) to ensure normal communication. If the slave selection is pulled low, a mode error will occur and the corresponding register SPI_SR.MODF will be set.

A mode error will cause the SPI to be forced to return to the disabled state (that is, SPI_CR0.EN is cleared), and the master-slave mode selection register (SPI_CR0.MSTR) will also be cleared (that is, switch to slave mode). After a mode error occurs, setting the corresponding error flag does not prevent the user program from re-enabling the SPI and performing normal data transmission, but the error flag will remain set until the user program clears it or the SPI is reset by the system control module.

It should be noted that the mode error MODF is an error that only appears in the specific configuration of the host mode. It will only be triggered when the slave selection is configured as an input and pulled low, and the error will not be triggered in other cases.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

**Master Mode Example with CPOL=1 and CPHA=1**

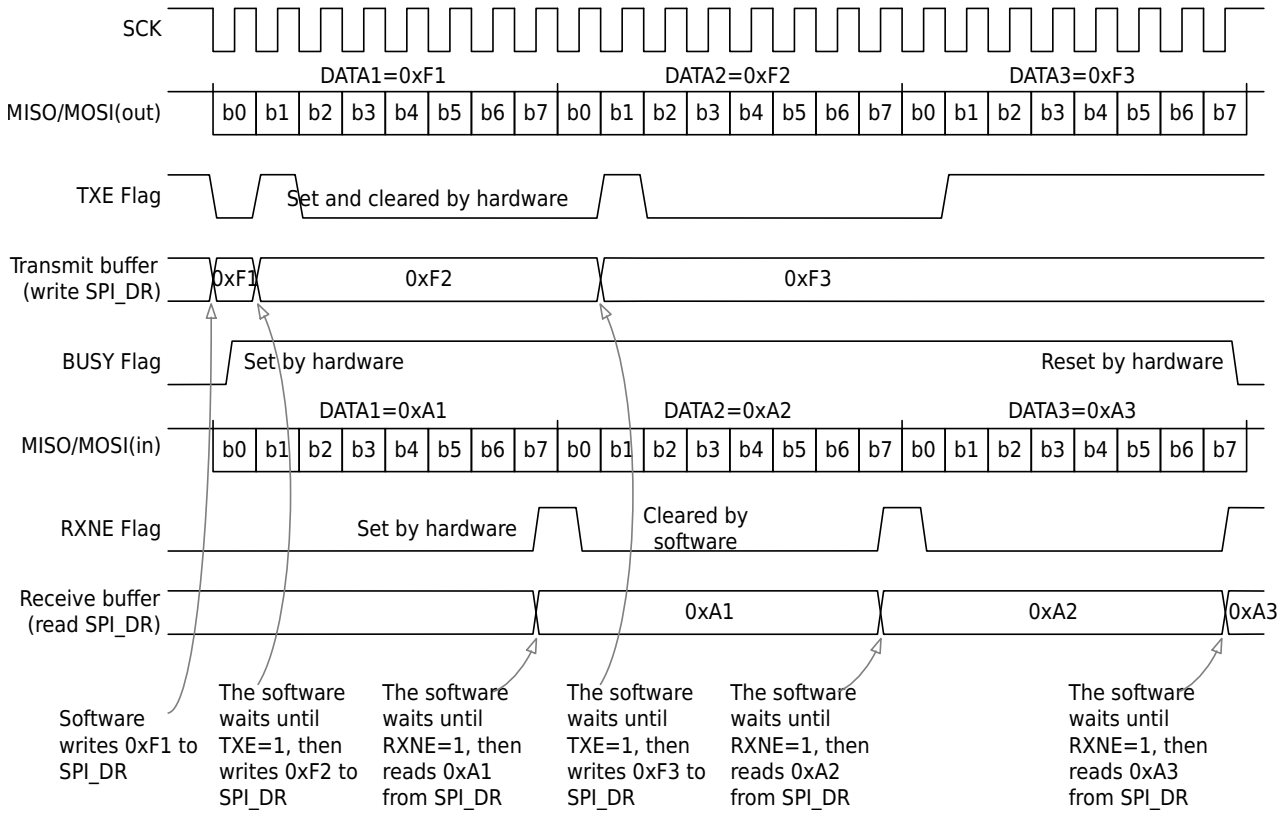



**Figure 21-9 Explanation Diagram of General Flags in SPI Master Mode**

**Slave mode example with CPOL=1 and CPHA=1**

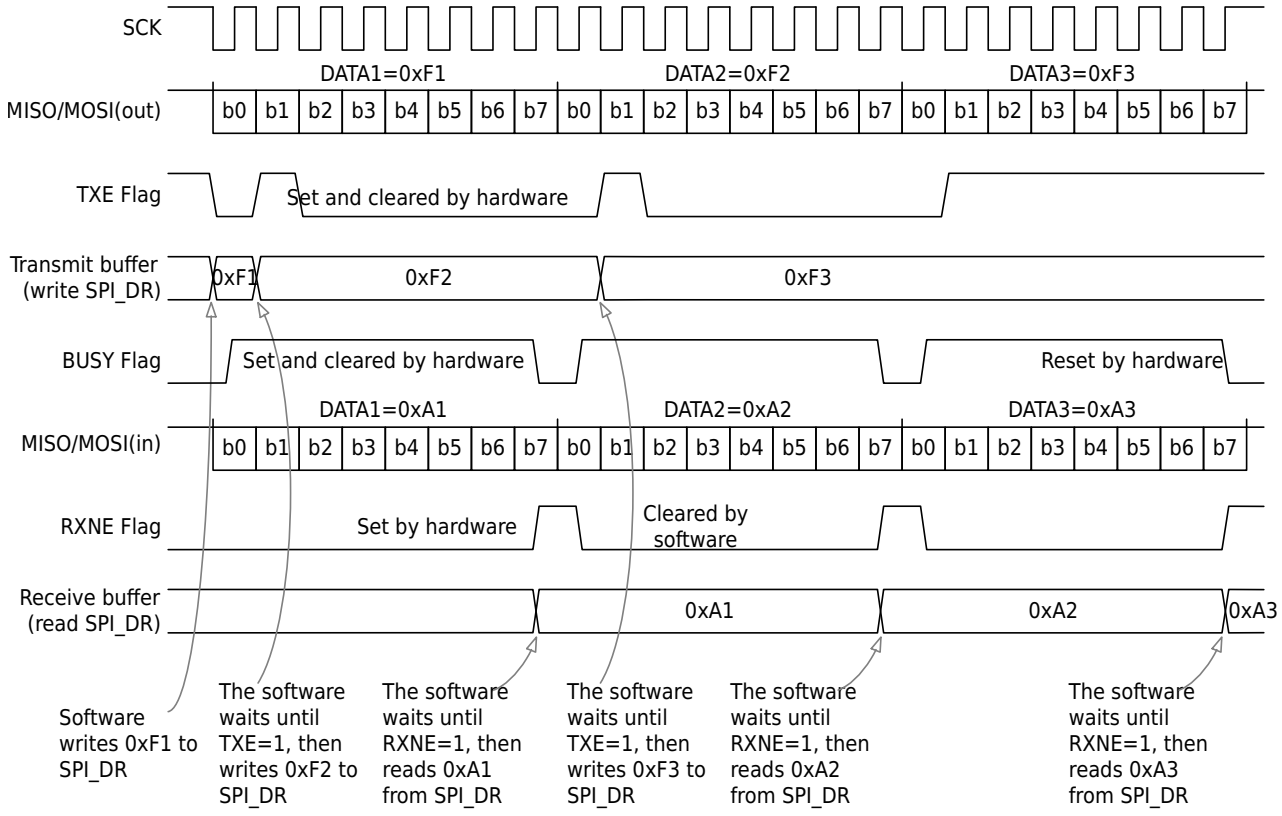



**Figure 21-10 SPI Slave Mode General Flag Bit Description Diagram**


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 21.3.10 Interrupt Generation and Clearing

SPI supports generating interrupt requests from different events, the specific information is as follows:

Table 21-1 SPI Interrupt

| Interrupt Event | Corresponding Flag Bit (SPI_SR) | Corresponding Interrupt Enable (SPI_CR1) | Method to Clear the Interrupt Flag |
|---|---|---|---|
| Slave select input rising edge | SSR | SSRIE | Write 0 to SPI_ICR.SSR |
| Slave select input falling edge | SSF | SSFIE | Write 0 to SPI_ICR.SSF |
| Send buffer empty | TXE | TXEIE | Write data to the data register DR |
| Receive buffer is not empty | RXNE | RXNEIE | Read data from the data register DR or write 0 to SPI_ICR.RXNE. |
| Slave selection error in slave mode | SSERR | SSERRIE | Write 0 to SPI_ICR.SSERR |
| Receive buffer overflow error | OVF | OVFIE | Write 0 to SPI_ICR.OVF |
| Slave mode transmit buffer underflow error | UDF | UDFIE | Write 0 to SPI_ICR.UDF |
| Mode error in master mode | MODF | MODFIE | Write 0 to SPI_ICR.MODF |

As shown in the above table, each interrupt event has an independent interrupt enable control, and the interrupt generated by the error event also has an independent interrupt clear register. For the above interrupt event (or request), no matter whether the interrupt is enabled or not, the method described in the table can be used to clear the interrupt flag.

## 21.4 Operation Example

### 21.4.1 Example of Sending and Receiving Data in Full-duplex by the Master

**Step1.** According to the relevant description of the pin digital multiplexing function in the chapter "General-Purpose I/O (GPIO)", map NSS, SCK, MOSI, and MISO to the required pins.

**Step2.** Configure SYSCTRL module register PERI_CLKEN0.SPIx to 1 to enable SPIx configuration clock and working clock.

**Step3.** Set CR0.MSTR to 1, select the master mode.

**Step4.** Set CR0.BRR to configure the communication baud rate.

**Step5.** Set CR0.CM to 0, select full-duplex communication mode.

**Step6.** Set CR0.SSM to 1 to configure the slave selection mode.

**Step7.** Set CR0.CPOL, CR0.CPHA, select serial clock polarity and phase.

**Step8.** Set CR0.WIDTH to configure the data width of each frame.

**Step9.** Set CR0.LSBF to configure the data sequence order when sending and receiving data.

**Step10.** Set CR0.EN to 1 to enable the SPI module.

**Step11.** Set SSI.SSI to 0, select the corresponding slave.

**Step12.** Write the data to be sent into the DR register, and the SPI automatically starts data transmission.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

**Step13.** Query and wait for SR.RXNE to become 1 to complete the sending and receiving of a frame of data.

**Step14.** Read the received data from the DR register and save it.

**Step15.** If the data to be sent and received has not been completed, jump to step12 to continue execution.

### 21.4.2 Example of Sending and Receiving Data in Full-duplex by the Slave

**Step1.** According to the relevant description of the pin digital multiplexing function in the chapter "General-Purpose I/O (GPIO)", map NSS, SCK, MOSI, and MISO to the required pins.

**Step2.** Configure SYSCTRL module register PERI_CLKEN0.SPIx to 1 to enable SPIx configuration clock and working clock.

**Step3.** Set CR0.MSTR to 0, select slave mode.

**Step4.** Set CR0.CM to 0, select full-duplex communication mode.

**Step5.** Set CR0.SSM to configure the slave selection mode.

**Step6.** Set CR0.CPOL, CR0.CPHA, select serial clock polarity and phase.

**Step7.** Set CR0.WIDTH to configure the data width of each frame.

**Step8.** Set CR0.LSBF to configure the data sequence order when sending and receiving data.

**Step9.** Set CR0.EN to 1 to enable the SPI module.

**Step10.** Set ICR.TXE to 0 to clear the send buffer.

**Step11.** The query waits for SR.SSLVL to become 0, and the SPI master selects this module.

**Step12.** Write the data to be sent into the DR register, and prepare the data to be sent.

**Step13.** Query and wait for SR.RXNE to become 1 to complete the sending and receiving of a frame of data.

**Step14.** Read the received data from the DR register and save it.

**Step15.** If the data to be sent and received has not been completed, jump to step12 to continue execution.

**Step16.** Wait for SR.SSLVL to become 1 to complete this communication.

### 21.4.3 Single-wire Half-duplex Sending and Receiving Data Example

**Step1.** According to the relevant description of the pin digital multiplexing function in the chapter "General-Purpose I/O (GPIO)", map NSS, SCK, MOSI of the master and NSS, SCK, MISO of the slave to the required pins .

**Step2.** Configure SYSCTRL module register PERI_CLKEN0.SPIx to 1 to enable SPIx configuration clock and working clock.

**Step3.** The master sets CR0.MSTR to 1, and the slave sets CR0.MSTR to 0.

**Step4.** The master sets CR0.BRR to configure the communication baud rate.

**Step5.** Set CR0.CM to 0b11, select single-line half-duplex communication mode.

**Step6.** Configure CR0.SSM to configure the slave selection mode.

**Step7.** Configure CR0.CPOL, CR0.CPHA, select serial clock polarity and phase.

**Step8.** Configure CR0.WIDTH to configure the data width of each frame.

**Step9.** Configure CR0.LSBF to configure the data sequence order when sending data.

**Step10.** Set CR0.EN to 1 to enable the SPI module.

**Step11.** The master sets SSI.SSI to 0 and selects the corresponding slave; the slave waits for SR.SSLVL to become 0 to select this module.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

**Step12.** The master sets HDOE.HDOE to 1 and switches to the sending mode; the slave sets HDOE.HDOE to 0 and switches to the receiving mode.

**Step13.** The master writes the data to be sent into the DR register, and the SPI automatically starts data transmission.

**Step14.** The master query waits for SR.BUSY to become 0 to complete the transmission of a frame of data; the slave query waits for SR.RXNE to become 1 to complete the reception of a frame of data.

**Step15.** The slave reads the received data from the DR register and saves it.

**Step16.** If the data to be sent by the master is not completed, then jump to step13 to continue execution.

**Step17.** The master sets HDOE.HDOE to 0 and switches to the receiving mode; the slave sets HDOE.HDOE to 1 and switches to the sending mode.

**Step18.** The slave query waits for SR.TXE to become 1, and the sending buffer is idle.

**Step19.** If the data to be sent by the slave is not completed, write a frame of data to the DR register.

**Step20.** Writes arbitrary data to the DR register, and the SPI master sends SCK for data transmission.

**Step21.** The master query waits for SR.RXNE to become 1, and completes the reception of a frame of data.

**Step22.** The master reads the received data from the DR register and saves it.

**Step23.** If the data to be sent by the slave is not completed, then jump to step18 to continue execution.

**Step24.** The slave waits for SR.SSLVL to become 1, sets HDOE.HDOE to 0, and completes this communication.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

## 21.5 Register Description

### 21.5.1 Register List

- SPI0 base address: 0x40000800
- SPI1 base address: 0x40008800

Table 21-2 Serial Peripheral Interface (SPI) Register Offset

| Offset | Register | Description | Width |
|---|---|---|---|
| 0x00 | SPI_CR0 | SPI Control Register 0 | 32 |
| 0x04 | SPI_CR1 | SPI Control Register 1 | 32 |
| 0x08 | SPI_HDOE | SPI Single-wire Half-duplex Output Enable Register | 32 |
| 0x0C | SPI_SSI | SPI Internal Slave Select Register | 32 |
| 0x10 | SPI_SR | SPI Status Register | 32 |
| 0x14 | SPI_ICR | SPI Interrupt Clear Register | 32 |
| 0x18 | SPI_DR | SPI Data Register | 32 |

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 21.5.2 SPI Control Register 0 (SPI_CR0)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000700 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | SMP | CM | | SSM | WIDTH | | | | LSBF | MSTR | CPOL | CPHA | BRR | | | EN |
| Access | | | | | | | | | | | | | | | | | RW | RW | | RW | RW | | | | RW | RW | RW | RW | RW | | | RW |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:16* | *Reserved* | - | - |
| 15 | SMP | Master mode delayed sample bit<br>● 0b0: In traditional SPI mode, the data received by the master is sampled according to the settings of CPOL and CPHA<br>● 0b1: Delayed sampling method, the sampling time is delayed by half an SCK cycle compared with the traditional method | RW |
| 14:13 | CM | Communication mode configuration<br>● 0b00: full-duplex bidirectional<br>● 0b01: simplex transmit only<br>● 0b10: simplex receive only<br>● 0b11: single-wire half-duplex | RW |
| 12 | SSM | Slave selection configuration<br>Master mode:<br>● 0b0: NSS pin determines the slave select input value<br>● 0b1: SSI register determines the slave select output value<br>Slave mode:<br>● 0b0: NSS pin determines the slave select input value<br>● 0b1: SSI register determines the slave select input value | RW |
| 11:8 | WIDTH | Frame data width configuration<br>● 0b0011: 4-bit<br>● 0b0100: 5-bit<br>● ...<br>● 0b1110: 15-bit<br>● 0b1111: 16-bit<br>● Others: Setting prohibited | RW |
| 7 | LSBF | Data frame high and low order selection<br>● 0b0: Most Significant Bit (MSB) transmitted and received first<br>● 0b1: Least Significant Bit (LSB) transmitted and received first | RW |
| 6 | MSTR | Operating mode configuration<br>● 0b0: Slave mode<br>● 0b1: Master mode | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 5 | CPOL | Serial clock polarity configuration<br>● 0b0: Low level during standby<br>● 0b1: High level during standby | RW |
| 4 | CPHA | Serial clock phase configuration<br>● 0b0: Front edge sampling/back edge shift<br>● 0b1: Front edge shift/back edge sampling | RW |
| 3:1 | BRR | Master mode baud rate configuration<br>● 0b000: PCLK/2<br>● 0b001: PCLK/4<br>● 0b010: PCLK/8<br>● 0b011: PCLK/16<br>● 0b100: PCLK/32<br>● 0b101: PCLK/64<br>● 0b110: PCLK/128<br>● 0b111: Setting prohibited | RW |
| 0 | EN | SPI module enable control<br>● 0b0: SPI module disabled<br>● 0b1: SPI module enabled | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 21.5.3 SPI Control Register 1 (SPI_CR1)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x04 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | DMATXEN | DMARXEN | MODFIE | SSERRIE | OVFIE | UDFIE | SSRIE | SSFIE | RXNEIE | TXEIE |
| Access | | | | | | | | | | | | | | | | | | | | | | | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:10* | *Reserved* | - | - |
| 9 | DMATXEN | Transmit DMA enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 8 | DMARXEN | Receive DMA enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 7 | MODFIE | Mode error interrupt enable control<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 6 | SSERRIE | Slave select error interrupt enable control in slave mode<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 5 | OVFIE | Receive buffer overflow error interrupt enable control<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 4 | UDFIE | Transmit buffer underflow error interrupt enable control in slave mode<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 3 | SSRIE | Slave select input rising edge interrupt enable control<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 2 | SSFIE | Slave select input falling edge interrupt enable control<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 1 | RXNEIE | Receive buffer non-empty interrupt enable control<br>● 0b0: Disable<br>● 0b1: Enable | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 0 | TXEIE | Transmit buffer empty interrupt enable control<br>● 0b0: Disable<br>● 0b1: Enable | RW |

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 21.5.4 SPI Single-wire Half-duplex Output Enable Register (SPI_HDOE)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x08 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | HDOE |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | RW |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:1* | *Reserved* | - | - |
| 0 | HDOE | MOSI/MISO transceiver switching control<br>● 0b0: Receive only<br>● 0b1: Transmit only<br>**Note**<br>Only valid in single-wire half-duplex communication. | RW |

### 21.5.5 SPI Internal Slave Select Register (SPI_SSI)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x0C | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000001 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | SSI |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | RW |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:1* | *Reserved* | - | - |
| 0 | SSI | Internal slave select<br>Slave select output value is determined by the register in master mode.<br>Slave select input value is determined by the register in slave mode.<br>**Note**<br>Valid when the SPI_CR0.SSM register is 1. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 21.5.6 SPI Status Register (SPI_SR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x10 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000201 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | SSLVL | BUSY | MODF | SSERR | OVF | UDF | SSR | SSF | RXNE | TXE |
| Access | | | | | | | | | | | | | | | | | | | | | | | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:10* | *Reserved* | - | - |
| 9 | SSLVL | State of slave select input<br>● 0b0: Slave select input is low level<br>● 0b1: Slave select input is high level<br>**Note**<br>This bit setting is invalid when in master mode and SPI_CR0.SSM=1. | RO |
| 8 | BUSY | Bus busy flag bit<br>● 0b0: Bus idle<br>● 0b1: Bus busy | RO |
| 7 | MODF | Mode error flag bit<br>● 0b0: Normal<br>● 0b1: Error<br>**Note**<br>MODF is triggered when the slave select input is low in master mode. | RO |
| 6 | SSERR | Slave selection error flag in slave mode<br>● 0b0: Normal<br>● 0b1: Error | RO |
| 5 | OVF | Receive buffer overflow error flag bit<br>● 0b0: Normal<br>● 0b1: Error | RO |
| 4 | UDF | Transmit buffer underflow error flag in slave mode<br>● 0b0: Normal<br>● 0b1: Error | RO |
| 3 | SSR | Slave select input rising edge flag<br>● 0b0: No rising edge detected<br>● 0b1: Rising edge detected<br>**Note**<br>This bit setting is invalid when in master mode and SPI_CR0.SSM=1. | RO |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 2 | SSF | Slave select input falling edge flag<br>● 0b0: No falling edge detected<br>● 0b1: Falling edge detected<br>**Note**<br>This bit setting is invalid when in master mode and SPI_CR0.SSM=1. | RO |
| 1 | RXNE | Receive buffer not empty flag<br>● 0b0: Receive buffer empty<br>● 0b1: Receive buffer not empty | RO |
| 0 | TXE | Send buffer empty flag<br>● 0b0: Transmit buffer not empty<br>● 0b1: Transmit buffer empty | RO |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 21.5.7 SPI Interrupt Clear Register (SPI_ICR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x14 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x000000FF | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | MODF | SSERR | OVF | UDF | SSR | SSF | RXNE | TXE |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | WO | WO | WO | WO | WO | WO | WO | WO |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:8* | *Reserved* | - | - |
| 7 | MODF | SR.MODF flag clear bit<br>Write 0 to clear flag, write 1 has no effect; read always returns 1. | WO |
| 6 | SSERR | SR.SSERR flag clear bit<br>Write 0 to clear flag, write 1 has no effect; read always returns 1. | WO |
| 5 | OVF | SR.OVF flag clear bit<br>Write 0 to clear flag, write 1 has no effect; read always returns 1. | WO |
| 4 | UDF | SR.UDF flag clear bit<br>Write 0 to clear flag, write 1 has no effect; read always returns 1. | WO |
| 3 | SSR | SR.SSR flag clear bit<br>Write 0 to clear flag, write 1 has no effect; read always returns 1. | WO |
| 2 | SSF | SR.SSF flag clear bit<br>Write 0 to clear flag, write 1 has no effect; read always returns 1. | WO |
| 1 | RXNE | SR.RXNE flag clear bit<br>Write 0 to clear flag, write 1 has no effect; read always returns 1. | WO |
| 0 | TXE | SR.TXE clear bit<br>Write 0 to set the SR.TXE flag to 1, write 1 has no effect; read always returns 1. | WO |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 21.5.8 SPI Data Register (SPI_DR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x18 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | DR | | | | | | | | | | | | | | | |
| Access | | | | | | | | | | | | | | | | | RW | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:16* | *Reserved* | - | - |
| 15:0 | DR | SPI data bits<br>When sending data, write the data to be sent into this register<br>When receiving data, read the received data from this register | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

# 22 ADC

## 22.1 Introduction

External analog signals need to be converted into digital signals to be further processed by the MCU. This product integrates an 12-bit high-precision, high-speed successive approximation register (SAR ADC) module.

## 22.2 Features

- 12-bit conversion accuracy
- A maximum of 2Msps conversion speed
- Up to 19 input channels are supported, including 16 external pin inputs, 1 internal temperature sensor voltage channel, 1 channel of 1/3 AVCC voltage, and 1 channel of internal 1.2V reference voltage VREF1P2
- 4 reference sources: AVCC voltage, EXVREF pins, internal 1.5V reference voltage, internal 2.5V reference voltage
- ADC voltage input range: 0-Vref
- Built-in signal follower, which can convert high-impedance signals
- Multiple conversion modes: single conversion, SQR scan conversion, and JQR scan conversion
- Software can configure ADC conversion rate
- Support on-chip peripherals to automatically trigger ADC conversion, effectively reducing chip power consumption and improving real-time conversion
- ADC conversion results are continuously accumulated
- Input channel voltage threshold monitoring


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

# 22.3 Functional Description

## 22.3.1 ADC Block Diagram

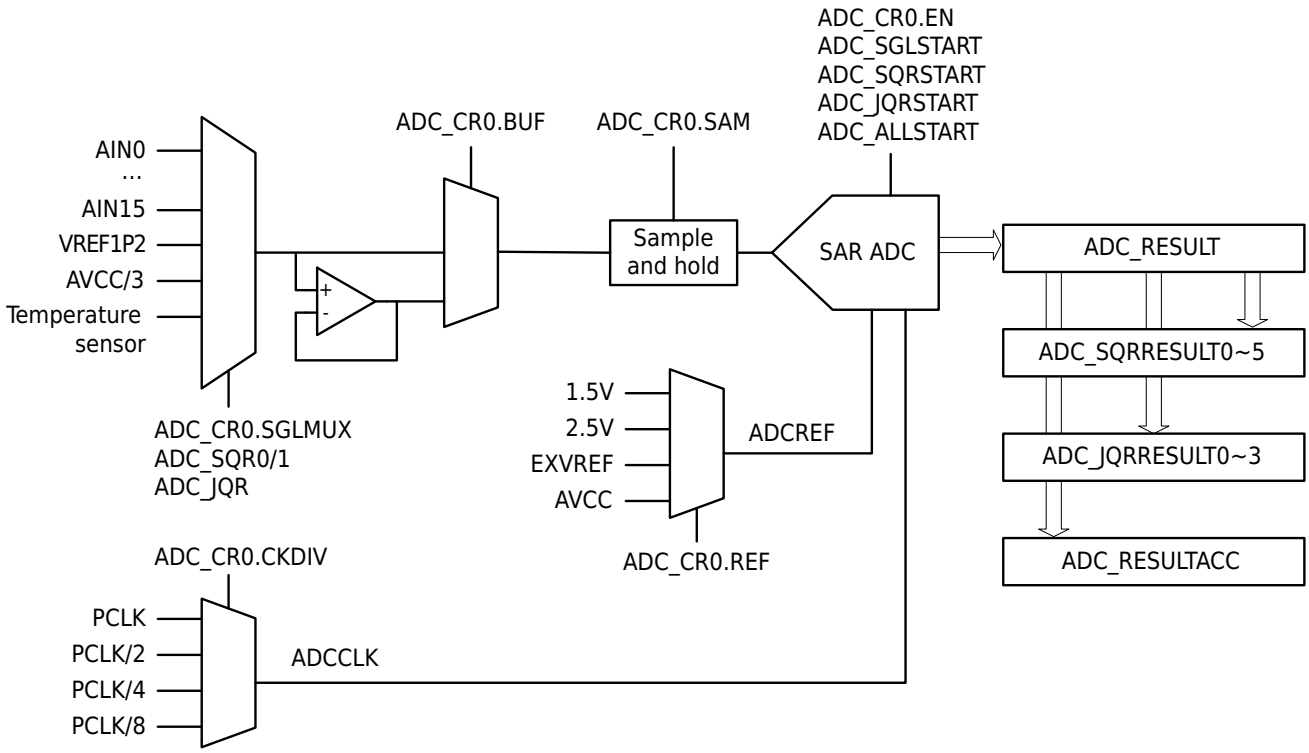



Figure 22-1 ADC Block Diagram

## 22.3.2 Conversion Timing and Conversion Speed

The ADC conversion timing is shown in the figure below: a complete ADC conversion consists of a sampling process and a successive comparison process. The sampling process requires 2-127 ADCCLK cycles, configured by ADC_CR0.SAM; the successive comparison process requires 12 ADCCLK cycles. Therefore, an ADC conversion requires a total of 14-139 ADCCLK cycles.

The ADC conversion speed is measured in sps (samples per second), indicating how many ADC conversions are performed per second. The ADC conversion speed is: the frequency of ADCCLK/the number of ADCCLKs required for one ADC conversion.

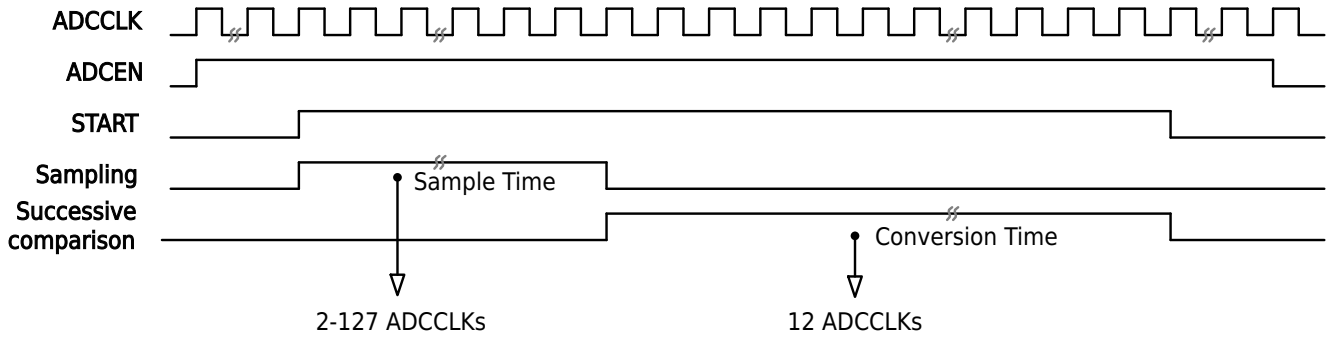



Figure 22-2 ADC Conversion Timing Diagram

ADC conversion speed is related to the ADC reference voltage and AVCC voltage. For more information, see the "Electrical Specifications" chapter in "Datasheet".


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 22.3.3 Single Conversion Mode

In the single conversion mode, after ADC startup, only one sampling conversion is performed, which can convert all ADC channels. This mode can be started either by setting the ADC_SGLSTART.START bit for software trigger or by configuring an external trigger via ADC_EXTTRIGGER0. Once the ADC conversion of the selected channel is completed, ADC_IFR.SGLIF bit is automatically set to 1, and the conversion result is saved in the ADC_RESULT register.

If continuous conversion of a specific channel is required, set the ADC_ALLSTART.START bit to 1. To stop conversion, set the ADC_ALLSTART.START bit to 0 or disable the ADC directly (ADC_CR0.EN = 0).

### 22.3.4 Scan Conversion Mode

Scan conversion mode is divided into two types: SQR scan conversion and JQR scan conversion. When operating independently, both modes can continuously perform conversions on multiple channels. The JQR sequence can preempt the SQR sequence during operation; for detailed behavior, see the Table 22-1 : Preemptive Behavior State Table.

#### SQR Scan Conversion Mode

SQR scan conversion mode supports up to 6 consecutive conversions. The total number of conversions is configured by ADC_SQR1.CNT, and all channels can be configured for conversion through ADC_SQRx.CHxMUX. This mode can be initiated either by setting the ADC_SQRSTART.START bit for software trigger or by setting the external trigger of ADC_EXTTRIGGER0. After conversion start, the ADC module sequentially converts the channels configured in ADC_SQR0.CHxMUX to ADC_SQR0.CH0MUX until the total conversion count is completed. After the ADC module completes the total number of conversions, the ADC_IFR.SQRIF bit will be automatically set to 1, and the conversion result will be saved in the ADC_SQRRESULTx-ADC_SQRRESULT0 register corresponding to the conversion channel. The following diagram illustrates the SQR scan conversion process for performing 6 conversions on AIN0, AIN1, and AIN5. In this configuration, SQR scan conversion channels 4 and 1 are set to AIN1, channels 3 and 0 to AIN0, and channels 5 and 2 to AIN5. After ADC_SQRSTART.START is set to 1, the ADC module will sequentially convert SQR scan channels 5 to 0.

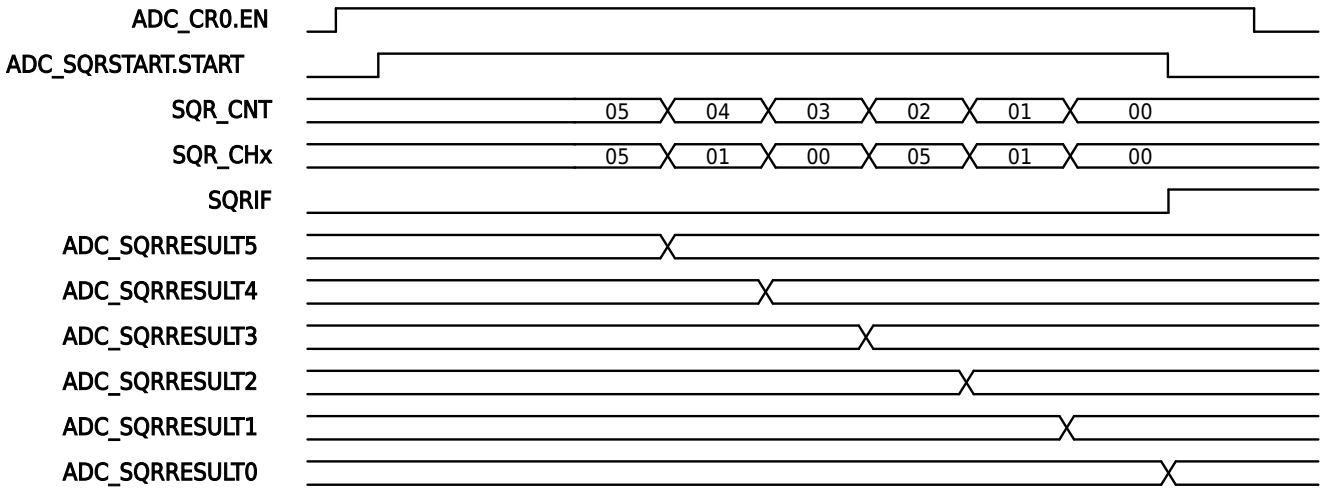



Figure 22-3 Examples of ADC SQR Scan Conversion Process

#### JQR Scan Conversion Mode

JQR scan conversion mode can perform up to 4 consecutive conversions, and the total number of conversions is configured by ADC_JQR.CNT; all channels can be configured for conversion, and the channel to be converted is configured by ADC_JQR.CHxMUX. This mode can be started either by setting the the


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

ADC_JQRSTART.START bit or by configuring an external trigger via ADC_EXTTRIGGER1. This mode can be inserted into the SQR scan conversion queue or started independently. After starting the conversion, the ADC module sequentially converts the channels configured from ADC_JQR.CHxMUX-ADC_JQR.CH0MUX, until the total number of conversions is completed. After the ADC module completes the total conversion count, the ADC_IFR. bit is automatically set to 1, and the conversion results are stored in the corresponding ADC_JQRRESULTx-ADC_JQRRESULT0 registers. The figure below demonstrates the process of performing 4 conversions of AIN0, AIN1, and AIN5 in the JQR scan conversion process. JQR scan conversion channels 3, 2, 1, and 0 are configured as AIN5, AIN0, AIN1, and AIN5, respectively. After ADC_JQRSTART.START is set to 1, the ADC module will sequentially convert the JQR scanning conversion channels 3 to 0.

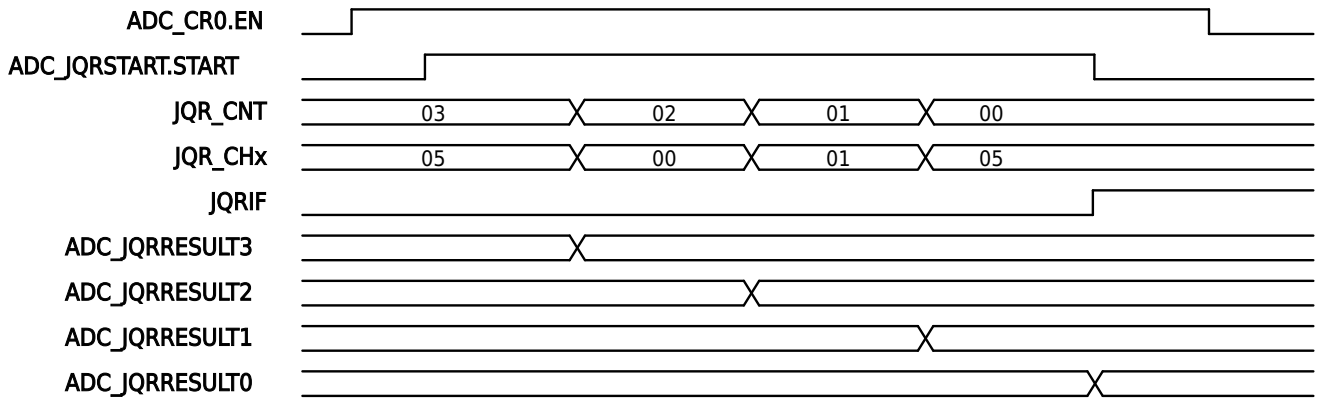



Figure 22-4 Examples of ADC JQR Scan Conversion Process

The execution priority of JQR scan conversion is higher than that of SQR scan conversion. When SQR scan conversion is in progress and a JQR scan conversion starts, SQR scan conversion pauses after completing the current channel conversion. Once JQR scan conversion finishes, the scan resumes and continues converting the remaining channels.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

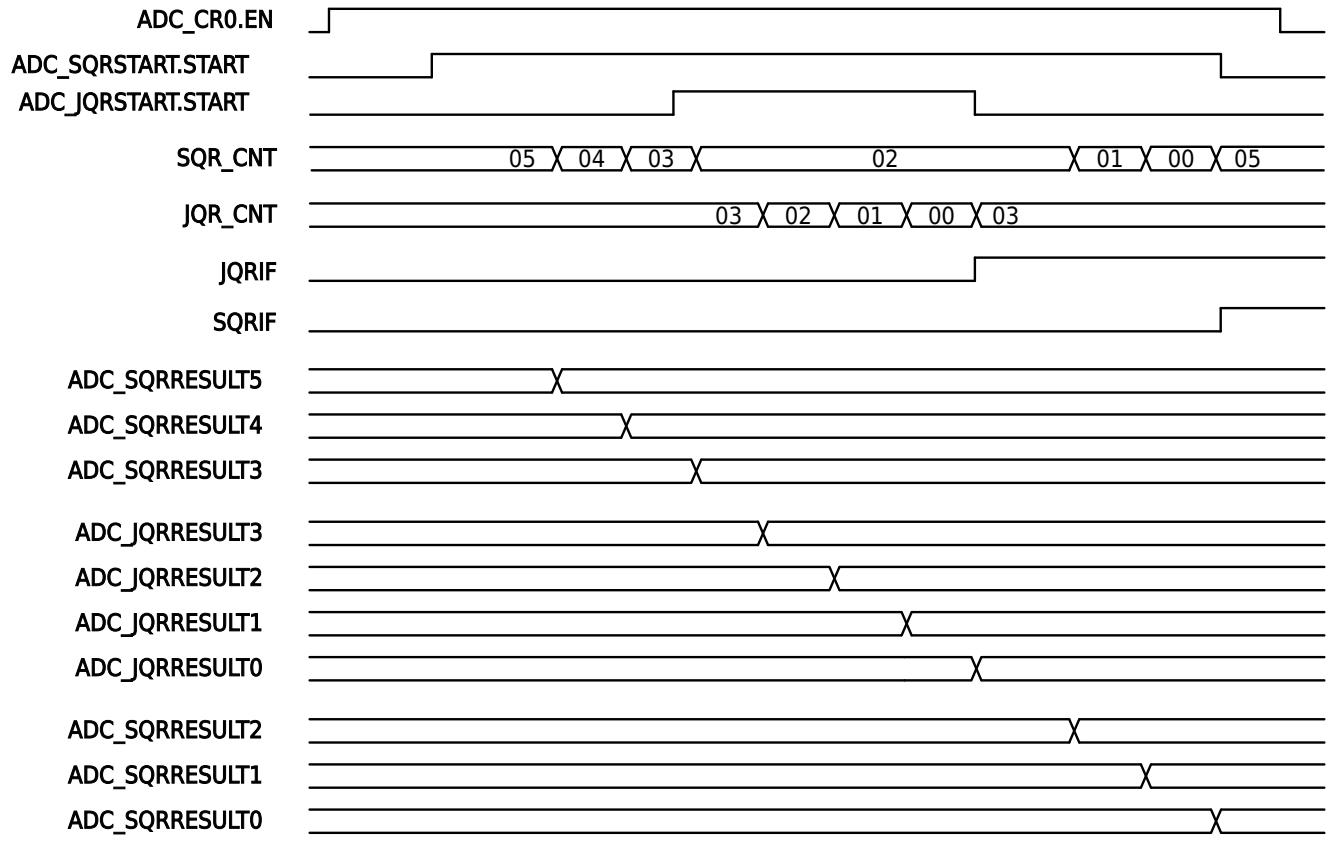



Figure 22-5 Examples of JQR Scan During ADC SQR Scan Process

Table 22-1 Preemptive Behavior State Table

| Current Queue Status | Preemptive Behavior | |
|---|---|---|
| | Start SQR Scan | Start JQR Scan |
| JQR scan in progress | No response; continue current JQR scan | No response; continue current JQR scan |
| SQR scan in progress | No response; continue current SQR scan | Switch to JQR scan after completing the current channel conversion; resume the previous SQR queue after JQR scan conversion is completed. |

**Note**

If both JQR and SQR scan conversions are triggered simultaneously, JQR scan takes priority. After JQR scan completes, the SQR scan will be executed.

### Scan Conversion Triggers DMA Read

After SQR scan and JQR scan conversions are completed, DMA can be automatically triggered to read the conversion results. In SQR scan mode, enable the function by setting ADC_CR1.DMASQR to 1; in JQR scan mode, enable the function by setting ADC_CR1.DMAJQR to 1.

## 22.3.5 ADC Conversion Result Accumulation

Enable the conversion accumulation function by setting the ADC_CR1.RACCEN bit. In this mode, the results of each ADC conversion are accumulated, and the accumulated conversion result is stored in the ADC_RESULTACC register. The following diagram illustrates the continuous accumulation process in


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

SQR mode, where a total of 6 consecutive conversions are performed on AIN0, AIN1, and AIN5. SQR scan conversion channels 3 and 0 are configured as AIN0; channels 5 and 2 as AIN1; channels 4 and 1 as AIN5. After ADC_SQRSTART.START is set to 1, the ADC module will sequentially perform conversions according to the SQR scan conversion channel configuration until the SQR_CNT count reaches 0. The ADC_RESULTACC register is automatically accumulated each time a conversion is completed. The conversion results of AIN0, AIN1, and AIN5 given in the figure are 0x010, 0x020, and 0x040 in turn.

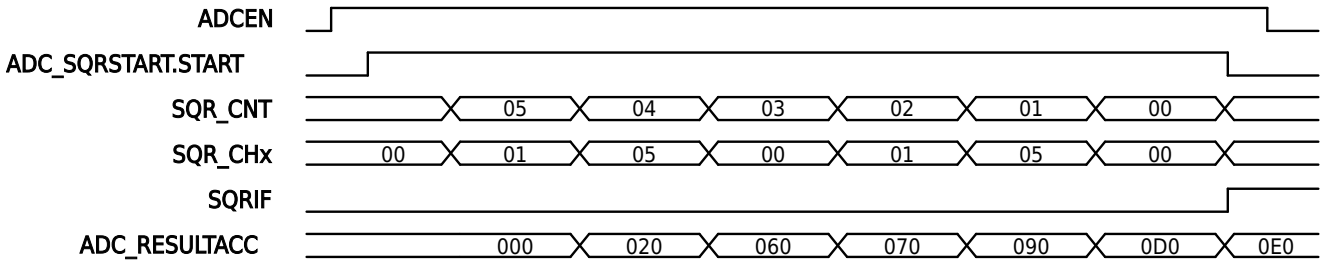



Figure 22-6 Examples of Continuous ADC Conversion Accumulation Process

### 22.3.6 ADC Conversion External Trigger Source

ADC conversions can be initiated either by software configuration or by an external trigger.

The ADC_EXTTRIGGER0 registers can be configured to set the external trigger source for single or SQR scan conversions.

The ADC_EXTTRIGGER1 register can be configured to set the external trigger source for ADC JQR scan conversions.

For detailed trigger sources, see the descriptions of the ADC_EXTTRIGGER0 and ADC_EXTTRIGGER1 registers.

### 22.3.7 ADC Conversion Result Comparison

When the ADC conversion is completed, the ADC conversion result can be compared with the threshold set by the user, supporting high threshold comparison, low threshold comparison, and interval value comparison. This function requires setting the corresponding control bits ADC_CR1.HTCMP, ADC_CR1.LTCMP and ADC_CR1.REGCMP to 1. This function can realize the automatic monitoring of the analog quantity, until the ADC conversion result meets the user's expectations, an interrupt is generated to apply for user program interface entry. Monitoring channel selection is configured through ADC_CR1.THCH.

- High Threshold Comparison: When the ADC conversion result falls within the range [ADC_HT, 4095], the ADC_IFR.HTIF bit is set to 1; writing 0 to ADC_ICR.HTIC clears ADC_IFR.HTIF.
- Low Threshold Comparison: If the ADC conversion result falls within the range [0, ADC_LT), ADC_IFR.LTIF is set to 1; writing 0 to ADC_ICR.LTIC clears ADC_IFR.LTIF.
- Interval Value Comparison: If the ADC conversion result falls within the range [ADC_LT, ADC_HT), ADC_IFR.REGIF is set to 1; writing 0 to ADC_ICR.REGIC clears ADC_IFR.REGIF.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

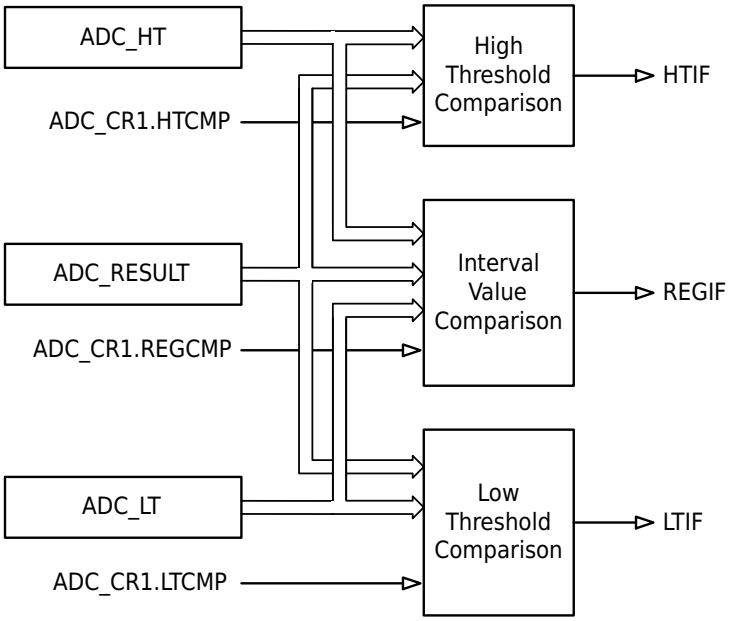



Figure 22-7 ADC Conversion Result Comparison Diagram

### 22.3.8 ADC Interrupt

The ADC interrupt request is shown in the table below:

Table 22-2 ADC Interrupt

| Interrupt Source | Interrupt Flag | Interrupt Enable |
|---|---|---|
| ADC JQR scan conversion complete | ADC_IFR.JQRIF | ADC_CR0.IE |
| ADC SQR scan conversion complete | ADC_IFR.SQRIF | |
| ADC conversion result is within the range (greater than or equal to the low threshold and less than the high threshold) | ADC_IFR.REGIF | |
| ADC conversion result is within the high threshold region (greater than or equal to the high threshold) | ADC_IFR.HTIF | |
| ADC conversion result is within the low threshold region (less than the low threshold) | ADC_IFR.LTIF | |
| ADC single conversion completed | ADC_IFR.SGLIF | |

### 22.3.9 Measure Ambient Temperature Using a Temperature Sensor

The output voltage of the temperature sensor will change with the change of the ambient temperature, so the corresponding ambient temperature can be calculated according to the output voltage of the temperature sensor. When the measurement channel of the ADC module selects the output voltage of the temperature sensor, the ambient temperature can be measured.

The calculation formula of ambient temperature $T_A$ is as follows:

$$T_A(℃) = \begin{cases} \frac{1}{16} \cdot T_0 + K \cdot V_{ref} \cdot (C_{1\_1P5ref} - C_{0\_1P5ref}), \ldots\ldots\ldots (V_{ref} = 1.5V) \\ \frac{1}{16} \cdot T_0 + K \cdot V_{ref} \cdot (C_{1\_2P5ref} - C_{0\_2P5ref}), \ldots\ldots\ldots (V_{ref} = 2.5V) \end{cases}$$

Where:

- Vref is the reference voltage of the current ADC module, with typical values of 1.5V or 2.5V.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

- $C_{1\_1P5ref}$ and $C_{1\_2P5ref}$ are the result of the ADC module measuring the output voltage of the temperature sensor, and the value is 0-4095.
- $T_0$ is representative values of the ambient temperature recorded during the temperature test, and their values are 16 times the absolute value of the actual temperature value.
- K represents the slope of the output curve for the temperature sensor. Due to the influence of manufacturing processes, the slopes of temperature curves vary across different chips. For typical values, please refer to the *Datasheet*.
- Both $C_{0\_1P5ref}$ and $C_{0\_2P5ref}$ are calibration values obtained through temperature testing, representing the results of ADC module measurements of the temperature sensor's output voltage under test conditions where the power supply voltage is 3.3V and the ambient temperature is $T_0$.

**Table 22-3 Temperature Sensor Calibration Value Information List**

| Temperature | Calibration Value Name | Calibration Value Storage Address | Width |
|---|---|---|---|
| Room temperature | $C_{0\_1P5ref}$ | 0x00101BE8-0x00101BE9 | 12 bits |
| | $C_{0\_2P5ref}$ | 0x00101BEA-0x00101BEB | 12 bits |
| | $T_0$ | 0x00101BEC-0x00101BED | 16 bits |

**An Example Calculation is as Follows:**

Condition 1: Vref=2.5, $C_{0\_2P5ref}$=1389, $C_{1\_2P5ref}$=1180, K=0.0798, $T_0$=400:

Temperature 1:

$$T_A(℃) = \frac{1}{16} \cdot T_0 + K \cdot V_{ref} \cdot (C_{1\_2P5ref} - C_{0\_2P5ref}) = 25 + 0.0798 \times 2.5 \times (1180 - 1389) = -16.7℃$$

Condition 2: Vref=1.5, $C_{0\_1P5ref}$=2317, $C_{1\_1P5ref}$=2858, K=0.0798, $T_0$=400:

Temperature 2:

$$T_A(℃) = \frac{1}{16} \cdot T_0 + K \cdot V_{ref} \cdot (C_{1\_1P5ref} - C_{0\_1P5ref}) = 25 + 0.0798 \times 1.5 \times (2858 - 2317) = 89.76℃$$

## 22.4 Operation Example

### 22.4.1 Start the ADC Single Conversion Operation Process through the ADC_SGLSTART.START Bit

#### Procedure

**Step1.** Configure the corresponding bits of PAADS-PCADS, and configure the ADC channel to be converted as an analog port.

**Step2.** Set PBADS[1] to 1 to configure the ADC external reference voltage pin as an analog port.

> **Note**
> If the ADC reference voltage does not select an external reference voltage pin, this step can be skipped.

**Step3.** Set ADC_BGR.BGR_EN to 1, to enable the BGR module.

**Step4.** Set ADC_CR0.EN to 1 to enable the ADC module.

**Step5.** Delay for the longer of $T_{ADCSTART}$ and $T_{stable(BGR)}$ (refer to the "Electrical CharacteristicsOperating ConditionsInternal Reference Voltage/ADC Characteristics" section in the "Datasheet" for specific values), and wait for the ADC and BGR module to complete startup.

**Step6.** Set ADC_CR1.MODE to 0 to select single conversion mode.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

**Step7.** Configure ADC_CR0.REF and select the reference voltage of the ADC.

**Step8.** Configure ADC_CR0.SAM and ADC_CR0.CLKDIV to set the conversion speed of ADC.

**Step9.** Configure ADC_CR0.SGLMUX to select the channel to be converted.

**Step10.** Set ADC_ICR.SGLIC to 0, and clear ADC_IFR.SGLIF flag.

**Step11.** Set ADC_SGLSTART.START to 1 to start ADC single conversion. If continuous conversion is still needed, set ADC_ALLSTART.START to 1.

**Step12.** Wait for ADC_IFR.SGLIF to become 1, read the ADC_RESULT register to get the ADC conversion result.

**Step13.** To convert other channels, repeat steps9-12. If continuous conversion has been enabled, set ADC_ALLSTART.START to 0 to stop it before performing conversions on other channels, then repeat steps9-12.

**Step14.** Set ADC_CR0.EN and ADC_BGR.BGR_EN to 0, turn off ADC module and BGR module.

### 22.4.2 ADC Single Conversion Operation Process through an External Trigger

#### Procedure

**Step1.** Configure the corresponding bits of PAADS-PCADS, and configure the ADC channel to be converted as an analog port.

**Step2.** Set PBADS[1] to 1 to configure the ADC external reference voltage pin as an analog port.

> **Note**
> If the ADC reference voltage does not select an external reference voltage pin, this step can be skipped.

**Step3.** Set ADC_BGR.BGR_EN to 1, to enable the BGR module.

**Step4.** Set ADC_CR0.EN to 1 to enable the ADC module.

**Step5.** Delay for the longer of $T_{ADCSTART}$ and $T_{stable(BGR)}$ (refer to the "Electrical CharacteristicsOperating ConditionsInternal Reference Voltage/ADC Characteristics" section in the "Datasheet" for specific values), and wait for the ADC and BGR module to complete startup.

**Step6.** Set ADC_CR1.MODE to 0 to select single conversion mode.

**Step7.** Set ADC_CR0.IE to 1 to enable ADC interrupt.

**Step8.** Enable the ADC interrupt in the NVIC interrupt vector table.

**Step9.** Configure ADC_CR0.REF and select the reference voltage of the ADC.

**Step10.** Configure ADC_CR0.SAM and ADC_CR0.CLKDIV to set the conversion speed of ADC.

**Step11.** Configure ADC_CR0.SGLMUX to select the channel to be converted.

**Step12.** Set ADC_ICR to 0x0, clear the ADC interrupt flag.

**Step13.** Configure ADC_EXTTRIGGER0 to select the external trigger condition.

**Step14.** When an external trigger condition triggers the ADC to complete the conversion, the ADC module will generate an interrupt. The user can read the ADC_RESULT register in the ADC interrupt service program to obtain the ADC conversion result.

**Step15.** To convert other channels, repeat steps11-14.

**Step16.** Set ADC_CR0.EN and ADC_BGR.BGR_EN to 0, turn off ADC module and BGR module.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 22.4.3 Start the ADC SQR Scan Conversion Operation Process through the ADC_SQRSTART.START Bit

#### Procedure

**Step1.** Configure the corresponding bits of PAADS-PCADS, and configure the ADC channel to be converted as an analog port.

**Step2.** Set PBADS[1] to 1 to configure the ADC external reference voltage pin as an analog port.

> **Note**
> If the ADC reference voltage does not select an external reference voltage pin, this step can be skipped.

**Step3.** Set ADC_BGR.BGR_EN to 1, to enable the BGR module.

**Step4.** Set ADC_CR0.EN to 1 to enable the ADC module.

**Step5.** Delay for the longer of $T_{ADCSTART}$ and $T_{stable(BGR)}$ (refer to the "Electrical CharacteristicsOperating ConditionsInternal Reference Voltage/ADC Characteristics" section in the "Datasheet" for specific values), and wait for the ADC and BGR module to complete startup.

**Step6.** Set ADC_CR1.MODE to 1 to select scan conversion mode.

**Step7.** Configure ADC_CR0.REF and select the reference voltage of the ADC.

**Step8.** Configure ADC_CR0.SAM and ADC_CR0.CLKDIV to set the conversion speed of ADC.

**Step9.** Configure ADC_SQR0.CHxMUX, and select SQR scan conversion channels.

**Step10.** Configure ADC_SQR1.CNT, and select the total number of conversions for SQR scan conversions.
The total number of conversions needs to be consistent with the number of conversion channels configured in step9.

**Step11.** Set ADC_ICR.SQRIC to 0, clear ADC_IFR.SQRIF flag.

**Step12.** Set ADC_SQRSTART.START to 1 to start ADC SQR scan conversion.

**Step13.** Wait for ADC_IFR.SQRIF to become 1, read ADC_SQRRESULTx-ADC_SQRRESULT0 registers to obtain the conversion result of the corresponding channel.

**Step14.** To convert other channels, repeat steps9-13.

**Step15.** Set ADC_CR0.EN and ADC_BGR.BGR_EN to 0, turn off ADC module and BGR module.

### 22.4.4 External Trigger Start ADC SQR Scan Conversion Operation Flow

#### Procedure

**Step1.** Configure the corresponding bits of PAADS-PCADS, and configure the ADC channel to be converted as an analog port.

**Step2.** Set PBADS[1] to 1 to configure the ADC external reference voltage pin as an analog port.

> **Note**
> If the ADC reference voltage does not select an external reference voltage pin, this step can be skipped.

**Step3.** Set ADC_BGR.BGR_EN to 1, to enable the BGR module.

**Step4.** Set ADC_CR0.EN to 1 to enable the ADC module.

**Step5.** Delay for the longer of $T_{ADCSTART}$ and $T_{stable(BGR)}$ (refer to the "Electrical CharacteristicsOperating ConditionsInternal Reference Voltage/ADC Characteristics" section in the "Datasheet" for specific values), and wait for the ADC and BGR module to complete startup.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

**Step6.** Set ADC_CR1.MODE to 1 to select scan conversion mode.

**Step7.** Set ADC_CR0.IE to 1 to enable ADC interrupt.

**Step8.** Enable the ADC interrupt in the NVIC interrupt vector table.

**Step9.** Configure ADC_CR0.REF and select the reference voltage of the ADC.

**Step10.** Configure ADC_CR0.SAM and ADC_CR0.CLKDIV to set the conversion speed of ADC.

**Step11.** Configure ADC_SQR0.CHxMUX, and select SQR scan conversion channels.

**Step12.** Configure ADC_SQR1.CNT, and select the total number of conversions for SQR scan conversions.
The total number of conversions needs to be consistent with the number of conversion channels configured in step11.

**Step13.** Set ADC_ICR to 0x0, clear the ADC interrupt flag.

**Step14.** Configure ADC_EXTTRIGGER0 to select the external trigger condition.

**Step15.** When an external trigger condition triggers the ADC to complete the conversion, the ADC module will generate an interrupt. Users can read ADC_SQRRESULTx-ADC_SQRRESULT0 registers in the ADC interrupt service routine to get the conversion result of the corresponding channel.

**Step16.** To convert other channels, repeat steps11-15.

**Step17.** Set ADC_CR0.EN and ADC_BGR.BGR_EN to 0, turn off ADC module and BGR module.

### 22.4.5 Start ADC JQR Scan Conversion Operation Flow via ADC_JQRSTART.START Bit

#### Procedure

**Step1.** Configure the corresponding bits of PAADS-PCADS, and configure the ADC channel to be converted as an analog port.

**Step2.** Set PBADS[1] to 1 to configure the ADC external reference voltage pin as an analog port.

> **Note**
> If the ADC reference voltage does not select an external reference voltage pin, this step can be skipped.

**Step3.** Set ADC_BGR.BGR_EN to 1, to enable the BGR module.

**Step4.** Set ADC_CR0.EN to 1 to enable the ADC module.

**Step5.** Delay for the longer of $T_{ADCSTART}$ and $T_{stable(BGR)}$ (refer to the "Electrical CharacteristicsOperating ConditionsInternal Reference Voltage/ADC Characteristics" section in the "Datasheet" for specific values), and wait for the ADC and BGR module to complete startup.

**Step6.** Set ADC_CR1.MODE to 1 to select scan conversion mode.

**Step7.** Configure ADC_CR0.REF and select the reference voltage of the ADC.

**Step8.** Configure ADC_CR0.SAM and ADC_CR0.CLKDIV to set the conversion speed of ADC.

**Step9.** Configure ADC_JQR.CHxMUX and select the JQR scan conversion channel.

**Step10.** Configure ADC_JQR.CNT to select the total number of conversions for the JQR scan conversion.
The total number of conversions needs to be consistent with the number of conversion channels configured in step9.

**Step11.** Set ADC_ICR.JQRIC to 0, clear ADC_IFR.JQRIF flag.

**Step12.** Set ADC_JQRSTART.START to 1 to start ADC JQR scan conversion.

**Step13.** Wait for ADC_IFR.JQRIF to become 1, read ADC_JQRRESULTx-ADC_JQRRESULT0 registers to obtain the conversion result of the corresponding channel.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

**Step14.** To convert other channels, repeat steps9-13.

**Step15.** Set ADC_CR0.EN and ADC_BGR.BGR_EN to 0, turn off ADC module and BGR module.

### 22.4.6 External Trigger Start ADC JQR Scan Conversion Operation Flow

#### Procedure

**Step1.** Configure the corresponding bits of PAADS-PCADS, and configure the ADC channel to be converted as an analog port.

**Step2.** Set PBADS[1] to 1 to configure the ADC external reference voltage pin as an analog port.

> **Note**
> If the ADC reference voltage does not select an external reference voltage pin, this step can be skipped.

**Step3.** Set ADC_BGR.BGR_EN to 1, to enable the BGR module.

**Step4.** Set ADC_CR0.EN to 1 to enable the ADC module.

**Step5.** Delay for the longer of $T_{ADCSTART}$ and $T_{stable(BGR)}$ (refer to the "Electrical CharacteristicsOperating ConditionsInternal Reference Voltage/ADC Characteristics" section in the "Datasheet" for specific values), and wait for the ADC and BGR module to complete startup.

**Step6.** Set ADC_CR1.MODE to 1 to select scan conversion mode.

**Step7.** Set ADC_CR0.IE to 1 to enable ADC interrupt.

**Step8.** Enable the ADC interrupt in the NVIC interrupt vector table.

**Step9.** Configure ADC_CR0.REF and select the reference voltage of the ADC.

**Step10.** Configure ADC_CR0.SAM and ADC_CR0.CLKDIV to set the conversion speed of ADC.

**Step11.** Configure ADC_JQR.CHxMUX and select the JQR scan conversion channel.

**Step12.** Configure ADC_JQR.CNT to select the total number of conversions for the JQR scan conversion. The total number of conversions needs to be consistent with the number of conversion channels configured in step11.

**Step13.** Set ADC_ICR to 0x0, clear the ADC interrupt flag.

**Step14.** Configure ADC_EXTTRIGGER1 to select the external trigger condition.

**Step15.** When an external trigger condition triggers the ADC to complete the conversion, the ADC module will generate an interrupt. Users can read the ADC_JQRRESULTx-ADC_JQRRESULT0 registers in the ADC interrupt service routine to obtain the conversion results of the corresponding channels.

**Step16.** To convert other channels, repeat steps11-15.

**Step17.** Set ADC_CR0.EN and ADC_BGR.BGR_EN to 0, turn off ADC module and BGR module.

### 22.4.7 Start the ADC Continuous Conversion Accumulation Operation Process through the ADC_SQRSTART.START Bit

#### Procedure

**Step1.** Configure the corresponding bits of PAADS-PCADS, and configure the ADC channel to be converted as an analog port.

**Step2.** Set PBADS[1] to 1 to configure the ADC external reference voltage pin as an analog port.

> **Note**
> If the ADC reference voltage does not select an external reference voltage pin, this step can be skipped.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

**Step3.** Set ADC_BGR.BGR_EN to 1, to enable the BGR module.

**Step4.** Set ADC_CR0.EN to 1 to enable the ADC module.

**Step5.** Delay for the longer of $T_{ADCSTART}$ and $T_{stable(BGR)}$ (refer to the "Electrical CharacteristicsOperating ConditionsInternal Reference Voltage/ADC Characteristics" section in the "Datasheet" for specific values), and wait for the ADC and BGR module to complete startup.

**Step6.** Set ADC_CR1.MODE to 1 to select scan conversion mode.

**Step7.** Set ADC_CR1.RACCEN to 1 to enable the automatic accumulation function of ADC conversion.

**Step8.** Configure ADC_CR0.REF and select the reference voltage of the ADC.

**Step9.** Configure ADC_CR0.SAM and ADC_CR0.CLKDIV to set the conversion speed of ADC.

**Step10.** Configure ADC_SQR0.CHxMUX, and select SQR scan conversion channels.

**Step11.** Configure ADC_SQR1.CNT, and select the total number of conversions for SQR scan conversions.
The total number of conversions needs to be consistent with the number of conversion channels configured in step10.

**Step12.** Set ADC_ICR.SQRIC to 0, clear ADC_IFR.SQRIF flag.

**Step13.** Set ADC_CR1.RACCCLR to 0 and clear the ADC_RESULTACC register.

**Step14.** Set ADC_SQRSTART.START to 1 to start ADC SQR scan conversion.

**Step15.** Wait for ADC_IFR.SQRIF to become 1, read ADC_RESULTACC registers to obtain accumulated value of the conversion result.

**Step16.** To convert other channels, repeat steps10-15.

**Step17.** Set ADC_CR0.EN and ADC_BGR.BGR_EN to 0, turn off ADC module and BGR module.

### 22.4.8 Start ADC Continuous Conversion Accumulation Operation Process through External Trigger

#### Procedure

**Step1.** Configure the corresponding bits of PAADS-PCADS, and configure the ADC channel to be converted as an analog port.

**Step2.** Set PBADS[1] to 1 to configure the ADC external reference voltage pin as an analog port.

> **Note**
> If the ADC reference voltage does not select an external reference voltage pin, this step can be skipped.

**Step3.** Set ADC_BGR.BGR_EN to 1, to enable the BGR module.

**Step4.** Set ADC_CR0.EN to 1 to enable the ADC module.

**Step5.** Delay for the longer of $T_{ADCSTART}$ and $T_{stable(BGR)}$ (refer to the "Electrical CharacteristicsOperating ConditionsInternal Reference Voltage/ADC Characteristics" section in the "Datasheet" for specific values), and wait for the ADC and BGR module to complete startup.

**Step6.** Set ADC_CR1.MODE to 1 to select scan conversion mode.

**Step7.** Set ADC_CR0.IE to 1 to enable ADC interrupt.

**Step8.** Enable the ADC interrupt in the NVIC interrupt vector table.

**Step9.** Set ADC_CR1.RACCEN to 1 to enable the automatic accumulation function of ADC conversion.

**Step10.** Configure ADC_CR0.REF and select the reference voltage of the ADC.

**Step11.** Configure ADC_CR0.SAM and ADC_CR0.CLKDIV to set the conversion speed of ADC.

**Step12.** Configure ADC_SQR0.CHxMUX, and select SQR scan conversion channels.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

**Step13.** Configure ADC_SQR1.CNT, and select the total number of conversions for SQR scan conversions.
The total number of conversions needs to be consistent with the number of conversion channels configured in step12.

**Step14.** Set ADC_ICR to 0x0, clear the ADC interrupt flag.

**Step15.** Set ADC_CR1.RACCCLR to 0 and clear the ADC_RESULTACC register.

**Step16.** Configure ADC_EXTTRIGGER0 to select the external trigger condition.

**Step17.** When an external trigger condition triggers the ADC to complete the conversion, the ADC module will generate an interrupt. The user can read the ADC_RESULTACC register in the ADC interrupt service routine to obtain the accumulated value of the conversion result.

**Step18.** To convert other channels, repeat steps12-17.

**Step19.** Set ADC_CR0.EN and ADC_BGR.BGR_EN to 0, turn off ADC module and BGR module.

### 22.4.9 Operation Process of Measuring Ambient Temperature through ADC

#### Procedure

**Step1.** Set ADC_BGR to 3, enable BGR module and temperature sensor module.

**Step2.** Set ADC_CR0.EN to 1 to enable the ADC module.

**Step3.** Delay for the longer of $T_{ADCSTART}$ and $T_{stable(BGR)}$ (refer to the "Electrical CharacteristicsOperating ConditionsInternal Reference Voltage/ADC Characteristics" section in the "Datasheet" for specific values), and wait for the ADC and BGR module to complete startup.

**Step4.** Set ADC_CR1.MODE to 0 to select single conversion mode.

**Step5.** Configure ADC_CR0.REF, select the reference voltage of ADC as internal 1.5V or internal 2.5V.

**Step6.** Configure ADC_CR0.SAM and ADC_CR0.CLKDIV to set the conversion speed of ADC.

**Step7.** Set ADC_CR0.SGLMUX to 0x10, select the channel to be converted as the output of the temperature sensor.

**Step8.** Set ADC_CR0.BUF to 1 to enable the input signal amplifier.

**Step9.** Set ADC_ICR.SGLIC to 0, and clear ADC_IFR.SGLIF flag.

**Step10.** Set ADC_SGLSTART.START to 1 to start ADC single conversion.

**Step11.** Wait for ADC_IFR.SGLIF to become 1, read ADC_RESULT registers to obtain accumulated value of the conversion result.

**Step12.** Set ADC_CR0.EN and ADC_BGR to 0, turn off the ADC module, BGR module, and temperature sensor module.

**Step13.** Read the calibration value of the temperature sensor and calculate the current ambient temperature according to the formula.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

## 22.5 Register Description

### 22.5.1 Register List

Base address: 0x40002400

Table 22-4 Analog-to-Digital Converter (ADC) Register Offset

| Offset | Register | Description | Width |
|---|---|---|---|
| 0x000 | ADC_BGR | BGR Control Register | 32 |
| 0x004 | ADC_CR0 | ADC Basic Configuration Register 0 | 32 |
| 0x008 | ADC_CR1 | ADC Basic Configuration Register 1 | 32 |
| 0x040 | ADC_SQR0 | ADC SQR Scan Conversion Channel Configuration Register 0 | 32 |
| 0x048 | ADC_SQR1 | ADC SQR Scan Conversion Channel Configuration Register 1 | 32 |
| 0x04C | ADC_JQR | ADC JQR Scan Conversion Channel Configuration Register | 32 |
| 0x050 | ADC_SQRRESULT0 | ADC SQR Scan Conversion Channel Configuration Register 0 | 32 |
| 0x054 | ADC_SQRRESULT1 | ADC SQR Scan Conversion Channel Configuration Register 1 | 32 |
| 0x058 | ADC_SQRRESULT2 | ADC SQR Scan Conversion Channel Configuration Register 2 | 32 |
| 0x05C | ADC_SQRRESULT3 | ADC SQR Scan Conversion Channel Configuration Register 3 | 32 |
| 0x060 | ADC_SQRRESULT4 | ADC SQR Scan Conversion Channel Configuration Register 4 | 32 |
| 0x064 | ADC_SQRRESULT5 | ADC SQR Scan Conversion Channel Configuration Register 5 | 32 |
| 0x090 | ADC_JQRRESULT0 | ADC JQR Scan Conversion Channel Configuration Register 0 | 32 |
| 0x094 | ADC_JQRRESULT1 | ADC JQR Scan Conversion Channel Configuration Register 1 | 32 |
| 0x098 | ADC_JQRRESULT2 | ADC JQR Scan Conversion Channel Configuration Register 2 | 32 |
| 0x09C | ADC_JQRRESULT3 | ADC JQR Scan Conversion Channel Configuration Register 3 | 32 |
| 0x0A0 | ADC_RESULT | ADC Conversion Results Register | 32 |
| 0x0A4 | ADC_RESULTACC | Accumulated Value of ADC Conversion Result Register | 32 |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Offset | Register | Description | Width |
|---|---|---|---|
| 0x0A8 | ADC_HT | ADC Comparison High Threshold Register | 32 |
| 0x0AC | ADC_LT | ADC Comparison Low Threshold Register | 32 |
| 0x0B0 | ADC_IFR | ADC Interrupt Flag Register | 32 |
| 0x0B4 | ADC_ICR | ADC Interrupt Clear Register | 32 |
| 0x0B8 | ADC_EXTTRIGGER0 | ADC External Interrupt Trigger Source Configuration Register for Single or SQRScan Conversion | 32 |
| 0x0BC | ADC_EXTTRIGGER1 | ADC JQR Scan Conversion External Interrupt Trigger Source Configuration Register | 32 |
| 0x0C0 | ADC_SGLSTART | ADC Single Conversion Start Control Register | 32 |
| 0x0C4 | ADC_SQRSTART | ADC SQR Scan Conversion Start Control Register | 32 |
| 0x0C8 | ADC_JQRSTART | ADC JQR Scan Conversion Start Control Register | 32 |
| 0x0CC | ADC_ALLSTART | ADC Continuous Conversion Start/Stop Control Register | 32 |

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 22.5.2 BGR Control Register (ADC_BGR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x000 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | TS_EN | BGR_EN |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | RW | RW |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:2* | *Reserved* | - | - |
| 1 | TS_EN | Built-in temperature sensor enable control<br>● 0b0: Disable internal temperature sensor<br>● 0b1: Enable internal temperature sensor, and BGR must be enabled<br>**Note**<br>The temperature sensor outputs a stable signal only after the $T_{stable\ (TS)}$ time is enabled (for specific values, see the "Electrical Specifications Operating Conditions Temperature Sensor Characteristics" section in the "Datasheet"). | RW |
| 0 | BGR_EN | BGR enable control<br>● 0b0: Disable BGR<br>● 0b1: Enable BGR<br>**Note**<br>● This register can only be operated when SYSCTRL module register PERI_CLKEN0.ADC is 1.<br>● The BGR outputs a stable high-precision reference voltage only after the $T_{stable\ (BGR)}$ time is enabled (for specific values, see the "Electrical Specifications Operating Conditions Internal Reference Voltage" section in the "Datasheet"). After the BGR is stable, it can be used by other modules, so the steps of waiting for the BGR to be stable should be added in the user operation.<br>● When using ADC, BGR must be enabled.<br>● When using VC, it is necessary to decide whether to enable BGR according to the configuration of the VC register.<br>● It is recommended to disable the BGR before the chip enters low-power mode. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 22.5.3 ADC Basic Configuration Register 0 (ADC_CR0)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x004 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x000427F0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | SAM | | | Reserved | IE | Reserved | | | | BUF | REF | SGLMUX | | | | CLKDIV | | Reserved | EN | |
| Access | | | | | | | | | | | | RW | | | | RW | | | | | RW | RW | RW | | | | RW | | | RW | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:21* | *Reserved* | - | - |
| 20:17 | SAM | ADC sampling period selection<br>● 0b0000: 2 ADCCLK cycles<br>● 0b0001: 3 ADCCLK cycles<br>● 0b0010: 4 ADCCLK cycles<br>● 0b0011: 6 ADCCLK cycles<br>● 0b0100: 8 ADCCLK cycles<br>● 0b0101: 10 ADCCLK cycles<br>● 0b0110: 12 ADCCLK cycles<br>● 0b0111: 16 ADCCLK cycles<br>● 0b1000: 20 ADCCLK cycles<br>● 0b1001: 28 ADCCLK cycles<br>● 0b1010: 36 ADCCLK cycles<br>● 0b1011: 48 ADCCLK cycles<br>● 0b1100: 64 ADCCLK cycles<br>● 0b1101: 72 ADCCLK cycles<br>● 0b1110: 96 ADCCLK cycles<br>● 0b1111: 127 ADCCLK cycles<br>**Note**<br>Can only be modified when ADC is idle. | RW |
| *16* | *Reserved* | - | - |
| 15 | IE | ADC interrupt control<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| *14:12* | *Reserved* | - | - |
| 11 | BUF | Built-in follower enable control<br>● 0b0: Disable the follower, and the external input signal is directly connected to the ADC<br>● 0b1: Turn on the follower, and the external input signal is connected to the ADC after passing through the follower to measure high-impedance signals | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 11 | BUF | **Note**<br>When this function is enabled, the maximum rate is 200ksps. The following conditions must enable this function:<br>● The output impedance of the signal to be converted is high<br>● The signals to be converted are: AVCC/3, internal temperature sensor output voltage, VREF1P2.<br>This bit is effective only when CR0.EN is set to 1. | RW |
| 10:9 | REF | ADC reference voltage selection<br>● 0b00: Internal 1.5V<br>● 0b01: Internal 2.5V (power supply voltage must exceed 2.8V)<br>● 0b10: External reference voltage EXVREF (PB01)<br>● 0b11: AVCC voltage | RW |
| 8:4 | SGLMUX | Single conversion mode conversion channel selection<br>● 0b00000: AIN0 (PA00)<br>● 0b00001: AIN1 (PA01)<br>● 0b00010: AIN2 (PA02)<br>● 0b00011: AIN3 (PA03)<br>● 0b00100: AIN4 (PA04)<br>● 0b00101: AIN5 (PA05)<br>● 0b00110: AIN6 (PA06)<br>● 0b00111: AIN7 (PA07)<br>● 0b01000: AIN8 (PB00)<br>● 0b01001: AIN9 (PB01)<br>● 0b01010: AIN10 (PC00)<br>● 0b01011: AIN11 (PC01)<br>● 0b01100: AIN12 (PC02)<br>● 0b01101: AIN13 (PC03)<br>● 0b01110: AIN14 (PC04)<br>● 0b01111: AIN15 (PC05)<br>● 0b10000: Internal temperature sensor output voltage<br>● 0b10001: AVCC/3<br>● 0b10010: VREF1P2<br>● Others: Setting prohibited<br>**Note**<br>● When set to 0b10000 to 0b10010, the ADCCLK frequency shall not exceed 2 MHz, and the sampling period shall be at least 36 ADCCLK cycles.<br>● Can only be modified when ADC is idle. | RW |
| 3:2 | CLKDIV | ADC clock selection<br>● 0b00: PCLK clock<br>● 0b01: PCLK clock divided by 2<br>● 0b10: PCLK clock divided by 4<br>● 0b11: PCLK clock divided by 8 | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 3:2 | CLKDIV | **Notice**<br>● When the system clock division selected by HCLK/PCLK is greater than 1, the CLKDIV division must be set to greater than 1.<br>● Can only be modified when ADC is idle. | RW |
| *1* | *Reserved* | - | - |
| 0 | EN | ADC enable control<br>Write 1 to enable ADC, and write 0 to turn off the ADC<br>Read:<br>● 0b0: ADC is OFF<br>● 0b1: ADC is in open state<br>**Note**<br>● It is recommended to disable the ADC when it is idle to avoid unexpected behavior.<br>● After this bit is set to 0, the ADC shutdown process will take some time. If this bit is set back to 1 during this period, the ADC will continue operating, which may cause unexpected behavior. It is recommended to read this bit before enabling the ADC and set it to 1 only after confirming it is 0. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 22.5.4 ADC Basic Configuration Register 1 (ADC_CR1)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x008 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00008000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | RACCCLR | REGCMP | HTCMP | LTCMP | RACCEN | MODE | DMAJQR | DMASQR | THCH | | | | | Reserved | | |
| Access | | | | | | | | | | | | | | | | | WO | RW | RW | RW | RW | RW | RW | RW | RW | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:16* | *Reserved* | - | - |
| 15 | RACCCLR | ADC conversion result accumulation register is cleared<br>Write 0 to clear the ADC_RESULTACC register, write 1 has no effect; read always returns 1. | WO |
| 14 | REGCMP | ADC interval comparison control<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 13 | HTCMP | ADC high threshold compare control<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 12 | LTCMP | ADC low threshold compare control<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 11 | RACCEN | ADC conversion result automatic accumulation control<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 10 | MODE | ADC conversion mode selection<br>● 0b0: Single conversion mode<br>● 0b1: Scan conversion mode | RW |
| 9 | DMAJQR | JQR scan trigger DMA read control<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 8 | DMASQR | SQR scan trigger DMA read control<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 7:3 | THCH | Threshold comparison channel selection<br>For more information, see ADC_CR0.SGLMUX | RW |
| *2:0* | *Reserved* | - | - |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 22.5.5 ADC SQR Scan Conversion Channel Configuration Register 0 (ADC_SQR0)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x040 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | CH5MUX | | | | | | CH4MUX | | | | | CH3MUX | | | | | CH2MUX | | | | | CH1MUX | | | | | CH0MUX | | | |
| Access | | RW | | | | | | RW | | | | | RW | | | | | RW | | | | | RW | | | | | RW | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:30* | *Reserved* | - | - |
| 29:25 | CH5MUX | SQR scan conversion channel 5 selection<br>For more information, see ADC_CR0.SGLMUX | RW |
| 24:20 | CH4MUX | SQR scan conversion channel 4 selection<br>For more information, see ADC_CR0.SGLMUX | RW |
| 19:15 | CH3MUX | SQR scan conversion channel 3 selection<br>For more information, see ADC_CR0.SGLMUX | RW |
| 14:10 | CH2MUX | SQR scan conversion channel 2 selection<br>For more information, see ADC_CR0.SGLMUX | RW |
| 9:5 | CH1MUX | SQR scan conversion channel 1 selection<br>For more information, see ADC_CR0.SGLMUX | RW |
| 4:0 | CH0MUX | SQR scan conversion channel 0 selection<br>For more information, see ADC_CR0.SGLMUX | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 22.5.6 ADC SQR Scan Conversion Channel Configuration Register 1 (ADC_SQR1)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x048 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | CNT | |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | RW | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:3* | *Reserved* | - | - |
| 2:0 | CNT | Number of SQR scan conversions<br>The actual number of conversions is CNT + 1, with a maximum valid setting of 0x5 (i.e., 6 conversions). When a value greater than 0x5 is set, 6 conversions are actually performed.<br>Channels selected by CHxMUX-CH0MUX (x=CNT) are scanned by SQR in sequence..<br>**Note**<br>Can only be modified when ADC is idle. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 22.5.7 ADC JQR Scan Conversion Channel Configuration Register (ADC_JQR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x04C | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | CNT | | CH3MUX | | | | | CH2MUX | | | | | CH1MUX | | | | | CH0MUX | | | |
| Access | | | | | | | | | | | RW | | RW | | | | | RW | | | | | RW | | | | | RW | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:22* | *Reserved* | - | - |
| 21:20 | CNT | Number of JQR scan conversions<br>The actual number of conversions is CNT + 1.<br>Channels selected by CHxMUX-CH0MUX (x=CNT) are scanned by JQR in sequence.<br>**Note**<br>Can only be modified when ADC is idle. | RW |
| 19:15 | CH3MUX | JQR scan conversion channel 3 selection<br>For more information, see ADC_CR0.SGLMUX | RW |
| 14:10 | CH2MUX | JQR scan conversion channel 2 selection<br>For more information, see ADC_CR0.SGLMUX | RW |
| 9:5 | CH1MUX | JQR scan conversion channel 1 selection<br>For more information, see ADC_CR0.SGLMUX | RW |
| 4:0 | CH0MUX | JQR scan conversion channel 0 selection<br>For more information, see ADC_CR0.SGLMUX | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 22.5.8 ADC SQR Scan Conversion Channel x Result Register (ADC_SQRRESULTx) (x=0-5)

Offset: 0x050 (SQRRESULT0), 0x054 (SQRRESULT1), 0x058 (SQRRESULT2), 0x05C (SQRRESULT3), 0x060 (SQRRESULT4), 0x064 (SQRRESULT5)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| - | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | RESULT | | | | | | | | | | | |
| Access | | | | | | | | | | | | | | | | | | | | | RO | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:12* | *Reserved* | - | - |
| 11:0 | RESULT | ADC SQR scan conversion channelx conversion result | RO |

### 22.5.9 ADC JQR Scan Conversion Channel x Result Register (ADC_JQRRESULTx) (x=0-3)

Offset: 0x090 (JQRRESULT0), 0x094 (JQRRESULT1), 0x098 (JQRRESULT2), 0x09C (JQRRESULT3)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| - | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | RESULT | | | | | | | | | | | |
| Access | | | | | | | | | | | | | | | | | | | | | RO | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:12* | *Reserved* | - | - |
| 11:0 | RESULT | ADC JQR scan conversion channelx conversion result | RO |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 22.5.10 ADC Conversion Result Register (ADC_RESULT)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x0A0 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | RESULT | | | | | | | | | | | |
| Access | | | | | | | | | | | | | | | | | | | | | RO | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:12* | *Reserved* | - | - |
| 11:0 | RESULT | ADC conversion results | RO |

### 22.5.11 ADC Conversion Result Accumulation Value Register (ADC_RESULTACC)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x0A4 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | RESULTACC | | | | | | | | | | | | | | | | | | | |
| Access | | | | | | | | | | | | | RO | | | | | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:20* | *Reserved* | - | - |
| 19:0 | RESULTACC | ADC conversion accumulated value | RO |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 22.5.12 ADC Compare High Threshold Register (ADC_HT)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x0A8 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000FFF | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | HT | | | | | | | | | | | |
| Access | | | | | | | | | | | | | | | | | | | | | RW | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:12* | *Reserved* | - | - |
| 11:0 | HT | ADC conversion result comparison high threshold | RW |

### 22.5.13 ADC Compare Low Threshold Register (ADC_LT)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x0AC | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | LT | | | | | | | | | | | |
| Access | | | | | | | | | | | | | | | | | | | | | RW | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:12* | *Reserved* | - | - |
| 11:0 | LT | ADC conversion result comparison low threshold | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 22.5.14 ADC Interrupt Flag Register (ADC_IFR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x0B0 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | JQRIF | SQRIF | REGIF | HTIF | LTIF | SGLIF |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | | RO | RO | RO | RO | RO | RO |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:6* | *Reserved* | - | - |
| 5 | JQRIF | ADC JQR scan conversion complete flag<br>● 0b0: Conversion not complete<br>● 0b1: Conversion complete | RO |
| 4 | SQRIF | ADC SQR scan conversion complete flag<br>● 0b0: Conversion not complete<br>● 0b1: Conversion complete | RO |
| 3 | REGIF | ADC conversion result comparison interval flag<br>● 0b0: ADC conversion result is outside the [ADC_LT, ADC_HT) range<br>● 0b1: ADC conversion result is within the range of [ADC_LT, ADC_HT) | RO |
| 2 | HTIF | ADC conversion result comparison upper threshold flag<br>● 0b0: ADC conversion result is outside [ADC_HT, 4095] range<br>● 0b1: ADC conversion result is within the range [ADC_HT, 4095] | RO |
| 1 | LTIF | ADC conversion result comparison lower threshold flag<br>● 0b0: ADC conversion result is outside the interval [0, ADC_LT)<br>● 0b1: ADC conversion result is within the range [0, ADC_LT) | RO |
| 0 | SGLIF | ADC single conversion complete flag<br>● 0b0: Conversion not complete<br>● 0b1: Conversion complete | RO |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 22.5.15 ADC Interrupt Clear Register (ADC_ICR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x0B4 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x0000003F | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | JQRIC | SQRIC | REGIC | HTIC | LTIC | SGLIC |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | | WO | WO | WO | WO | WO | WO |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:6* | *Reserved* | - | - |
| 5 | JQRIC | ADC JQR scan conversion complete flag clear<br>Write 0 to clear flag, write 1 has no effect; read always returns 1. | WO |
| 4 | SQRIC | ADC SQR scan conversion complete flag clear<br>Write 0 to clear flag, write 1 has no effect; read always returns 1. | WO |
| 3 | REGIC | ADC conversion result comparison interval flag clear<br>Write 0 to clear flag, write 1 has no effect; read always returns 1. | WO |
| 2 | HTIC | ADC conversion result comparison upper threshold flag clear<br>Write 0 to clear flag, write 1 has no effect; read always returns 1. | WO |
| 1 | LTIC | ADC conversion result comparison lower threshold flag clear<br>Write 0 to clear flag, write 1 has no effect; read always returns 1. | WO |
| 0 | SGLIC | ADC single conversion complete flag clear<br>Write 0 to clear flag, write 1 has no effect; read always returns 1. | WO |

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 22.5.16 ADC Single or SQR Scan Conversion External Interrupt Trigger Source Configuration Register (ADC_EXTTRIGGER0)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x0B8 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | EXTTRIGEN | | | |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | | | | RW | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:4* | *Reserved* | - | - |
| 3:0 | EXTTRIGEN | ADC trigger signal selection<br>● 0b0000: No peripheral trigger<br>● 0b0001: TIM0_TRADC rising edge<br>● 0b0010: TIM1_TRADC rising edge<br>● 0b0011: TIM2_TRADC rising edge<br>● 0b0100: TIM3_TRADC rising edge<br>● 0b0101: LPTIM0_OV rising edge<br>● 0b0110: USART0 interrupt request<br>● 0b0111: LPUART0 interrupt request<br>● 0b1000: VC0 interrupt request<br>● 0b1001: RTC interrupt request<br>● 0b1010: SPI0 interrupt request<br>● 0b1011: PA03 interrupt request<br>● 0b1100: PB03 interrupt request<br>● 0b1101: PC03 interrupt request<br>● 0b1110: PD03 interrupt request<br>● 0b1111: PA11 interrupt request | RW |

**Note**

- TIM0/1/2/3 trigger ADC automatic conversion without the need to enable the corresponding interrupt. Simply configure the corresponding trigger enable registers of the TIM modules. For detailed operation, see the "Function Description Timer Trigger ADC" chapter in the "General Timers TIM0/1/2/3".
- Except for TIM0/1/2/3, LPTIM0, other peripherals trigger the ADC using the rising edge of their respective interrupt flags. If repeated triggering is required, the interrupt flag needs to be cleared. If the interrupt service routine is not required, the NVIC interrupt enable shall not be enabled.
- To ensure that LPTIM0_OV can properly trigger the ADC, the LPTIM clock frequency must be lower than PCLK.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 22.5.17 ADC JQR Scan Conversion External Interrupt Trigger Source Configuration Register (ADC_EXTTRIGGER1)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x0BC | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | EXTTRIGEN | | | |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | | | | RW | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:4* | *Reserved* | - | - |
| 3:0 | EXTTRIGEN | ADC trigger signal selection<br>● 0b0000: No peripheral trigger<br>● 0b0001: TIM0_TRGO rising edge<br>● 0b0010: TIM1_TRGO rising edge<br>● 0b0011: TIM2_TRGO rising edge<br>● 0b0100: TIM3_CMP3A rising edge<br>● 0b0101: LPTIM1_OV rising edge<br>● 0b0110: USART1 interrupt request<br>● 0b0111: LPUART1 interrupt request<br>● 0b1000: VC1 interrupt request<br>● 0b1001: DMA interrupt request<br>● 0b1010: SPI1 interrupt request<br>● 0b1011: PA07 interrupt request<br>● 0b1100: PB07 interrupt request<br>● 0b1101: PC07 interrupt request<br>● 0b1110: PD07 interrupt request<br>● 0b1111: PA15 interrupt request | RW |

**Note**

- TIM0/1/2/3 trigger ADC automatic conversion without the need to enable the corresponding interrupt. Simply configure the corresponding trigger enable registers of the TIM modules. For detailed operation, see the "Function Description Timer Trigger ADC" section in the "General Timers TIM0/1/2/3".
- Except for TIM0/1/2/3, LPTIM1, other peripherals trigger the ADC using the rising edge of their respective interrupt flags. If repeated triggering is required, the interrupt flag needs to be cleared. If the interrupt service routine is not required, the NVIC interrupt enable shall not be enabled.
- To ensure that LPTIM1_OV can properly trigger the ADC, the LPTIM clock frequency must be lower than PCLK.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 22.5.18 ADC Single Conversion Start Control Register (ADC_SGLSTART)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x0C0 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | START |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | RW |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:1* | *Reserved* | - | - |
| 0 | START | ADC Single conversion start control<br>Write 1 to start a single ADC conversion; writing 0 is prohibited. The hardware automatically clears this bit to 0 after the single conversion completes.<br>Read:<br>● 0b0: ADC is in idle state<br>● 0b1: ADC single conversion in progress<br>**Note**<br>● Write 1 is invalid when START is 1.<br>● This bit is valid only when CR1.MODE is set to 0. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 22.5.19 ADC SQR Scan Conversion Start Control Register (ADC_SQRSTART)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x0C4 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | START | |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | RW | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:1* | *Reserved* | - | - |
| 0 | START | ADC SQR scan conversion start control<br>Write 1 to start ADC SQR scan conversion; write 0 is prohibited. The hardware automatically clears this bit after the SQR scan completes.<br>Read:<br>● 0b0: ADC is in idle state<br>● 0b1: SQR conversion in progress<br>**Note**<br>● Write 1 is invalid when START is 1.<br>● Writing 1 is effective only when the ADC is in idle state.<br>● This bit is valid only when CR1.MODE is set to 1. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 22.5.20 ADC JQR Scan Conversion Start Control Register (ADC_JQRSTART)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x0C8 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | START | |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | RW | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:1* | *Reserved* | - | - |
| 0 | START | ADC JQR scan conversion start control<br>Write 1 to start ADC JQR scan conversion; write 0 is prohibited. The hardware automatically clears this bit after the JQR scan completes.<br>Read:<br>● 0b0: ADC is in idle state<br>● 0b1: JQR conversion in progress<br>**Note**<br>● Write 1 is invalid when START is 1.<br>● Writing 1 is effective only when the ADC is in idle state.<br>● This bit is valid only when CR1.MODE is set to 1. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 22.5.21 ADC Continuous Conversion Start/Stop Control Register (ADC_ALLSTART)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x0CC | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | START |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | RW |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:1* | *Reserved* | - | - |
| 0 | START | ADC continuous start of single conversion control<br>Write 1 to start continuous ADC conversion; write 0 to stop ADC conversion.<br>Read:<br>● 0b0: ADC is in idle state<br>● 0b1: ADC continuous conversion in progress<br>**Note**<br>This bit is valid only when CR1.MODE is set to 0. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

# 23 LCD

## 23.1 Introduction

The LCD controller is a digital controller/driver designed for monochrome passive liquid crystal display (LCD), featuring up to 8 common terminals (COM) to drive 160 (4x40), 228 (6x38) or 288 (8x36) LCD pixel elements. The exact number of terminals depends on the device pinout as stated in the Datasheet.

LCDs are made up of segments (either pixels or complete symbols) that can be either on or off. Each segment contains a layer of liquid crystal molecules aligned between two electrodes. When a voltage higher than the threshold voltage is applied to the liquid crystal, the corresponding segment becomes visible. The segment voltage must be AC to avoid electrophoretic effects in the liquid crystal (which will affect the display). Afterwards, waveforms must be generated across the segment to avoid DC.

### Glossary

- Liquid Crystal Display(LCD): A passive display panel with terminals leading directly to the segment.
- Common (COM): Electrical connection terminals connected to multiple sections.
- Bias (BIAS): The voltage level used when driving the LCD, defined as 1/(the number of voltage levels that drive the LCD display-1).
- Segment (SEG): The smallest visual unit (the smallest constituent element, line or point on an LCD display).
- Duty Cycle (DUTY): A number defined as 1/(number of common terminals on the LCD display).
- Frame: One period of the waveform written to the segment.
- Frame Rate: Frames per second, ie the number of times an LCD segment is excited per second.

## 23.2 Features

- Highly flexible frame rate control
- Support static, 1/2, 1/3, 1/4, 1/6 and 1/8 duty cycle
- Support 1/2, 1/3, 1/4 offset
- LCD data RAM with up to 16 registers
- The contrast of LCD can be configured through software
- Three drive waveform generation methods
  - Internal resistor divider, external resistor divider, external capacitor divider method
  - The power consumption of the internal resistor divider method can be configured through software to match the capacitive charge required by the LCD panel
- Support low power mode: LCD controller can display in Active, Sleep, DeepSleep mode
- Configurable frame interrupt
- Support LCD blinking function and configurable multiple blinking frequencies
- The unused LCD segments and common pins can be configured as digital or analog functions
- Support dead zone insertion


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

## 23.3 Functional Description

### 23.3.1 Function Block Diagram

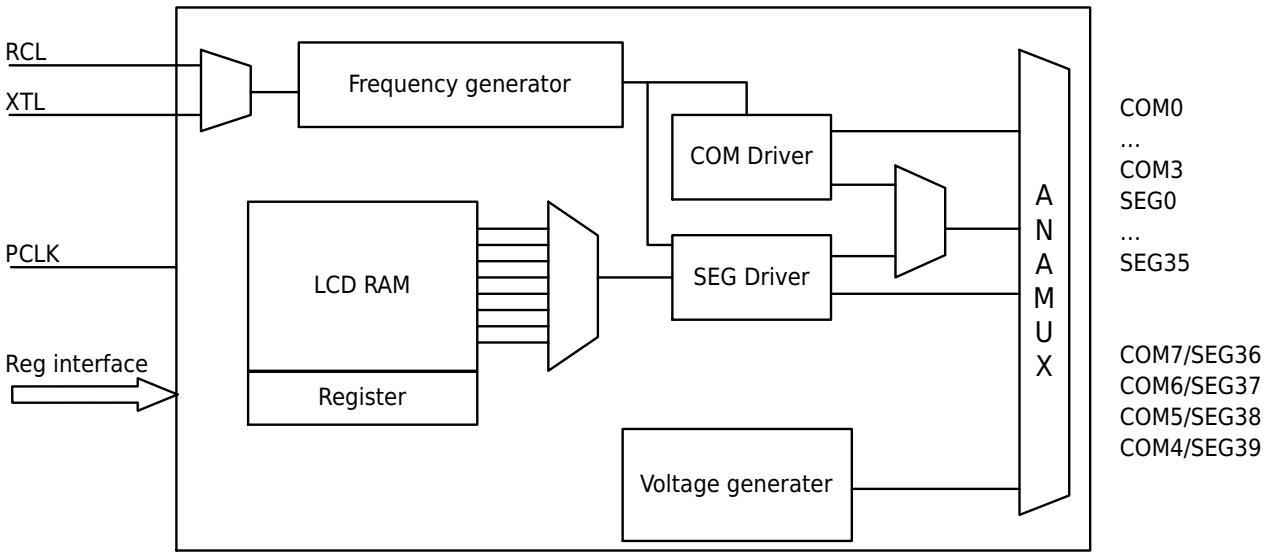



**Figure 23-1 LCD Function Block Diagram**

- LCD RAM and registers serve as the data and control registers for the LCD, containing SEG data and controlling LCD operations. They are configurable via the APB bus and can generate CPU interrupts.
- Frequency generator is an internal LCD frequency generation module that generates clocks to drive COM and SEG outputs. It selects source clocks from either the RCL or XTL clock sources.
- Voltage generator is an analog voltage generation module that drives voltages on COM and SEG lines. It operates through the analog matrix ANAMUX to enable LCD display.

### 23.3.2 Frame Rate

The LCD controller utilizes the clock signal output from the integrated clock divider to generate timing sequences for SEG and COM lines. The LCD clock frequency is configured via the LCD_CR1.PSC and LCD_CR1.DIV bits, calculated as follows:

$f_{LCD} = f_{in\_clk}/(2^{(PSC + 1)} \times (DIV + 16))$

$f_{LCD}$ is the clock reference for the LCD controller, equivalent to the phase clock. $f_{in\_clk}$ is the input clock to the clock generator, selectable between RCL and XTL via LCD_CR1.CLKSRC.

The LCD clock frequency corresponds to the phase frequency. A TYPE B frame is either an odd or even frame, both containing as many phases as there are active COM. The duty cycle is defined as 1/(the number of COM required for the SLCD screen display), $f_{frame}$ frequency is calculated as follows:

$f_{frame} = f_{LCD} \times$ Duty

### 23.3.3 LCD Driving Waveform

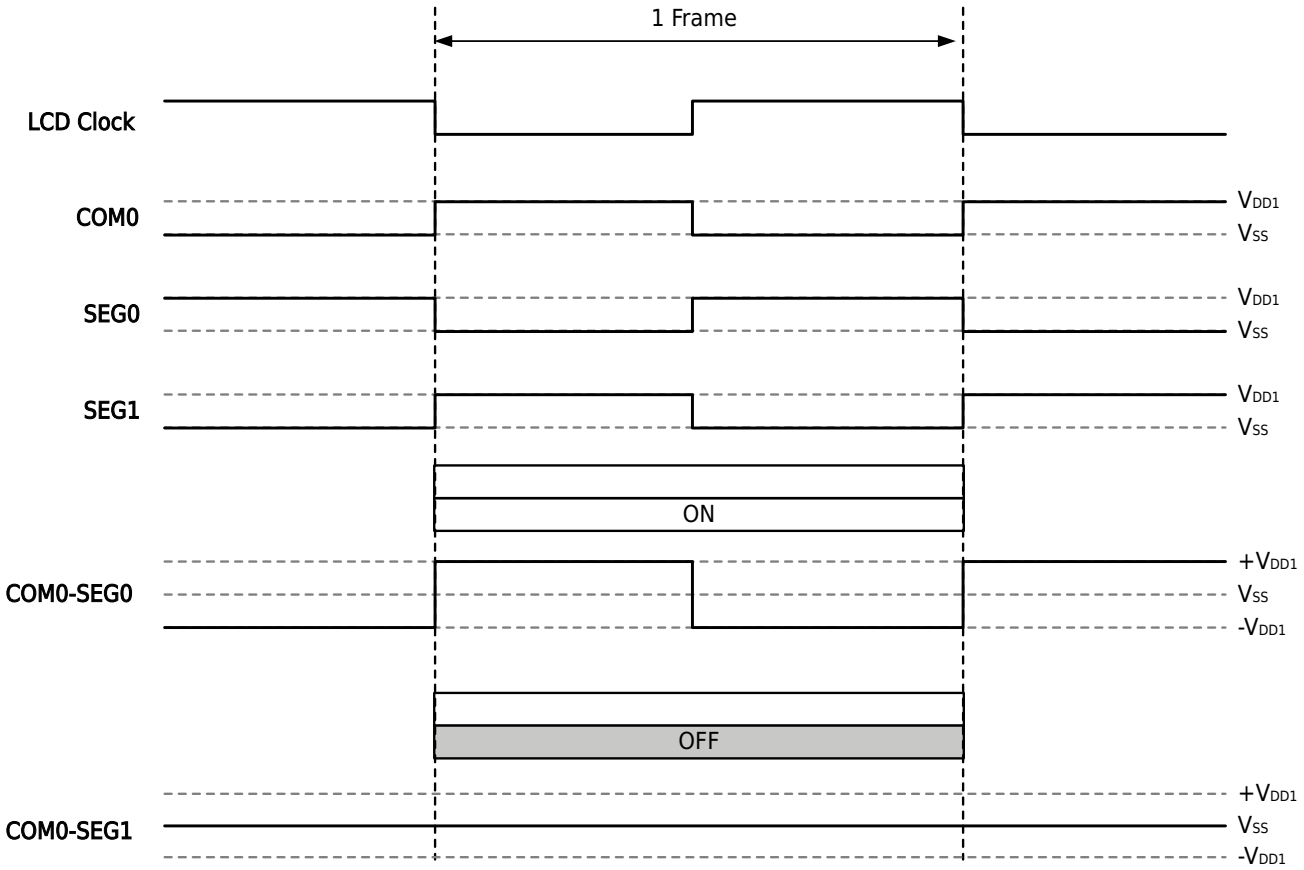
LCD supports 6 driving waveforms with duty cycle (Duty): Static, 1/2, 1/3, 1/4, 1/6 and 1/8, which are set by LCD_CR0.DUTY. LCD supports 3 types of bias drive waveforms: 1/2, 1/3 and 1/4, configured via the LCD_CR0.BIAS register.

The drive waveforms in each mode are as follows: when the number of COM is greater than or equal to 4, a subset of COM is partially omitted. COM are sequentially activated. When a COM is active, the SEG outputs an appropriate voltage level, and jointly applied to the LCD panel with the COM level. If


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

this differential exceeds the LCD turn-on voltage, the corresponding pixel of LCD panel is illuminated (ON state); otherwise, the corresponding pixel of LCD panel remains unilluminated (OFF state)

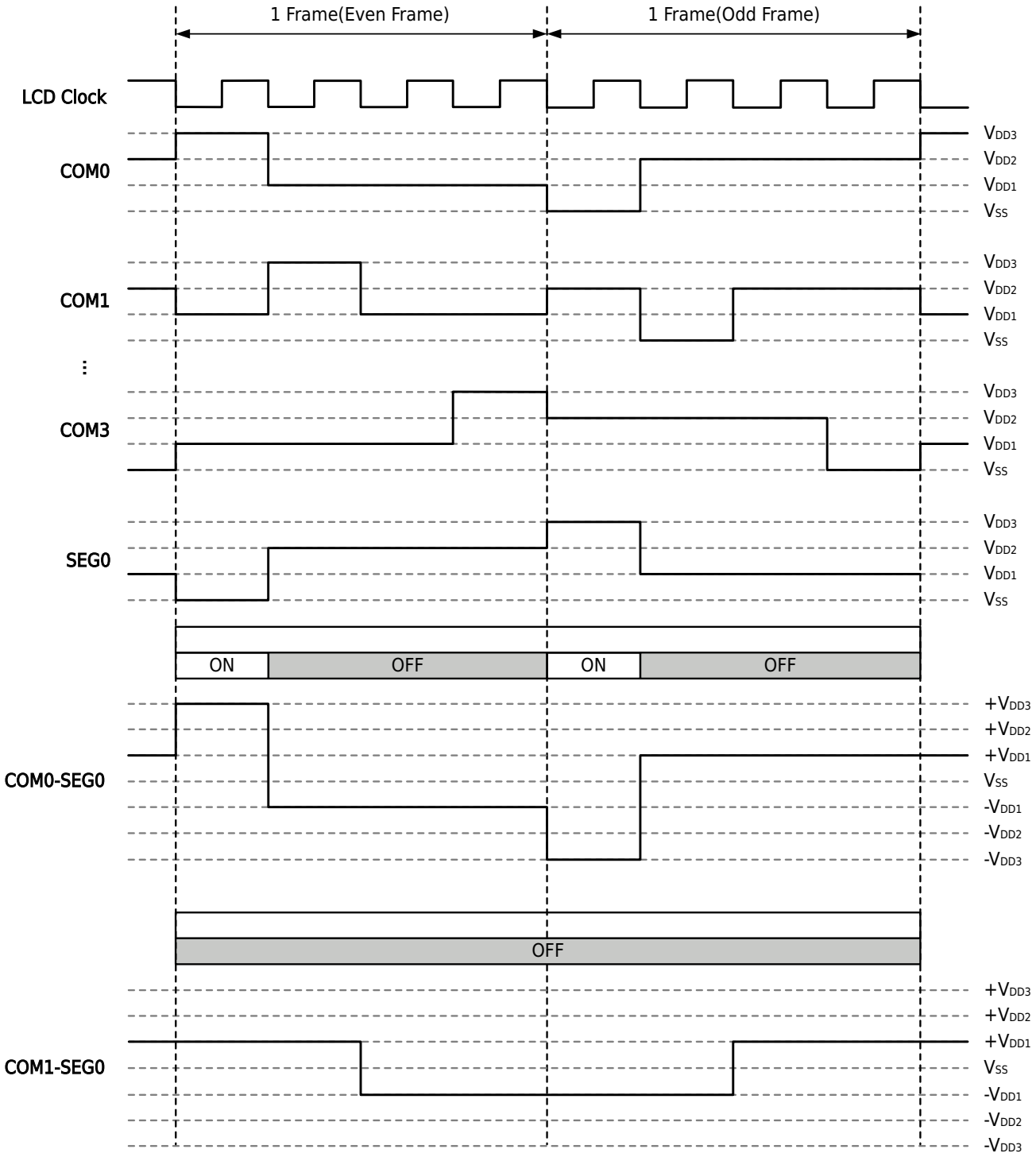
### Static Drive Waveform





All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

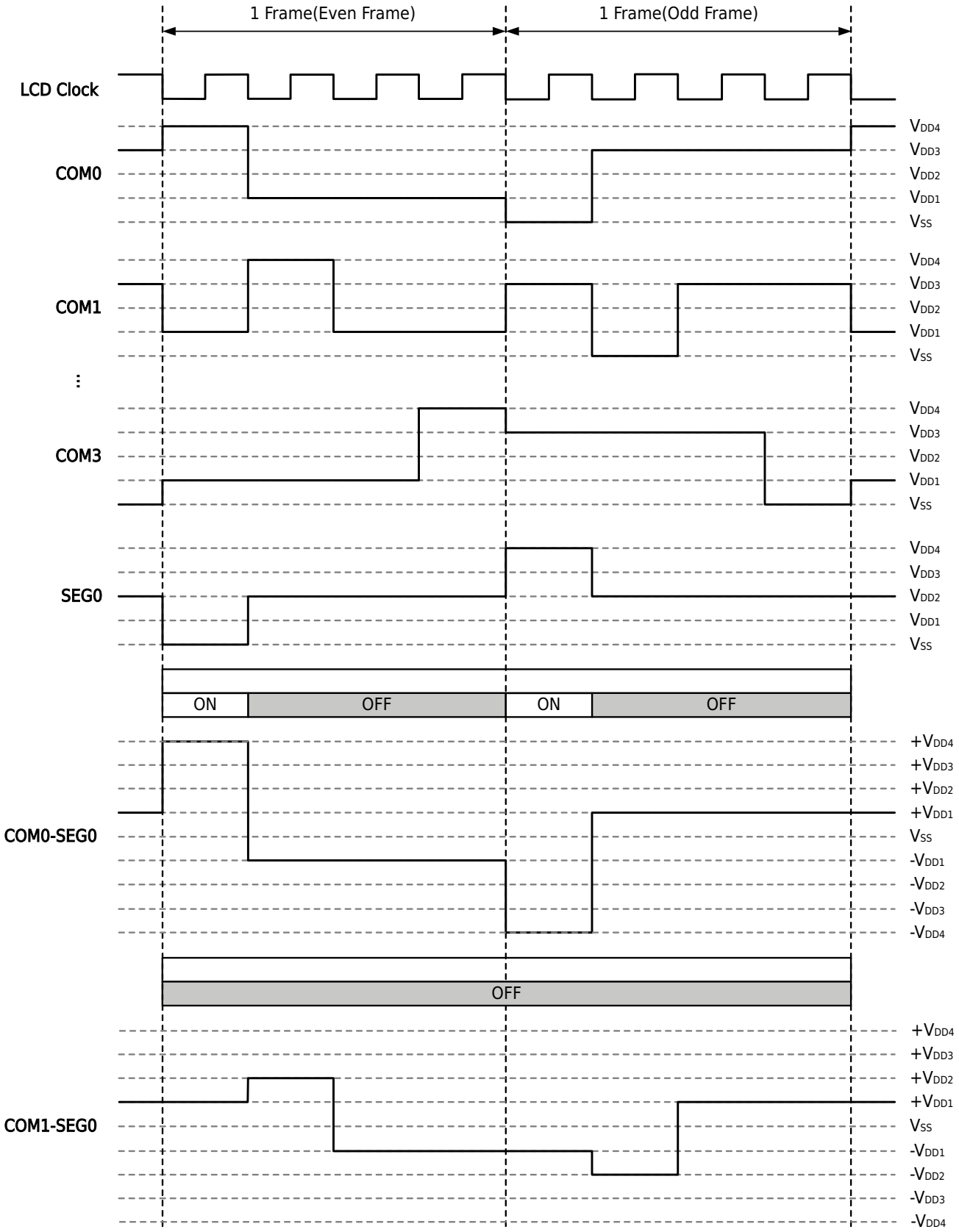
### 1/4Duty 1/3Bias Type B Drive Waveform





All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 1/4Duty 1/4Bias Type B Drive Waveform

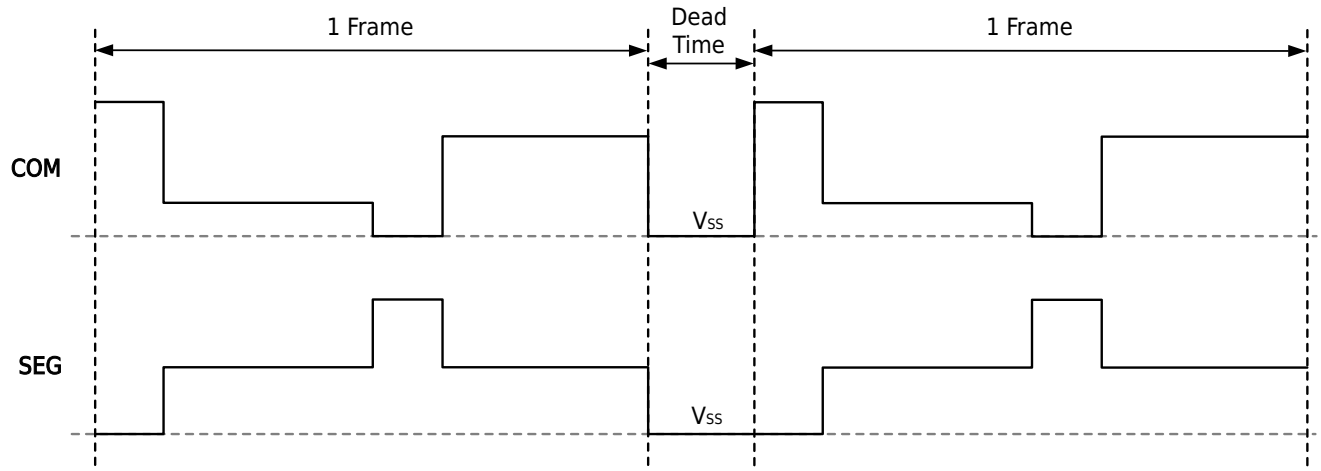



## 23.3.4 Dead Zone Insertion

Dead time can be inserted between frames to adjust contrast. Throughout the dead time, all COM and SEG lines are connected to VSS. The duration of the dead time is regulated by LCD_CR1.DEAD. For TYPE


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

B frames, dead time can only be inserted after even-numbered frames. The waveform for dead time insertion is shown below:




### 23.3.5 LCD Bias Generation Voltage

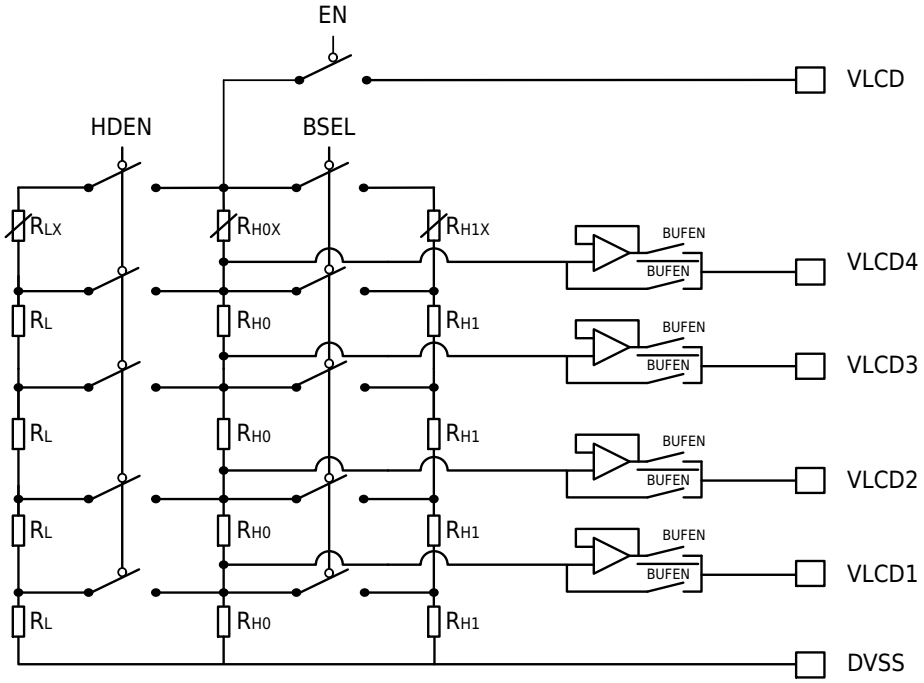
LCD Bias voltage has three sources: internal resistor divider, external resistor divider, and external capacitor divider.

1. When the internal resistance voltage division is selected, the chip will automatically switch to the internal circuit, and the driving capability of this mode is weak. If the LCD screen to be driven is large, it is recommended to select the external resistor voltage division or external capacitor voltage division mode. Drive capability can also be enhanced by configuring the LCD_CR1.DRVRES register. Resistance values decrease progressively across high, medium, and low power consumption modes. Under identical power consumption configurations, resistance values increase sequentially among 1/2 bias, 1/3 bias, and 1/4 bias settings.
2. When selecting an external resistor to divide the voltage, adjust the resistance value according to the system power consumption and LCD driving capability, and it is recommended to use a 500kΩ resistor. When selecting an external capacitor for voltage division, the drive capability is stronger, and a 100nF capacitor is recommended. However, these two circuits require users to build related circuits on the peripheral pins of the chip and occupy 5 IO ports.

#### Internal Resistance Mode

The figure below shows the schematic diagram of the 1/4 bias internal resistor.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.




Internal resistance mode VLCDH, VLCD1-VLCD4 can be used as LCD SEG output or IO port.

During transitions, low-value resistors ($R_L$) are employed to increase current for rapid stabilization. After that, the low-value resistor is turned off, and the high-value resistor ($R_H$) is used to reduce power consumption. The duration of $R_L$ usage is determined by LCD_CR1.DRVRES configuration. $R_L$ can be persistently enabled by configuring LCD_CR1.DRVRES ≥ 0xE.

Two sets of $R_H$ ($R_{H0}$ and $R_{H1}$) are used to select different power consumption modes. The selection of high, medium, and low power consumption is achieved by configuring LCD_CR0.BSEL.

### Enhanced Mode

The LCD module integrates a configurable voltage output driver, where the enhanced mode can be activated by configuring the LCD_CR0.VODEN register bit. Enabling the voltage output driver can reduce voltage interference caused by the LCD capacitive load on the bridge circuit, thereby providing a stable voltage supply and achieving the goal of enhancing LCD driving capability.

In enhanced mode, the high-value resistor bridge (RHN) generates an intermediate voltage, the LCD_CR1.DRVRES configuration is overridden, and the low-value resistor bridge (RLN) is automatically disabled, thereby reducing power consumption.

**Note**

The voltage output driver can only be configured when the LCD controller is inactive.

It is recommended to enable the buffer only when the maximum LCD drive voltage is guaranteed to be no lower than 1.8V; otherwise, LCD display quality may be compromised.

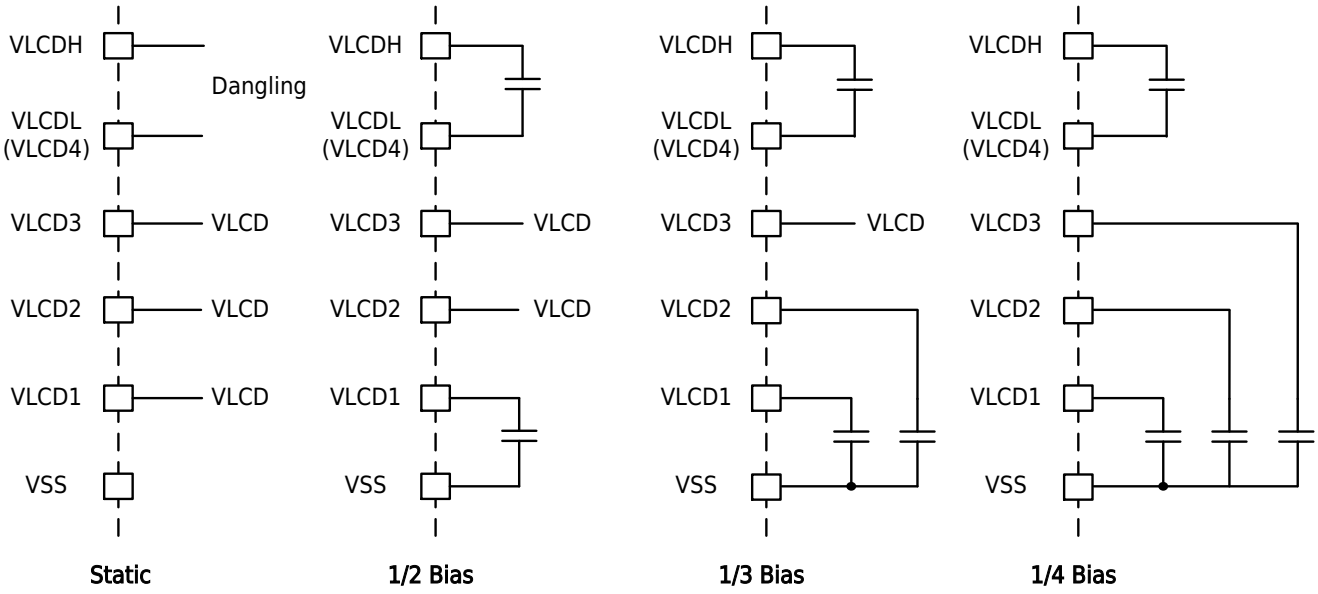
In internal resistance mode, the driving voltage of LCD is controlled by CR0.CONTRAST, as shown in the table below:

| CR0.CONTRAST | VLCD4/VLCD |
|---|---|
| 0 | 1.000 |
| 1 | 0.947 |
| 2 | 0.900 |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| CR0.CONTRAST | VLCD4/VLCD |
|---|---|
| 3 | 0.857 |
| 4 | 0.818 |
| 5 | 0.782 |
| 6 | 0.750 |
| 7 | 0.720 |
| 8 | 0.692 |
| 9 | 0.666 |
| 10 | 0.643 |
| 11 | 0.621 |
| 12 | 0.600 |
| 13 | 0.581 |
| 14 | 0.562 |
| 15 | 0.545 |

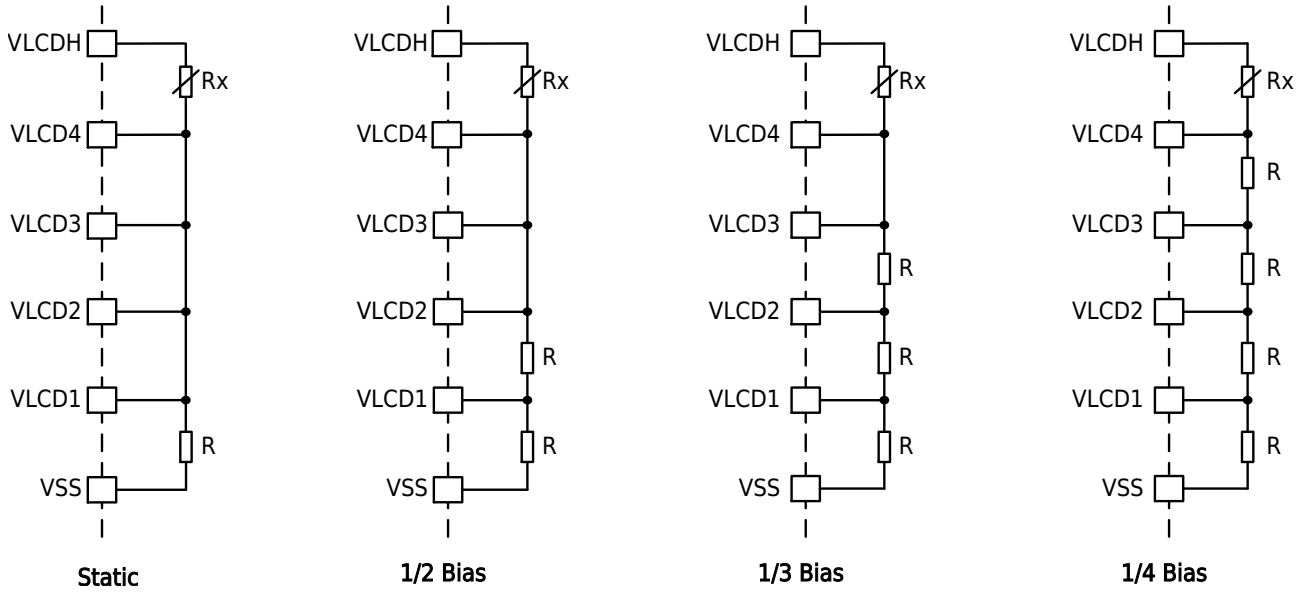
### External Capacitor Mode





All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### External Resistor Mode




**Note**

- Rx is an adjustable resistor for adjusting the contrast of LCD display.
- Select the proper resistor R according to the LCD screen used.

## 23.3.6 DMA

LCD supports software and hardware to trigger DMA data transmission, which can automatically move the content to be displayed from RAM or ROM to LCD display RAM. The hardware trigger uses the frame interrupt signal.

## 23.3.7 Interrupt

When the LCD setting is valid, the LCD interrupt can be configured to generate an interrupt for the frame number.

**Note**

When the LCD is configured in the static mode (LCD_CR0.DUTY set to 0b000), the interrupt function is not supported.

## 23.3.8 LCD Display Mode

The LCD supports two display modes. One uses COM as the display unit, and all COM segments of the same SEG are in the same byte (mode 0). The other is that different SEG of the same COM are in the same byte (mode 1). Selecting the appropriate display mode according to the LCD panel can simplify the program operation.

**Note**

SEG0-SEG31 support two display modes, whereas SEG32-SEG39 only support display mode 1. If SEG32-SEG39 are utilized, configure RAM data of SEG32~SEG39 strictly according to Display Mode 1.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

## LCD Display Mode 1 (MODE=1)

### 1/8 Duty

| | Bit0 | Bit1 | Bit2 | Bit3 | Bit4 | Bit5 | Bit6 | Bit7 | Bit8 | Bit9 | Bit10 | Bit11 | Bit12 | Bit13 | Bit14 | Bit15 | Bit16 | Bit17 | Bit18 | Bit19 | Bit20 | Bit21 | Bit22 | Bit23 | Bit24 | Bit25 | Bit26 | Bit27 | Bit28 | Bit29 | Bit30 | Bit31 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SEG0 | SEG1 | SEG2 | SEG3 | SEG4 | SEG5 | SEG6 | SEG7 | SEG8 | SEG9 | SEG10 | SEG11 | SEG12 | SEG13 | SEG14 | SEG15 | SEG16 | SEG17 | SEG18 | SEG19 | SEG20 | SEG21 | SEG22 | SEG23 | SEG24 | SEG25 | SEG26 | SEG27 | SEG28 | SEG29 | SEG30 | SEG31 | |
| COM0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAM0 |
| COM1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAM1 |
| COM2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAM2 |
| COM3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAM3 |
| COM4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAM4 |
| COM5 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAM5 |
| COM6 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAM6 |
| COM7 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAM7 |
| | SEG32 | SEG33 | SEG34 | SEG35 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| COM0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAM8 |
| COM1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAM9 |
| COM2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAMA |
| COM3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAMB |
| COM4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAMC |
| COM5 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAMD |
| COM6 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAME |
| COM7 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAMF |

### 1/6 Duty

| | Bit0 | Bit1 | Bit2 | Bit3 | Bit4 | Bit5 | Bit6 | Bit7 | Bit8 | Bit9 | Bit10 | Bit11 | Bit12 | Bit13 | Bit14 | Bit15 | Bit16 | Bit17 | Bit18 | Bit19 | Bit20 | Bit21 | Bit22 | Bit23 | Bit24 | Bit25 | Bit26 | Bit27 | Bit28 | Bit29 | Bit30 | Bit31 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SEG0 | SEG1 | SEG2 | SEG3 | SEG4 | SEG5 | SEG6 | SEG7 | SEG8 | SEG9 | SEG10 | SEG11 | SEG12 | SEG13 | SEG14 | SEG15 | SEG16 | SEG17 | SEG18 | SEG19 | SEG20 | SEG21 | SEG22 | SEG23 | SEG24 | SEG25 | SEG26 | SEG27 | SEG28 | SEG29 | SEG30 | SEG31 | |
| COM0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAM0 |
| COM1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAM1 |
| COM2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAM2 |
| COM3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAM3 |
| COM4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAM4 |
| COM5 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAM5 |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAM6 |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAM7 |
| | SEG32 | SEG33 | SEG34 | SEG35 | SEG36 | SEG37 | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| COM0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAM8 |
| COM1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAM9 |
| COM2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAMA |
| COM3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAMB |
| COM4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAMC |
| COM5 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAMD |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAME |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAMF |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

## 1/4 Duty

| | Bit0 | Bit1 | Bit2 | Bit3 | Bit4 | Bit5 | Bit6 | Bit7 | Bit8 | Bit9 | Bit10 | Bit11 | Bit12 | Bit13 | Bit14 | Bit15 | Bit16 | Bit17 | Bit18 | Bit19 | Bit20 | Bit21 | Bit22 | Bit23 | Bit24 | Bit25 | Bit26 | Bit27 | Bit28 | Bit29 | Bit30 | Bit31 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SEG0 | SEG1 | SEG2 | SEG3 | SEG4 | SEG5 | SEG6 | SEG7 | SEG8 | SEG9 | SEG10 | SEG11 | SEG12 | SEG13 | SEG14 | SEG15 | SEG16 | SEG17 | SEG18 | SEG19 | SEG20 | SEG21 | SEG22 | SEG23 | SEG24 | SEG25 | SEG26 | SEG27 | SEG28 | SEG29 | SEG30 | SEG31 | |
| COM0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAM0 |
| COM1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAM1 |
| COM2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAM2 |
| COM3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAM3 |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAM4 |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAM5 |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAM6 |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAM7 |
| | SEG32 | SEG33 | SEG34 | SEG35 | SEG36 | SEG37 | SEG38 | SEG39 | | | | | | | | | | | | | | | | | | | | | | | | | |
| COM0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAM8 |
| COM1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAM9 |
| COM2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAMA |
| COM3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAMB |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAMC |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAMD |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAME |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAMF |

## 1/3 Duty

| | Bit0 | Bit1 | Bit2 | Bit3 | Bit4 | Bit5 | Bit6 | Bit7 | Bit8 | Bit9 | Bit10 | Bit11 | Bit12 | Bit13 | Bit14 | Bit15 | Bit16 | Bit17 | Bit18 | Bit19 | Bit20 | Bit21 | Bit22 | Bit23 | Bit24 | Bit25 | Bit26 | Bit27 | Bit28 | Bit29 | Bit30 | Bit31 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SEG0 | SEG1 | SEG2 | SEG3 | SEG4 | SEG5 | SEG6 | SEG7 | SEG8 | SEG9 | SEG10 | SEG11 | SEG12 | SEG13 | SEG14 | SEG15 | SEG16 | SEG17 | SEG18 | SEG19 | SEG20 | SEG21 | SEG22 | SEG23 | SEG24 | SEG25 | SEG26 | SEG27 | SEG28 | SEG29 | SEG30 | SEG31 | |
| COM0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAM0 |
| COM1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAM1 |
| COM2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAM2 |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAM3 |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAM4 |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAM5 |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAM6 |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAM7 |
| | SEG32 | SEG33 | SEG34 | SEG35 | SEG36 | SEG37 | SEG38 | SEG39 | | | | | | | | | | | | | | | | | | | | | | | | | |
| COM0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAM8 |
| COM1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAM9 |
| COM2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAMA |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAMB |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAMC |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAMD |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAME |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAMF |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

## 1/2 Duty

| | Bit0 | Bit1 | Bit2 | Bit3 | Bit4 | Bit5 | Bit6 | Bit7 | Bit8 | Bit9 | Bit10 | Bit11 | Bit12 | Bit13 | Bit14 | Bit15 | Bit16 | Bit17 | Bit18 | Bit19 | Bit20 | Bit21 | Bit22 | Bit23 | Bit24 | Bit25 | Bit26 | Bit27 | Bit28 | Bit29 | Bit30 | Bit31 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SEG0 | SEG1 | SEG2 | SEG3 | SEG4 | SEG5 | SEG6 | SEG7 | SEG8 | SEG9 | SEG10 | SEG11 | SEG12 | SEG13 | SEG14 | SEG15 | SEG16 | SEG17 | SEG18 | SEG19 | SEG20 | SEG21 | SEG22 | SEG23 | SEG24 | SEG25 | SEG26 | SEG27 | SEG28 | SEG29 | SEG30 | SEG31 | |
| COM0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAM0 |
| COM1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAM1 |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAM2 |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAM3 |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAM4 |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAM5 |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAM6 |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAM7 |
| | SEG32 | SEG33 | SEG34 | SEG35 | SEG36 | SEG37 | SEG38 | SEG39 | | | | | | | | | | | | | | | | | | | | | | | | | |
| COM0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAM8 |
| COM1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAM9 |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAMA |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAMB |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAMC |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAMD |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAME |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LCDRAMF |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

## LCD Display Mode 0 (MODE=0)

### 1/8 Duty

| Bit0 | Bit1 | Bit2 | Bit3 | Bit4 | Bit5 | Bit6 | Bit7 | Bit8 | Bit9 | Bit10 | Bit11 | Bit12 | Bit13 | Bit14 | Bit15 | Bit16 | Bit17 | Bit18 | Bit19 | Bit20 | Bit21 | Bit22 | Bit23 | Bit24 | Bit25 | Bit26 | Bit27 | Bit28 | Bit29 | Bit30 | Bit31 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| COM0 | COM1 | COM2 | COM3 | COM4 | COM5 | COM6 | COM7 | COM0 | COM1 | COM2 | COM3 | COM4 | COM5 | COM6 | COM7 | COM0 | COM1 | COM2 | COM3 | COM4 | COM5 | COM6 | COM7 | COM0 | COM1 | COM2 | COM3 | COM4 | COM5 | COM6 | COM7 | |
| SEG0 | | | | | | | | SEG1 | | | | | | | | SEG2 | | | | | | | | SEG3 | | | | | | | | LCDRAM0 |
| SEG4 | | | | | | | | SEG5 | | | | | | | | SEG6 | | | | | | | | SEG7 | | | | | | | | LCDRAM1 |
| SEG8 | | | | | | | | SEG9 | | | | | | | | SEG10 | | | | | | | | SEG11 | | | | | | | | LCDRAM2 |
| SEG12 | | | | | | | | SEG13 | | | | | | | | SEG14 | | | | | | | | SEG15 | | | | | | | | LCDRAM3 |
| SEG16 | | | | | | | | SEG17 | | | | | | | | SEG18 | | | | | | | | SEG19 | | | | | | | | LCDRAM4 |
| SEG20 | | | | | | | | SEG21 | | | | | | | | SEG22 | | | | | | | | SEG23 | | | | | | | | LCDRAM5 |
| SEG24 | | | | | | | | SEG25 | | | | | | | | SEG26 | | | | | | | | SEG27 | | | | | | | | LCDRAM6 |
| SEG28 | | | | | | | | SEG29 | | | | | | | | SEG30 | | | | | | | | SEG31 | | | | | | | | LCDRAM7 |

### 1/6 Duty

| Bit0 | Bit1 | Bit2 | Bit3 | Bit4 | Bit5 | Bit6 | Bit7 | Bit8 | Bit9 | Bit10 | Bit11 | Bit12 | Bit13 | Bit14 | Bit15 | Bit16 | Bit17 | Bit18 | Bit19 | Bit20 | Bit21 | Bit22 | Bit23 | Bit24 | Bit25 | Bit26 | Bit27 | Bit28 | Bit29 | Bit30 | Bit31 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| COM0 | COM1 | COM2 | COM3 | COM4 | COM5 | | | COM0 | COM1 | COM2 | COM3 | COM4 | COM5 | | | COM0 | COM1 | COM2 | COM3 | COM4 | COM5 | | | COM0 | COM1 | COM2 | COM3 | COM4 | COM5 | | | |
| SEG0 | | | | | | | | SEG1 | | | | | | | | SEG2 | | | | | | | | SEG3 | | | | | | | | LCDRAM0 |
| SEG4 | | | | | | | | SEG5 | | | | | | | | SEG6 | | | | | | | | SEG7 | | | | | | | | LCDRAM1 |
| SEG8 | | | | | | | | SEG9 | | | | | | | | SEG10 | | | | | | | | SEG11 | | | | | | | | LCDRAM2 |
| SEG12 | | | | | | | | SEG13 | | | | | | | | SEG14 | | | | | | | | SEG15 | | | | | | | | LCDRAM3 |
| SEG16 | | | | | | | | SEG17 | | | | | | | | SEG18 | | | | | | | | SEG19 | | | | | | | | LCDRAM4 |
| SEG20 | | | | | | | | SEG21 | | | | | | | | SEG22 | | | | | | | | SEG23 | | | | | | | | LCDRAM5 |
| SEG24 | | | | | | | | SEG25 | | | | | | | | SEG26 | | | | | | | | SEG27 | | | | | | | | LCDRAM6 |
| SEG28 | | | | | | | | SEG29 | | | | | | | | SEG30 | | | | | | | | SEG31 | | | | | | | | LCDRAM7 |

### 1/4 Duty

| Bit0 | Bit1 | Bit2 | Bit3 | Bit4 | Bit5 | Bit6 | Bit7 | Bit8 | Bit9 | Bit10 | Bit11 | Bit12 | Bit13 | Bit14 | Bit15 | Bit16 | Bit17 | Bit18 | Bit19 | Bit20 | Bit21 | Bit22 | Bit23 | Bit24 | Bit25 | Bit26 | Bit27 | Bit28 | Bit29 | Bit30 | Bit31 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| COM0 | COM1 | COM2 | COM3 | | | | | COM0 | COM1 | COM2 | COM3 | | | | | COM0 | COM1 | COM2 | COM3 | | | | | COM0 | COM1 | COM2 | COM3 | | | | | |
| SEG0 | | | | | | | | SEG1 | | | | | | | | SEG2 | | | | | | | | SEG3 | | | | | | | | LCDRAM0 |
| SEG4 | | | | | | | | SEG5 | | | | | | | | SEG6 | | | | | | | | SEG7 | | | | | | | | LCDRAM1 |
| SEG8 | | | | | | | | SEG9 | | | | | | | | SEG10 | | | | | | | | SEG11 | | | | | | | | LCDRAM2 |
| SEG12 | | | | | | | | SEG13 | | | | | | | | SEG14 | | | | | | | | SEG15 | | | | | | | | LCDRAM3 |
| SEG16 | | | | | | | | SEG17 | | | | | | | | SEG18 | | | | | | | | SEG19 | | | | | | | | LCDRAM4 |
| SEG20 | | | | | | | | SEG21 | | | | | | | | SEG22 | | | | | | | | SEG23 | | | | | | | | LCDRAM5 |
| SEG24 | | | | | | | | SEG25 | | | | | | | | SEG26 | | | | | | | | SEG27 | | | | | | | | LCDRAM6 |
| SEG28 | | | | | | | | SEG29 | | | | | | | | SEG30 | | | | | | | | SEG31 | | | | | | | | LCDRAM7 |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

## 1/3 Duty

| Bit0 | Bit1 | Bit2 | Bit3 | Bit4 | Bit5 | Bit6 | Bit7 | Bit8 | Bit9 | Bit10 | Bit11 | Bit12 | Bit13 | Bit14 | Bit15 | Bit16 | Bit17 | Bit18 | Bit19 | Bit20 | Bit21 | Bit22 | Bit23 | Bit24 | Bit25 | Bit26 | Bit27 | Bit28 | Bit29 | Bit30 | Bit31 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| COM0 | COM1 | COM2 | | | | | | COM0 | COM1 | COM2 | | | | | | COM0 | COM1 | COM2 | | | | | | COM0 | COM1 | COM2 | | | | | | |
| SEG0 | | | | | | | | SEG1 | | | | | | | | SEG2 | | | | | | | | SEG3 | | | | | | | | LCDRAM0 |
| SEG4 | | | | | | | | SEG5 | | | | | | | | SEG6 | | | | | | | | SEG7 | | | | | | | | LCDRAM1 |
| SEG8 | | | | | | | | SEG9 | | | | | | | | SEG10 | | | | | | | | SEG11 | | | | | | | | LCDRAM2 |
| SEG12 | | | | | | | | SEG13 | | | | | | | | SEG14 | | | | | | | | SEG15 | | | | | | | | LCDRAM3 |
| SEG16 | | | | | | | | SEG17 | | | | | | | | SEG18 | | | | | | | | SEG19 | | | | | | | | LCDRAM4 |
| SEG20 | | | | | | | | SEG21 | | | | | | | | SEG22 | | | | | | | | SEG23 | | | | | | | | LCDRAM5 |
| SEG24 | | | | | | | | SEG25 | | | | | | | | SEG26 | | | | | | | | SEG27 | | | | | | | | LCDRAM6 |
| SEG28 | | | | | | | | SEG29 | | | | | | | | SEG30 | | | | | | | | SEG31 | | | | | | | | LCDRAM7 |

## 1/2 Duty

| Bit0 | Bit1 | Bit2 | Bit3 | Bit4 | Bit5 | Bit6 | Bit7 | Bit8 | Bit9 | Bit10 | Bit11 | Bit12 | Bit13 | Bit14 | Bit15 | Bit16 | Bit17 | Bit18 | Bit19 | Bit20 | Bit21 | Bit22 | Bit23 | Bit24 | Bit25 | Bit26 | Bit27 | Bit28 | Bit29 | Bit30 | Bit31 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| COM0 | COM1 | | | | | | | COM0 | COM1 | | | | | | | COM0 | COM1 | | | | | | | COM0 | COM1 | | | | | | | |
| SEG0 | | | | | | | | SEG1 | | | | | | | | SEG2 | | | | | | | | SEG3 | | | | | | | | LCDRAM0 |
| SEG4 | | | | | | | | SEG5 | | | | | | | | SEG6 | | | | | | | | SEG7 | | | | | | | | LCDRAM1 |
| SEG8 | | | | | | | | SEG9 | | | | | | | | SEG10 | | | | | | | | SEG11 | | | | | | | | LCDRAM2 |
| SEG12 | | | | | | | | SEG13 | | | | | | | | SEG14 | | | | | | | | SEG15 | | | | | | | | LCDRAM3 |
| SEG16 | | | | | | | | SEG17 | | | | | | | | SEG18 | | | | | | | | SEG19 | | | | | | | | LCDRAM4 |
| SEG20 | | | | | | | | SEG21 | | | | | | | | SEG22 | | | | | | | | SEG23 | | | | | | | | LCDRAM5 |
| SEG24 | | | | | | | | SEG25 | | | | | | | | SEG26 | | | | | | | | SEG27 | | | | | | | | LCDRAM6 |
| SEG28 | | | | | | | | SEG29 | | | | | | | | SEG30 | | | | | | | | SEG31 | | | | | | | | LCDRAM7 |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

## 23.4 Operation Example

### 23.4.1 Internal Resistance Mode Operating Process

**Step1.** Enable the LCD clock and GPIO clock.
**Step2.** Configure the pins used by the LCD as analog functions.
**Step3.** Configure LCD_CR1.CLKSRC to select the work clock source.
**Step4.** Configure LCD_CR0.BIAS to select the bias ratio.
**Step5.** Configure LCD_CR0.DUTY to select the duty cycle.
**Step6.** Configure LCD_CR1.PSC and LCD_CR1.DIV to set the frame frequency.
**Step7.** Configure LCD_CR1.MODE to select the display mode.
**Step8.** Configure LCD_CR0.VODEN to select whether enable/disable the voltage output driver. If enabled, skip step9.
**Step9.** Configure LCD_CR1.DRVRES to set the drive pulse duration.
**Step10.** Configure LCD_POEN0 and LCD_POEN1 to enable SEG/COM functions for relevant pins.
**Step11.** Set LCD_CR0.EN as 1 to enable the LCD controller.
**Step12.** Write the display content to the LCD RAM.

**Notice**
When the pins are not used as the SEG or COM pins of the LCD, the SEG or COM functions of the pins must not be enabled.

### 23.4.2 External Capacitor Mode Operating Process

**Step1.** Build the corresponding circuits based on the different expected configurations on the peripheral pins of the chip.
**Step2.** Enable the LCD clock and GPIO clock.
**Step3.** Configure the pins used by the LCD as analog functions.
**Step4.** Configure LCD_CR1.CLKSRC to select the work clock source.
**Step5.** Configure LCD_CR0.BIAS to select the bias ratio.
**Step6.** Configure LCD_CR0.DUTY to select the duty cycle.
**Step7.** Configure LCD_CR1.PSC and LCD_CR1.DIV to set the frame frequency.
**Step8.** Configure LCD_CR0.CPCLK to select the capacitor voltage division mode for driving the frequency (this mode requires a certain amount of time to complete the capacitor voltage division. When the display starts, there may be a brief period of poor display performance).
**Step9.** Configure LCD_POEN0 and LCD_POEN1 to enable SEG and COM functions for relevant pins.
**Step10.** Set LCD_CR0.EN as 1 to enable the LCD controller.
**Step11.** Write the display content to the LCD RAM.

**Notice**
When the pins are not used as the SEG or COM pins of the LCD, the SEG or COM functions of the pins must not be enabled.

### 23.4.3 External Resistor Mode Operating Process

**Step1.** Build the corresponding circuits based on the different expected configurations on the peripheral pins of the chip.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

**Step2.** Enable the LCD clock and GPIO clock.

**Step3.** Configure the pins used by the LCD as analog functions.

**Step4.** Configure LCD_CR1.CLKSRC to select the work clock source.

**Step5.** Configure LCD_CR0.BIAS to select the bias ratio.

**Step6.** Configure LCD_CR0.DUTY to select the duty cycle.

**Step7.** Configure LCD_CR1.PSC and LCD_CR1.DIV to set the frame frequency.

**Step8.** Configure LCD_POEN0 and LCD_POEN1 to enable SEG and COM functions for relevant pins.

**Step9.** Set LCD_CR0.EN as 1 to enable the LCD controller.

**Step10.** Write the display content to the LCD RAM.

**Notice**

When the pins are not used as the SEG or COM pins of the LCD, the SEG or COM functions of the pins must not be enabled.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

## 23.5 Register Description

### 23.5.1 Register List

Base address: 0x40009C00

Table 23-12 Liquid Crystal Display Controller (LCD) Register Offset

| Offset | Register | Description | Width |
|---|---|---|---|
| 0x000 | LCD_CR0 | LCD Configuration Register 0 | 32 |
| 0x004 | LCD_CR1 | LCD Configuration Register 1 | 32 |
| 0x008 | LCD_SR | LCD Status Register | 32 |
| 0x00C | LCD_INTCLR | LCD Interrupt Clear Register | 32 |
| 0x010 | LCD_POEN0 | LCD Output Configuration Register 0 | 32 |
| 0x014 | LCD_POEN1 | LCD Output Configuration Register 1 | 32 |
| 0x040 | LCD_RAM0 | LCD RAM0 | 32 |
| 0x044 | LCD_RAM1 | LCD RAM1 | 32 |
| 0x048 | LCD_RAM2 | LCD RAM2 | 32 |
| 0x04C | LCD_RAM3 | LCD RAM3 | 32 |
| 0x050 | LCD_RAM4 | LCD RAM4 | 32 |
| 0x054 | LCD_RAM5 | LCD RAM5 | 32 |
| 0x058 | LCD_RAM6 | LCD RAM6 | 32 |
| 0x05C | LCD_RAM7 | LCD RAM7 | 32 |
| 0x060 | LCD_RAM8 | LCD RAM8 | 32 |
| 0x064 | LCD_RAM9 | LCD RAM9 | 32 |
| 0x068 | LCD_RAMA | LCD RAM10 | 32 |
| 0x06C | LCD_RAMB | LCD RAM11 | 32 |
| 0x070 | LCD_RAMC | LCD RAM12 | 32 |
| 0x074 | LCD_RAMD | LCD RAM13 | 32 |
| 0x078 | LCD_RAME | LCD RAM14 | 32 |
| 0x07C | LCD_RAMF | LCD RAM15 | 32 |

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 23.5.2 Configuration Register 0 (LCD_CR0)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x000 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x000000DC | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | VODEN | CONTRAST | | | | BSEL | | | DUTY | | | TYPE | CPCLK | | BIAS | EN |
| Access | | | | | | | | | | | | | | | | RW | RW | | | | RW | | | RW | | | RW | RW | | RW | RW |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:17* | *Reserved* | - | - |
| 16 | VODEN | Voltage output drive enable<br>● 0b0: Disable<br>● 0b1: Enable<br>**Note**<br>This bit can only be set when LCD_CR0.EN is 0.<br>Only valid when the BIAS voltage source is selected as an internal resistor divider. | RW |
| 15:12 | CONTRAST | LCD contrast adjustment<br>The larger the CONTRAST value is, the smaller the amplitude of the LCD waveform will be.<br>● 0b0000: The LCD waveform has the largest amplitude and the largest contrast<br>● .....<br>● 0b1111: The LCD waveform amplitude is the smallest, and the contrast is the smallest<br>**Note**<br>It is only valid when the BIAS voltage source selects internal resistor voltage divider. | RW |
| 11:9 | BSEL | BIAS voltage source selection<br>● 0b000: External resistance mode,requires external circuit cooperation<br>● 0b001: Capacitive voltage divider mode, requires external circuit cooperation<br>● 0b010: Internal resistor divider, medium power consumption mode<br>● 0b011: Setting prohibited<br>● 0b100: Internal resistor divider, low power consumption mode<br>● 0b101: Setting prohibited<br>● 0b110: Internal resistor divider, high power consumption mode<br>● 0b111: Setting prohibited | RW |
| 8:6 | DUTY | LCD duty configuration | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

<table>
<tr><th>Bit</th><th>Name</th><th>Functional Description</th><th>Access</th></tr>
<tr><td>8:6</td><td>DUTY</td><td>
<ul>
<li>0b000: Static</li>
<li>0b001: 1/2 duty</li>
<li>0b010: 1/3 duty</li>
<li>0b011: 1/4 duty</li>
<li>0b100: Setting prohibited</li>
<li>0b101: 1/6 duty</li>
<li>0b110: Setting prohibited</li>
<li>0b111: 1/8 duty</li>
</ul>
Table 23-15 Configuration of COMy/SEGx PAD Functions Under Different Duty Ratios
<table>
<tr><th>PAD</th><th>1/4 duty</th><th>1/6 duty</th><th>1/8 duty</th></tr>
<tr><td>COM4/SEG39</td><td>SEG39</td><td>COM4</td><td>COM4</td></tr>
<tr><td>COM5/SEG38</td><td>SEG38</td><td>COM5</td><td>COM5</td></tr>
<tr><td>COM6/SEG37</td><td>SEG37</td><td>SEG37</td><td>COM6</td></tr>
<tr><td>COM7/SEG36</td><td>SEG36</td><td>SEG36</td><td>COM7</td></tr>
</table>
</td><td>RW</td></tr>
<tr><td>5</td><td>TYPE</td><td>LCD TYPE configuraton
<ul>
<li>0b0: Waveform B</li>
<li>0b1: Setting prohibited</li>
</ul>
</td><td>RW</td></tr>
<tr><td>4:3</td><td>CPCLK</td><td>Voltage pump clock frequency selection
<ul>
<li>0b00: $f_{in\_clk}/16$</li>
<li>0b01: $f_{in\_clk}/8$</li>
<li>0b10: $f_{in\_clk}/4$</li>
<li>0b11: $f_{in\_clk}/2$</li>
</ul>
</td><td>RW</td></tr>
<tr><td>2:1</td><td>BIAS</td><td>LCD BIAS configuration
<ul>
<li>0b00: 1/2 bias</li>
<li>0b01: 1/3 bias</li>
<li>0b10: 1/4 bias</li>
<li>0b11: Setting prohibited</li>
</ul>
</td><td>RW</td></tr>
<tr><td>0</td><td>EN</td><td>LCD enable control
<ul>
<li>0b0: Disable</li>
<li>0b1: Enable</li>
</ul>
</td><td>RW</td></tr>
</table>


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 23.5.3 Configuration Register 1 (LCD_CR1)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x004 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | DEAD | | | PSC | | | DIV | | | DRVRES | | | | Reserved | DMAEN | IE | MODE | CLKSRC | BLINKEN | BLINKCNT | | | | | |
| Access | | | | | | | | RW | | | RW | | | RW | | | RW | | | | | RW | RW | RW | RW | RW | RW | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:25* | *Reserved* | - | - |
| 24:22 | DEAD | Dead zone time configuration<br>● 0b000: No dead zone time<br>● 0b001: 1-phase cycle dead zone time<br>● 0b010: 2-phase cycle dead zone time<br>● 0b011: 3-phase cycle dead zone time<br>● 0b100: 4-phase cycle dead zone time<br>● 0b101: 5-phase cycle dead zone time<br>● 0b110: 6-phase cycle dead zone time<br>● 0b111: 7-phase cycle dead zone time | RW |
| 21:19 | PSC | LCD clock prescaler<br>● 0b000: $f_{PSC}= f_{in\_clk}/2$<br>● 0b001: $f_{PSC}= f_{in\_clk}/4$<br>● 0b010: $f_{PSC}= f_{in\_clk}/8$<br>● 0b011: $f_{PSC}= f_{in\_clk}/16$<br>● 0b100: $f_{PSC}= f_{in\_clk}/32$<br>● Others: Setting prohibited | RW |
| 18:15 | DIV | LCD clock divider<br>● 0b0000: $f_{LCD}= f_{PSC}/16$<br>● 0b0001: $f_{LCD}= f_{PSC}/17$<br>● .....<br>● 0b1110: $f_{LCD}= f_{PSC}/30$<br>● 0b1111: $f_{LCD}= f_{PSC}/31$ | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 14:12 | DRVRES | Internal resistor mode enhanced drive pulse time control<br>● 0b000: OFF, disable enhanced drive pulse<br>● 0b001: $1/f_{in\_clk}$<br>● 0b010: $2/f_{in\_clk}$<br>● 0b011: $4/f_{in\_clk}$<br>● 0b100: $8/f_{in\_clk}$<br>● 0b101: $16/f_{in\_clk}$<br>● 0b110: $32/f_{in\_clk}$<br>● 0b111: ALL ON | RW |
| *11* | *Reserved* | - | - |
| 10 | DMAEN | DMA hardware trigger enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 9 | IE | Interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 8 | MODE | LCD RAM display mode selection<br>● 0b0: Mode 0<br>● 0b1: Mode 1 | RW |
| 7 | CLKSRC | LCD clock source selection<br>● 0b0:RCL<br>● 0b1:XTL | RW |
| 6 | BLINKEN | LCD splash screen configuration<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 5:0 | BLINKCNT | Splash screen frequency and LCD interrupt interval setting<br>LCD blinking frequency = LCD frame frequency/(BlinkCnt + 1)<br>LCD interrupt interval = (1/LCD frame frequency) × (BlinkCnt + 1) | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 23.5.4 Status Register (LCD_SR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x008 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | INTF |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | RO |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:1* | *Reserved* | - | - |
| 0 | INTF | Interrupt flag<br>● 0b0: No interrupt generated<br>● 0b1: Generate an interrupt | RO |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 23.5.5 Interrupt Clear Register (LCD_INTCLR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x00C | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | INTF |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | WO |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:1* | *Reserved* | - | - |
| 0 | INTF | Interrupt flag clear<br>Write 1 to clear the interrupt flag, write 0 has no effect; read always returns 0. | WO |

### 23.5.6 Output Configuration Register 0 (LCD_POEN0)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x010 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0xFFFFFFFF | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | S31 | S30 | S29 | S28 | S27 | S26 | S25 | S24 | S23 | S22 | S21 | S20 | S19 | S18 | S17 | S16 | S15 | S14 | S13 | S12 | S11 | S10 | S9 | S8 | S7 | S6 | S5 | S4 | S3 | S2 | S1 | S0 |
| Access | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 31:0 | Sx (x = 31-0) | SEGx output control bit<br>● 0b0: SEG output enable<br>● 0b1: SEG output is off, you can use other functions, such as IO, analog input and output | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 23.5.7 Output Configuration Register 1 (LCD_POEN1)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x014 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x01F000FF | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | C3 | C2 | C1 | C0 | MUX | Reserved | | | | | | | | | | | | S39C4 | S38C5 | S37C6 | S36C7 | S35 | S34 | S33 | S32 |
| Access | | | | | | | | RW | RW | RW | RW | RW | | | | | | | | | | | | | RW | RW | RW | RW | RW | RW | RW | RW |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:25* | *Reserved* | - | - |
| 24:21 | Cx (x = 3-0) | COMx output control bit<br>● 0b0: COM output enable<br>● 0b1: COM output is off, you can use other functions, such as IO, analog input and output | RW |
| 20 | MUX | SEG32-SEG35 port function selection<br>Parameter selection details (See table below) | RW |
| *19:8* | *Reserved* | - | - |
| 7 | S39C4 | SEG39/COM4 output control bit<br>SEG/COM pin function selection is determined by CR0.DUTY<br>● 0b0: SEG/COM output enable<br>● 0b1: SEG/COM output is off, you can use other functions, such as IO, analog input and output | RW |
| 6 | S38C5 | SEG38/COM5 output control bit<br>SEG/COM pin function selection is determined by CR0.DUTY<br>● 0b0: SEG/COM output enable<br>● 0b1: SEG/COM output is off, you can use other functions, such as IO, analog input and output | RW |
| 5 | S37C6 | SEG37/COM6 output control bit<br>SEG/COM pin function selection is determined by CR0.DUTY<br>● 0b0: SEG/COM output enable<br>● 0b1: SEG/COM output is off, you can use other functions, such as IO, analog input and output | RW |
| 4 | S36C7 | SEG36/COM7 output control bit<br>SEG/COM pin function selection is determined by CR0.DUTY<br>● 0b0: SEG/COM output enable<br>● 0b1: SEG/COM output is off, you can use other functions, such as IO, analog input and output | RW |
| 3:0 | Sx (x = 35-32) | SEGx output control bit<br>● 0b0: SEG output enable<br>● 0b1: SEG output is off, you can use other functions, such as IO, analog input and output | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| VLCDy/SEGx PAD State Description | | | | | | | Register configuration | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Status | | VLCDH/ SEG35 | VLCD4/ SEG34 | VLCD3/ SEG33 | VLCD2/ SEG32 | VLCD1/ SEG31 | MUX | S <35:31> | BSEL | | |
| When VLCDy/SEGx PAD selects GPIO and LCD disable (LCD_ON=0) | | IO | IO | IO | IO | IO | - | - | - | - | - |
| VLCDy/SEGx PAD selects GPIO and LCD enable (LCD_ON=1), only the internal resistance working mode can be selected at this time | Small resistance (high current) mode | IO | IO | IO | IO | IO | 0b1 | 0b11111 | 1 | 1 | 0 |
| | Medium resistance (medium current) mode | IO | IO | IO | IO | IO | 0b1 | 0b11111 | 0 | 1 | 0 |
| | Large resistance (low current) mode | IO | IO | IO | IO | IO | 0b1 | 0b11111 | 1 | 0 | 0 |
| When VLCDy/SEGx selects GPIO or LCD special pins and LCD enable (LCD_ON = 1), it selects the external resistor operation mode and disconnects the internal resistor. | | VLCDH | VLCD4 | VLCD3 | VLCD2 | VLCD1 | 0b0 | 0bX0000 | 0 | 0 | 0 |
| When VLCDy/SEGx PAD selects GPIO or LCD special pins and LCD enable (LCD_ON=1), internal resistor mode can be selected | Small resistance (high current) mode | SEG35/IO | SEG34/IO | SEG33/IO | SEG32/IO | SEG31/IO | 0b0 | - | 1 | 1 | 0 |
| | Medium resistance (medium current) mode | SEG35/IO | SEG34/IO | SEG33/IO | SEG32/IO | SEG31/IO | 0b0 | - | 0 | 1 | 0 |
| | Large resistance (low current) mode | SEG35/IO | SEG34/IO | SEG33/IO | SEG32/IO | SEG31/IO | 0b0 | - | 1 | 0 | 0 |
| When VLCDy/SEGx selects LCD special pins and LCD enable (LCD_ON = 1), it selects the external capacitance operation mode and disconnects the internal resistor. | | VLCDH | VLCDL | VLCD3 | VLCD2 | VLCD1 | 0b0 | 0b00000 | 0 | 0 | 1 |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 23.5.8 LCD_RAMx (x=0-7)

Offset: 0x040 (LCDRAM0), 0x044 (LCDRAM1), 0x048 (LCDRAM2), 0x04C (LCDRAM3), 0x050 (LCDRAM4), 0x054 (LCDRAM5), 0x058 (LCDRAM6), 0x05C (LCDRAM7)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| - | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0xXXXXXXXX | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | D31 | D30 | D29 | D28 | D27 | D26 | D25 | D24 | D23 | D22 | D21 | D20 | D19 | D18 | D17 | D16 | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| Access | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 31:0 | Dn (n = 31-0) | LCD point output<br>For specific display, refer to the description of LCD Display Mode.<br>● 0b0: Corresponding SEG-COM crossover point not lit<br>● 0b1: Corresponding SEG-COM crossover point lit | RW |

### 23.5.9 LCD_RAMx (x=8-F)

Offset: 0x060 (LCDRAM8), 0x064 (LCDRAM9), 0x068 (LCDRAMA), 0x06C (LCDRAMB), 0x070 (LCDRAMC), 0x074 (LCDRAMD), 0x078 (LCDRAME), 0x07C (LCDRAMF)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| - | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0xXXXXXXXX | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | D23 | D22 | D21 | D20 | D19 | D18 | D17 | D16 | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| Access | | | | | | | | | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:24* | *Reserved* | - | - |
| 23:0 | Dn (n = 23-0) | LCD point output<br>For specific display, refer to the description of LCD Display Mode.<br>● 0b0: Corresponding SEG-COM crossover point not lit<br>● 0b1: Corresponding SEG-COM crossover point lit | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

# 24 VC

## 24.1 Introduction

The analog voltage comparator VC is used to compare the size of two input analog voltages, and output high/low level according to the comparison result.

- When the "+" input voltage is higher than the "-" input voltage, the voltage comparator output is high;
- When the "+" input voltage is lower than the "-" input voltage, the voltage comparator output is low.

## 24.2 Features

- Support 64-stage voltage divider reference
- Support 8 internal and external input signals
- Support 2 output polarities
- Support window compare function
- A variety of filter time configurations can effectively enhance the anti-interference ability of the chip
- Support 3 interrupt triggering methods: high level/rising edge/falling edge
- The output of the voltage comparator can be used as the input of the timer
- Support working in DeepSleep mode

**Note**

BGR needs to be enabled when using VC see "Other Analog Registers" chapter to BGR register related description.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

# 24.3 Functional Description

## 24.3.1 Function Block Diagram

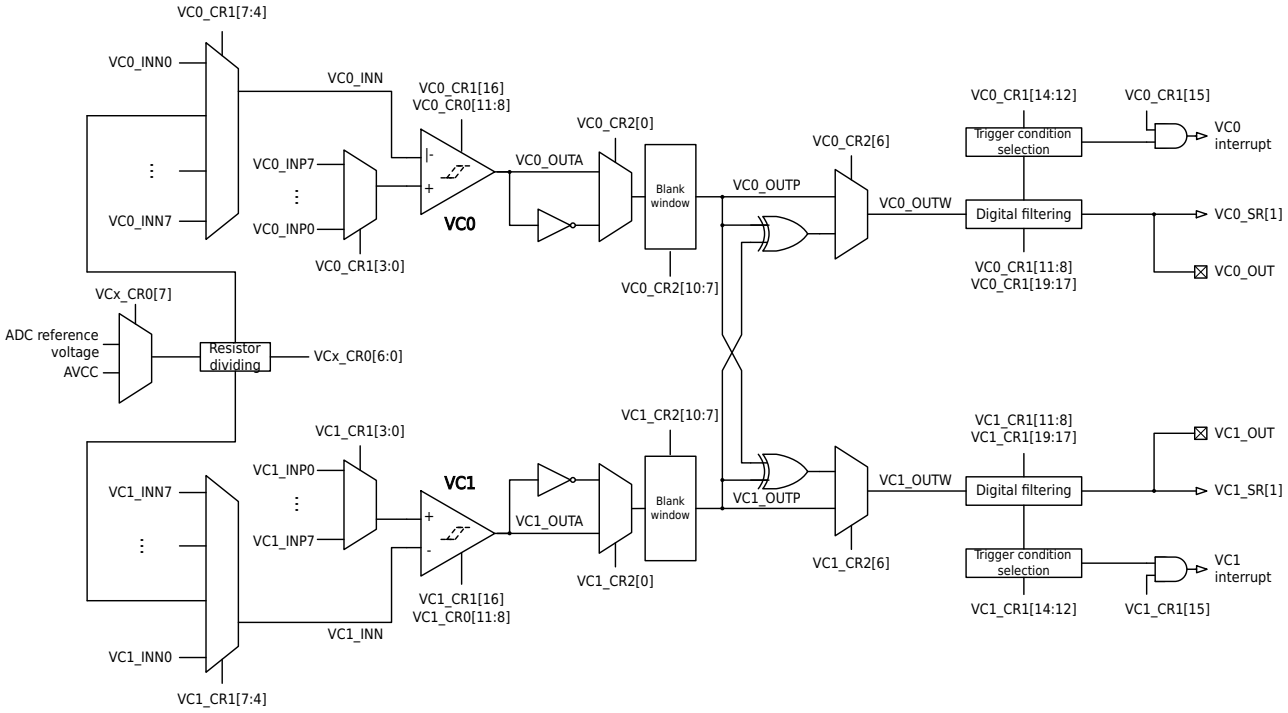



Figure 24-1 VC Function Block Diagram

## 24.3.2 Setup/Response Time

When using a voltage comparator, the time from when VC is enabled or when the input voltage at both ends of VC changes to when correct result is output is determined by the VCx_CR0.BIAS bit. The larger the current, the faster the VC response. For the specific time, see the description in the "Electrical Specifications-Operating Conditions-VC Features" chapter of the "Datasheet".

When VCx_CR0.BIAS = 0b10/0b11, the BGR needs to be started (For the stabilization time $T_{stable(BGR)}$ after the BGR is started, see the description in the "Electrical Specifications-Operating Conditions-VC Features" chapter of the "Datasheet"). The voltage comparator needs to wait until the internal BGR is stable before it can output normally.

When the resistor voltage-dividing circuit is enabled (VCx_CR0.DIVEN=0b1), the voltage comparator needs to wait until the resistor voltage-dividing circuit is stable ($T_{VDAC\_setup}$) before it can output normally. If the ADC reference voltage is used as the comparison reference voltage (VCx_CR0.REF = 0b1), the voltage comparator needs to wait for the maximum value of $T_{VDAC\_setup}$ and $T_{stable\ (REFBUF)}$ before it can output normally.

**Note**

For the stabilization time $T_{stable(REFBUF)}$ after the REFBUF is started, see the description in the "Electrical Specifications-Operating Conditions-Built-In Reference Voltage" section of the "Datasheet". For the stabilization time $T_{VDAC\_setup}$ of the dividing voltage output after the built-in DAC in the voltage comparator is turned on, see the description in the "Electrical Specifications-Operating Conditions-VC Features" section of the "Datasheet".


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 24.3.3 Digital Filtering

In addition to the inherent setup/response time of the voltage comparator, the user can set a longer filter time to filter out system noise, such as high current noise when the motor stops.

The digital filtering signal level can be read from the register VCx_SR[1]; when the GPIO function is configured as VCx_OUT, the digital filtering signal can be output from the GPIO for easy measurement.

Enable the digital filter module and select the filter time as 2 filter clock cycles, the filter illustration is as follows:

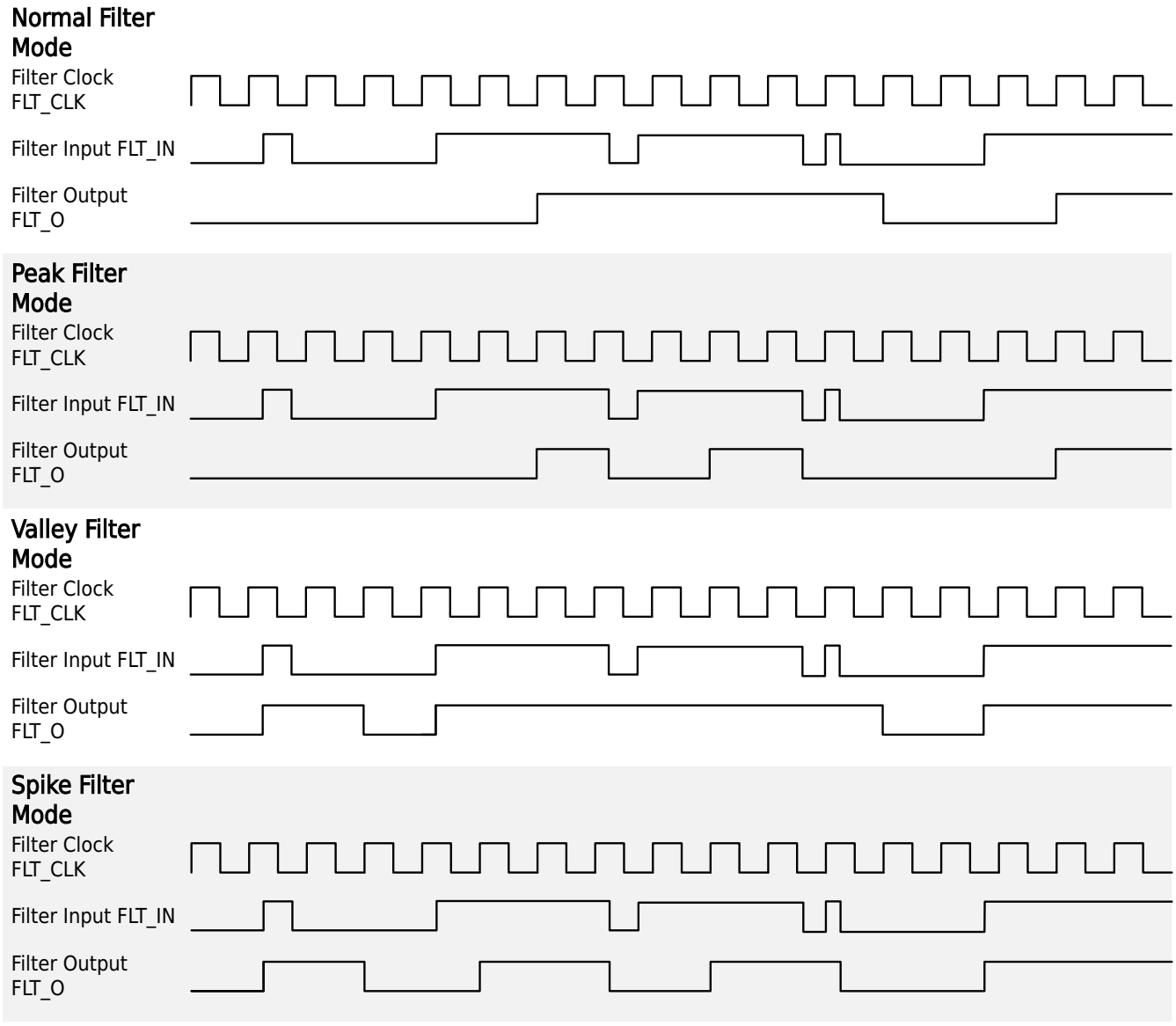



**Figure 24-2 VC Filter Response Time**

### 24.3.4 Hysteresis Function

The hysteresis function can be selected for the voltage comparator, and the diagram after the hysteresis function is enabled is as follows:


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

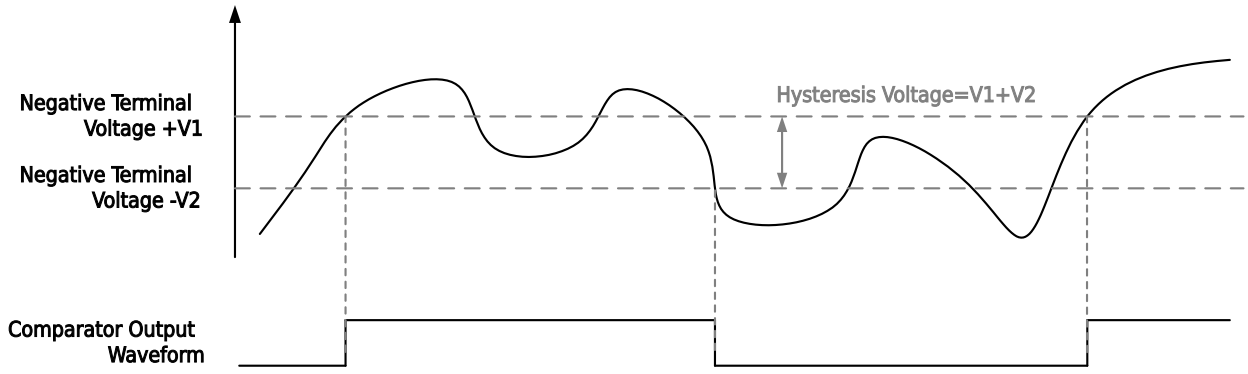



Figure 24-3 VC Hysteresis Function

### 24.3.5 Configuration of Comparison Result Polarity

When VCx_CR2.POL is configured as 0, the level of VCx_OUTA and VCx_OUTP is the same.

When VCx_CR2.POL is configured as 1, the level of VCx_OUTA and VCx_OUTP is opposite.

### 24.3.6 Window Comparison Function

When VCx_CR2.WINDOW is configured to 0, the VCx_OUTW signal is the same as the VCx_OUTP signal.

When VCx_CR2.WINDOW is configured to 1, the VCx_OUTW signal is the XOR value of VC0_OUTP and VC1_OUTP.

### 24.3.7 Blank Window

The PWM signals output by timer TIM0/3 can be used as blank window signal to control the output of the comparator. Configure VCx_CR2.BLANK to enable the blank window function or select the corresponding PWM blank window signal.

The purpose of the blank window is to prevent abnormal VC comparison results caused by current glitches that occur at the start of the PWM cycle. Blank window function diagram is as follows:

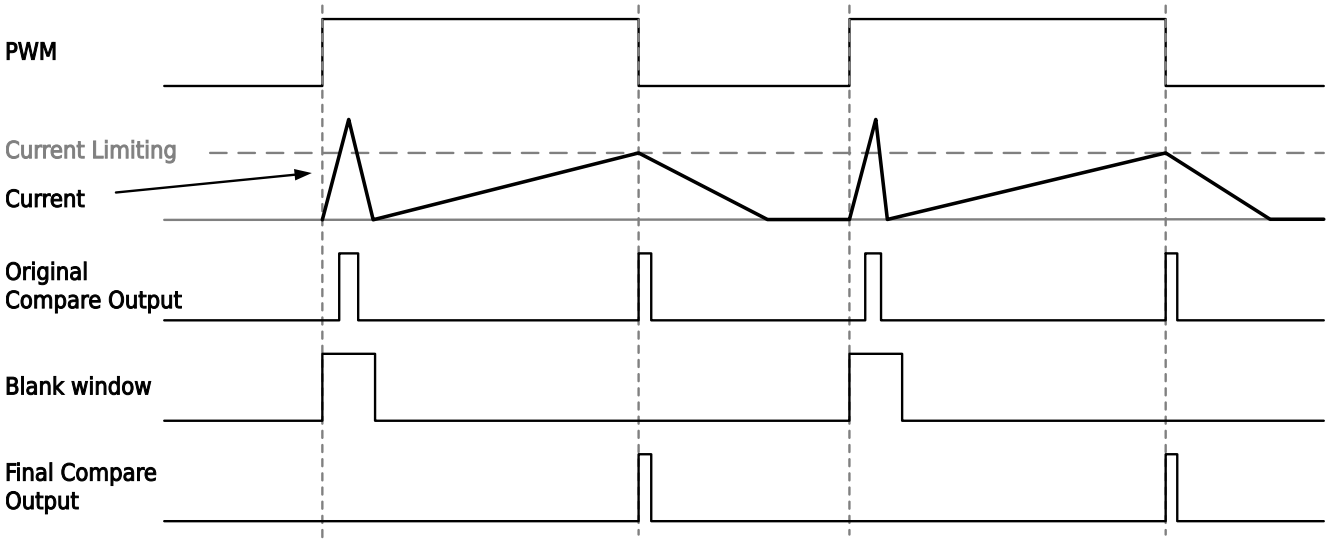



Figure 24-4 VC Blank Window


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

## 24.4 Register Description

### 24.4.1 Register List

Base address: 0x40002800

Table 24-1 Analog Voltage Comparator (VC) Register Offset

| Offset | Register | Description | Width |
|---|---|---|---|
| 0x00 | VC0_CR0 | VC0 Control Register 0 | 32 |
| 0x04 | VC0_CR1 | VC0 Control Register 1 | 32 |
| 0x08 | VC0_CR2 | VC0 Control Register 2 | 32 |
| 0x0C | VC0_SR | VC0 Status Register | 32 |
| 0x10 | VC1_CR0 | VC1 Control Register 0 | 32 |
| 0x14 | VC1_CR1 | VC1 Control Register 1 | 32 |
| 0x18 | VC1_CR2 | VC1 Control Register 2 | 32 |
| 0x1C | VC1_SR | VC1 Status Register | 32 |
| 0x20 | VC_TRIM | VC Offset Trim Register | 32 |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 24.4.2 Control Register 0 (VCx_CR0) (x=0-1)

Offset: 0x00 (VC0_CR0), 0x10 (VC1_CR0)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| - | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000020 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | HYS | | BIAS | | REF | DIVEN | DIV | | | | | |
| Access | | | | | | | | | | | | | | | | | | | | | RW | | RW | | RW | RW | RW | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:12* | *Reserved* | - | - |
| 11:10 | HYS | Hysteresis function configuration<br>● 0b00: No hysteresis<br>● 0b01: Hysteresis voltage approximately 10mV<br>● 0b10: Hysteresis voltage approximately 20mV<br>● 0b11: Hysteresis voltage approximately 30mV | RW |
| 9:8 | BIAS | Power selection<br>● 0b00: Low power consumption level<br>● 0b01: Medium and low power consumption level<br>● 0b10: Medium power consumption level<br>● 0b11: High power consumption level<br>**Note**<br>The larger the power consumption level, the faster the response speed. For the the specifications of specific gears, see the description in "Electrical Specifications-VC Features" section of the "Datasheet".<br>When 0b10 and 0b11 are selected, the BGR needs to be turned on. For the waiting time, see the description in the "Datasheet". | RW |
| 7 | REF | Selection of reference voltage for resistor voltage dividing circuit<br>● 0b0: AVCC<br>● 0b1: The reference voltage of the ADC.<br>**Note**<br>VC0/VC1 share this bit setting. | RW |
| 6 | DIVEN | Resistor voltage divider circuit enable control<br>● 0b0: Disable<br>● 0b1: Enable<br>**Note**<br>VC0/VC1 share this bit setting. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 5:0 | DIV | Resistor voltage divider circuit output voltage configuration<br>● 0b000000: 0/64 Vref<br>● 0b000001: 1/64 Vref<br>● ...<br>● 0b111110: 62/64 Vref<br>● 0b111111: 63/64 Vref<br>**Note**<br>VC0/VC1 share this bit setting, Vref source determined by the REF control bit. | RW |

**Note**

When VC is enabled, if you modify the VC control register in the above table, you need to wait for the $T_{setup}$ time before the VC can work normally. For the value of $T_{setup}$, detailed see the "Datasheet".


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 24.4.3 Control Register 1 (VCx_CR1) (x=0-1)

Offset: 0x04 (VC0_CR1), 0x14 (VC1_CR1)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| - | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x000000FF | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | FLT_MODE | | FLTCLK | EN | IE | LEVEL | RISING | FALLING | FLTTIME | | | | NSEL | | | PSEL | | | |
| Access | | | | | | | | | | | | | RW | | RW | RW | RW | RW | RW | RW | RW | | | | RW | | | RW | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:20* | *Reserved* | - | - |
| 19:18 | FLT_MODE | Filter pattern selection<br>● 0b00: Standard Filtering Mode. Filter output toggles only if input remains stable for a defined hold time after transition.<br>● 0b01: Peak Filtering Mode. Output sets to logic 1 only after the input holds logic 1 for a stable duration; sets to 0 immediately on input set to 0. The purpose of this mode is to filter out input peaks that are not long enough.<br>● 0b10: Valley Filtering Mode. Output sets to logic 0 only after input holds logic 0 for a stable duration; sets to 1 immediately on input set to 1. The purpose of this mode is to filter out input vallys that are not long enough.<br>● 0b11: Glitch Filtering Mode. Output toggles immediately on input transition, then ignores subsequent input changes for a hold time. In this mode, the filter output will closely follow the input, and at the same time, it will prevent the output signal from having glitches. | RW |
| 17 | FLTCLK | Filter clock selection<br>● 0b0: RC256K<br>● 0b1: PCLK | RW |
| 16 | EN | VC enable control<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 15 | IE | VC interrupt enable control<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 14 | LEVEL | Filter signal high level trigger interrupt enable control<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 13 | RISING | Filter signal rising edge trigger interrupt enable control<br>● 0b0: Disable<br>● 0b1: Enable | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 12 | FALLING | Filter signal falling edge trigger interrupt enable control<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 11:8 | FLTTIME | Filter time configuration<br>N represents filter clock cycles<br>● 0x0: Digital filter disabled<br>● 0x1: N = 2<br>● 0x2: N = 4<br>● 0x3: N = 8<br>● 0x4: N = 16<br>● 0x5: N = 32<br>● 0x6: N = 64<br>● 0x7: N = 128<br>● 0x8: N = 256<br>● 0x9: N = 512<br>● 0xA: N = 1024<br>● 0xB: N = 2048<br>● Other: N = 4096 | RW |
| 7:4 | NSEL | Voltage comparator negative input signal selection<br>● 0b0000: VCx_INN0 (x=0, PA00; x=1, PA02)<br>● 0b0001: VCx_INN1 (PA04)<br>● 0b0010: VCx_INN2 (PA05)<br>● 0b0011: VCx_INN3 (x=0, PB01; x=1, PB03)<br>● 0b0100: VCx_INN4 ( VCAP voltage)<br>● 0b0101: VCx_INN5 ( Built-in resistor voltage divider output voltage)<br>● 0b0110: VCx_INN6 (built-in temperature sensor output voltage)<br>● 0b0111: VCx_INN7 (PA04)<br>● Others: Setting prohibited | RW |
| 3:0 | PSEL | Voltage comparator positive input signal selection<br>● 0b0000: VCx_INP0 (PA01)<br>● 0b0001: VCx_INP1 (PA03)<br>● 0b0010: VCx_INP2 (PA06)<br>● 0b0011: VCx_INP3 (PA07)<br>● 0b0100: VCx_INP4 ( PB04)<br>● 0b0101: VCx_INP5 ( PB05)<br>● 0b0110: VCx_INP6 (PB06)<br>● 0b0111: VCx_INP7 (PB07)<br>● Others: Setting prohibited | RW |

**Note**

When VC is enabled, if you modify NSEL/PSEL, you need to wait for $T_{setup}$ before VC can work normally. For the value of $T_{setup}$, details see the "Datasheet".


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 24.4.4 Control Register 2 (VCx_CR2) (x=0-1)

Offset: 0x08 (VC0_CR2), 0x18 (VC1_CR2)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| - | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | BLANK | | | | WINDOW | TIMBK | TIM3CLR | TIM2CLR | TIM1CLR | TIM0CLR | POL |
| Access | | | | | | | | | | | | | | | | | | | | | | RW | | | | RW | RW | RW | RW | RW | RW | RW |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:11* | *Reserved* | - | - |
| 10:7 | BLANK | VCx blank window source selection<br>● 0b0000: No blank window<br>● 0bxxx1: TIM0 OCREFB<br>● 0bxx1x: TIM3 OCREF0B<br>● 0bx1xx: TIM3 OCREF1B<br>● 0b1xxx: TIM3 OCREF2B | RW |
| 6 | WINDOW | Window comparison function configuration<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 5 | TIMBK | VCx filtered signal output to the TIM0/1/2/3 brake enable control<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 4 | TIM3CLR | VCx filtered signal output to the TIM3 REFCLR enable control<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 3 | TIM2CLR | VCx filtered signal output to the TIM2 REFCLR enable control<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 2 | TIM1CLR | VCx filtered signal output to the TIM1 REFCLR enable control<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 1 | TIM0CLR | VCx filtered signal output to the TIM0 REFCLR enable control<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 0 | POL | VCx_OUTP signal polarity configuration<br>● 0b0: VCx_OUTP is high when the positive terminal is greater than the negative terminal<br>● 0b1: VCx_OUTP is low when the positive terminal is greater than the negative terminal | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 24.4.5 Status Register (VCx_SR) (x=0-1)

Offset: 0x0C (VC0_SR), 0x1C (VC1_SR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| - | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | FLTV | INTF |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | RO | RW |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:2* | *Reserved* | - | - |
| 1 | FLTV | Filtered signal level<br>● 0b0: Low level<br>● 0b1: High level | RO |
| 0 | INTF | Interrupt flag<br>Write 0 to clear the flag, write 1 has no effect<br>Read:<br>● 0b0: No interrupt occurred<br>● 0b1: Interrupt has occurred | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 24.4.6 Offset Trim Register (VC_TRIM)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x20 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 (Peripheral reset is invalid) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | TRIM1 | | | | Reserved | | | | TRIM0 | | | |
| Access | | | | | | | | | | | | | | | | | | | | | RW | | | | | | | | RW | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:12* | *Reserved* | - | - |
| 11:8 | TRIM1 | VC1 Trim Value<br>2 sets of trim values are stored in Flash. Select the appropriate Flash address based on the actual situation, read the trim value stored in that address, and then write it into VC_TRIM.TRIM1.<br>● When the reference voltage at the positive terminal or negative terminal of the VC is less than or equal to 1/2 AVCC: 0x00101BC2<br>● When the reference voltage at the positive terminal or negative terminal of the VC is greater than 1/2 AVCC: 0x00101BC3 | RW |
| *8:4* | *Reserved* | - | - |
| 3:0 | TRIM0 | VC0 Trim Value<br>2 sets of trim values are stored in Flash. Select the appropriate Flash address based on the actual situation, read the trim value stored in that address, and then write it into VC_TRIM.TRIM0.<br>● When the reference voltage at the positive terminal or negative terminal of the VC is less than or equal to 1/2 AVCC: 0x00101BC0<br>● When the reference voltage at the positive terminal or negative terminal of the VC is greater than 1/2 AVCC: 0x00101BC1 | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

# 25 LVD

## 25.1 Introduction

LVD can be used to monitor the voltage of DVCC and chip pins. When the comparison result of the monitored voltage and the LVD threshold meets the trigger condition, the LVD will generate an interrupt or reset signal, and the user can perform some urgent tasks according to the signal.

## 25.2 Features

- 4 monitoring sources
- 16-stage threshold voltage, 1.8-4.8V optional
- 8 trigger conditions, combinations of high level, rising edge and falling edge
- 2 trigger results: reset and interrupt
- 12-stage filter configuration to prevent false triggering
- With hysteresis function, strong anti-interference

## 25.3 Functional Description

### 25.3.1 LVD Block Diagram

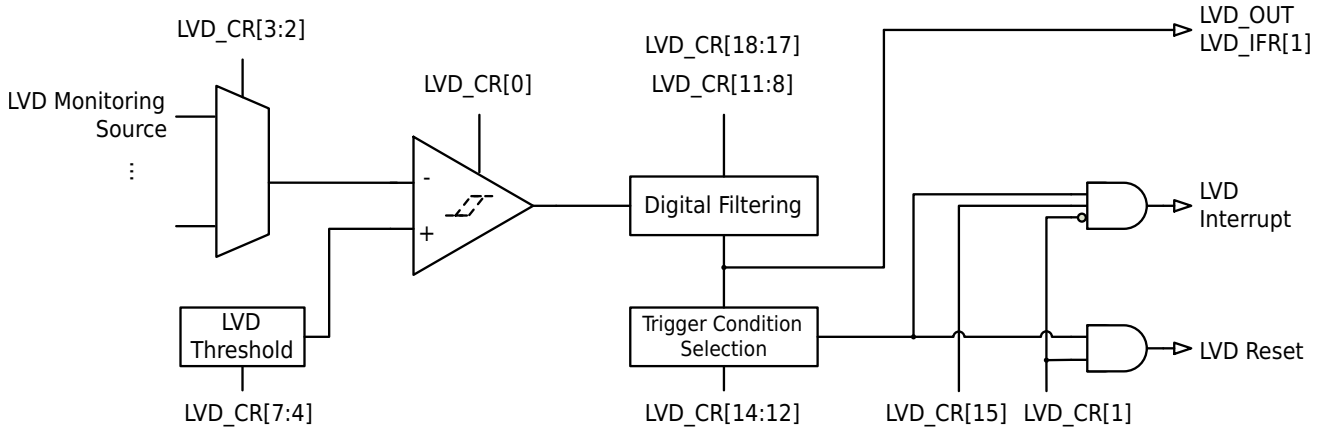



**Figure 25-1 LVD Block Diagram**

**Note**

For the selection of LVD monitoring sources, please refer to the descriptions of the SOURCE bit in "LVD Configuration Register (LVD_CR)".

### 25.3.2 Digital Filtering

If the working environment of the chip is bad, the output of the hysteresis comparator will appear noise signal. The noise signal whose pulse width is less than LVD_CR.DEBOUNCE_TIME in the output waveform of the hysteresis comparator can be filtered out. If the digital filter module is disabled, the input and output signals of the digital filter module are the same. The filter clock used by the digital filter module is a built-in RC oscillator with an oscillation frequency of about 256kHz.

The digitally filtered signal level can be read from the register LVD_IFR[1]; when the GPIO function is configured as LVD_OUT, the digitally filtered signal can be output from the GPIO for easy measurement.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Enable the digital filter module, and select the filter time as 2 filter clock cycles, the filter illustration is as follows:

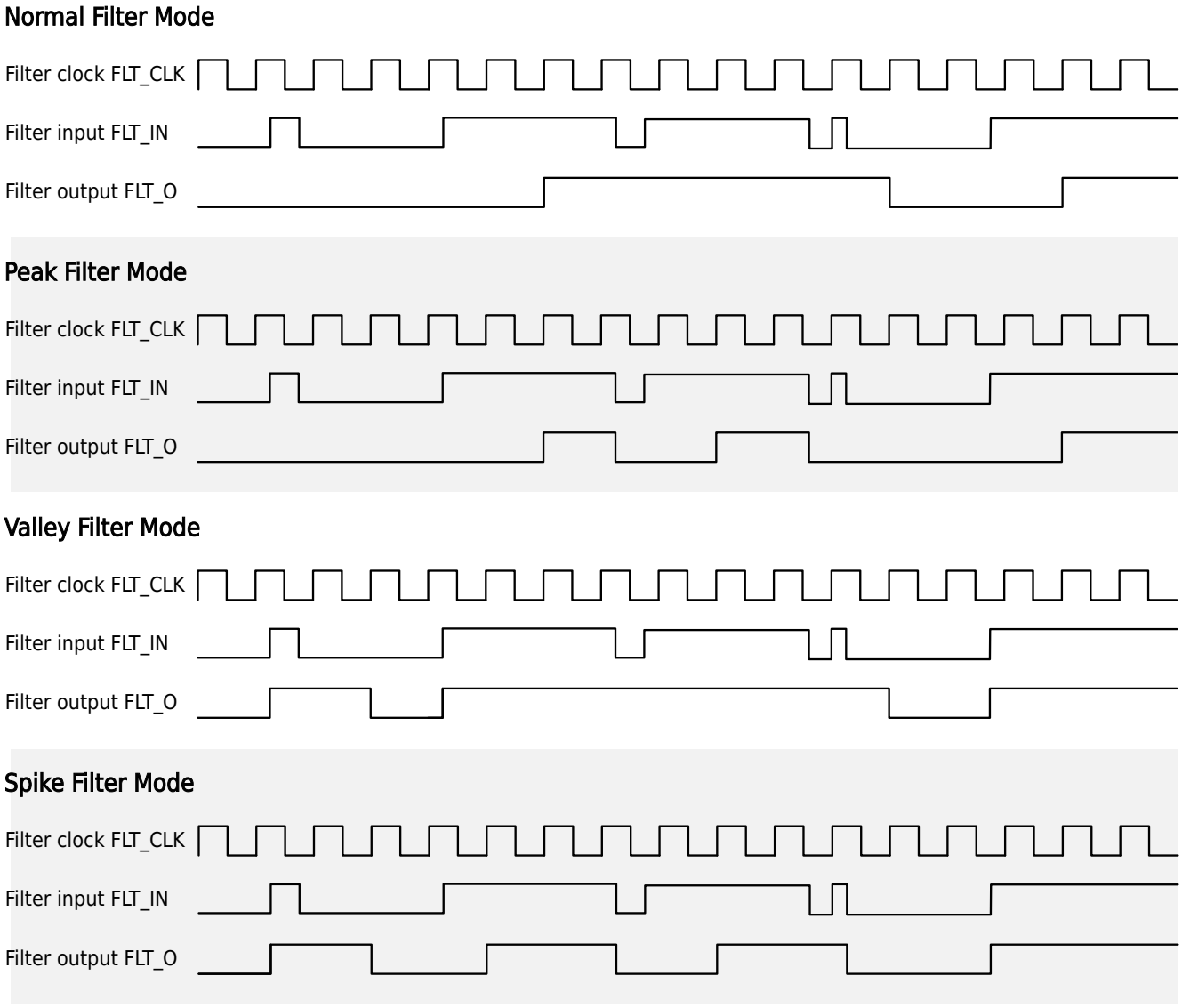



Figure 25-2 LVD Filter Output

### 25.3.3 Hysteresis Function

The built-in voltage comparator of LVD has a hysteresis function, which can enhance the anti-interference ability of the chip. The output signal of LVD will not flip until the input signal is higher or lower than the threshold voltage by the hysteresis voltage/2. The input and output signal diagram of LVD is as follows:


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

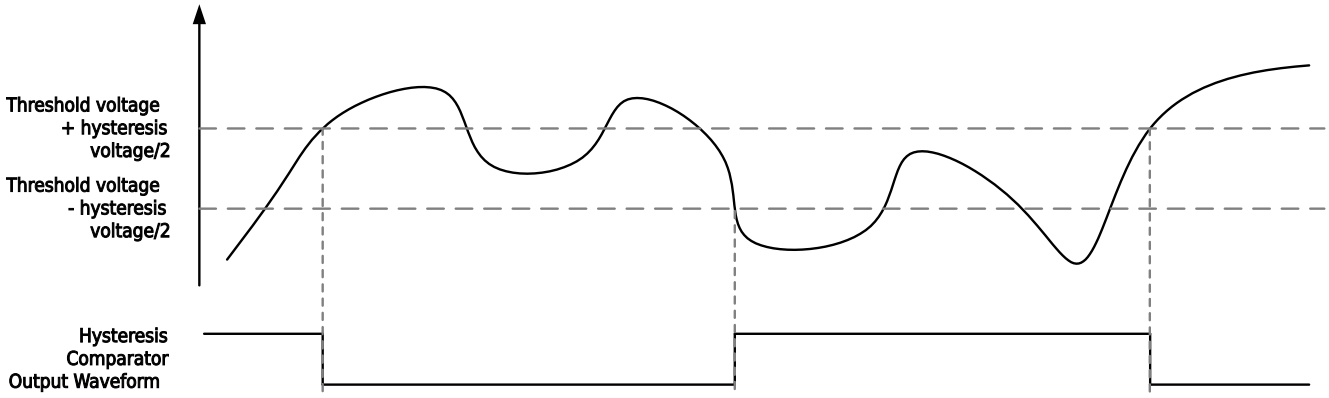



Figure 25-3 LVD Hysteresis Response

## 25.4 Operation Example

### 25.4.1 LVD Configured as Low Voltage Reset

#### Context

In this mode, the MCU is reset when the monitored voltage is lower than the threshold voltage.

#### Procedure

**Step1.** Configure LVD_CR.SOURCE_SEL to select the voltage source to be monitored.

**Step2.** Configure LVD_CR.VTDS to select the threshold voltage of LVD.

**Step3.** Configure LVD_CR.DEBOUNCE_TIME to select the LVD filter time.

**Step4.** Configure LVD_CR.FLT_MODE to select the filter mode.

**Step5.** Set LVD_CR.HTEN to 1, select high level to trigger LVD action.

**Step6.** Set LVD_CR.ACT to 1, select LVD action as reset.

**Step7.** Set LVD_CR.LVDEN to 1 to enable LVD.

### 25.4.2 LVD Configured as Voltage Change Interrupt

#### Context

In this mode, an interrupt is generated when the monitored voltage goes above or below the threshold voltage.

#### Procedure

**Step1.** Configure LVD_CR.SOURCE_SEL to select the voltage source to be monitored.

**Step2.** Configure LVD_CR.VTDS to select the threshold voltage of LVD.

**Step3.** Configure LVD_CR.DEBOUNCE_TIME to select the LVD filter time.

**Step4.** Configure LVD_CR.FLT_MODE to select the filter mode.

**Step5.** Set LVD_CR.RTEN and LVD_CR.FTEN to 1, select level change to trigger LVD action.

**Step6.** Set LVD_CR.ACT to 0, select LVD action as interrupt.

**Step7.** Set LVD_CR.LVDEN to 1 to enable LVD.

**Step8.** Wait for the maximum setup time ($T_{setup}$) plus the configured filter time.

**Step9.** Set LVD_CR.IE to 1 to enable LVD interrupt.

**Step10.** Enable the LVD interrupt in the NVIC interrupt vector table.

**Step11.** Execute the operations required by the user in the LVD interrupt service routine; write 0x00 to LVD_IFR to clear the interrupt flag before exiting the interrupt service routine.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

## 25.5 Register Description

### 25.5.1 Register List

Base address: 0x40002C00

Table 25-1 Low Voltage Detector (LVD) Register Offset

| Offset | Register | Description | Width |
|---|---|---|---|
| 0x000 | LVD_CR | LVD Configuration Register | 32 |
| 0x004 | LVD_IFR | LVD Interrupt Flag Register | 32 |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 25.5.2 LVD Configuration Register (LVD_CR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x000 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | FLT_MODE | | Reserved | | IE | HTEN | RTEN | FTEN | DEBOUNCE_TIME | | | | VTDS | | | SOURCE_SEL | | ACT | LVDEN |
| Access | | | | | | | | | | | | | | | | RW | | | | RW | RW | RW | RW | RW | | | | RW | | | RW | | RW | RW |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:19* | *Reserved* | - | - |
| 18:17 | FLT_MODE | Filter pattern selection<br>● 0b00: Standard Filtering Mode. Filter output toggles only if input remains stable for a defined hold time after transition.<br>● 0b01: Peak Filtering Mode. Output sets to logic 1 only after input holds logic 1 for a stable duration; sets to 0 immediately on input 0. The purpose of this mode is to filter out input peaks that are not long enough.<br>● 0b01: Valley Filtering Mode. Output sets to logic 0 only after input holds logic 0 for a stable duration; sets to 1 immediately on input 1. The purpose of this mode is to filter out input valleys that are not long enough.<br>● 0b11: Glitch Filtering Mode. Output toggles immediately on input transition, then ignores subsequent input changes for a hold time. In this mode, the filter output will closely follow the input, and at the same time, it will prevent the output signal from having spikes. | RW |
| *16* | *Reserved* | - | - |
| 15 | IE | LVD interrupt enable<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 14 | HTEN | High level trigger enable<br>The monitored voltage is lower than the threshold voltage<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 13 | RTEN | Rising edge trigger enable<br>The monitored voltage changes from above the threshold voltage to below the threshold voltage<br>● 0b0: Disable<br>● 0b1: Enable | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 12 | FTEN | Falling edge trigger enable<br>The monitored voltage changes from below the threshold voltage to above the threshold voltage<br>● 0b0: Disable<br>● 0b1: Enable | RW |
| 11:8 | DE-BOUNCE_TIME | Digital filter time configuration<br>N represents filter clock cycles<br>● 0x0: Digital filter disabled<br>● 0x1: N = 2<br>● 0x2: N = 4<br>● 0x3: N = 8<br>● 0x4: N = 16<br>● 0x5: N = 32<br>● 0x6: N = 64<br>● 0x7: N = 128<br>● 0x8: N = 256<br>● 0x9: N = 512<br>● 0xA: N = 1024<br>● 0xB: N = 2048<br>● Other: N = 4096 | RW |
| 7:4 | VTDS | LVD threshold voltage selection<br>● 0b0000: 1.8V<br>● 0b0001: 2.0V<br>● 0b0010: 2.2V<br>● 0b0011: 2.4V<br>● 0b0100: 2.6V<br>● 0b0101: 2.8V<br>● 0b0110: 3.0V<br>● 0b0111: 3.2V<br>● 0b1000: 3.4V<br>● 0b1001: 3.6V<br>● 0b1010: 3.8V<br>● 0b1011: 4.0V<br>● 0b1100: 4.2V<br>● 0b1101: 4.4V<br>● 0b1110: 4.6V<br>● 0b1111: 4.8V | RW |
| 3:2 | SOURCE_SEL | LVD monitoring source selection<br>● 0b00: LVD_IN0 (DVCC voltage)<br>● 0b01: LVD_IN1 (PC13 port input voltage)<br>● 0b10: LVD_IN2 (PB08 port input voltage)<br>● 0b11: LVD_IN3 (PB07 port input voltage) | RW |
| 1 | ACT | LVD trigger action selection<br>● 0b0: NVIC interrupt<br>● 0b1: System reset | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 0 | LVDEN | LVD enable control<br>● 0b0: Disable<br>● 0b1: Enable | RW |

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 25.5.3 LVD Interrupt Flag Register (LVD_IFR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x004 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | FILTER | INTF |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | RO | RW |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:2* | *Reserved* | - | - |
| 1 | FILTER | Level value of digital filter output<br>● 0b0: Digital filter output low level<br>● 0b1: Digital filter output high level | RO |
| 0 | INTF | LVD interrupt flag<br>Write 0 to clear the interrupt flag, write 1 has no effect.<br>Read:<br>● 0b0: No interrupt occurred<br>● 0b1: Interrupt occurred | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

# 26 CRC

## 26.1 Introduction

In many applications, CRC algorithms are required to verify the integrity and correctness of data. Especially in data transmission, CRC checking is widely used. This module can use 6 algorithms, including CRC16/CCITT, CRC16/CCITT-FALSE, CRC16/XMODE, CRC16/X25, CRC32, CRC32/MPEG-2, to calculate and check the data.

## 26.2 Functional Description

### 26.2.1 Function Block Diagram

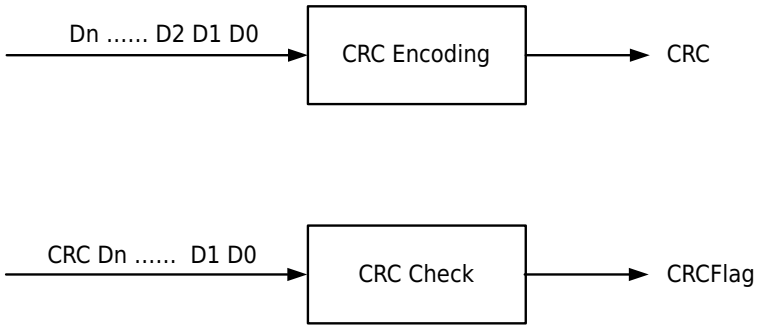



**Figure 26-1 CRC Application Diagram**

### 26.2.2 CRC Parameter Model

The CRC algorithm of this module complies with the definition of ISO/IEC13239, using 16-bit, 32-bit CRC respectively. The corresponding parameter models for each CRC algorithm are shown in the table below.

**Table 26-1 CRC Parameter Model**

| CRC Algorithm Name | Polynomial Formula | POLY | INIT | XOROUT | REFIN | REFOUT | Checksum Value |
|---|---|---|---|---|---|---|---|
| CRC16/X25 | $X^{16}+X^{12}+X^5+1$ | 0x1021 | 0xFFFF | 0xFFFF | TRUE | TRUE | 0xF0B8 |
| CRC16/CCITT | $X^{16}+X^{12}+X^5+1$ | 0x1021 | 0x0000 | 0x0000 | TRUE | TRUE | 0x0000 |
| CRC16/CCITT-FALSE | $X^{16}+X^{12}+X^5+1$ | 0x1021 | 0xFFFF | 0x0000 | FALSE | FALSE | 0x0000 |
| CRC16/XMODEM | $X^{16}+X^{12}+X^5+1$ | 0x1021 | 0x0000 | 0x0000 | FALSE | FALSE | 0x0000 |
| CRC32 | $X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X+1$ | 0x04C11DB7 | 0xFFFFFFFF | 0xFFFFFFFF | TRUE | TRUE | 0xDEBB20E3 |
| CRC32/MPEG-2 | $X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X+1$ | 0x04C11DB7 | 0xFFFFFFFF | 0x00000000 | FALSE | FALSE | 0x00000000 |

- POLY: Hexadecimal polynomial.
- INIT: CRC initial value, consistent with bit width.
- XOROUT: The final CRC code is obtained by performing XOR operation on the calculation result with this parameter, consistent with bit width.
- REFIN: Whether to flip the high and low bits of the original data, TRUE for flip, FALSE for no flip.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

- REFOUT: Whether to flip the high and low bits of the obtained CRC code after encoding ends, TRUE for flip, FALSE for no flip.

When using this module, it is necessary to configure the CR.SEL bit according to the encoding method requirements and write the initial value INIT. Other parameters within the algorithm have been pre-configured.

The functions of this module include:

- CRC encoding and CRC check
- 3 bit width access methods: 8-bit, 16-bit, 32-bit:
  - Example of input data with 8-bit bit width: 0x00, 0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x77;
  - Example of input data with 16-bit bit width: 0x1100, 0x3322, 0x5544, 0x7766;
  - Example of input data with 32-bit bit width: 0x33221100, 0x77665544.

### 26.2.3 CRC16 Encoding Mode

CRC encoding is the process of encoding the original data to calculate its CRC value. The operating procedures are as follows:

1. Write 0b0 to CRC_CR.CR and write corresponding data to CRC_CR.SEL, selecting the corresponding CRC16 specification algorithm.
2. Initialize CRC calculation by writing 0x0000 or 0xFFFF to CRC_RESULT[15:0] according to the CRC parameter model.
3. Organize the original data to be encoded into 8-bit/16-bit/32-bit formats and write them into CRC_DATA register sequentially.

   **Note**
   When writing the CRC value in 8-bit format, write the low bits first and then the high bits.

4. Read CRC_RESULT[15:0] to obtain the 16-bit CRC encoded value.

### 26.2.4 CRC16 Check Mode

The check mode can verify whether the encoded data has been tampered. The operating procedures are as follows:

1. Write 0b0 to CRC_CR.CR and write corresponding data to CRC_CR.SEL, selecting the corresponding CRC16 specification algorithm.
2. Initialize CRC calculation by writing 0x0000 or 0xFFFF to CRC_RESULT[15:0] according to the CRC parameter model.
3. Write the encoded data into CRC_DATA register sequentially in 8-bit, 16-bit, or 32-bit organization.

   **Note**
   When writing the CRC value in 8-bit format, write the low bits first and then the high bits.

4. Write the check code into the CRC_DATA register with a 16-bit bit width.

   **Note**
   For CRC16/CCITT-FALSE and CRC16/XMODEM, the check code needs to be swapped between the high 8 bits and low 8 bits.

   For example: 0x1234-> 0x3412


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

5. Read the value of CRC_CR.FLAG. A value of 1 indicates check success, and a value of 0 indicates check failure.

### 26.2.5 CRC32 Encoding Mode

CRC encoding is the process of encoding the original data to calculate its CRC value. The operating procedures are as follows:

1. Write 0b1 to CRC_CR.CR and write corresponding data to CRC_CR.SEL, selecting the corresponding CRC32 specification algorithm.
2. Write 0xFFFFFFFF to CRC_RESULT[31:0] to initialize CRC calculation.
3. Organize the original data to be encoded into 8-bit/16-bit/32-bit formats and write them into CRC_DATA register sequentially.

   **Note**
   When writing the CRC value in 8-bit format, write the low bits first and then the high bits.

4. Read CRC_RESULT[31:0] to obtain the 32-bit CRC encoded value.

### 26.2.6 CRC32 Check Mode

The check mode can verify whether the encoded data has been tampered with. The operating procedures are as follows:

1. Write 0b1 to CRC_CR.CR and write corresponding data to CRC_CR.SEL, selecting the corresponding CRC32 specification algorithm.
2. Write 0xFFFFFFFF to CRC_RESULT[31:0] to initialize CRC calculation.
3. Write the encoded data into CRC_DATA register sequentially in 8-bit, 16-bit, or 32-bit organization.

   **Note**
   When writing the CRC value in 8-bit format, write the low bits first and then the high bits.

4. Write the check code into the CRC_DATA register in 32-bit organization.

   **Note**
   For CRC32/MPEG-2, during check, the high 16 bits and low 16 bits need to be swapped, and within each 16-bit section, the high 8 bits and low 8 bits also need to be swapped.

   For example: 0x12345678->0x78563412

5. Read the value of CRC_CR.FLAG. A value of 1 indicates check success, and a value of 0 indicates check failure.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

## 26.3 Register Description

### 26.3.1 Register List

Base address: 0x40020800

Table 26-2 CRC Operation (CRC) Register Offset

| Offset | Register | Description | Width |
|---|---|---|---|
| 0x00 | CRC_CR | CRC Control Register | 32 |
| 0x04 | CRC_RESULT | CRC Result Register | 32 |
| 0x80-0xFC | CRC_DATA | CRC Data Register | 32 |

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 26.3.2 Control Register (CRC_CR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000001 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | SEL | | | FLAG | CR |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | | | RW | | | RO | RW |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:5* | *Reserved* | - | - |
| 4:2 | SEL | CRC Selection Bit<br>**When CR=0b0, Bit4-2 correspond to CRC algorithm selection:**<br>● 0b000: CRC16/X25<br>● 0b001: CRC16/CCITT<br>● 0b010: CRC16/CCITT-FALSE<br>● 0b011: CRC16/XMODEM<br>● Others: Setting prohibited<br>**When CR=0b1, Bit4-2 correspond to CRC algorithm selection:**<br>● 0b000: CRC32<br>● 0b001: CRC32/MPEG-2<br>● Others: Setting prohibited | RW |
| 1 | FLAG | Check Result Bit<br>● 0b0: Current check error<br>● 0b1: Current check correct | RO |
| 0 | CR | Operational Control Bit<br>● 0b0: CRC16<br>● 0b1: CRC32 | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 26.3.3 Result Register (CRC_RESULT)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x04 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | RESULT | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Access | RW | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 31:0 | RESULT | Result bit<br>● When CRC16 is selected, use RESULT[15:0].<br>● When CRC32 is selected, use RESULT[31:0]. | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 26.3.4 Data Register (CRC_DATA)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x80-0xFC | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Access | WO | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 31:0 | DATA | Data for operation<br>Write the data to be operated on. | WO |

**Note**

CRC_DATA register is at the 0x80 offset from the CRC base address. The 0x84-0xFC offset from the base address are the mapping addresses of the CRC_DATA register, with the same access and operation permissions as the CRC_DATA register.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

# 27 TRNG

## 27.1 Introduction

The true random number module generates 64-bit true random numbers.

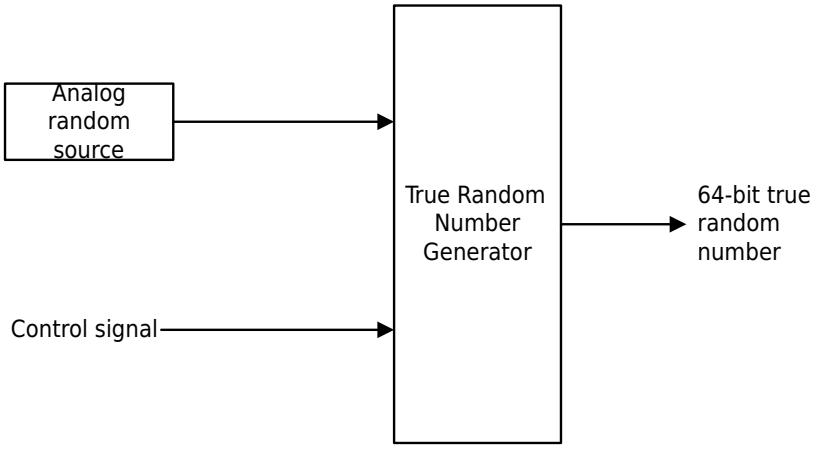



Figure 27-1 True Random Number Generator Function Block Diagram

## 27.2 Functional Description

### 27.2.1 True Random Number Generation Principle

This module uses an internal analog random source, which can generate 64bits true random numbers every time it is started. In addition, software configuration can be performed on the way of true random number generation. For details, refer to the Register Description section. The generated 64-bit true random numbers are stored in DATA0 and DATA1 registers respectively.

## 27.3 Operation Example

### 27.3.1 Generating 64bits True Random Number (The First Time After Power-On)

#### Procedure

**Step1.** Enable the random source circuit: Set TRNG_CR.TRNGCIR_EN to 0b1 to start the random source circuit, which begins outputting serial true random numbers.

**Step2.** Select reload initial value: Set TRNG_MODE.LOAD to 0b1 so that the initial value of newly generated true random numbers is obtained from the random source.

**Step3.** Select direct feedback mode for PRNG64: Set TRNG_MODE.FDBK to 0b1 to XOR the feedback signal with the random source before inputting it into the PRNG.

**Step4.** Select the shift count for PRNG64: Set TRNG_MODE.CNT to 0b110 to choose 256 shifts.

**Step5.** Generate true random numbers: Set TRNG_CR.TRNG_RUN to 0b1, and the hardware performs operations based on the true random number generation configuration. Upon completion, the hardware automatically clears TRNG_CR.TRNG_RUN.

**Step6.** Select not to reload the initial value: Set TRNG_MODE.LOAD to 0b0.

**Step7.** Select direct feedback mode for PRNG64: Set TRNG_MODE.FDBK to 0b0 to directly input the feedback signal into the PRNG.

**Step8.** Select the shift count for PRNG64: Set TRNG_MODE.CNT to 0b100 to choose 64 shifts.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

**Step9.** Generate true random numbers: Set TRNG_CR.TRNG_RUN to 0b1, and the hardware performs operations based on the true random number generation configuration. Upon completion, the hardware automatically clears TRNG_CR.TRNG_RUN.

**Step10.** Read true random numbers: After detecting that TRNG_CR.TRNG_RUN has changed to "0," the software reads TRNG_DATA0 and TRNG_DATA1 to obtain a 64-bit true random number.

**Step11.** After completing the generation of true random numbers, it is recommended to disable the random source circuit to save power: Set TRNG_CR.TRNGCIR_EN to 0b0 to turn off the random source circuit.

### 27.3.2 Generating 64bits True Random Number (Not Generated for the First Time After Power-on)

#### Procedure

**Step1.** Enable the random source circuit: Set TRNG_CR.TRNGCIR_EN to 0b1 to start the random source circuit, which begins outputting serial true random numbers.

**Step2.** Select not to reload the initial value: Set TRNG_MODE.LOAD to 0b0.

**Step3.** Select direct feedback mode for PRNG64: Set TRNG_MODE.FDBK to 0b1 to XOR the feedback signal with the random source before inputting it into the PRNG.

**Step4.** Select the shift count for PRNG64: Set TRNG_MODE.CNT to 0b110 to choose 256 shifts.

**Step5.** Generate true random numbers: Set TRNG_CR.TRNG_RUN to 0b1, and the hardware performs operations based on the true random number generation configuration. Upon completion, the hardware automatically clears TRNG_CR.TRNG_RUN.

**Step6.** Select direct feedback mode for PRNG64: Set TRNG_MODE.FDBK to 0b0 to directly input the feedback signal into the PRNG.

**Step7.** Select the shift count for PRNG64: Set TRNG_MODE.CNT to 0b100 to choose 64 shifts.

**Step8.** Read true random numbers: After detecting that TRNG_CR.TRNG_RUN has changed to "0," the software reads TRNG_DATA0 and TRNG_DATA1 to obtain a 64-bit true random number.
If new true random numbers need to be generated continuously, proceed to step 2 until the requirements are met.

**Step9.** After completing the generation of true random numbers, it is recommended to disable the random source circuit to save power: Set TRNG_CR.TRNGCIR_EN to 0b0 to turn off the random source circuit.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

## 27.4 Register Description

### 27.4.1 Register List

Base address: 0x40008C00

Table 27-1 True Random Number Generator (TRNG) Register Offset

| Offset | Register | Description | Width |
|---|---|---|---|
| 0x00 | TRNG_CR | Control register | 32 |
| 0x04 | TRNG_MODE | Mode Register | 32 |
| 0x0C | TRNG_DATA0 | Data Register 0 | 32 |
| 0x10 | TRNG_DATA1 | Data Register 1 | 32 |

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 27.4.2 Control Register (TRNG_CR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | TRNG_RUN | TRNGCIR_EN |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | RW | RW |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:2* | *Reserved* | - | - |
| 1 | TRNG_RUN | True random number generation control<br>The software writes "1" and starts to generate a new 64bits random number; after the operation is completed, the hardware is cleared;<br>● 0b0: The random number generation is completed<br>● 0b1: Write 1 to start random number generation, read 1 to indicate that random number is being generated | RW |
| 0 | TRNGCIR_EN | Random source circuit enable<br>● 0b0: Turn off the random source<br>● 0b1: Turn on the random source | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 27.4.3 Mode Register (TRNG_MODE)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x04 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000010 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | CNT | | | FDBK | LOAD |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | | | RW | | | RW | RW |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:5* | *Reserved* | - | - |
| 4:2 | CNT | Feedback shift times of 64-bit TRNG<br>● 0b000: Shift 0 times (that is, output the sampling value of the random source)<br>● 0b001: Shift 8 times<br>● 0b010: Shift 16 times<br>● 0b011: Shift 32 times<br>● 0b100: Shift 64 times<br>● 0b101: Shift 128 times<br>● 0b110: Shift 256 times<br>● 0b111: Setting prohibited | RW |
| 1 | FDBK | During the shift operation, whether the feedback signal of the 64-bit TRNG is XORed with the random source<br>● 0b0: Do not perform XOR operation<br>● 0b1: Perform XOR operation | RW |
| 0 | LOAD | When generating a new random number, whether the 64-bit TRNG obtains a new initial value from the random source<br>● 0b0: Do not load new initial values (generating pseudo-random numbers)<br>● 0b1: Load a new initial value (generating true random numbers) | RW |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 27.4.4 Data Register 0 (TRNG_DATA0)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x0C | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0xXXXXXXXX | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | DATA0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Access | RO | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 31:0 | DATA0 | Low 32-bit random number | RO |

### 27.4.5 Data Register 1 (TRNG_DATA1)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x10 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0xXXXXXXXX | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | DATA1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Access | RO | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 31:0 | DATA1 | High 32-bit random number | RO |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

# 28 AES

## 28.1 Overview

AES (The Advanced Encryption Standard) is a new data encryption standard officially announced by the National Institute of Standards and Technology (NIST) on October 2, 2000.

The block size of AES is fixed at 128 bits, while the key length supports 128, 192, and 256 bits. For encryption, the input is a plaintext block and a key, and the output is a ciphertext block; for decryption, the input is a ciphertext block and a key, and the output is a plaintext block. The process is as follows:

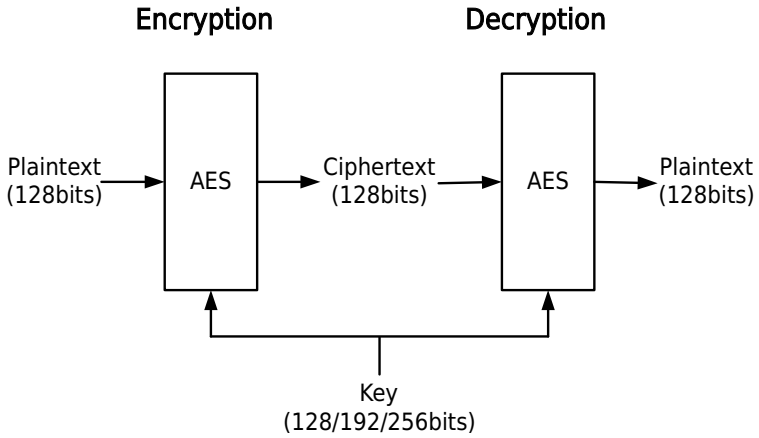



Figure 28-1 AES Encryption and Decryption Schematic Diagram

The basic unit of processing in the AES algorithm is the byte. A 128-bit message is divided into 16 bytes, which are copied into a 4×4 matrix called the state. All transformations in AES are based on transformations of the state matrix, which holds the intermediate results of the computation.

AES is a key-iterated block cipher that involves repeated applications of a round function to the state. The round transformations in AES consist of four operations: SubBytes, ShiftRows, MixColumns, and AddRoundKey. Specifically, SubBytes involves finding the modular inverse of each byte in GF($2^8$) and an affine transformation; ShiftRows is a byte substitution that cyclically shifts the rows in the state by different offsets; MixColumns performs a linear transformation on each column of the state; and AddRoundKey performs a bitwise XOR operation between each byte in the state and the round key. The encryption process of AES is as follows:

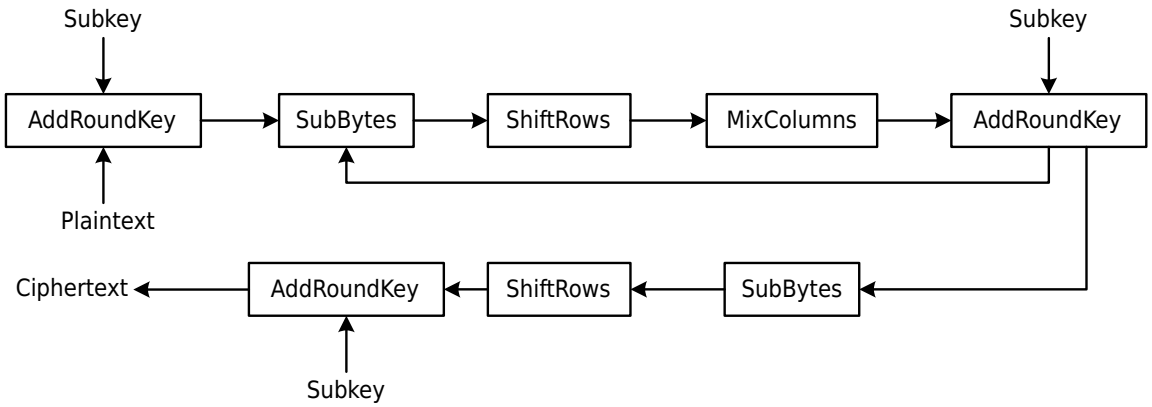



Figure 28-2 AES Encryption Flowchart


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

The subkeys used in the diagram need to be derived from the initial key expansion, and the key expansion process is synchronized with the encryption process.

Since the plaintext is fixed at 128 bits, the number of rounds in the encryption process depends on the length of the key. For example, when the key is 128 bits, there are 10 rounds; when the key is 192 bits, there are 12 rounds; and when the key is 256 bits, there are 14 rounds. Except for the last round, which lacks the MixColumns transformation, all other rounds perform the complete round transformation operations.

The decryption process differs from the encryption process. First, all key expansions must be completed, and the decryption process starts from the last round subkey of the expansion and proceeds backward. Then, the four operations of the round transformation become their corresponding inverse operations: InvSubBytes, InvShiftRows, InvMixColumns, and AddRoundKey. The modular inverse operation in InvSubBytes remains unchanged, but the affine transformation is changed to inverse transformation; InvShiftRows and InvMixColumns become corresponding inverse transformations; AddRoundKey remains unchanged.

In the direct decryption process, the call order of the four operations in the round transformation is: InvShiftRows, InvSubBytes, AddRoundKey, InvMixColumns, which is inconsistent with the call order in the encryption process, but the keys used are the same as in the encryption process. In the equivalent decryption process, the call order of the four operations in the round transformation is: InvSubBytes, InvShiftRows, InvMixColumns, AddRoundKey, which is exactly the same as the call order in the encryption process, except that the subkey of each round needs to undergo the InvMixColumns operation.

For detailed algorithm descriptions, refer to the standard "FIPS PUB 197".

## 28.2 Features

- The execution results of the encryption and decryption processes of the AES algorithm standard fully comply with the description of the algorithm principles in "FIPS PUB 197".
- Supports 128-bit, 192-bit and 256-bit keys

## 28.3 Functional Description

### 28.3.1 Exception Mechanism

- Only 32-bit access is supported. Access with other bit widths will cause a system exception and enter the hardware exception interrupt.
- Accessing the AES module with an offset greater than or equal to 0x80 will cause a system exception and enter the hardware exception interrupt.

### 28.3.2 Runtime Instructions

The time required for the module to complete an operation (AES_CR.START written to 1) to the end of the operation (AES_CR.START restored to 0) is shown in the table below:

**Table 28-1 AES Encryption and Decryption Running Time**

| Operation | 128-Bit Key | 192-Bit Key | 256-Bit Key |
|---|---|---|---|
| Encryption | 220 cycles | 260 cycles | 300 cycles |
| Decryption | 290 cycles | 332 cycles | 398 cycles |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

## 28.4 Operation Example

This module has two functions: encryption and decryption. The operations of these two functions share some common characteristics. Below, we will first introduce their common points, and then separately introduce the standard operating procedures for each function.

### 28.4.1 Common Points of Encryption and Decryption Operations

1. During AES encryption and decryption, the data register will change. If the data to be operated on next time is the result of the current operation, there is no need to rewrite the data.
2. Supports 128-bit, 192-bit, and 256-bit keys. A 128-bit key is written to the offset 0x20-0x2C, a 192-bit key is written to the offset 0x20-0x34, and a 256-bit key is written to the offset 0x20-0x3C.
3. Method to determine the end of the module's operation: Continuously read AES_CR.START. If its value becomes 0, it indicates the end of the operation.

### 28.4.2 Encryption Operation Procedures

#### Procedure

**Step1.** Write the 128-bit data to be encrypted into the data register (AES_DATA).

**Step2.** Write the encryption key into the key register (AES_KEY).

**Step3.** Set AES_CR.KEYSIZE according to the key length.

**Step4.** Set AES_CR.MODE to 0 to start encryption mode.

**Step5.** Write 1 to AES_CR.START in the control register to start the module's operation. Steps3-5 can be performed simultaneously.

**Step6.** Query AES_CR.START and wait for AES_CR.START to revert to 0, indicating the end of the module's operation.

**Step7.** Read the data register (AES_DATA) to obtain the 128-bit ciphertext.

### 28.4.3 Decryption Operation Procedures

#### Procedure

**Step1.** Write 128-bit data to be decrypted into the data register (AES_DATA).

**Step2.** Write the decryption key into the key register (AES_KEY).

**Step3.** Set AES_CR.KEYSIZE according to the key length.

**Step4.** Set AES_CR.MODE to 1 to enable decryption mode.

**Step5.** Write 1 to AES_CR.START in the control register to start the module's operation. Steps3-5 can be performed simultaneously.

**Step6.** Query AES_CR.START and wait for AES_CR.START to revert to 0, indicating the end of the module's operation.

**Step7.** Read the data register (AES_DATA) to obtain the 128-bit plaintext.

### 28.4.4 Data Example

#### 128-Bit Key

128-bit plaintext: 0xFFEEDDCCBBAA99887766554433221100

128-bit key: 0x0F0E0D0C0B0A09080706050403020100

128-bit ciphertext: 0x5AC5B47080B7CDD830047B6AD8E0C469


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Table 28-2 128-Bit Operation Register Example

| Before Encryption | | | |
|---|---|---|---|
| Register | Value (Key) | Register | Value (Plaintext) |
| AES_KEY0 | 0x03020100 | AES_DATA0 | 0x33221100 |
| AES_KEY1 | 0x07060504 | AES_DATA1 | 0x77665544 |
| AES_KEY2 | 0x0B0A0908 | AES_DATA2 | 0xBBAA9988 |
| AES_KEY3 | 0x0F0E0D0C | AES_DATA3 | 0xFFEEDDCC |
| After Encryption | | | |
| Register | Value (Key) | Register | Value (Ciphertext) |
| AES_KEY0 | 0x03020100 | AES_DATA0 | 0xD8E0C469 |
| AES_KEY1 | 0x07060504 | AES_DATA1 | 0x30047B6A |
| AES_KEY2 | 0x0B0A0908 | AES_DATA2 | 0x80B7CDD8 |
| AES_KEY3 | 0x0F0E0D0C | AES_DATA3 | 0x5AC5B470 |

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 192-Bit Key

128-bit plaintext: 0xFFEEDDCCBBAA99887766554433221100

192-bit key: 0x17161514131211100F0E0D0C0B0A09080706050403020100

128-bit ciphertext: 0x91710DECA070AF6EE0DF4C86A47CA9DD

**Table 28-3 192-Bit Operation Register Example**

| Before Encryption | | | |
|---|---|---|---|
| **Register** | **Value (Key)** | **Register** | **Value (Plaintext)** |
| AES_KEY0 | 0x03020100 | AES_DATA0 | 0x33221100 |
| AES_KEY1 | 0x07060504 | AES_DATA1 | 0x77665544 |
| AES_KEY2 | 0x0B0A0908 | AES_DATA2 | 0xBBAA9988 |
| AES_KEY3 | 0x0F0E0D0C | AES_DATA3 | 0xFFEEDDCC |
| AES_KEY4 | 0x13121110 | | |
| AES_KEY5 | 0x17161514 | | |
| **After Encryption** | | | |
| **Register** | **Value (Key)** | **Register** | **Value (Ciphertext)** |
| AES_KEY0 | 0x03020100 | AES_DATA0 | 0xA47CA9DD |
| AES_KEY1 | 0x07060504 | AES_DATA1 | 0xE0DF4C86 |
| AES_KEY2 | 0x0B0A0908 | AES_DATA2 | 0xA070AF6E |
| AES_KEY3 | 0x0F0E0D0C | AES_DATA3 | 0x91710DEC |
| AES_KEY4 | 0x13121110 | | |
| AES_KEY5 | 0x17161514 | | |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 256-Bit Key

128-bit plaintext: 0xFFEEDDCCBBAA99887766554433221100

256-bit key: 0x1F1E1D1C1B1A191817161514131211100F0E0D0C0B0A09080706050403020100

128-bit ciphertext: 0x8960494B9049FCAEBF456751CAB7A28E

**Table 28-4 256-Bit Operation Register Example**

| Before Encryption | | | |
|---|---|---|---|
| **Register** | **Value (Key)** | **Register** | **Value (Plaintext)** |
| AES_KEY0 | 0x03020100 | AES_DATA0 | 0x33221100 |
| AES_KEY1 | 0x07060504 | AES_DATA1 | 0x77665544 |
| AES_KEY2 | 0x0B0A0908 | AES_DATA2 | 0xBBAA9988 |
| AES_KEY3 | 0x0F0E0D0C | AES_DATA3 | 0xFFEEDDCC |
| AES_KEY4 | 0x13121110 | | |
| AES_KEY5 | 0x17161514 | | |
| AES_KEY6 | 0x1B1A1918 | | |
| AES_KEY7 | 0x1F1E1D1C | | |
| **After Encryption** | | | |
| **Register** | **Value (Key)** | **Register** | **Value (Ciphertext)** |
| AES_KEY0 | 0x03020100 | AES_DATA0 | 0xCAB7A28E |
| AES_KEY1 | 0x07060504 | AES_DATA1 | 0xBF456751 |
| AES_KEY2 | 0x0B0A0908 | AES_DATA2 | 0x9049FCAE |
| AES_KEY3 | 0x0F0E0D0C | AES_DATA3 | 0x8960494B |
| AES_KEY4 | 0x13121110 | | |
| AES_KEY5 | 0x17161514 | | |
| AES_KEY6 | 0x1B1A1918 | | |
| AES_KEY7 | 0x1F1E1D1C | | |

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

## 28.5 Register Description

### 28.5.1 Register List

Base address: 0x40021400

Table 28-5 Advanced Encryption Standard Module (AES) Register Offset

| Offset | Register | Description | Width |
|---|---|---|---|
| 0x00 | AES_CR | Control Register | 32 |
| 0x10+0x4*n | AES_DATAn | Data Register n (n=0-3) | 32 |
| 0x20+0x4*n | AES_KEYn | Key Register n (n=0-7) | 32 |
| 0x40 | AES_CRDM | Control Register Mapping Address | 32 |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 28.5.2 Control Register (AES_CR)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | KEYSIZE | | Reserved | MODE | START |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | | | RW | | | RW | RW |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:5* | *Reserved* | - | - |
| 4:3 | KEYSIZE | Key length<br>● 0b00: Key length is 128 bits<br>● 0b01: Key length is 192 bits<br>● 0b10: Key length is 256 bits<br>● 0b11: Key length is 128 bits | RW |
| *2* | *Reserved* | - | - |
| 1 | MODE | Operation mode<br>● 0b0: Encryption operation<br>● 0b1: Decryption operation | RW |
| 0 | START | Operation start control<br>● 0b0: Operation ended or not started<br>● 0b1: Start operation | RW |

**Note**

- The operation method for the AES_CR.START bit is: After software writes 1 to this bit, the module will start running. After the current operation ends, the module's hardware will automatically clear this bit to 0. When software queries this bit and finds it to be 0, it indicates that the current operation is complete.
- Writes to this register can only be performed when the module is not in an operation state (for example, when AES_CR.START=0). Otherwise, the hardware will automatically ignore the write operation. Read operations are not subject to this restriction.
- The AES_CR register bit is located at the AES base address with an offset of 0. The mapping address for the AES_CR register is at an offset of 0x40 from the base address, with the same access and operation permissions as the AES_CR register.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 28.5.3 Data Register (AES_DATAn) (n=0-3)

Offset: 0x10+0x4*n

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| - | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Access | RW | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 31:0 | DATA | Plaintext/ciphertext data<br>Stores 128-bit plaintext/ciphertext for the AES algorithm | RW |

The data register consists of four 32-bit registers that form a 128-bit data used to store plaintext to be encrypted or ciphertext to be decrypted before the module's operation, and to store the encrypted ciphertext or decrypted plaintext after the operation is completed.

| Encryption Operation | | Decryption Operation | |
|---|---|---|---|
| Before Operation | After Operation | Before Operation | After Operation |
| 128-bit Plaintext | 128-bit Ciphertext | 128-bit Ciphertext | 128-bit Plaintext |

The four 32-bit registers are concatenated to form a 128-bit data. Read and write operations require separate operations on each of the four registers. The fill-in data for the data register is as follows (using the data 0xFFEEDDCCBBAA99887766554433221100 as an example):

**Table 28-11 Fill-in data for Register**

| AES_DATA0 | AES_DATA1 | AES_DATA2 | AES_DATA3 |
|---|---|---|---|
| 0x33221100 | 0x77665544 | 0xBBAA9988 | 0xFFEEDDCC |

**Note**

- Writes to this register can only be performed when the module is not in an operation state (for example, when AES_CR.START=0). Otherwise, the hardware will automatically ignore writes to this register.
- Reads from this register can only be performed when the module is not in an operation state (for example, when AES_CR.START=0). Otherwise, reads from this register will result in all zeros.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 28.5.4 Key Register (AES_KEYn) (n=0-7)

Offset: 0x20+0x4*n

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| - | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | KEY | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Access | RW | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 31:0 | KEY | AES key<br>Stores 128 (KEY0-KEY3), 192 (KEY0-KEY5), 256 (KEY0-KEY7)-bit keys for the AES algorithm | RW |

The key register consists of eight 32-bit registers that store the input initial key. Write operations require separate operations on each of the eight 32-bit registers. The corresponding fill-in data for register is shown in the examples below.

**Example of 128-bit key**: 0x0F0E0D0C_0B0A0908_07060504_03020100

**Table 28-14 Fill-in data for Register - 128-bit key**

| AES_KEY0 | AES_KEY1 | AES_KEY2 | AES_KEY3 |
|---|---|---|---|
| 0x03020100 | 0x07060504 | 0x0B0A0908 | 0x0F0E0D0C |

**Example of 192-bit key**: 0x17161514_13121110_0F0E0D0C_0B0A0908_07060504_03020100

**Table 28-15 Fill-in data for Register - 192-bit key**

| AES_KEY0 | AES_KEY1 | AES_KEY2 | AES_KEY3 | AES_KEY4 | AES_KEY5 |
|---|---|---|---|---|---|
| 0x03020100 | 0x07060504 | 0x0B0A0908 | 0x0F0E0D0C | 0x13121110 | 0x17161514 |

**Example of 256-bit key**:
0x1F1E1D1C_1B1A1918_17161514_13121110_0F0E0D0C_0B0A0908_07060504_03020100

**Table 28-16 Fill-in data for Register - 256-bit key**

| AES_KEY0 | AES_KEY1 | AES_KEY2 | AES_KEY3 | AES_KEY4 | AES_KEY5 | AES_KEY6 | AES_KEY7 |
|---|---|---|---|---|---|---|---|
| 0x03020100 | 0x07060504 | 0x0B0A0908 | 0x0F0E0D0C | 0x13121110 | 0x17161514 | 0x1B1A1918 | 0x1F1E1D1C |

**Note**

- Writes to this register can only be performed when the module is not in an operation state (for example, when AES_CR.START=0). Otherwise, the hardware will automatically ignore writes to this register.
- Reads from this register can only be performed when the module is not in an operation state (for example, when AES_CR.START=0). Otherwise, reads from this register will result in all zeros.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 28.5.5 Control Register Mapping Address (AES_CRDM)

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x40 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | KEYSIZE | | Reserved | MODE | START |
| Access | | | | | | | | | | | | | | | | | | | | | | | | | | | | RW | | | RW | RW |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| *31:5* | *Reserved* | - | - |
| 4:3 | KEYSIZE | Key length<br>● 0b00: Key length is 128 bits<br>● 0b01: Key length is 192 bits<br>● 0b10: Key length is 256 bits<br>● 0b11: Key length is 128 bits | RW |
| *2* | *Reserved* | - | - |
| 1 | MODE | Operation mode<br>● 0b0: Encryption operation<br>● 0b1: Decryption operation | RW |
| 0 | START | Operation start control<br>● 0b0: Operation ended or not started<br>● 0b1: Start operation | RW |

**Note**

- The operation method for the AES_CR.START bit is: After software writes 1 to this bit, the module will start running. After the current operation ends, the module's hardware will automatically clear this bit to 0. When software queries this bit and finds it to be 0, it indicates that the current operation is complete.
- Writes to this register can only be performed when the module is not in an operation state (for example, when AES_CR.START=0). Otherwise, the hardware will automatically ignore the write operation. Read operations are not subject to this restriction.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

# 29 SWD Debug Interface

## 29.1 Introduction

This series of products uses an ARM Cortex-M0+ core, which features a hardware debugging module SWD that supports complex debugging operations. The hardware debug module allows the core to stop when fetching instructions (instruction breakpoints) or accessing data (data breakpoints). When the core is stopped, both the internal state of the core and the external state of the system can be queried in the IDE. After the query is complete, the core and peripherals can be restored and program execution continues. When this series of MCUs are connected to a debugger and start debugging, the debugger will use the hardware debugging module of the core to perform debugging operations.

**Note**

SWD can wake up DeepSleep, please be careful not to disconnect the SWD interface as the state of Deep-Sleep.

## 29.2 Functional Description

### 29.2.1 Debug System Diagram

This product uses an ARM Cortex-M0+ core and includes hardware extensions for advanced debugging function; therefore, the debugging function of this product are consistent with Cortex-M0+. Debug extensions allow the core to halt the core when fetching instructions (instruction breakpoints) or fetching accessing data (data breakpoints). When the core stops, you can query the internal state of the core and the external state of the system. After querying, the core and system are resumed, and program execution is restored.

When the debug host is connected to the MCU and debugged, the debug function will be used.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

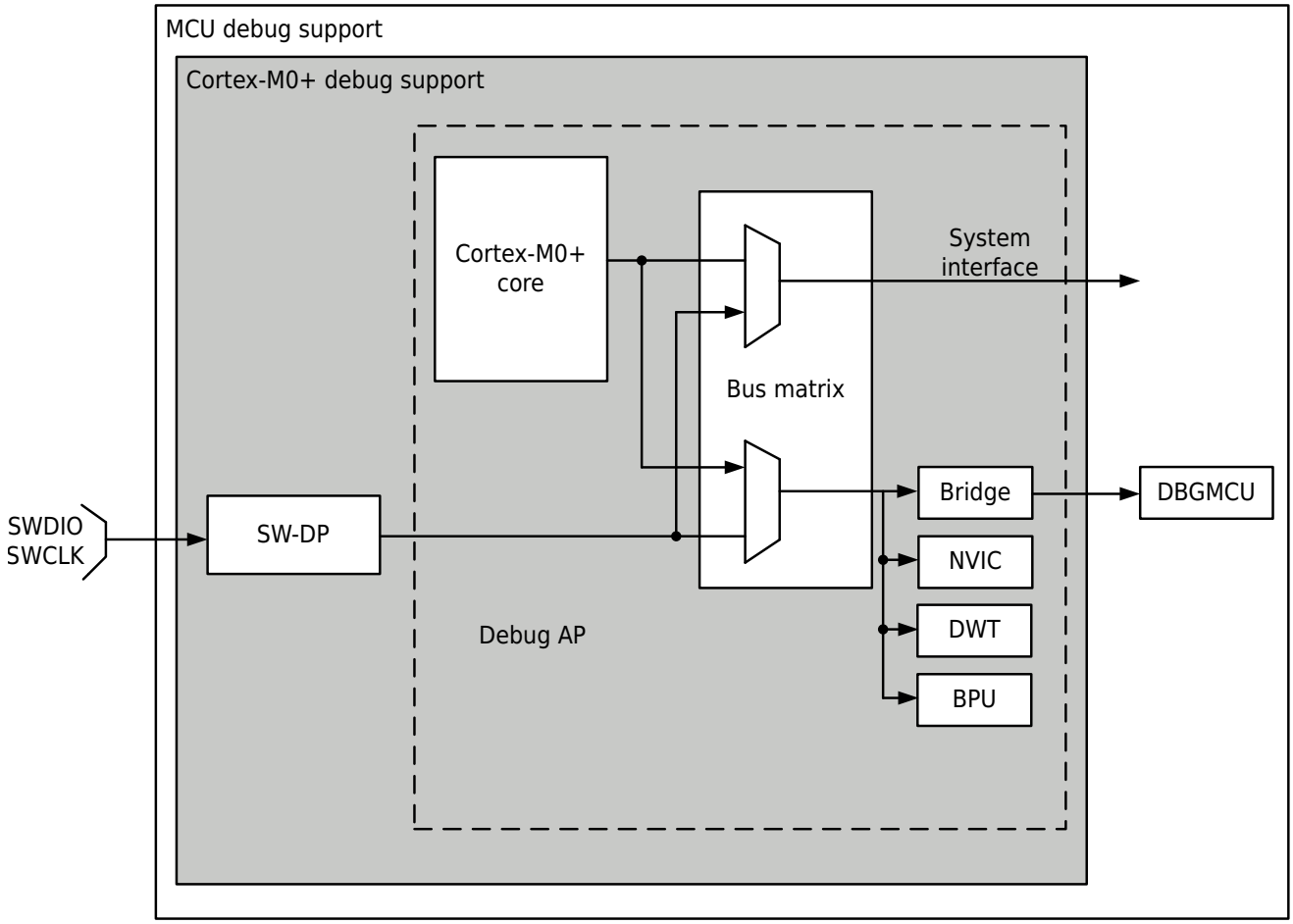



Figure 29-1 Debug Support Block Diagram

The built-in debugging function in the Cortex-M0+ core are part of the ARM® CoreSight design suite.

ARM® Cortex-M0+ core provides integrated on-chip debugging support, including:

- SW-DP: Serial line
- BPU: Breakpoint unit
- DWT: Data watchpoint trigger

**Note**

For detailed information on the debugging function supported by the ARM® Cortex-M0+ core, see "Cortex-M0+ Technical Reference Manual".

## 29.2.2 ARM® Reference Documentation

- Cortex-M0+ Technical Reference Manual (TRM)
  Available at developer.arm.com.
- ARM® debug interface V5
- ARM® CoreSight Design Suite Version r1p1 Technical Reference Manual
- ARMv6-M Architecture Reference Manual

## 29.2.3 Debug Port Pins

### SWD Port Pins

The SWD interface of this series requires 2 pins, as shown in the table below.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| SWD Port Name | Debugging Function | Pin Assignment |
|---|---|---|
| SWCLK | Serial clock | PA14 |
| SWDIO | Serial data input/output | PA13 |

#### SW-DP Pin Assignment

If the [ Encryption Chip] option is enabled when programming the program, the SWD debugging function will be disabled after power-on. If the [Encryption Chip] option is not enabled during programming, the PA13/PA14 pins will be initialized as dedicated pins for debugger use after power-on. Users can set SYSCTRL1.SWD_USE_IO register to disable the SWD pin debugging function, and the SWD pin will be released to be used as a normal GPIO. The configuration and function of SWD pins are summarized in the following table:

| [Encryption Chip] Option | SWD_USE_IO Configuration | PA13/PA14 Function |
|---|---|---|
| Encryption | 0 | - |
| Encryption | 1 | GPIO |
| No encryption | 0 | SWD |
| No encryption | 1 | GPIO |

#### Internal Pull-Up on SWD Pins

After the user software releases the SW I/O, the GPIO controller takes control of these pins. The reset state of the GPIO control register will put the I/O into the equivalent state:

- SWDIO: input pull-up
- SWCLK: input pull-up

No external resistors need to be added due to built-in pull-up and pull-down resistors.

### 29.2.4 SWD Port

#### 29.2.4.1 Introduction to SWD Protocol

This synchronous serial protocol uses two pins:

- SWCLK: Clock from host to target
- SWDIO: Bidirectional

Using this protocol, two register sets (DPACC register set and APACC register set) can be read and written simultaneously. LSB is ahead when transferring data.

For SWDIO bidirectional management, the line must be pulled up on the circuit board (ARM® recommends using 100k). These pull-up resistors can be configured internally, so external pull-up resistors are not required.

Every time the SWDIO direction is changed in the protocol, the conversion time is inserted, and the line is not driven by the host or the target. By default, this conversion time is one bit time, but can be adjusted by configuring the SWCLK frequency.

#### 29.2.4.2 SWD Protocol Sequence

Each sequence consists of three phases:

1. Host transmitted data packet request (8-bit)


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

2. Target transmitted acknowledgment response (3-bit)
3. Data transfer phase transmitted by host or target (33-bit)

Table 29-3 Host Transmitted Data Packet Request

| Bit | Name | Description |
|---|---|---|
| 0 | Start | Must be 0b1 |
| 1 | APnDP | ● 0b0: DP access<br>● 0b1: AP access |
| 2 | RnW | ● 0b0: Write request<br>● 0b1: Read request |
| 4:3 | A[3:2] | Address field of DP or AP register |
| 5 | Parity check | Unit parity for the first few bits |
| 6 | Stop | 0b0 |
| 7 | Reside | Not driven by the host. Due to the presence of the pull-up, the target must read it as 0b1. |

For detailed information on the DPACC and APACC registers, please refer to "Cortex-M0+ Technical Reference Manual".

The packet request is always followed by a conversion time (1 bit by default), at which point neither the host nor the target will drive.

Table 29-4 Target Transmitted Acknowledgment Response

| Bit | Name | Description |
|---|---|---|
| 2:0 | ACK | ● 0b001: FAULT<br>● 0b010: WAIT<br>● 0b100: OK |

The ACK response must be followed by conversion time only when a READ transaction occurs or a WAIT or FAULT acknowledgment is received.

Table 29-5 Data Transfers Transmitted by Host or Target

| Bit | Name | Description |
|---|---|---|
| 31:0 | WDATA or RDATA | Write or read data |
| 32 | Parity check | Single parity for 32 data bits |

The conversion time must be after the DATA transfer only when a READ transaction occurs.

### 29.2.4.3 SW-DP State Machine (Reset, Idle State, ID Code)

The SW-DP state machine includes an internal ID code to identify the SW-DP. This code complies with the JEP-106 standard and is set to the default ARM® code **0x0BC11477** (which corresponds to the Cortex®-M0+).


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

**Note**

The SW-DP state machine is inactive until the target reads the ID code.

- The SW-DP state machine is in reset state after power-on reset or after the line has been at high level for more than 50 cycles.
- If the line is low for at least two cycles after the reset state, the SW-DP state machine is in the idle state.
- After the reset state, the state machine must first enter the idle state and then perform a read access to the DP-SW ID CODE register. FAULT acknowledgment response on another transaction.

For more detailed information about the SW-DP state machine, please refer to "Cortex-M0+ Technical Reference Manual" and "ARM® CoreSight Design Suite Version r1p1 Technical Reference Manual".

#### 29.2.4.4 DP and AP Read/Write Access

- Read access to DP is not delayed: the target response can be sent immediately (if ACK=OK) or delayed (if ACK=WAIT).
- Delay read access to AP. This means that access results will be returned on the next transfer. If the next access to be performed is not an AP access, the DP-RDBUFF register must be read to obtain the result.
- Every time an AP read access or RDBUFF read request is made, the READOK flag of the DP-CTRL/STAT register will be updated to know whether the AP read access is successful.
- SW-DP has a write buffer (for DP or AP writes) so that write operations can be accepted even while other operations are still pending. If the write buffer is full, the target acknowledgment response is WAIT. Except for IDCODE read, CTRL/STAT read or ABORT write, these operations will also be accepted when the write buffer is full.
- Due to the asynchronous clock domains SWCLK and HCLK, two additional SWCLK cycles are required after the write operation (after the parity bit) for the write operation to take effect internally. These cycles should be applied while the line is being driven low (idle state).

This is especially important when writing to the CTRL/STAT register to initiate a power-on request. Otherwise the next operation (one that is only valid after the core is powered on) will be executed immediately, which will cause a failure.

#### 29.2.4.5 SW-DP Register

These registers can be accessed when APnDP=0:

| A[3:2] | RW | CTRLSEL Bit of the SELECT Register | Register | Description |
|---|---|---|---|---|
| 0b00 | Read | - | IDCODE | The manufacturer code is set to the default ARM® code for Cortex®-M0+. **0x0BC11477** (identifies SW-DP) |
| 0b00 | Write | - | ABORT | - |
| 0b01 | Read/ write | 0 | DP-CTRL/STAT | Purpose:<br>● Request system or debug power-on<br>● Configure transport operations for AP access<br>● Control compare and verify operations<br>● Read some status flags (overflow and power-on acknowledgment) |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| A[3:2] | RW | CTRLSEL Bit of the SELECT Register | Register | Description |
|---|---|---|---|---|
| 0b01 | Read/ write | 1 | WIRE CON-TROL | Used to configure the physical serial port protocol (such as switching time duration) |
| 0b10 | Read | - | READ RESEND | Allow recovery of reads data from corrupt-ed debug software transfers without re-peating the original AP transmission. |
| 0b10 | Write | - | SELECT | 4-word register window used to select the currently accessed port and active |
| 0b11 | Read/ write | - | READ BUFFER | Since an AP access has been issued, this read buffer is very useful (it provides the result of the AP read request during the execution of the next AP transaction). This read buffer captures data from the AP and displays the result of the previous read without requiring a new operation to be initiated |

#### 29.2.4.6 SW-AP Register

These registers can be accessed when APnDP=1:

There are multiple AP registers, which are addressed in the following combinations:

- Shift value A[3:2]
- Current value of DP SELECT register

| Address | A[3:2] Value | Description |
|---|---|---|
| 0x0 | 0b00 | Reserved, must be kept at the reset value. |
| 0x4 | 0b01 | DP CTRL/STAT register. Used for:<br>● Request system or debug power-on<br>● Configure transport operations for AP access<br>● Control compare and verify operations<br>● Read some status flags (overflow and power-on ac-knowledgment) |
| 0x8 | 0b10 | DP SELECT register: used to select the currently accessed port and active 4-word register window.<br>● Bit[31:24]: APSEL, select the current AP<br>● Bit[23:8]: Reserved<br>● Bit[7:4]: APBANKSEL, select the active 4-word register window within the current AP<br>● Bit[3:0]: Reserved |
| 0xC | 0b11 | DP RDBUFF Register: Used by the debugger to retrieve the final result after performing a series of operations (without initiating a new JTAG-DP request). |

### 29.2.5 Core Debugging

Debug the core through the core debug registers. Debug access to these registers is through the debug access port. It consists of four registers:


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

| Register | Description |
|---|---|
| DHCSR | 32-bit debug stop control and status register<br>This register provides information about the state of the processor, enables the core to enter a debug stop state, and provides processor stepping capabilities. |
| DCRSR | 17-bit debug core register selector registers<br>This register selects the processor register to be read or written. |
| DCRDR | 32-bit debug core register data registers<br>This register holds data read and written between the register and the processor selected by the DCRSR (selector) register. |
| DEMCR | 32 -bit debug exception and monitor control registers<br>This register provides vector capture and debug monitor control. |

These registers are not reset on system reset. They can only be reset by a power-on reset. For more information, see "Cortex-M0+ Technical Reference Manual".

In order to put the core into the debug stop state immediately after reset, it is necessary to:

- Enable Debug and Exception Monitoring Control Register bit 0 (VC_CORRESET)
- Enable Debug Halt Control and Status Register bit 0 (C_DEBUGEN)

### 29.2.6 BPU (Breakpoint Unit)

Cortex-M0+ BPU implementation provides 4 breakpoint registers.

Processor breakpoint implements PC-based breakpoint functionality.

For more information about the BPU CoreSight identification registers, including their addresses and access types, please see "ARMv6-M Architecture Reference Manual" and "ARM® CoreSight Component Technical Reference Manual".

### 29.2.7 DWT (Data Watchpoint)

The Cortex-M0+ DWT implementation provides 2 watchpoint register groups.

#### DWT Function

The processor watchpoint implements data address and PC-based watchpoint functions (i.e., PC sampling registers) and includes comparator address masking. For detailed information, please refer to the "ARMv6-M Architecture Reference Manual".

#### DWT Program Counter Sampling Register

Processors implementing the data watchpoint unit also implement the ARMv6-M optional DWT Program Counter Sampling Register (DWT_PCSR). This register allows the debugger to periodically sample the PC without stopping the processor, enabling coarse-grained analysis. For more information, see "ARMv6-M Architecture Reference Manual".

Cortex-M0+ DWT_PCSR records instructions that pass condition codes as well as instructions that do not pass condition codes.

### 29.2.8 MCU Debug Component (DBG)

MCU debug component helps the debugger provide support for:

- Low power mode
- Timer and watchdog clock control during breakpoints


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### Debug Support for Low Power Mode

To enter low power mode, the instruction WFI or WFE must be executed.

The MCU supports several low power modes that disable the CPU clock or reduce CPU power consumption.

The core does not allow FCLK or HCLK to be turned off during a debugging session. They must remain active as they are required for debug connections during debugging. The MCU incorporates special methods that allow users to debug software in low power mode.

### Debug Support for Timer and Watchdog

During a breakpoint, the behavior of the counters of the timer and watchdog must be selected:

- The counter continues to count while the breakpoint is generated. This method is required when using PWM to control motors.
- The counter stops to count while the breakpoint is generated. This method is required when using the watchdog.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

# 30 Device Electronic Signature

## 30.1 Introduction

The electronic signature is stored in the system storage area of the flash memory module and can be read by SWD or CPU. The chip identification information is programmed during manufacturing. User firmware or external devices can read the electronic signature to automatically match MCUs with different configurations.

## 30.2 Functional Description

### 30.2.1 Product Unique Identifier (UID)

Typical application scenarios for unique identifier (UID):

- Used as a serial number
- UID is used as a security key when used in conjunction with software cryptographic primitives and protocols to increase the security of code in Flash prior to programming the internal Flash
- Activation of the secure bootstrap process, etc.

The UID provides a reference number that is unique to any device and in any context. The user can never change these bits. The UID can also be read in different ways like single byte/halfword/word and then concatenated using a custom algorithm.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

## 30.3 Register Description

### 30.3.1 Product Unique Identifier (UID) Register (80 bits)

Base address: 0x00101E8C

| Offset | Description | UID Bits (80bits) | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | Y Coordinate on the wafer | UID[7:0] | | | | | | | |
| 1 | X Coordinate on the wafer | UID[15:8] | | | | | | | |
| 2 | Wafer Number | UID[23:16] | | | | | | | |
| 3 | Lot Number | UID[31:24] | | | | | | | |
| 4 | | UID[39:32] | | | | | | | |
| 5 | | UID[47:40] | | | | | | | |
| 6 | | UID[55:48] | | | | | | | |
| 7 | | UID[63:56] | | | | | | | |
| 8 | | UID[71:64] | | | | | | | |
| 9 | | UID[79:72] | | | | | | | |

### 30.3.2 Product Model Register

0x00101B40-0x00101B4F store the ASCII code representing the product model. If the product model is less than 16 bytes, fill it with 0x00.

Example: The product model represented by 484333324C3132364B41544200000000 is HC32L126KATB.


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 30.3.3 FLASH Capacity Register

Base address: 0x00101B60

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| - | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | - | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | FLASHSIZE | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Access | RO | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 31:0 | FLASHSIZE | Product built-in Flash capacity<br>In unit of Byte<br>0x00020000 means the Flash capacity is 128K Byte | RO |

### 30.3.4 RAM Capacity Register

Base address: 0x00101B64

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| - | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | - | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | RAMSIZE | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Access | RO | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 31:0 | RAMSIZE | Product built-in RAM capacity<br>In unit of Byte<br>0x00004000 means the RAM capacity is 16K Byte | RO |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

### 30.3.5 Pin Count/Flash Sector Size Configuration Register

Base address: 0x00101B68

| Offset | Bit Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| - | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset | - | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | PINCOUNT | | | | | | | | | | | | | | | | FLASHSECTORSIZE | | | | | | | | | | | | | | | |
| Access | RO | | | | | | | | | | | | | | | | RO | | | | | | | | | | | | | | | |

| Bit | Name | Functional Description | Access |
|---|---|---|---|
| 31:16 | PINCOUNT | Product pins count<br>0x0040 means the number of product pins is 64<br>..... | RO |
| 15:0 | FLASHSEC-TORSIZE | Flash sector size<br>In unit of Byte<br>0x0200 represents 512Byte. | RO |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

# A Glossary

| Terminology | Full Name | description |
|---|---|---|
| Byte | - | Bytes, 8-bit data |
| Half Word | - | Half word, 16-bit data |
| Word | - | Words, 32-bit data |
| Double Word | - | Double word, 64 -bit data. |
| IAP | - | It refers to reprogram the MCU's Flash during the running of a user program (programming in the application). |
| ICP | - | It means that the MCU's Flash of the MCU can be programmed by using JTAG protocol, SWD protocol or bootstrap program when the device is installed on the user application circuit board (in-circuit/online programming). |
| AHB | Advanced High-performance Bus | Advanced High-Performance Bus |
| APB | Advanced Peripheral Bus | Advanced peripheral bus |
| DMA | Direct Memory Access | Direct memory access |
| TIM/TMR | - | Timer |

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

# Revision History

| Revision/Date | Changes |
|---|---|
| Rev1.20<br>Aug. 28, 2026 | 1. RTC: In the "Timestamp Time Register (RTC_TSTR)" chapter, add note "This bit uses 24-hour format and is not affected by RTC_CR0.AMPM" to HOURH/HOURL bit.<br>2. LPUART: In the "Multi-Machine Communication" chapter, the description of Slave Address Configuration is optimized.<br>3. TIM0/1/2/3: In the "Mode2/3 Compare Capture Mode - Capture Input" chapter, the capture input description of CH0B is optimized. When MSCR.IB0S is configured to 1, MSCR.TS does not support selecting the CH0B filter signal.<br>4. ADC: The built-in signal follower function is added.<br>5. SWD Debug Interface: The internal ID of SW-DP is modified to "0x0BC11477". |
| Rev1.10<br>Feb. 12, 2026 | 1. SYSCTRL: In the "Example of Switching from Other Clocks to XTH" chapter, the software delay waiting time in Step 7 is removed; in the "XTH Control Register (XTH_CR)" chapter, the software delay waiting time for the STABLE bit is removed. |
| Rev1.00<br>Sep. 12, 2025 | First official release. |


All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.