XIAOHUA SEMICONDUCTOR

HC32F052 Series

32-bit ARM® Cortex®-M0+ Microcontroller

Reference Manual

Rev1.50

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) 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 HC32F052 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 26
2 Storage and Architectures 27
2.1 Overview 27
2.2 System Address Division 28
2.3 Memory and Peripheral Address Allocation 29
3 Operating Mode 31
3.1 Overview 31
3.2 Working Mode Switching 31
3.3 Active Mode 32
3.4 Sleep Mode 32
3.5 DeepSleep Mode 32
3.6 Functional/Module Working Status 33
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 38
4.2 System Clock Switching 39
4.2.1 System Clock Switching Introduction 39
4.2.2 Switching Process 39
4.2.3 Examples of Switching 40
4.2.4 Clock Switching Diagram 42
4.3 On-Chip Peripheral Clock Control 42
4.4 Interrupt Wakeup Control 42
4.4.1 Executing Interrupt Service after Waking from DeepSleep mode 43
4.4.2 Interrupt Service Not Executed after Waking from DeepSleep mode 43
4.4.3 Using Sleep-on-exit Feature 44
4.5 Reset Control 44
4.5.1 Reset Controller Overview 44
4.5.2 Reset Function 45
4.6 Register Description 46
4.6.1 Register List 46
4.6.2 System Control Register 0 (SYSCTRL_CR0) 47
4.6.3 System Control Register 1 (SYSCTRL_CR1) 48
4.6.4 System Control Register 2 (SYSCTRL_CR2) 50
4.6.5 System Control Register 3 (SYSCTRL_CR3) 52
4.6.6 RCH Control Register (SYSCTRL_RCHCR) 53
4.6.7 XTH Control Register (SYSCTRL_XTHCR) 54
4.6.8 RCL Control Register (SYSCTRL_RCLCR) 56

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

4.6.9 XTL Control Register (SYSCTRL_XTLCR)....57
4.6.10 PLL Control Register (SYSCTRL_PLLCR)....58
4.6.11 RCH Trim Value Register (SYSCTRL_RCHTRIM)....60
4.6.12 RCL Trim Value Register (SYSCTRL_RCLTRIM)....61
4.6.13 Clock Status Register (SYSCTRL_STATE)....62
4.6.14 Peripheral Module Clock Control Register 0 (SYSCTRL_PERI_CLKEN0)....64
4.6.15 Peripheral Module Clock Control Register 1 (SYSCTRL_PERI_CLKEN1)....67
4.6.16 Peripheral Module Reset Control Register 0 (SYSCTRL_PERI_RESET0)....68
4.6.17 Peripheral Module Reset Control Register 1 (SYSCTRL_PERI_RESET1)....71
4.6.18 Reset Flag Register (SYSCTRL_RSTFLAG)....72
4.6.19 Peripheral Debug Function Enable Register (SYSCTRL_DBG_STOP)....74
4.6.20 GPIO Auxiliary Control Register 0 (SYSCTRL_IOAUX0)....76
4.6.21 GPIO Auxiliary Control Register 1 (SYSCTRL_IOAUX1)....77
4.6.22 GPIO Auxiliary Control Register 2 (SYSCTRL_IOAUX2)....78
5 NVIC....79
5.1 Introduction....79
5.2 Functional Description....79
5.2.1 Interrupt Priority....79
5.2.2 Interrupt Digraph....80
5.2.3 Interrupt Input and Pending Behavior....81
5.2.4 Interrupt Waiting....83
5.2.5 Interrupt Source....83
5.2.6 Interrupt Structure Diagram....85
5.3 Operation Example....87
5.3.1 External Interrupt Enable....87
5.3.2 NVIC Interrupt Enable and Clear Enable....87
5.3.3 NVIC Interrupt Pending and Clear Pending....87
5.3.4 NVIC Interrupt Priority....87
5.3.5 NVIC Interrupt Mask....87
5.4 Register Description....89
5.4.1 Register List....89
5.4.2 Interrupt Enable Set Register (NVIC_ISER)....90
5.4.3 Interrupt Enable Clear Register (NVIC_ICER)....90
5.4.4 Interrupt Pending State Set Register (NVIC_ISPR)....91
5.4.5 Interrupt Pending State Clear Register (NVIC_ICPR)....91
5.4.6 Interrupt Priority Register (NVIC_IPR0)....92
5.4.7 Interrupt Priority Register (NVIC_IPR1)....92
5.4.8 Interrupt Priority Register (NVIC_IPR2)....93
5.4.9 Interrupt Priority Register (NVIC_IPR3)....93
5.4.10 Interrupt Priority Register (NVIC_IPR4)....94
5.4.11 Interrupt Priority Register (NVIC_IPR5)....94
5.4.12 Interrupt Priority Register (NVIC_IPR6)....95
5.4.13 Interrupt Priority Register (NVIC_IPR7)....95
5.4.14 Interrupt Mask Special Register (PRIMASK)....96
6 GPIO....97

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

6.1 Introduction..... 97
6.2 Features..... 97
6.3 Functional Description..... 98
6.3.1 Port Circuit Block Diagram..... 98
6.3.2 Port Mode Configuration..... 98
6.3.3 Port Reset Status..... 99
6.3.4 Port Analog Function..... 99
6.3.5 Port General Input/Output Function..... 99
6.3.6 Port Multiplexing Function..... 102
6.3.7 Port Clock Output..... 104
6.3.8 Port External Interrupt..... 104
6.4 Register Description..... 106
6.4.1 Register List..... 106
6.4.2 Port Digital-to-Analog Configuration Register (GPIO_ADS)..... 107
6.4.3 Port Input/Output Direction Configuration Register (GPIO_DIR)..... 108
6.4.4 Port Output Type Register (GPIO_OD)..... 109
6.4.5 Port Output Driving Capability Register (GPIO_DR)..... 109
6.4.6 Port Pull-Up Pull-Down Register (GPIO_PUPD)..... 110
6.4.7 Port Data Input Register (GPIO_IN)..... 110
6.4.8 Port Data Output Register (GPIO_OUT)..... 112
6.4.9 Port Reset Register (GPIO_BRR)..... 113
6.4.10 Port Set Reset Register (GPIO_BSRR)..... 114
6.4.11 Port Multiplexing Function Low Register (GPIO_AFRL)..... 115
6.4.12 Port Multiplexing Function High Register (GPIO_AFRH)..... 116
6.4.13 Port High Level Interrupt Enable Configuration Register (GPIO_HIE)..... 117
6.4.14 Port Low Level Interrupt Enable Configuration Register (GPIO_LIE)..... 118
6.4.15 Port Rising Edge Interrupt Enable Configuration Register (GPIO_RIE)..... 119
6.4.16 Port Falling Edge Interrupt Enable Configuration Register (GPIO_FIE)..... 120
6.4.17 Port Interrupt Status Register (GPIO_IFR)..... 121
6.4.18 Port Interrupt Clear Register (GPIO_ICR)..... 121
7 I2C..... 122
7.1 Introduction..... 122
7.2 Features..... 122
7.3 Protocol Description..... 122
7.3.1 Data Transmission on the I2C Bus..... 122
7.3.2 Acknowledgment Signal on the I2C Bus..... 123
7.3.3 Arbitration on the I2C Bus..... 124
7.4 Functional Description..... 125
7.4.1 Function Block Diagram..... 126
7.4.2 Serial Clock Generator..... 126
7.4.3 Input Filter..... 127
7.4.4 Address Comparator..... 127
7.4.5 Acknowledgement Flag Bit..... 127
7.4.6 Interrupt Generator..... 127
7.4.7 Operating Mode..... 127

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

7.4.8 Status Code Expression....132
7.5 Operation Example....134
7.5.1 Master Sending Example....134
7.5.2 Master Receiving Example....134
7.5.3 Slave Sending Example....135
7.5.4 Slave Receiving Example....136
7.6 Register Description....137
7.6.1 Register List....137
7.6.2 Baud Rate Generator Enable Register (I2C_TMRUN)....138
7.6.3 Baud Rate Counter Configuration Register (I2C_TM)....138
7.6.4 Configuration Register (I2C_CR)....139
7.6.5 Data Register (I2C_DATA)....141
7.6.6 Status Register (I2C_STAT)....141
7.6.7 Slave Address Register 0 (I2C_ADDR0)....142
7.6.8 Slave Address Register 1 (I2C_ADDR1)....142
7.6.9 Slave Address Register 2 (I2C_ADDR2)....143
7.6.10 Slave Address Match Register (I2C_MATCH)....144
8 SPI....145
8.1 Introduction....145
8.2 Features....145
8.3 Functional Description....145
8.3.1 Function Block Diagram....145
8.3.2 Communication Format and Timing....146
8.3.3 Slave Select Pin Configuration....147
8.3.4 Full-duplex Communication....148
8.3.5 Single-wire Half-duplex Communication....149
8.3.6 Simplex Communication....149
8.3.7 Multi-machine Communication....150
8.3.8 Master Mode Delayed Sampling....151
8.3.9 Status Flags and Error Flags....152
8.3.10 Interrupt Generation and Clearing....155
8.4 Operation Example....155
8.4.1 Example of Sending and Receiving Data in Full-duplex by the Master....155
8.4.2 Example of Sending and Receiving Data in Full-duplex by the Slave....156
8.4.3 Single-wire Half-duplex Sending and Receiving Data Example....156
8.5 Register Description....158
8.5.1 Register List....158
8.5.2 SPI Control Register 0 (SPI_CR0)....159
8.5.3 SPI Control Register 1 (SPI_CR1)....161
8.5.4 SPI Single-wire Half-duplex Output Enable Register (SPI_HDOE)....163
8.5.5 SPI Internal Slave Select Register (SPI_SSI)....163
8.5.6 SPI Status Register (SPI_SR)....164
8.5.7 SPI Interrupt Clear Register (SPI_ICR)....166
8.5.8 SPI Data Register (SPI_DR)....167
9 CTRIM....168

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

9.1 Introduction....168
9.2 Features....168
9.3 Functional Description....168
9.3.1 RCH Clock Calibration Mode....168
9.3.2 RCL Clock Calibration Mode....169
9.3.3 Low Power Timer Mode....170
9.4 Operation Example....171
9.4.1 RCH Clock Automatic Calibration Example....171
9.4.2 RCL Clock Automatic Calibration Example....172
9.5 Register Description....173
9.5.1 Register List....173
9.5.2 Autoload Register (CTRIM_ARR)....174
9.5.3 Error Counter Register (CTRIM_CNT)....174
9.5.4 Control Register 0 (CTRIM_CR0)....175
9.5.5 Control Register 1 (CTRIM_CR1)....177
9.5.6 Interrupt Enable Register (CTRIM_IER)....179
9.5.7 Interrupt and Status Register (CTRIM_ISR)....180
9.5.8 Interrupt Status Flag Register (CTRIM_ICR)....181
9.5.9 Error Result Register (CTRIM_FCAP)....182
9.5.10 TRIM Result Register (CTRIM_TVAL)....183
9.5.11 Error Upper Limit Register (CTRIM_FLIM)....183
10 FLASH....184
10.1 Introduction....184
10.2 Functional Description....184
10.2.1 Capacity Division....184
10.2.2 Read Wait Period....184
10.2.3 Flash Security Protection....184
10.3 Operation Example....186
10.3.1 Sector Erase....186
10.3.2 Chip Erase....186
10.3.3 Program....187
10.3.4 Cont-Program....189
10.3.5 Read....191
10.4 Register Description....192
10.4.1 Register List....192
10.4.2 Control Register (FLASH_CR)....193
10.4.3 Waiting Period Setting Register (FLASH_WAIT)....194
10.4.4 LOCK Status Register (FLASH_LOCK_ST)....194
10.4.5 Interrupt Enable Register (FLASH_IER)....195
10.4.6 Interrupt Flag Register (FLASH_IFR)....196
10.4.7 Interrupt Flag Clear Register (FLASH_ICR)....197
10.4.8 BYPASS Sequence Register (FLASH_BYPASS)....197
10.4.9 Sector Erase/Write Protection Register 0 (FLASH_SLOCK0)....198
10.4.10 Sector Erase/Write Protection Register 1 (FLASH_SLOCK1)....199
11 RAM....200

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

11.1 Introduction..... 200
11.2 Functional Description..... 200
11.2.1 RAM Address Range..... 200
11.2.2 Read and Write Bit Width..... 200
11.2.3 Parity Check..... 200
11.3 Register Description..... 201
11.3.1 Register List..... 201
11.3.2 Control Register (RAM_CR)..... 202
11.3.3 Parity Check Error Address Register (RAM_ERRADDR)..... 202
11.3.4 Error Interrupt Flag Register (RAM_IFR)..... 203
11.3.5 Error Interrupt Flag Clear Register (RAM_ICLR)..... 203
12 DMAC..... 204
12.1 Introduction..... 204
12.2 Features..... 204
12.3 Functional Description..... 205
12.3.1 Function Block Diagram..... 205
12.3.2 DMAC Transfer Mode..... 206
12.3.3 DMA Transfer Pause..... 209
12.3.4 Other Configurations of DMA..... 210
12.3.5 DMA Interrupt..... 211
12.4 Register Description..... 212
12.4.1 Register List..... 212
12.4.2 All Channel Configuration Register (DMAC_CONF)..... 213
12.4.3 Channel x Configuration A Register (DMAC_CONFAx) (x=0-4)..... 214
12.4.4 Channel x Configuration B Register (DMAC_CONFBx) (x=0-4)..... 216
12.4.5 Channel x Transfer Source Address Register (DMAC_SRCADRx) (x=0-4)..... 219
12.4.6 Channel x Transfer Destination Address Register (DMAC_DSTADRx) (x=0-4)..... 219
13 ATIM..... 220
13.1 Introduction..... 220
13.2 Features..... 220
13.2.1 ATIM0/1/2 Features..... 220
13.2.2 ATIM3 Features..... 221
13.3 Functional Description..... 222
13.3.1 Timer Clock..... 222
13.3.2 Timer Counter..... 222
13.3.3 Timer Prescaler..... 222
13.3.4 Mode 0 Count Timer Function..... 222
13.3.5 Mode 1 Pulse Width Measurement PWC..... 225
13.3.6 Mode2/3 Compare Capture Mode..... 228
13.3.7 Mode2/3 Slave Mode..... 246
13.3.8 Quadrature Encoder Counting Function..... 248
13.3.9 Timer Triggers ADC..... 250
13.3.10 Cycle-By-Cycle Current Limiting and Brake Control..... 251
13.3.11 Interrupt..... 252
13.3.12 Timer Interconnection..... 252

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

13.3.13 Input Interconnection....253
13.4 Operation Example....253
13.4.1 Mode0 Counting Timer Function Configuration Example....253
13.4.2 Mode1 Pulse Width Measurement PWC Configuration Example....254
13.4.3 Mode2/3 Compare Capture Mode Configuration Example....254
13.4.4 Complementary PWM Output B Compare as ADC Trigger Function....256
13.5 Register Description....257
13.5.1 Register List....257
13.5.2 Mode0 Count Timer Register....259
13.5.3 Mode1 Pulse Width Measurement PWC Register....265
13.5.4 Mode2/3 Compare Capture Mode Register....272
14 CTIM....297
14.1 Introduction....297
14.2 Features....298
14.2.1 BTIM Features....298
14.2.2 GTIM Features....298
14.3 Functional Description....299
14.3.1 BTIM Functional Description....299
14.3.2 GTIM Functional Description....304
14.4 Register Description....312
14.4.1 Register List....312
14.4.2 BTIM Register....314
14.4.3 GTIM Register....320
15 LPTIM....330
15.1 Introduction....330
15.2 Features....330
15.3 Functional Description....330
15.3.1 Function Block Diagram....330
15.3.2 Counting Unit....331
15.3.3 Counting Clock Configuration....331
15.3.4 Internal Clock Timing Function....331
15.3.5 External Pin Counting Function....332
15.3.6 Gating Function....332
15.4 Register Description....333
15.4.1 Register List....333
15.4.2 Counter Count Value Register (LPTIM_CNT)....334
15.4.3 Reload Register (LPTIM_ARR)....334
15.4.4 Control Register (LPTIM_CR)....335
15.4.5 Interrupt Flag Register (LPTIM_IFR)....336
15.4.6 Interrupt Flag Clear Register (LPTIM_ICLR)....337
16 RTC....338
16.1 Introduction....338
16.2 Features....338
16.3 Functional Description....339
16.3.1 Function Block Diagram....340

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

16.3.2 Power-on Setting....340
16.3.3 System Low Power Mode Switching....340
16.3.4 Clock Error Compensation....341
16.3.5 Timestamp Function....342
16.3.6 Intrusion Detection....342
16.3.7 Wake-up Timer....342
16.3.8 RTC Interrupt....342
16.4 Operation Example....343
16.4.1 RTC Count Start Setting....343
16.4.2 Read Count Register....343
16.4.3 Write Count Register....344
16.4.4 Alarm Clock Setting....344
16.4.5 1Hz Output....344
16.5 Register Description....346
16.5.1 Register List....346
16.5.2 Control Register 0 (RTC_CR0)....347
16.5.3 Control Register 1 (RTC_CR1)....349
16.5.4 Second Count Register (RTC_SEC)....350
16.5.5 Minute Count Register (RTC_MIN)....350
16.5.6 Hour Count Register (RTC_HOUR)....351
16.5.7 Day Count Register (RTC_DAY)....353
16.5.8 Week Count Register (RTC_WEEK)....354
16.5.9 Month Count Register (RTC_MON)....355
16.5.10 Year Count Register (RTC_YEAR)....356
16.5.11 Clock Error Compensation Register (RTC_COMPEN)....357
16.5.12 Control Register 2 (RTC_CR2)....358
16.5.13 Alarm A Register (RTC_ALMA)....359
16.5.14 Alarm B Register (RTC_ALMB)....361
16.5.15 Timestamp Time Register (RTC_TSTR)....363
16.5.16 Timestamp Date Register (RTC_TSDR)....364
16.5.17 Wake-Up Timer Register (RTC_WUTR)....365
16.5.18 Interrupt Enable Register (RTC_IER)....365
16.5.19 Interrupt Flag Register (RTC_IFR)....367
16.5.20 Interrupt Clear Register (RTC_ICR)....368
16.5.21 Protection Register (RTC_WPR)....369
17 IWDT....370
17.1 Introduction....370
17.2 Features....370
17.3 Functional Description....370
17.3.1 Function Block Diagram....370
17.3.2 Window Watchdog Mode....370
17.3.3 Independent Watchdog Mode....371
17.3.4 Watchdog Overflow Handling....371
17.3.5 Watchdog Pause....371
17.3.6 Watchdog Stop and Run Again....371

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

17.3.7 Register Write Protection....371
17.4 Operation Example....371
17.4.1 Configure IWDT as Window Watchdog....371
17.4.2 Configure IWDT as Independent Watchdog....372
17.4.3 Watchdog Timeout Time (32kHz Independent Clock)....372
17.5 Register Description....373
17.5.1 Register List....373
17.5.2 Key Value Register (IWDT_KR)....374
17.5.3 Control Register (IWDT_CR)....375
17.5.4 Reload Register (IWDT_ARR)....376
17.5.5 Status Register (IWDT_SR)....377
17.5.6 Window Register (IWDT_WINR)....378
18 WWDT....379
18.1 Introduction....379
18.2 Features....379
18.3 Functional Description....379
18.3.1 Function Block Diagram....379
18.3.2 Enable Watchdog....380
18.3.3 Configure Watchdog Timeout....380
18.3.4 Update Down Counter....380
18.3.5 Pre-overflow Interrupt....380
18.4 Operation Example....380
18.4.1 Example of Configuring Watchdog Window Overflow Period....380
18.5 Register Description....381
18.5.1 Register List....381
18.5.2 Control Register 0 (WWDT_CR0)....382
18.5.3 Control Register 1 (WWDT_CR1)....383
18.5.4 Status Register (WWDT_SR)....384
19 USART....385
19.1 Introduction....385
19.2 Features....385
19.2.1 Asynchronous Serial Port Features....385
19.2.2 Synchronous Serial Port Features....385
19.2.3 Smart Card Interface Features....385
19.2.4 Infrared Communication Features....385
19.2.5 LIN Features....386
19.3 Functional Description....386
19.3.1 USART General Function....386
19.3.2 Asynchronous Serial Port Mode....395
19.3.3 Synchronous Serial Port Mode....399
19.3.4 Smart Card (7816) Mode....403
19.3.5 Infrared Mode....406
19.3.6 LIN Mode....408
19.4 Operation Example....413
19.4.1 Programming Example of Sending Data in Asynchronous Serial Port Mode....413

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

19.4.2 Programming Example of Receiving Data in Asynchronous Serial Port Mode....413
19.4.3 Infrared Mode 1: Transmission Data Setting....414
19.4.4 Infrared Mode 1: Reception Data Setting....414
19.5 Register Description....415
19.5.1 Register List....415
19.5.2 Control Register 1 (USART_CR1)....416
19.5.3 Control Register 2 (USART_CR2)....420
19.5.4 Control Register 3 (USART_CR3)....423
19.5.5 Baud Rate Register (USART_BRR)....426
19.5.6 Timer Counter Register (USART_TOR)....427
19.5.7 Status Register (USART_SR)....428
19.5.8 Transmit Data Register (USART_TDR)....431
19.5.9 Interrupt Enable Register (USART_IER)....432
19.5.10 Interrupt Flag Clear Register (USART_ICR)....434
19.5.11 Receive Data Register (USART_RDR)....435
19.5.12 Address Register (USART_ADDR)....435
19.5.13 Address Mask Register (USART_ADDMR)....436
19.5.14 Baud Rate Detection Value Register (USART_BDR)....436
19.5.15 Infrared Modulation Register (USART_IR)....437
20 LPUART....438
20.1 Introduction....438
20.2 Features....438
20.3 Functional Description....439
20.3.1 Function Block Diagram....439
20.3.2 Clock Description....439
20.3.3 Working Mode....439
20.3.4 Baud Rate Generation....443
20.3.5 Frame Error Detection....446
20.3.6 Multi-Machine Communication....446
20.3.7 Hardware Flow Control....447
20.3.8 DMA Transmission....448
20.4 Operation Example....449
20.4.1 Transmit Data Example....449
20.4.2 Receive Data Example....449
20.5 Register Description....450
20.5.1 Register List....450
20.5.2 Data Register (LPUART_SBUF)....451
20.5.3 Control Register (LPUART_SCON)....452
20.5.4 Address Register (LPUART_SADDR)....455
20.5.5 Address Mask Register (LPUART_SADEN)....455
20.5.6 Interrupt Flag Bit Register (LPUART_ISR)....456
20.5.7 Interrupt Flag Bit Clear Register (LPUART_ICR)....457
20.5.8 Baud Rate Register (LPUART_SCNT)....457
21 CAN....458
21.1 Introduction....458

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

21.2 Features....458
21.3 Functional Description....459
21.3.1 System Block Diagram....459
21.3.2 Pin Description....459
21.3.3 Operating Mode....459
21.3.4 Baud Rate Setting....460
21.3.5 Transmission Processing....461
21.3.6 Reception Processing....463
21.3.7 Error Handling....464
21.3.8 BUS OFF....465
21.3.9 Arbitration lost Position Capture....465
21.3.10 Time-Triggered TTCAN....465
21.3.11 Upward Compatibility with CAN-FD Function....467
21.3.12 Software Reset Function....467
21.3.13 Interrupt....469
21.4 Register Description....470
21.4.1 Register List....470
21.4.2 CAN Receive Buffer Register (CAN_RBUF)....471
21.4.3 CAN Transmit Buffer Register (CAN_TBUF)....473
21.4.4 CAN Configuration and Status Register (CAN_CFG_STAT)....475
21.4.5 CAN Command Register (CAN_TCMD)....477
21.4.6 CAN Transmit Control Register (CAN_TCTRL)....480
21.4.7 CAN Receive Control Register (CAN_RCTRL)....482
21.4.8 CAN Receive and Transmit Interrupt Enable Register (CAN_RTIE)....483
21.4.9 CAN Receive and Transmit Interrupt Flag Register (CAN_RTIF)....484
21.4.10 CAN Error Interrupt Enable and Flag Register (CAN_ERRINT)....486
21.4.11 CAN Bit Timing Register (CAN_BT)....487
21.4.12 CAN Error and Arbitration Lost Capture Register (CAN_EALCAP)....488
21.4.13 CAN Warning Limits Register (CAN_LIMIT)....489
21.4.14 CAN Receive Error Counter Register (CAN_RECNT)....489
21.4.15 CAN Transmit Error Counter Register (CAN_TECNT)....490
21.4.16 CAN Acceptance Filter Control Register (CAN_ACFCTRL)....491
21.4.17 CAN Acceptance Filter Enable Register (CAN_ACFEN)....492
21.4.18 CAN Acceptance Code and Mask Register (CAN_ACF)....493
21.4.19 TTCAN TB Slot Point Register (CAN_TBSLOT)....494
21.4.20 TTCAN Time Trigger Configuration Register (CAN_TTCFG)....495
21.4.21 TTCAN Reference Message Register (CAN_REF_MSG)....497
21.4.22 TTCAN Trigger Configuration Register (CAN_TRG_CFG)....498
21.4.23 TTCAN Trigger Time Register (CAN_TRG_TRIG)....499
21.4.24 TTCAN Watch Trigger Time Register (CAN_TRG_WTRIG)....499
21.5 Precautions....500
21.5.1 CAN Bus Anti-Interference Measures....500
21.5.2 CAN Controller Noise Constraints....500
22 CRC....501
22.1 Introduction....501

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

22.2 Features....501
22.3 Functional Description....501
22.3.1 Working Mode....501
22.3.2 Encoding Method....501
22.3.3 Write Bit Width....502
22.4 Operation Example....502
22.4.1 CRC-16 Encoding Mode....502
22.4.2 CRC-16 Check Mode....502
22.4.3 CRC-32 Encoding Mode....502
22.4.4 CRC-32 Check Mode....502
22.5 Register Description....503
22.5.1 Register List....503
22.5.2 Control and Status Register (CRC_CSR)....504
22.5.3 Result Register (CRC_RESULT)....504
22.5.4 Data Register (CRC_DATA)....505
23 I2CSLV....506
23.1 Introduction....506
23.2 Features....506
23.3 Protocol Description....506
23.4 Functional Description....507
23.4.1 Function Block Diagram....507
23.4.2 Auto Address Recognition....508
23.4.3 Operating Mode....508
23.4.4 Status Decoding....511
23.4.5 Filtering....513
23.4.6 Sampling Location....513
23.4.7 Edge Shaping Output....513
23.4.8 Timeout Function....513
23.4.9 Device ID Reading....514
23.4.10 Interrupt....514
23.5 Operation Example....514
23.5.1 Master Transmitting and Slave Receiving Data in Non-High-Speed Mode....514
23.5.2 Master Receiving and Slave Transmitting Data in Non-High-Speed Mode....515
23.5.3 Master Transmitting and Slave Receiving Data in High-Speed Mode....515
23.5.4 Master Receiving and Slave Transmitting Data in High-Speed Mode....515
23.6 Register Description....517
23.6.1 Register List....517
23.6.2 Status Register (I2CSLV_SR)....518
23.6.3 Data Transmit Register (I2CSLV_DOUTR)....521
23.6.4 Slave Receive Address Register (I2CSLV_ADIR)....521
23.6.5 Slave Address n Register (I2CSLV_SLADRn) (n=0-3)....522
23.6.6 Data Receive Register (I2CSLV_DINR)....522
23.6.7 Control Register (I2CSLV_CR)....523
23.6.8 Slave Address 0 Mask Register (I2CSLV_ADMR0)....525
23.6.9 Filter Register (I2CSLV_FLT)....526

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

23.6.10 Timeout Register (I2CSLV_TOR)....526
23.6.11 Edge Shaping Register (I2CSLV_ODDR)....527
23.6.12 Receive Sampling Location Register (I2CSLV_RSR)....528
23.6.13 Interrupt Enable Register (I2CSLV_IER)....529
23.6.14 Interrupt Flag Clear Register (I2CSLV_ICR)....531
23.6.15 Device ID Register (I2CSLV_DIDR)....533
24 EAU....534
24.1 Introduction....534
24.2 Features....534
24.3 Functional Description....534
24.3.1 Function Block Diagram....534
24.3.2 Operation Definition....535
24.3.3 Status Flags....535
24.4 Operation Example....536
24.4.1 Division Operation....536
24.4.2 Square Root Operation....537
24.5 Register Description....538
24.5.1 Register List....538
24.5.2 Control and Status Register (EAU_CSR)....539
24.5.3 Dividend/Square Root Operand Register (EAU_DIVIDEND)....540
24.5.4 Divisor Register (EAU_DIVISOR)....541
24.5.5 Quotient/Square Root Register (EAU_QUOTIENT)....541
24.5.6 Remainder Register (EAU_REMAINDER)....542
25 ADC....543
25.1 Introduction....543
25.2 Features....543
25.3 Functional Description....544
25.3.1 ADC Block Diagram....544
25.3.2 Conversion Timing and Conversion Speed....544
25.3.3 Single Conversion Mode....545
25.3.4 Scan Conversion Mode....545
25.3.5 Continuous Conversion Accumulation Mode....547
25.3.6 ADC Conversion External Trigger Source....548
25.3.7 ADC Conversion Result Comparison....549
25.3.8 ADC Interrupt....550
25.3.9 Measure Ambient Temperature Using a Temperature Sensor....550
25.4 Operation Example....551
25.4.1 Start the ADC Single Conversion Operation Process through the ADC_SGLSTART.START Bit....551
25.4.2 ADC Single Conversion Operation Process through an External Trigger....552
25.4.3 Start the ADC SQR Scan Conversion Operation Process through the ADC_SQRSTART.START Bit....552
25.4.4 External Trigger Start ADC SQR Scan Conversion Operation Flow....553
25.4.5 Start ADC JQR Scan Conversion Operation Flow via ADC_JQRSTART.START Bit....554
25.4.6 External Trigger Start ADC JQR Scan Conversion Operation Flow....555
25.4.7 Start the ADC Continuous Conversion Accumulation Operation Process through the ADC_SQRSTART.START Bit....555
25.4.8 Start ADC Continuous Conversion Accumulation Operation Process through External Trigger....556

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

25.4.9 Operation Process of Measuring Ambient Temperature through ADC....557
25.5 Register Description....558
25.5.1 Register List....558
25.5.2 ADC Analog Device Select Register (ADC_BGR)....560
25.5.3 ADC Basic Configuration Register 0 (ADC_CR0)....561
25.5.4 ADC Basic Configuration Register 1 (ADC_CR1)....563
25.5.5 ADC SQR Scan Conversion Channel Configuration Register 0 (ADC_SQR0)....565
25.5.6 ADC SQR Scan Conversion Channel Configuration Register 1 (ADC_SQR1)....566
25.5.7 ADC SQR Scan Conversion Channel Configuration Register 2 (ADC_SQR2)....567
25.5.8 ADC JQR Scan Conversion Channel Configuration Register (ADC_JQR)....568
25.5.9 ADC SQR Scan Conversion Channel x Result Register (ADC_SQRRESULTx) (x=0-3)....569
25.5.10 ADC JQR Scan Conversion Channel x Result Register (ADC_JQRRESULTx) (x=0-3)....569
25.5.11 ADC Conversion Result Register (ADC_RESULT)....570
25.5.12 ADC Conversion Result Accumulation Value Register (ADC_RESULTACC)....570
25.5.13 ADC Compare High Threshold Register (ADC_HT)....571
25.5.14 ADC Compare Low Threshold Register (ADC_LT)....571
25.5.15 ADC Interrupt and DMA Trigger Enable Register (ADC_IER)....572
25.5.16 ADC Interrupt Flag Register (ADC_IFR)....573
25.5.17 ADC Interrupt Clear Register (ADC_ICR)....575
25.5.18 ADC Single or SQR Scan Conversion External Interrupt Trigger Source Configuration Register (ADC_EXTTRIGGER0)....576
25.5.19 ADC JQR Scan Conversion External Interrupt Trigger Source Configuration Register (ADC_EXTTRIGGER1)....578
25.5.20 ADC Single Conversion Start Control Register (ADC_SGLSTART)....580
25.5.21 ADC SQR Scan Conversion Start Control Register (ADC_SQRSTART)....581
25.5.22 ADC JQR Scan Conversion Start Control Register (ADC_JQRSTART)....582
26 VC....583
26.1 Introduction....583
26.2 Features....583
26.3 Functional Description....583
26.3.1 Function Block Diagram....583
26.3.2 Setup/Response Time....584
26.3.3 Digital Filtering....584
26.3.4 Hysteresis Function....584
26.3.5 Configuration of Comparison Result Polarity....584
26.3.6 Window Comparison Function....584
26.4 Register Description....585
26.4.1 Register List....585
26.4.2 Control Register 0 (VCx_CR0) (x=0-1)....586
26.4.3 Control Register 1 (VCx_CR1) (x=0-1)....588
26.4.4 Control Register 2 (VCx_CR2) (x=0-1)....590
26.4.5 Status Register (VCx_SR) (x=0-1)....591
27 LVD....592
27.1 Introduction....592
27.2 Features....592
27.3 Functional Description....592

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

27.3.1 LVD Block Diagram....592
27.3.2 Digital Filtering....592
27.3.3 Hysteresis Function....593
27.4 Operation Example....593
27.4.1 LVD Configured as Low Voltage Reset....593
27.4.2 LVD Configured as Voltage Change Interrupt....593
27.5 Register Description....595
27.5.1 Register List....595
27.5.2 LVD Configuration Register (LVD_CR)....596
27.5.3 LVD Status Register (LVD_SR)....598
28 OPA....599
28.1 Introduction....599
28.2 Features....599
28.3 Functional Description....599
28.3.1 Function Block Diagram....599
28.3.2 OPA Zero Calibration....599
28.4 Register Description....601
28.4.1 Register List....601
28.4.2 OPAx Control Register (OPAx_CR) (x=0-1)....602
28.4.3 OPA Zero Calibration Control Register (OPA_AZ)....603
29 SWD Debug Interface....605
29.1 Introduction....605
29.2 Functional Description....605
29.2.1 Debug System Diagram....605
29.2.2 ARM® Reference Documentation....606
29.2.3 Debug Port Pins....606
29.2.4 SWD Port....607
29.2.5 Core Debugging....610
29.2.6 BPU (Breakpoint Unit)....611
29.2.7 DWT (Data Watchpoint)....611
29.2.8 MCU Debug Component (DBG)....611
30 Device Electronic Signature....613
30.1 Introduction....613
30.2 Functional Description....613
30.2.1 Product Unique Identifier (UID)....613
30.3 Register Description....614
30.3.1 Product Unique Identifier (UID) Register (80 bits)....614
30.3.2 Product Model Register....614
30.3.3 FLASH Capacity Register....615
30.3.4 RAM Capacity Register....615
30.3.5 Pin Count/Flash Sector Size Configuration Register....616
31 SysTick Timer....617
31.1 Introduction....617
31.2 Operation Example....617

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

31.2.1 Set SysTick....617
31.3 Register Description....618
31.3.1 Register List....618
31.3.2 SysTick Control and Status Register (CTRL)....619
31.3.3 SysTick Reload Register (LOAD)....620
31.3.4 SysTick Current Value Register (VAL)....621
31.3.5 SysTick Calibration Value Register (CALIB)....621
A Glossary....622
Revision History....623

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

List of Figures

Figure 2-1 System Architecture Schematic Diagram....27
Figure 3-1 Working Mode State Transition Diagram....31
Figure 4-1 Clock Architecture Diagram....36
Figure 4-2 PLL System Block Diagram....38
Figure 4-3 Crystal Oscillator Clock Startup Diagram....39
Figure 4-4 Clock Switching Diagram....42
Figure 4-5 Reset Source Diagram....45
Figure 5-1 Interrupt Vector Table....81
Figure 5-2 Interrupt Active and Pending States....82
Figure 5-3 Interrupt Pending State is Cleared and Then Reasserted....82
Figure 5-4 When the Interrupt Exits, If the Interrupt Request Remains High, It Will Cause the Interrupt Processing to Be Executed Again....82
Figure 5-5 Interrupt Pending Interrupts Generated During Interrupt Processing Can Also Be Acknowledged....83
Figure 5-6 Interrupt Structure Diagram....86
Figure 6-1 General Port Circuit Diagram....98
Figure 6-2 HiZ-Analog Configuration Diagram....99
Figure 6-3 AHB/FASTIO Bus Port Changes with System Clock....100
Figure 6-4 Input Floating/Pull-up/Pull-down Configuration Diagram....101
Figure 6-5 Read Port Pin Data Synchronization Diagram....101
Figure 6-6 Output Configuration Diagram....102
Figure 6-7 Multiplexed Function Configuration Diagram....103
Figure 6-8 Port External Interrupt Diagram....105
Figure 7-1 I2C Transmission Protocol....122
Figure 7-2 START, Repeated START and STOP....123
Figure 7-3 Bit Transfer on I2C Bus....123
Figure 7-4 Acknowledgment Signal on I2C Bus....124
Figure 7-5 Arbitration on the I2C Bus....125
Figure 7-6 I2C Function Block Diagram....126
Figure 7-7 Data Synchronization Diagram of Master Sending Mode....128
Figure 7-8 I2C Master Sending State Diagram....128
Figure 7-9 Data Synchronization Diagram of Master Receiving Mode....129
Figure 7-10 I2C Master Receiving State Diagram....129
Figure 7-11 Slave Receiving Mode Data Synchronization Diagram....130
Figure 7-12 Slave Receiving State Diagram....130
Figure 7-13 Slave Sending Mode Data Synchronization Diagram....131
Figure 7-14 I2C Slave Sending State Diagram....131
Figure 7-15 I2C Broadcast Call State Diagram....132
Figure 8-1 SPI Block Diagram....145
Figure 8-2 SPI Communication Format Timing Diagram....147
Figure 8-3 Slave Select NSS Pin Configuration....148
Figure 8-4 SPI Full-duplex Communication....149
Figure 8-5 SPI Single-wire Half-duplex Communication....149
Figure 8-6 SPI Simplex Transmit Communication Diagram....150
Figure 8-7 SPI Simplex Receive Communication Diagram....150
Figure 8-8 SPI Multi-machine System....151
Figure 8-9 Explanation Diagram of General Flags in SPI Master Mode....154
Figure 8-10 SPI Slave Mode General Flag Bit Description Diagram....154
Figure 9-1 RCH Clock Calibration Operational Block Diagram....168
Figure 9-2 RCL Clock Calibration Operational Block Diagram....170
Figure 9-3 General Low Power Timer Scenario Operational Block Diagram....171

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Figure 12-1 DMA Controller Functional Block Diagram....205
Figure 12-2 DMA Software Block Transfer Mode....206
Figure 12-3 DMA Software Burst Transfer Mode....207
Figure 12-4 DMA Hardware Block Transfer Mode....208
Figure 12-5 DMA Hardware Burst Transfer Mode....208
Figure 13-1 ATIM0/1/2 Block Diagram....220
Figure 13-2 ATIM3 Block Diagram....221
Figure 13-3 Mode0 Count Timer Free Counting Block Diagram....223
Figure 13-4 Mode0 Count Timer Reload Counting Block Diagram....223
Figure 13-5 Mode0 Count Timer 16-Bit Reload Counting Waveform....224
Figure 13-6 Mode0 Count Timer 32-Bit Free Counting Waveform....224
Figure 13-7 Mode0 Count Timer Free Counting Timing Diagram....225
Figure 13-8 Mode0 Count Timer Reload Counting Timing Diagram (Prescaler=2)....225
Figure 13-9 Mode1 Pulse Width Measurement PWC Function Block Diagram....226
Figure 13-10 Mode1: Pulse Width Measurement PWC High-Level Pulse Width Measurement....227
Figure 13-11 Mode1: Pulse Width Measurement PWC Falling Edge to Falling Edge Period Measurement....227
Figure 13-12 Mode1: Pulse Width Measurement PWC Rising Edge to Rising Edge Period Measurement....227
Figure 13-13 Mode1: Pulse Width Measurement PWC Rising Edge to Rising Edge Period Measurement Single Mode....228
Figure 13-14 Mode 2/3 Compare Capture Mode Counter Block Diagram....228
Figure 13-15 Mode 2 Compare Capture Mode Up-Counting Without Prescaler....229
Figure 13-16 Mode 2 Compare Capture Mode Up-Counting With Prescaler....229
Figure 13-17 Mode 2 Compare Capture Mode Down-Counting Without Prescaler....230
Figure 13-18 Mode 2 Compare Capture Mode Down-Counting With Prescaler....230
Figure 13-19 Mode2 Edge-Aligned Timer Waveform (M23CR.DIR=0)....230
Figure 13-20 Mode2 Edge-Aligned Timer Waveform (M23CR.DIR=1)....230
Figure 13-21 Mode3 Compare Capture Mode Up/Down-Counting with Prescaler....231
Figure 13-22 Mode3 Center-Aligned Counter Waveform....231
Figure 13-23 Repetition Counter Generates Update Timing (RCR.OV=0, RCR.UD=0)....232
Figure 13-24 Update Timing Generated by Repetition Counter when Overflow Is Masked (RCR.OV=1, RCR.UD=0)....232
Figure 13-25 Update Timing Generated by Repetition Counter when Underflow Is Masked (RCR.OV=0, RCR.UD=1)....232
Figure 13-26 Mode 3 Triangular Wave Mode with Cache Enable....233
Figure 13-27 Mode 3 Triangular Wave Mode with Cache Invalid....233
Figure 13-28 Mode 2 Sawtooth Wave Mode Up-Counting with Cache Enable....234
Figure 13-29 Mode 2 Sawtooth Wave Mode Up-Counting with Cache Invalid....234
Figure 13-30 Mode 2 Sawtooth Wave Mode Down-Counting with Cache Enable....234
Figure 13-31 Mode 2 Sawtooth Wave Mode Down-Counting with Cache Invalid....235
Figure 13-32 Mode2 Sawtooth Wave Down-Counting Mode with Compare Cache Enable....235
Figure 13-33 OCREF Output Block Diagram....236
Figure 13-34 Sawtooth Counting Single Point Comparison OCREF Output Waveform (OCMyx=0b111)....237
Figure 13-35 Triangular Counting Single Point Comparison OCREF Output Waveform (OCMyx=0b111)....237
Figure 13-36 Sawtooth Counting Dual Point Comparison OCREF Output Waveform (OCMyx=0b111)....237
Figure 13-37 Triangular Counting Dual Point Comparison OCREF Output Waveform (OCMyx=0b111)....238
Figure 13-38 Independent PWM Output Block Diagram....239
Figure 13-39 PWM Output Waveform When CCPy=0....239
Figure 13-40 PWM Output Waveform When CCPy=1....239
Figure 13-41 Complementary PWM Output Block Diagram....240
Figure 13-42 Complementary PWM Output Waveform....240
Figure 13-43 Complementary PWM Output Waveform (with Dead Time)....241
Figure 13-44 Dead Time....241
Figure 13-45 Triangular Wave Mode Single Pulse Counting....242
Figure 13-46 Sawtooth Up-Counting Single Pulse Mode....243
Figure 13-47 Sawtooth Down-Counting Single Pulse Mode....243

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Figure 13-48 Compare Interrupt Diagram....244
Figure 13-49 Capture Function Block Diagram....244
Figure 13-50 Capture Timing Diagram....245
Figure 13-51 ATIM3 CH0A Capture Input Port Selection....245
Figure 13-52 CH0B Capture Input Port Selection....246
Figure 13-53 Mode2/3 Slave Mode Diagram....246
Figure 13-54 Mode2/3 Slave Mode Gating Function....247
Figure 13-55 Mode2/3 Slave Mode Trigger and Reset Function....248
Figure 13-57 Quadrature Encoder Counting Mode3 Counting Diagram....250
Figure 13-58 Timer Triggers ADC....251
Figure 14-1 BTIM Block Diagram....299
Figure 14-2 BTIM Filter Unit Diagram....299
Figure 14-3 BTIM Counter Waveform....300
Figure 14-4 BTIM Timing Mode Block Diagram....300
Figure 14-5 BTIM Counter Mode Block Diagram....301
Figure 14-6 BTIM Trigger Start Mode Block Diagram....301
Figure 14-7 BTIM Trigger Start Mode Counting Diagram....302
Figure 14-8 BTIM Gating Mode Block Diagram....302
Figure 14-9 BTIM Gating Mode Counting Diagram....303
Figure 14-10 BTIM Timer Cascade Diagram....303
Figure 14-11 GTIM Block Diagram....304
Figure 14-12 GTIM Filter Unit Diagram....304
Figure 14-13 GTIM Counter Waveform....305
Figure 14-14 GTIM Timing Mode Block Diagram....306
Figure 14-15 GTIM Counter Mode Block Diagram....306
Figure 14-16 GTIM Trigger Start Mode Block Diagram....307
Figure 14-17 GTIM Trigger Start Mode Counting Diagram....307
Figure 14-18 GTIM Gating Mode Block Diagram....308
Figure 14-19 GTIM Gating Mode Counting Diagram....308
Figure 14-20 GTIM Capture Function Diagram....309
Figure 14-21 GTIM Compare Function Diagram....310
Figure 15-1 Low Power Timer Mode 1....330
Figure 15-2 Low Power Timer Mode 2....331
Figure 16-1 RTC Block Diagram....340
Figure 17-1 IWDT Function Block Diagram....370
Figure 18-1 WWDT Function Block Diagram....379
Figure 19-1 Asynchronous Serial Port Mode Hardware Flow Control Example....391
Figure 19-2 CTS Hardware Flow Control Signal....392
Figure 19-3 RTS Hardware Flow Control Signal....392
Figure 19-4 Asynchronous Serial Port Function Block Diagram....395
Figure 19-5 Asynchronous Serial Port Mode Data Format....396
Figure 19-6 Transmit Data Example....396
Figure 19-7 Receive Data Example....397
Figure 19-8 Multiprocessor Communication Data Format....398
Figure 19-9 Synchronize Serial Port Mode Data Format....399
Figure 19-10 Synchronous Serial Port Mode Sending Data Legend 1 (LSB/No CTS Function)....400
Figure 19-11 Synchronous Serial Port Mode to Send Data Legend 2 (LSB/CTS Function)....400
Figure 19-12 Synchronous Serial Port Mode Receiving Data Legend 1 (LSB/No RTS Function)....401
Figure 19-13 Synchronous Serial Port Mode Receiving Data Legend 2 (LSB/RTS Function)....402
Figure 19-14 Precautions for Synchronous Serial Port Mode....403
Figure 19-15 Smart Card Connection Schematic Diagram....403
Figure 19-16 Smart Card Mode Internal Synchronization Timing and Sample Timing Diagram....404

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Figure 19-17 Smart Card Mode Data Format....404
Figure 19-18 Smart Card Mode Data Format Example....404
Figure 19-19 Smart Card Mode Transmit Data Legend....405
Figure 19-20 Smart Card Mode Receive Data Legend....406
Figure 19-21 Infrared Data Modulation....406
Figure 19-22 38K Infrared Modulation....407
Figure 19-23 LIN Bus Data and Frame Structure....408
Figure 20-1 LPUART Function Block Diagram....439
Figure 20-2 Mode0 Data Transmitting....441
Figure 20-3 Mode0 Data Receiving....441
Figure 20-4 Mode1 Data Transmitting....442
Figure 20-5 Mode1 Data Receiving....442
Figure 20-6 Mode2/Mode3 Data Transmitting....442
Figure 20-7 Mode2/Mode3 Data Receiving....443
Figure 20-8 LPUART Hardware Flow Control....447
Figure 20-9 RTS Hardware Flow Control Signal....447
Figure 20-10 CTS Hardware Flow Control Signal....448
Figure 21-1 CAN System Block Diagram....459
Figure 21-2 CAN LBMI and LBME Schematic Diagram....460
Figure 21-3 CAN Bit Time Definition Diagram....461
Figure 21-4 CAN TBUF Register Write Transmit Buffer and Schematic Diagram....462
Figure 21-5 CAN RBUF Register Read Receive Buffer Schematic Diagram....463
Figure 21-6 CAN ACF Register Access Filter Group Schematic Diagram....464
Figure 23-1 I2C SLAVE Module Block Diagram....507
Figure 23-2 Example of I2C Master Transmitting and Slave Receiving Sequence....509
Figure 23-3 I2C Master Transmitting and Slave Receiving Operation Flow Diagram....509
Figure 23-4 Example of I2C Master Receiving and Slave Transmitting Sequence....510
Figure 23-5 I2C Master Receiving and Slave Transmitting Operation Flow Diagram....511
Figure 24-1 EAU Block Diagram....534
Figure 25-1 ADC Block Diagram....544
Figure 25-2 ADC Conversion Timing Diagram....544
Figure 25-3 Examples of ADC SQR Scan Conversion Process....546
Figure 25-4 Examples of ADC JQR Scan Conversion Process....547
Figure 25-5 Examples of Continuous ADC Conversion Accumulation Process....548
Figure 25-6 Diagram of External Trigger Sources for ADC Single and SQR Scan Conversions....548
Figure 25-7 Diagram of External Trigger Sources for ADC JQR Scan Conversion....549
Figure 25-8 ADC Conversion Result Comparison Diagram....549
Figure 26-1 VC Function Block Diagram....583
Figure 26-2 VC Filter Response Time....584
Figure 26-3 VC Hysteresis Function....584
Figure 27-1 LVD Block Diagram....592
Figure 27-2 LVD Filter Output....592
Figure 27-3 LVD Hysteresis Response....593
Figure 28-1 OPA Function Block Diagram....599
Figure 29-1 Debug Support Block Diagram....606

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

List of Tables

Table 2-1 Address Division Table....29
Table 2-2 Peripheral Address Division Table....29
Table 3-1 Reset Type Response Capabilities in Different Operational Mode....31
Table 3-2 Peripheral Working Status Table in Each Mode....33
Table 4-2 System Controller (SYSCTRL) Register Offset....46
Table 5-1 Cortex-M0+ Processor Interrupt List....79
Table 5-4 Interrupt Controller (NVIC) Register Offset Address....89
Table 6-1 Port Function Configuration Table....98
Table 6-2 Port Digital Multiplexed Function....103
Table 6-3 IOAUX1 Multiplexed Function....104
Table 6-4 IOAUX2 Multiplexed Function....104
Table 6-5 General-Purpose I/O (GPIO) Register Offset....106
Table 7-1 I2C Clock Signal Baud Rate....126
Table 7-2 I2C Status Code Expression....132
Table 7-3 I2C Bus (I2C) Register Offset....137
Table 8-1 SPI Interrupt....155
Table 8-2 Serial Peripheral Interface (SPI) Register Offset....158
Table 9-1 Clock Trim Module (CTRIM) Register Offset....173
Table 10-1 Flash Capacity Division Table....184
Table 10-3 Flash Controller (FLASH) Register Offset....192
Table 11-1 RAM Address Mapping....200
Table 11-2 RAM Controller (RAM) Register Offset....201
Table 12-1 DMAC Transfer Mode Comparison Table....206
Table 12-2 DMA Controller (DMA) Register Offset....212
Table 13-6 CH0A Capture Input Signal....245
Table 13-11 ATIM0/1/2/3 Interrupt List....252
Table 13-12 Timer Interconnection Signal List....252
Table 13-13 ATIM0 ETR Signal Source List....253
Table 13-14 ATIM0 CHA Input Signal Source List....253
Table 13-15 ATIM3 ETR and CH2B Input Signal Source List....253
Table 13-16 Mode0 Timer Register Offset....257
Table 13-17 Mode1 Pulse Width Measurement (PWM) Register Offset....257
Table 13-18 Mode2/3 Compare-Capture Mode Register Offset....257
Table 14-1 Comparison Table of Pin Function and Timer Function....297
Table 14-2 BTIM Timer Interrupt....303
Table 14-3 GTIM TRS Trigger Source List....310
Table 14-4 GTIM ETR Source List....310
Table 14-5 GTIM CH3 Signal Source List....310
Table 14-6 BTIM Register Offset....312
Table 14-7 GTIM Register Offset....312
Table 15-1 Low Power Timer (LPTIM) Register Offset....333
Table 16-1 Basic Features of RTC....338
Table 16-2 Real Time Counter (RTC) Register Offset....346
Table 17-1 IWDT Timeout Time (32kHz Independent Clock)....372
Table 17-2 Independent Watchdog (IWDT) Register Offset....373
Table 18-1 Window Watchdog (WWDT) Register Offset....381
Table 19-1 Each Mode Sample Count and Maximum Baud Rate....386
Table 19-2 Baud Rate Programming Example with PCLK at 4MHz....387
Table 19-3 Baud Rate Programming Example with PCLK at 8MHz....388
Table 19-4 Baud Rate Programming Example with PCLK at 16MHz....388

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Table 19-5 Baud Rate Programming Example with PCLK at 24MHz....388
Table 19-6 Baud Rate Programming Example with PCLK at 32MHz....389
Table 19-7 Baud Rate Programming Example with PCLK at 48MHz....389
Table 19-8 Multi-Function Timing Counter Function....393
Table 19-9 Each Mode Interrupt Validity....394
Table 19-10 Universal Synchronous Asynchronous Receiver/Transmitter (USART) Register Offset....415
Table 20-1 LPUART Mode0/1/2/3 Data Structure....440
Table 20-2 DR8 Data Meaning....440
Table 20-3 SCLK = 4MHz Baud Rate Calculation Table....443
Table 20-4 SCLK = 8MHz Baud Rate Calculation Table....444
Table 20-5 SCLK = 16MHz Baud Rate Calculation Table....444
Table 20-6 SCLK = 24MHz Baud Rate Calculation Table....445
Table 20-7 SCLK = 32MHz Baud Rate Calculation Table....445
Table 20-8 SCLK = 48MHz Baud Rate Calculation Table....445
Table 20-9 Low Power Universal Asynchronous Receiver/Transmitter (LPUART) Register Offset....450
Table 21-1 CAN Pin Description....459
Table 21-2 CAN Bit Time Setting Rules....461
Table 21-3 Software Reset Range Table....467
Table 21-4 CAN Interrupt Table....469
Table 21-5 Controller Area Network (CAN) Register Offset....470
Table 21-6 Standard Format CAN Receive Mailbox Format....471
Table 21-7 Extended Format CAN Receive Mailbox Format....472
Table 21-9 DLC Relationship with CAN 2.0 Data length....472
Table 21-10 Standard Format CAN Transmit Mailbox Format....473
Table 21-11 Extended Format CAN Transmit Mailbox Format....474
Table 22-1 Cyclic Redundancy Check (CRC) Register Offset....503
Table 23-1 I2C Status Decoding....512
Table 23-2 I2CSLV Slave Module (I2CSLV) Register Offset....517
Table 24-1 Extended Arithmetic Unit (EAU) Register Offset....538
Table 25-2 ADC Interrupt....550
Table 25-3 Temperature Sensor Calibration Value Information List....550
Table 25-4 Analog-to-Digital Converter (ADC) Register Offset....558
Table 26-1 Analog Voltage Comparator (VC) Register Offset....585
Table 27-1 Low Voltage Detector (LVD) Register Offset....595
Table 28-1 Operational Amplifier (OPA) Register Offset....601
Table 29-3 Host Transmitted Data Packet Request....608
Table 29-4 Target Transmitted Acknowledgment Response....608
Table 29-5 Data Transfers Transmitted by Host or Target....608
Table 31-2 SysTick Timer Register Offset....618

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

1 Product Introduction

HC32F052 Series with a maximum frequency of 64MHz and Flash capacity up to 128KB, it offers rich digital and analog peripherals. It is an MCU with a wide voltage operating range.

This product integrates USART (LIN-compatible), SPI, I2C, CAN, and other communication peripherals, featuring high noise immunity and reliability, suitable for general-purpose MCU markets.

This product integrates one 12-bit high-precision SAR ADC, two OPAs, and two comparators. Combined with a hardware division and square root unit and a high-performance motor-specific PWM timer, it can realize complex high-performance and highly reliable motor control solutions.

This product adopts the Cortex-M0+ core and is combined with the mature Keil & IAR debugging and development software, supporting C language, assembly language and assembly instructions.

Typical Application

- IoT
- Home appliance main control
- Motor control
- 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.

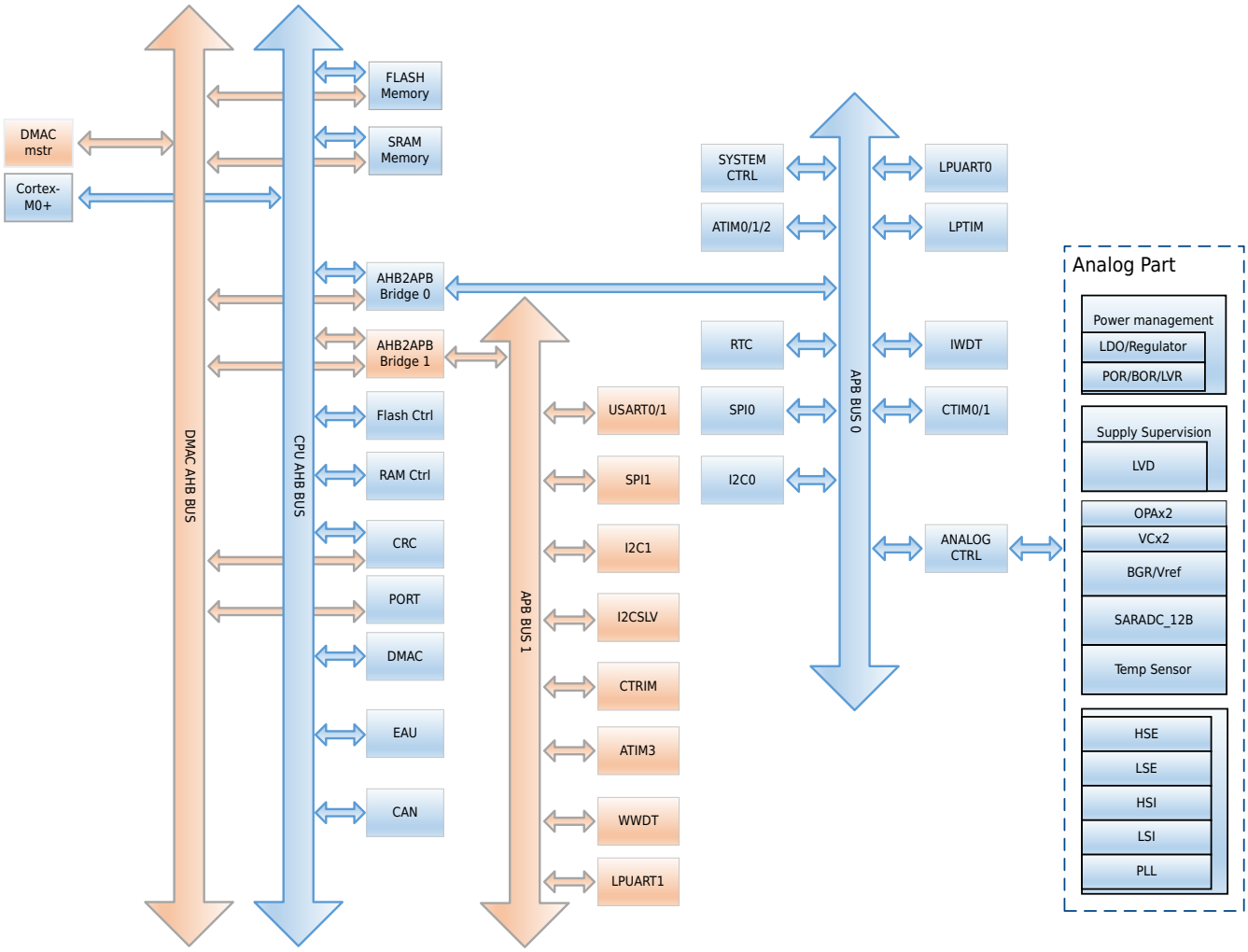
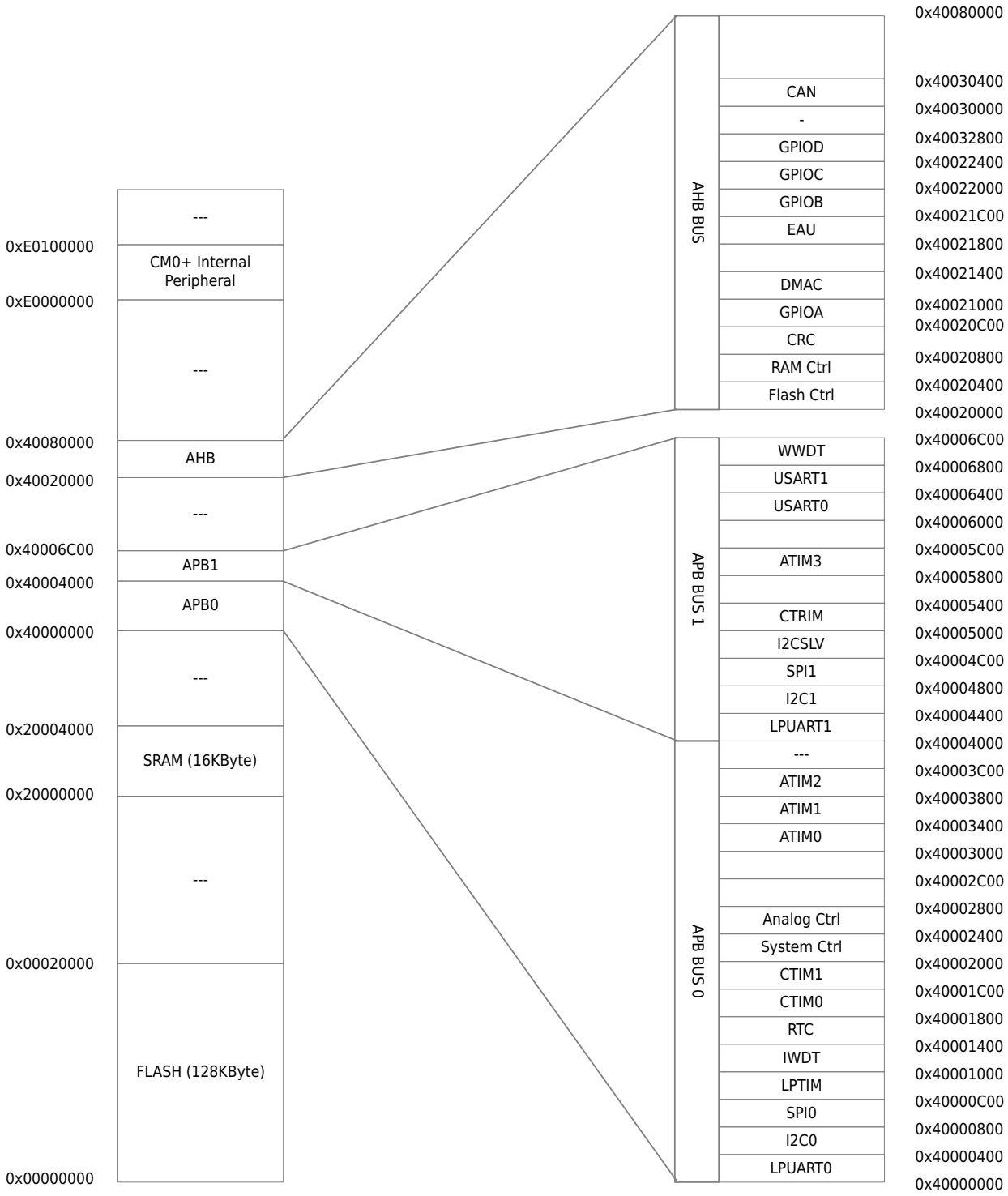


Figure 2-1 System Architecture Schematic Diagram

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

2.2 System Address Division

System of address region division:



Note

GPIOA and GPIOB/C/D modules use two separate address spaces.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

2.3 Memory and Peripheral Address Allocation

Table 2-1 Address Division Table

Categories	Boundary Address	Size	Memory Area
Memory	0x00000000-0x0001FFFF	128 KByte **Note** The maximum capacity is 128KByte.	FLASH
	0x20000000-0x20003FFF	16K Byte	SRAM
APB peripherals	0x40000000-0x40007FFF	32K Byte	APB0
	0x40008000-0x4000FFFF	32K Byte	APB1
AHB peripherals	0x40020000-0x40033FFF	16K Byte	AHB

Table 2-2 Peripheral Address Division Table

Categories	Base Address	Size	Peripheral
APB0	0x40000000	0x400	LPUART0
	0x40000400	0x400	I2C0
	0x40000800	0x400	SPI0
	0x40000C00	0x400	LPTIM
	0x40001000	0x400	IWDT
	0x40001400	0x400	RTC
	0x40001800	0x400	CTIM0
	0x40001C00	0x400	CTIM1
	0x40002000	0x400	SYSCTRL
	0x40002400	0x400	ANACTRL
	0x40003000	0x400	ATIM0
	0x40003400	0x400	ATIM1
	0x40003800	0x400	ATIM2
APB1	0x40004000	0x400	LPUART1
	0x40004400	0x400	I2C1
	0x40004800	0x400	SPI1
	0x40004C00	0x400	I2CSLV
	0x40005000	0x400	CTRIM
	0x40005400	0x400	-
	0x40005800	0x400	ATIM3
	0x40005C00	0x400	-

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Categories	Base Address	Size	Peripheral
APB1	0x40006000	0x400	USART0
	0x40006400	0x400	USART1
	0x40006800	0x400	WWDT
AHB	0x40020000	0x400	FLASHCTRL
	0x40020400	0x400	RAMCTRL
	0x40020800	0x400	CRC
	0x40020C00	0x400	GPIOA
	0x40021000	0x400	DMAC
	0x40021400	0x400	-
	0x40021800	0x400	EAU
	0x40021C00	0x400	GPIOB
	0x40022000	0x400	GPIOC
	0x40022400	0x400	GPIOD
	0x40022800	0x800	-
	0x40030000	0x400	CAN

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, on-chip peripherals running.
2. Sleep mode: CPU halted, on-chip peripherals running.
3. DeepSleep mode: CPU halted, low-power on-chip peripherals running.

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.

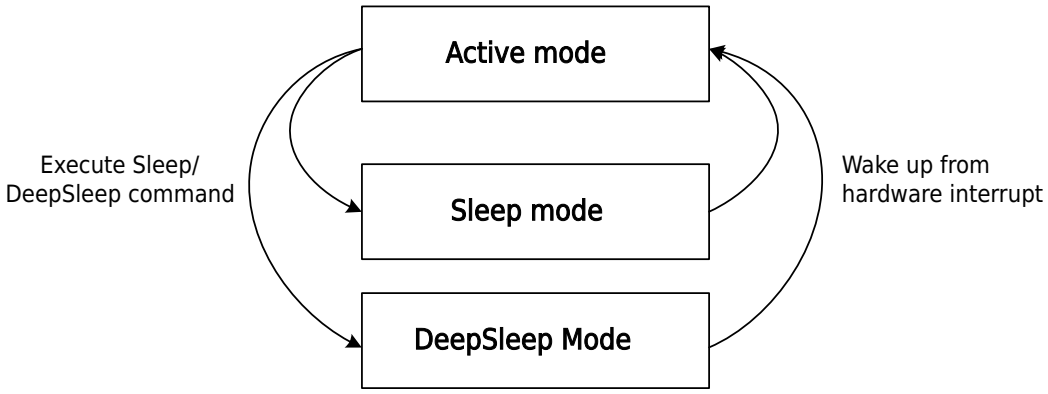


Figure 3-1 Working Mode State Transition Diagram

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

Reset Vector Index Reference	Reset Source	Active Mode	Sleep Mode	DeepSleep Mode
[0]	Power-on Reset (POR)/Brown-out Reset (BOR)	✓	✓	✓
[1]	Hardware Reset RESETB Pin	✓	✓	✓
[2]	Hardware Reset LVD	✓	✓	✓
[3]	Hardware Reset IWDT	✓	✓	✓
	Hardware Reset WWDT	✓	✓	×
[4]	Hardware Reset Cortex-M0+ LOCKUP	✓	×	×
[5]	Software Reset Cortex-M0+ SYSRESETREQ	✓	×	×

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

3.3 Active Mode

In Active mode, all functional modules of the MCU work normally, and the power consumption is the largest.

After the system is reset or woken up from an interrupt, the MCU will work in the running state.

In Active mode, certain measures can be taken to reduce the running power consumption:

- To select a lower frequency for the system clock (SystemClk), modify SYSCTRL_CR0.CLKSRC.
- To reduce the frequency of the system core clock (HCLK), modify SYSCTRL_CR0.HCLKPRS.
- To reduce the frequency of the system on-chip peripheral clock (PCLK), modify SYSCTRL_CR0.PCLKPRS.
- Turn off the clocks of unused on-chip peripherals and modify SYSCTRL_PERI_CLKENx.

3.4 Sleep Mode

In Sleep mode, the CPU halted, and other on-chip peripherals work normally, and the power consumption is slightly lower than that in the Active mode.

In Active mode, set SCB_SC R.SLEEPDEEP to 0 and execute WFI command to make MCU work in Sleep mode. When the MCU responds to the hardware interrupt, it can exit the Sleep mode and return to the Active mode.

In Sleep mode, certain measures can be taken to reduce operating power consumption:

- To select a lower frequency for the system clock (SystemClk), modify SYSCTRL_CR0.CLKSRC.
- To reduce the frequency of the system core clock (HCLK), modify SYSCTRL_CR0.HCLKPRS.
- To reduce the frequency of the system on-chip peripheral clock (PCLK), modify SYSCTRL_CR0.PCLKPRS.
- Turn off the clocks of unused on-chip peripherals and modify SYSCTRL_PERI_CLKENx.
- Set SCB_SCR.SLEEPONEXIT to 1, so that the MCU will automatically enter Sleep mode after executing the interrupt service routine.

3.5 DeepSleep Mode

In DeepSleep mode, the CPU halted, and the high-speed clock (RCH) is automatically turned off. If the source of HCLK is a high-speed clock, the on-chip high-speed peripherals stop running, and only the on-chip low-power peripherals work normally; if the source of HCLK is a low-speed clock (XTL, RCL), all on-chip peripherals work normally. The power consumption in this mode is much lower than that in Sleep mode. This system does not support entering this mode when the system clock is PLL, it is recommended to switch the system clock to RCHCLKD before entering this mode. RCHCLKD refers to the clock generated by the 12M RCH clock after being configured by the RCHCR.DIV register. For example, if RCHCR.DIV=2, the RCHCLKD frequency is 4 MHz.

If SYSCTRL_CR1.WAKEUPBYRCH is set to 1, the system clock source when the MCU exits DeepSleep mode will be the RCH to achieve fast wakeup.

In Active mode, set SCB_SCR.SLEEPDEEP to 1 and execute WFI command to make MCU work in Deep-Sleep mode. When the MCU responds to the hardware interrupt, it can exit the DeepSleep mode and return to the Active mode.

In DeepSleep mode, certain measures can be taken to reduce operating power consumption:

- Select a system clock (SystemClk), modify SYSCTRL_CR0.CLKSRC.
- Turn off the clocks of unused on-chip low-power peripherals and modify SYSCTRL_PERI_CLKENx.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

- Set SCB_SCR.SLEEPONEXIT to 1 to make the MCU automatically enter DeepSleep mode after completing interrupt service routine execution.

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+	✓	×	×
DMA	✓	✓	×
GPIO	✓	✓	✓
VC	✓	✓	✓(1)
LVD	✓	✓	✓
ADC	✓	✓	×
OPA	✓	✓	×
ATIM	✓	✓	×
CTIM	✓	✓	×
LPTIM	✓	✓	✓(2)
SPI	✓	✓	×
I2C	✓	✓	×
I2CSLV	✓	✓	×
USART	✓	✓	×
LPUART	✓	✓	✓(2)
RTC	✓	✓	✓
CTRIM	✓	✓	✓(2)
WWDT	✓	✓	×
IWDT	✓	✓	✓
CRC	✓	✓	×

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, wakeup from DeepSleep mode is not supported.

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 ● 0b1: A new event is generated upon each interrupt pending. If WFE is used to enter Sleep mode, this event can wake up the processor	RW
3	*Reserved*	-	-
2	SLEEPDEEP	Sleep Mode Selection ● 0b0: Executing WFI enters Sleep/Idle mode ● 0b1: Executing WFI enters DeepSleep mode	RW
1	SLEEPONEXIT	Auto-Sleep Mark ● 0b0: This feature disabled ● 0b1: When exiting exception handling and returning to thread mode, the processor automatically enters Sleep 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:12MHz)
- 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
- PLL - Phase-locked loop 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 clock signal from the PC14 pin. XTH can not be connected to the crystal oscillator, and directly input the clock signal from the PD00 pin.

This product also contains the following auxiliary clocks:

- RC150K - Internal 150kHz clock, for LVD and VC modules.

The PCLK clock signal can be output through the PCLK_OUT pin; the RCH/RCL/XTH/XTL/PLL/RC150K clock signal can be output through the MCO pin.

4.1.1 Clock Architecture Diagram

The clock architecture of this product is shown as follows:

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

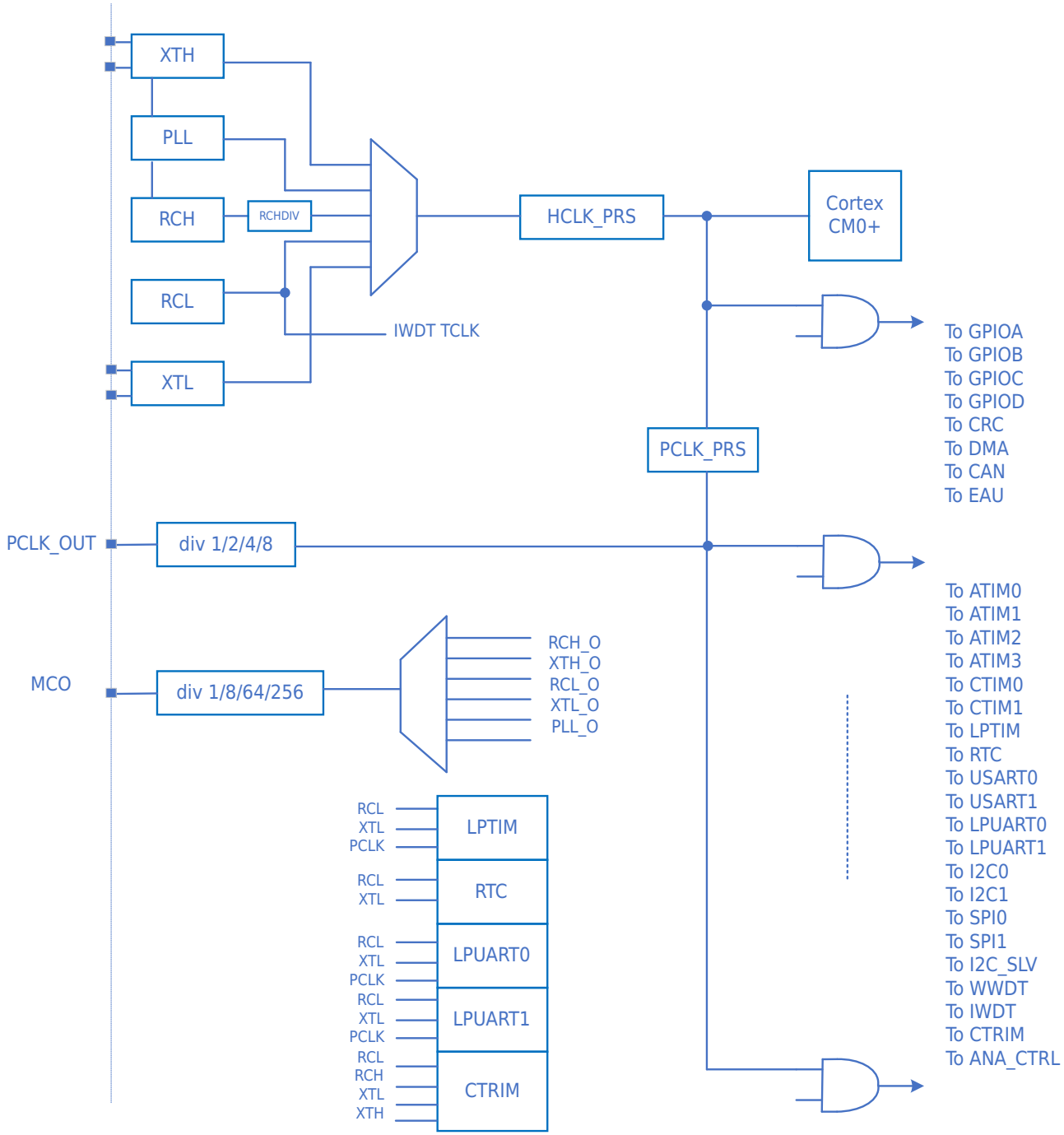


Figure 4-1 Clock Architecture Diagram

4.1.2 Clock Types

4.1.2.1 RCH

The default clock source after chip power-up or reset is RCHCLKD with a frequency of 4 MHz; this high-speed clock automatically shuts down when the system enters DeepSleep. The internal high-speed clock takes a very short time from wake-up to stabilization (please refer to the Datasheet for specific times), switching the system clock source to RCHCLKD before entering DeepSleep mode allows the system to respond quickly to interrupts.

The oscillation frequency of RCH can be fine-tuned by changing the value of the register SYSCTRL_RCHTRIM. Every time the register value increases by 1, the oscillation frequency of RCH will increase by about 0.08%. The RCH oscillation frequency is 12 MHz. The chip has a pre-adjusted

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

frequency point of 12 MHz when it leaves the factory; if other frequencies are required, please manually adjust the value of this register. If the RCH real-time calibration function is enabled in the CTRIM module, the accuracy of the RCH in the entire working range can reach 0.1%, see Clock Trim Module (CTRIM) for details.

The division ratio of RCH oscillator frequency and the RCHCLKD can be adjusted by changing the value of the SYSCTRL.RCHCR register, the division ratio can be set to 1, 2, 3, or 4.

4.1.2.2 RCL

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

The clock frequency can be configured through SYSCTRL_RCLTRIM 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.

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 of a 32.768kHz 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;
- Set SYSCTRL_CR2.EXXTL to 1.

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 8-24MHz. When the system enters DeepSleep mode, this clock will shut down automatically.

XTH supports an external active clock as input, allowing the input 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 Number input;
- Set SYSCTRL_CR2.EXXTH to 1.

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.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

4.1.2.5 PLL

The built-in PLL supports 12-64 MHz clock output, and the reference clock source of the PLL can be selected from RCH and XTH. This system does not support entering DeepSleep when the system clock is PLL. It is recommended to switch the system clock to another clock before entering DeepSleep.

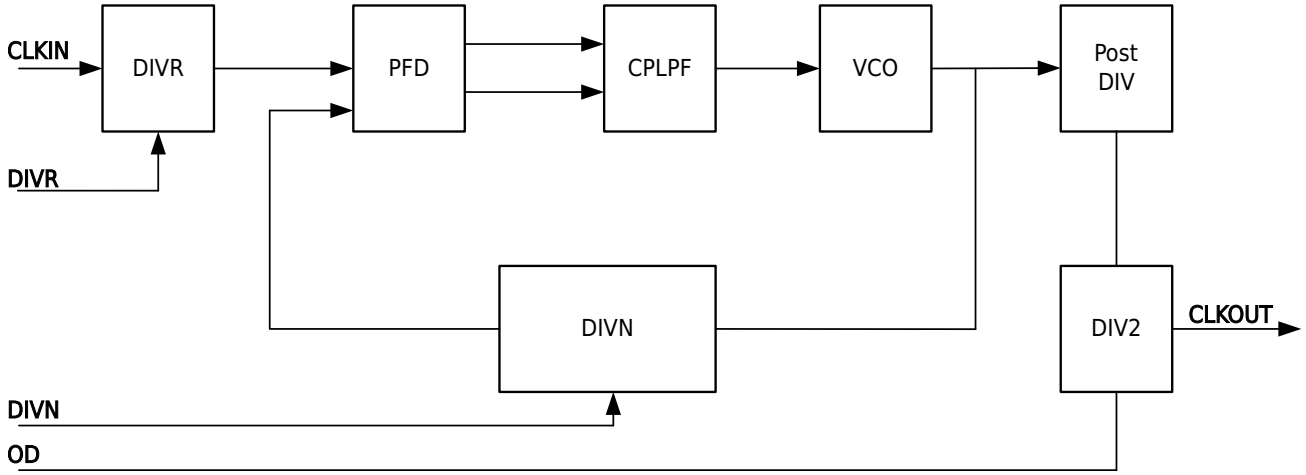


Figure 4-2 PLL System Block Diagram

The working frequency of the VCO is recommended to be set to 384MHz-512MHz. Beyond this range, the PLL may not work properly.

$$F_{VCO} = \frac{F_{in}}{DIVR+1} \times \left((DIVN+2) + \frac{DIVF}{8}\right)$$

The VCO frequency is determined by the formula above. Select an appropriate input frequency F_{in} and pre-divider ratio DIVR such that $F_{in}/(DIVR + 1)$ falls within the range of 4M-12MHz. The recommended range of DIVN is between 35 and 123. Beyond this range, the stability of the PLL loop cannot be guaranteed.

For example, F_{in}=24M, set PLLCR.DIVR=1, PLLCR.DIVN=38, PLLCR.DIVF=1, then F_{vco}=481.5MHz.

For another example, F_{in}=8M, set PLLCR.DIVR=0, PLLCR.DIVN=62, PLLCR.DIVF=0, then F_{vco}=(8/1)x(62+2), which is 512MHz.

If there are no strict requirements, it is recommended to set PLLCR.DIVF=0, because fractional frequency division will degrade jitter performance.

The VCO frequency is divided by POSTDIV and then divided by two to obtain the PLL output frequency F_{out}. The output frequency calculation formula is:

$$F_{OUT} = \frac{F_{VCO}}{2 \times (POSTDIV+2)}$$

For example, F_{vco}=512M, set PLLCR.POSTDIV=2, then F_{out}=64MHz.

For another example, F_{vco}=450M, set PLLCR.POSTDIV=10, then F_{out}= 18.75MHz.

See PLL Control Register for the PLLCR.DIVR, PLLCR.DIVN, PLLCR.DIVF and PLLCR.POSTDIV mentioned above.

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.

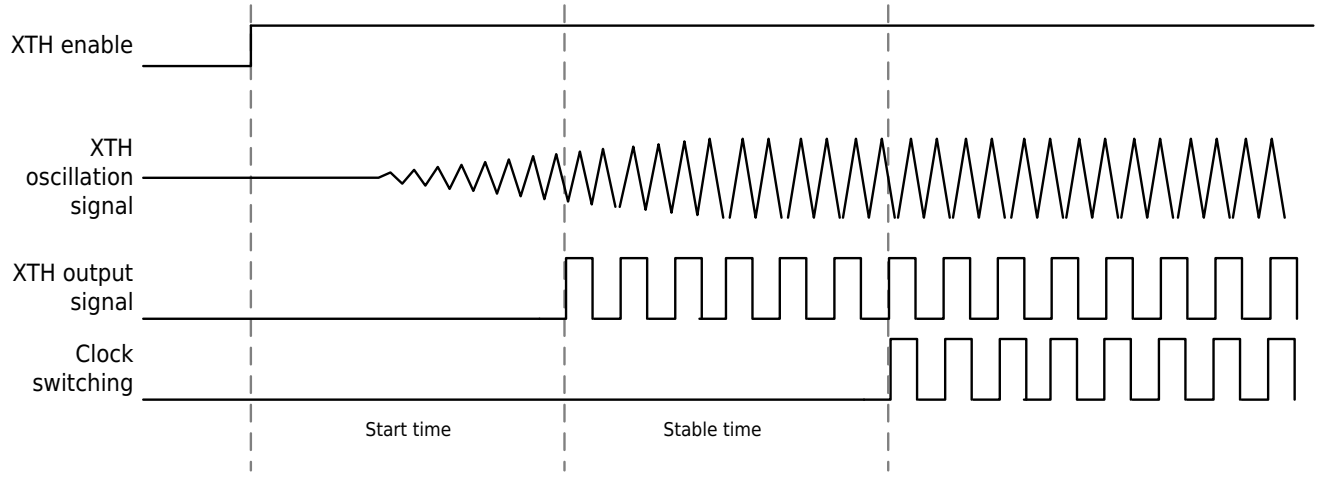


Figure 4-3 Crystal Oscillator Clock Startup Diagram

4.2 System Clock Switching

4.2.1 System Clock Switching Introduction

The system clock can choose 5 kinds of clock sources: RCH, RCL, XTL, XTH, and PLL, controlled by the register SYSCTRL_CR0.CLKSRC. Any two of the five clock sources RCH, RCL, XTL, PLL, and XTH can be switched between each other. When FLASH_WAIT is 0, it is necessary to ensure that HCLK is not greater than 24MHz during the switching process. It is recommended to perform clock switching according to the clock switching process described below to reduce the probability of clock switching errors.

The system clock source can be used as HCLK after system division, an appropriate frequency division value can be configured via SYSCTRL_CR0.HCLKPRS.

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

- When HCLK≤24MHz, FLASH_WAIT should be set to 0;
- When 24MHz<HCLK≤48MHz, FLASH_WAIT should be set to 1;
- When HCLK>48MHz, FLASH_WAIT should be set to 2.

Note

For detailed instructions on setting FLASH_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.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Step4. If the frequency of the new clock source is higher than that of the previous one, configure the FLASH_WAIT register 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.

Step6. Configure SYSCTRL_CR0.CLKSRC 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_WAIT register 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 RCHCLKD Switching Process between Different Oscillation Frequencies

Procedure

Configure SYSCTRL_RCHCR.DIV to select the frequency division ratio between RCH oscillation frequency and RCHCLKD frequency.

4.2.3 Examples of Switching

4.2.3.1 Example of Switching from Other Clocks to RCL

Procedure

Step1. Configure SYSCTRL_RCLTRIM.TRIM.

Step2. Write 0x5A5A and 0xA5A5 to SYSCTRL_CR3 register in sequence to enable register rewriting.

Step3. Set SYSCTRL_CR2.RCL to 1 to enable the RCL oscillation circuit.

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

Step5. Write 0x5A5A and 0xA5A5 to SYSCTRL_CR3 register in sequence to enable register rewriting.

Step6. Set SYSCTRL_CR0.CLKSRC to 2, switch the system clock to RCL.

Step7. See the chapter FLASH Controller to set FLASH_WAIT to 0.

Step8. Write 0x5A5A and 0xA5A5 to SYSCTRL_CR3 register in sequence to enable register rewriting.

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

4.2.3.2 Example of Switching from Other Clocks to RCHCLKD

Procedure

Step1. Configure SYSCTRL_RCHTRIM and SYSCTRL_RCHCR.DIV.

Step2. Write 0x5A5A and 0xA5A5 to SYSCTRL_CR3 register in sequence to enable register rewriting.

Step3. Set SYSCTRL_CR2.RCH to 1 to enable the RCH oscillation circuit.

Step4. Query and wait for SYSCTRL_STATE.RCH flag to become 1, and the RCH outputs a stable clock.

Step5. Write 0x5A5A and 0xA5A5 to SYSCTRL_CR3 register in sequence to enable register rewriting.

Step6. Set SYSCTRL_CR0.CLKSRC to 0, switch the system clock to RCHCLKD.

Step7. According to the frequency of RCHCLKD, set the FLASH_WAIT according to the procedures in the chapter FLASH Controller.

Step8. Write 0x5A5A and 0xA5A5 to SYSCTRL_CR3 register in sequence to enable register rewriting.

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

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

4.2.3.3 Example of Switching between PLL and RCHCLKD, the Reference Clock is RCH

Context

> **Notice**
> SYSCTRL_RCHTRIM must not be changed during switching.

Procedure

- Switch from RCHCLKD to PLL
  - **Step1.** Enable the BGR and delay the $T_{stable(BGR)}$ period (refer to the "Electrical Specifications - Operating Conditions - Internal Reference Voltage" section in the Datasheet for the specific value), then wait for the BGR to start up and stabilize.
  - **Step2.** Set SYSCTRL_PLLCR.SRC to 0 select the PLL clock source as RCH.
  - **Step3.** According to the PLL output frequency, configure SRC, DIVR, WAITCYCLE, DIVN, DIVF, POSTDIV in SYSCTRL_PLLCR. Required F_{IN}/(DIVR+1) range: 4MHz-12MHz.
  - **Step4.** Configure SYSCTRL_PLLCR.WAITCYCLE to select an appropriate PLL stable time.
  - **Step5.** Write 0x5A5A and 0xA5A5 to SYSCTRL_CR3 register in sequence to enable register rewriting.
  - **Step6.** Set SYSCTRL_CR2.PLL to 1 to enable the PLL oscillation circuit.
  - **Step7.** Configure FLASH_WAIT based on the PLL output frequency in terms of FLASH Controller.
  - **Step8.** Query and wait for SYSCTRL_STATE.PLL flag to become 1, and the PLL outputs a stable clock.
  - **Step9.** Write 0x5A5A and 0xA5A5 to SYSCTRL_CR3 register in sequence to enable register rewriting.
  - **Step10.** Set SYSCTRL_CR0.CLKSRC to 4, switch the system clock to PLL.
- Switch from PLL to RCHCLKD
  - **Step1.** Configure SYSCTRL_RCHTRIM and SYSCTRL_RCHCR.DIV.
  - **Step2.** Write 0x5A5A and 0xA5A5 to SYSCTRL_CR3 register in sequence to enable register rewriting.
  - **Step3.** Set SYSCTRL_CR2.RCH to 1 to enable the RCH oscillation circuit.
  - **Step4.** Query and wait for the SYSCTRL_STATE.RCH flag to become 1, and the RCH outputs a stable clock.
  - **Step5.** Write 0x5A5A and 0xA5A5 to SYSCTRL_CR3 register in sequence to enable register rewriting.
  - **Step6.** Set SYSCTRL_CR0.CLKSRC to 0,switch the system clock to RCHCLKD.
  - **Step7.** According to the frequency of RCHCLKD, set the FLASH_WAIT according to the procedures in the chapter FLASH Controller.
  - **Step8.** Write 0x5A5A and 0xA5A5 to SYSCTRL_CR3 register in sequence to enable register rewriting.
  - **Step9.** Set SYSCTRL_CR2.xxx to 0 to turn off unused clock sources.
  - **Step10.** Decide whether to disable the BGR based on the actual usage of other modules (such as ADC, etc.).

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

4.2.4 Clock Switching Diagram

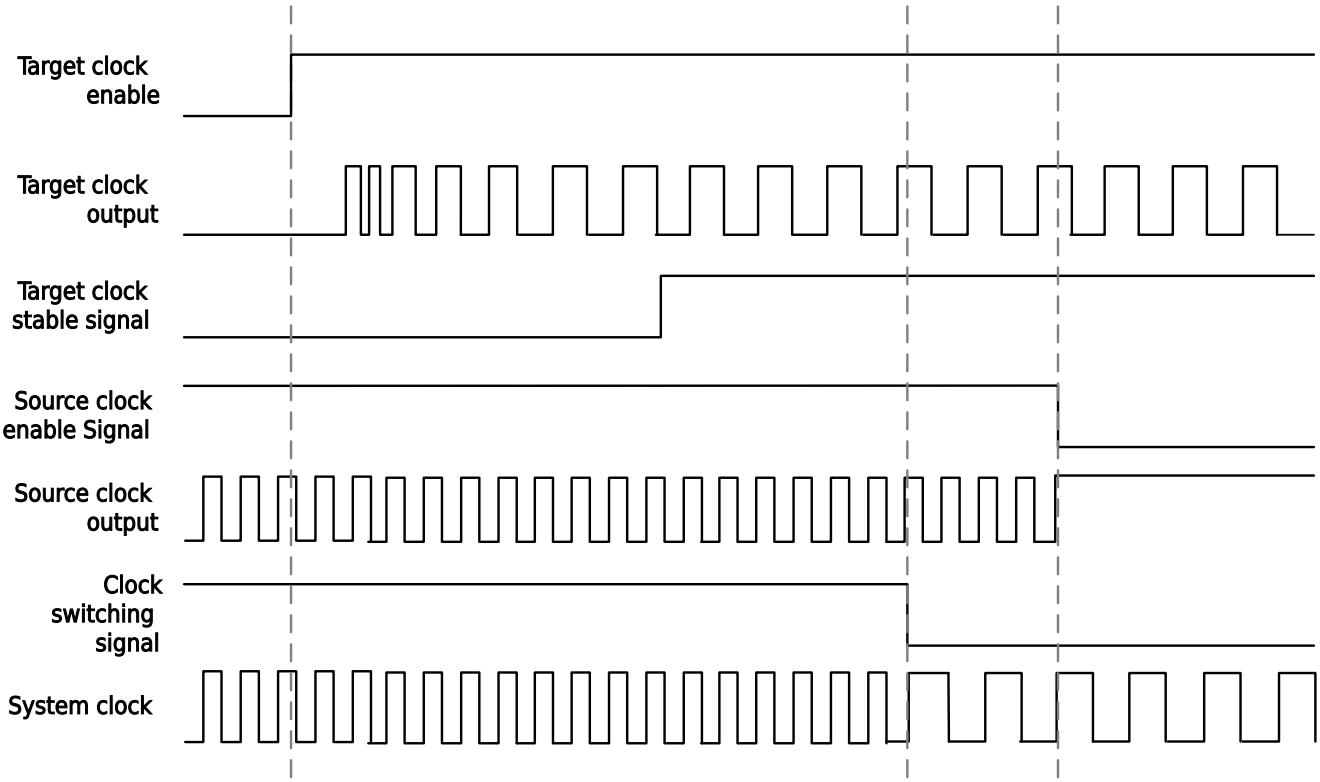


Figure 4-4 Clock Switching Diagram

4.3 On-Chip Peripheral Clock Control

Most on-chip peripherals require a clock to function properly. When the configuration clock (PCLK) of the on-chip peripherals is enabled, the user program can read and write the registers of the on-chip peripherals normally; when the working clock of the on-chip peripherals is enabled, the on-chip peripherals can work normally. The working clock of some on-chip peripherals is fixed to PCLK, and the working clock of some on-chip peripherals can be configured by the user program.

By setting the control bit of SYSCTRL_PERI_CLKENx to 1, the configuration clock of the corresponding on-chip peripherals can be enabled.

4.4 Interrupt Wakeup Control

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

The behavior of the processor in the Sleep 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

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

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

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 mode whenever exception handling is completed and returns to thread mode. With the sleep-on-exit feature, the processor can be in Sleep mode as much as possible.

Cortex-M0+ take use of the sleep-on-exit feature to enter Sleep. 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; when the reset is completed, query the RSTFLAG register to obtain the last reset source. Whenever the RSTFLAG register is queried, the register should be cleared so that the register can normally indicate its reset source during the next reset.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

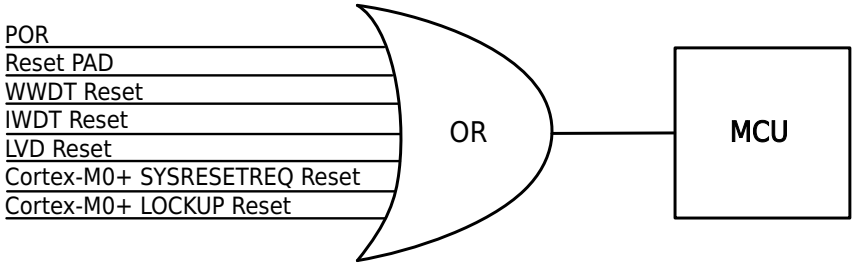


Figure 4-5 Reset Source Diagram

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 VCORE area. All analog modules and IO work in the VCC area; other modules work in the VCORE 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 BOR threshold voltage, POR5V signal is generated.
- When the VCORE area is powered on, when the VCORE voltage is lower than the POR threshold voltage, POR12V signal is generated; when the VCORE area is powered off, when the VCORE voltage is lower than the BOR threshold voltage, POR12V signal is generated.

Both the POR5V signal and the POR12V 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.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

4.6 Register Description

4.6.1 Register List

Base address: 0x40002000

Table 4-2 System Controller (SYSCTRL) Register Offset

Offset	Register	Description	Width
0x000	SYSCTRL_CR0	System Control Register 0	32
0x004	SYSCTRL_CR1	System Control Register 1	32
0x008	SYSCTRL_CR2	System Control Register 2	32
0x00C	SYSCTRL_CR3	System Control Register 3	32
0x010	SYSCTRL_RCHCR	RCH Control Register	32
0x014	SYSCTRL_XTHCR	XTH Control Register	32
0x018	SYSCTRL_RCLCR	RCL Control Register	32
0x01C	SYSCTRL_XTLCR	XTL Control Register	32
0x020	SYSCTRL_PLLCR	PLL Control Register	32
0x024	SYSCTRL_RCHTRIM	RCH Trim Value Register	32
0x028	SYSCTRL_RCLTRIM	RCL Trim Value Register	32
0x02C	SYSCTRL_STATE	Clock Status Register	32
0x030	SYSCTRL_PERI_CLKEN0	Peripheral Module Clock Control Register 0	32
0x034	SYSCTRL_PERI_CLKEN1	Peripheral Module Clock Control Register 1	32
0x038	SYSCTRL_PERI_RESET0	Peripheral Module Reset Control Register 0	32
0x03C	SYSCTRL_PERI_RESET1	Peripheral Module Reset Control Register 1	32
0x040	SYSCTRL_RSTFLAG	Reset Flag Register	32
0x044	SYSCTRL_DBG_STOP	Peripheral Debug Function Enable Register	32
0x070	SYSCTRL_IOAUX0	GPIO Auxiliary Control Register 0	32
0x074	SYSCTRL_IOAUX1	GPIO Auxiliary Control Register 1	32
0x078	SYSCTRL_IOAUX2	GPIO Auxiliary Control Register 2	32

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

4.6.2 System Control Register 0 (SYSCTRL_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																															
Name	Reserved																								PCLKPRS		HCLKPRS			CLKSRC		
Access																									RW		RW			RW		

Bit	Name	Functional Description	Access
31:8	*Reserved*	-	-
7:6	PCLKPRS	PCLK clock source selection ● 0b00: HCLK ● 0b01: HCLK/2 ● 0b10: HCLK/4 ● 0b11: HCLK/8	RW
5:3	HCLKPRS	HCLK clock source selection ● 0b000: SystemClk ● 0b001: SystemClk/2 ● 0b010: SystemClk/4 ● 0b011: SystemClk/8 ● 0b100: SystemClk/16 ● 0b101: SystemClk/32 ● 0b110: SystemClk/64 ● 0b111: SystemClk/128	RW
2:0	CLKSRC	SystemClk clock source selection ● 0b000: Internal high-speed clock RCH ● 0b001: External high-speed crystal oscillator XTH ● 0b010: Internal low-speed clock RCL ● 0b011: External low-speed crystal oscillator XTL ● 0b100: Internal PLL ● Others: Setting prohibited	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

4.6.3 System Control Register 1 (SYSCTRL_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																			FASTIOENB	GTIM1CFG	GTIM0CFG	XTHFLT	SPIPCTRL	AUTOSWITCH	RTCADJ		RTC_LPW	WAKEUPBYRCH	LOCKUP_EN	SWD_USE_IO
Access																				RW	RW	RW	RW	RW	RW	RW		RW	RW	RW	RW

Bit	Name	Functional Description	Access
31:13	*Reserved*	-	-
12	FASTIOENB	FASTIO function selection ● 0b0: FASTIO ● 0b1: General IO	RW
11	GTIM1CFG	GTIM1 function selection ● 0b0: GTIM1 is configured as a general timer ● 0b1: GTIM1 is configured as three basic timers	RW
10	GTIM0CFG	GTIM0 function selection ● 0b0: GTIM0 is configured as a general timer ● 0b1: GTIM0 is configured as three basic timers	RW
9	XTHFLT	XTH clock filter control ● 0b0: XTH filtering ● 0b1: No filtering **Note** When the system clock is XTH, operating this control bit is prohibited.	RW
8	SPIPCTRL	SPI port direction control ● 0b0: SPI control ● 0b1: GPIO control	RW
7	AUTOS-WITCH	When the system clock source is crystal and the crystal fails, automatically switch the system clock source configuration ● 0b0: Keep the system clock source unchanged (when PLL is used as the system clock, it needs to be set to 1) ● 0b1: Switch the system clock source to RCH	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Bit	Name	Functional Description	Access
6:4	RTCADJ	RTC high-speed clock compensation clock frequency selection ● 0b000: 4M ● 0b001: 6M ● 0b010: 8M ● 0b011: 12M ● 0b100: 16M ● 0b101: 20M ● 0b110: 24M ● 0b111: 32M	RW
3	RTC_LPW	RTC module low power control ● 0b0: Low power mode disabled ● 0b1: Low power mode enabled **Note** After enabling, RTC module register writing and RTC counting are not synchronized. After enabling, writing to the RTC register is prohibited.	RW
2	WAKEUP-BYRCH	System clock source control bit after wake-up ● 0b0: After waking up from DeepSleep, do not change the system clock source ● 0b1: After waking up from DeepSleep, the system clock source is RCH, and the original clock continues to be enabled	RW
1	LOCKUP_EN	Cortex-M0+ LockUp function configuration ● 0b0: Disable ● 0b1: Enable **Note** When enabled, the CPU will reset the MCU when it reads an invalid command, which can enhance system reliability.	RW
0	SWD_USE_IO	SWD port function configuration ● 0b0: SWD port ● 0b1: GPIO port	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

4.6.4 System Control Register 2 (SYSCTRL_CR2)

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	0x00000081																															
Name	Reserved																								XTLLOCK	EXXTL	EXXTH	PLL	XTL	RCL	XTH	RCH
Access																									RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Functional Description	Access
31:8	*Reserved*	-	-
7	XTLLOCK	XTL advanced enable control ● 0b0: CR2.XTL can be set and cleared ● 0b1: CR2.XTL can only be set	RW
6	EXXTL	External XTL clock input control ● 0b0: XTL output clock is generated by crystal oscillator ● 0b1: XTL output clock sourced from PC14 input **Note** When using PC14 to input clock, PC14 needs to be set as a digital input port, and CR2.XTL is 0.	RW
5	EXXTH	External XTH input control ● 0b0: XTH output clock is generated by crystal oscillator ● 0b1: XTH output clock sourced from PD00 input **Note** When using PD00 to input clock, PD00 needs to be set as a digital input port, and CR2.XTH is 0.	RW
4	PLL	PLL enable control ● 0b0: Disable ● 0b1: Enable **Note** PLL can only be enabled after BGR is enabled and stable.	RW
3	XTL	External low-speed crystal oscillator XTL enable control ● 0b0: Disable ● 0b1: Enable **Note** PC14 and PC15 need to be set as analog ports.	RW
2	RCL	Internal low-speed clock RCL enable control ● 0b0: Disable ● 0b1: Enable	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Bit	Name	Functional Description	Access
1	XTH	External high-speed crystal oscillator XTH enable control ● 0b0: Disable ● 0b1: Enable **Note** When the system enters DeepSleep, this high-speed clock will be automatically turned off.	RW
0	RCH	Internal high-speed clock RCH enable signal ● 0b0: Disabled ● 0b1: Enable **Note** When the system enters DeepSleep, this high-speed clock will be automatically turned off.	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

4.6.5 System Control Register 3 (SYSCTRL_CR3)

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																CR3															
Access																	RW															

Bit	Name	Functional Description	Access
31:16	*Reserved*	-	-
15:0	CR3	Register CR0/1/2 protection series control register Write 0x5A5A followed by 0xA5A5 to CR3 to enable write access to registers CR0/1/2. Once any write operation is performed on CR0/1/2, the protection bit automatically reverts to the protected state. Subsequent write operations to CR0/1/2 require repeating the same procedure.	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

4.6.6 RCH Control Register (SYSCTRL_RCHCR)

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	0x00000002																															
Name	Reserved																														DIV	
Access																															RW	

Bit	Name	Functional Description	Access
31:3	*Reserved*	-	-
2:0	DIV	RCH clock frequency division control ● 0b00: No division ● 0b01: 2 division frequency ● 0b10: 3 division frequency ● 0b11: 4 division frequency	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

4.6.7 XTH Control Register (SYSCTRL_XTHCR)

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	0x00000022																															
Name	Reserved																							DETIE	DETTIME	DETEN	STARTUP		RANGE		DRIVER	
Access																								RW	RW	RW	RW		RW		RW	

Bit	Name	Functional Description	Access
31:9	*Reserved*	-	-
8	DETIE	Failed interrupt enable control ● 0b0: Disable Fault and Fail interrupts ● 0b1: Enable Fault and Fail interrupts	RW
7	DETTIME	Crystal failure detection interval selection ● 0b0: Gear 1, 2ms detection interval ● 0b1: Gear 2, 8ms detection interval	RW
6	DETEN	Oscillation failure detection and work failure detection enabling control ● 0b0: Disable ● 0b1: Enable	RW
5:4	STARTUP	External high-speed clock XTH stable time selection ● 0b00: 8192 cycles ● 0b01: 32768 cycles ● 0b10: 131072 cycles ● 0b11: 262144 cycles **Note** It is highly recommended to set the stable time of XTH to 0b11. If the XTH stable time is insufficient, the system will not work stably when performing clock switching or waking up from DeepSleep.	RW
3:2	RANGE	XTH crystal oscillator operating frequency selection ● 0b00: 8M-12M ● 0b01: 8M-16M ● 0b10: 16M-24M ● 0b11: 20M-24M **Note** The above frequency range configuration is for reference only, users should make appropriate adjustments according to actual applications.	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Bit	Name	Functional Description	Access
1:0	DRIVER	XTH crystal oscillator driving capability selection ● 0b00: Weakest driving capability ● 0b01: Weak driving capability ● 0b10: Default driving capability (recommended value) ● 0b11: Strongest driving capability **Note** 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.

4.6.8 RCL Control Register (SYSCTRL_RCLCR)

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																				STARTUP		Reserved									
Access																					RW											

Bit	Name	Functional Description	Access
31:12	*Reserved*	-	-
11:10	STARTUP	Internal low-speed clock RCL stable time selection ● 0b00: 4 cycles ● 0b01: 16 cycles ● 0b10: 64 cycles ● 0b11: 256 cycles	RW
9:0	*Reserved*	-	-

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

4.6.9 XTL Control Register (SYSCTRL_XTLCR)

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	0x00000021																															
Name	Reserved																							DETIE	DETTIME	DETEN	STARTUP		AMP_SEL		DRIVER	
Access																								RW	RW	RW	RW		RW		RW	

Bit	Name	Functional Description	Access
31:9	*Reserved*	-	-
8	DETIE	Failed interrupt enable control ● 0b0: Disable Fault and Fail interrupts ● 0b1: Enable Fault and Fail interrupts	RW
7	DETTIME	Crystal failure detection interval selection ● 0b0: Gear 1, 2ms detection interval ● 0b1: Gear 2, 8ms detection interval	RW
6	DETEN	Oscillation failure detection and work failure detection enabling control ● 0b0: Disable ● 0b1: Enable	RW
5:4	STARTUP	External low-speed crystal oscillator XTL stable time selection ● 0b00: 256 cycles ● 0b01: 1024 cycles ● 0b10: 4096 cycles ● 0b11: 16384 cycles	RW
3:2	AMP_SEL	XTL crystal oscillation amplitude adjustment ● 0b00: Minimum amplitude ● 0b01: Normal amplitude ● 0b10: Larger amplitude (Recommended value) ● 0b11: Maximum amplitude	RW
1:0	DRIVER	XTL crystal oscillator driving capability selection ● 0b00: Weakest driving capability ● 0b01: Normal driving capability (Recommended value) ● 0b10: Stronger driving capability ● 0b11: Strongest driving capability **Note** 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.

4.6.10 PLL Control Register (SYSCTRL_PLLCR)

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	0x00802600																															
Name	Reserved								POSTDIV				Reserved		DIVF		Reserved			DIVN					Reserved		WAITCYCLE		Reserved		DIVR	SRC
Access									RW						RW					RW							RW			RW	RW

Bit	Name	Functional Description	Access
31:24	*Reserved*	-	-
23:20	POSTDIV	Output frequency division coefficient Out of PLL loop ● 0b0000: 2 ● 0b0001: 3 ● 0b0010: 4 ● 0b0011: 5 ● .... ● 0b1111: 17 **Note** The final output frequency division coefficient= the value of POSTDIV + 2, and F_{out}= F_{vco}/(2 × final output frequency division coefficient). For example, if F_{vco}= 384MHz and POSTDIV is set to 1, then the final PLL output F_{out}= 384/(2 × (1 + 2)) = 64MHz.	RW
19	*Reserved*	-	-
18:16	DIVF	Decimal division ● 0b000:0 ● 0b001:1/8 ● ..... ● 0b111:7/8	RW
15	*Reserved*	-	-

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Bit	Name	Functional Description	Access
14:8	DIVN	Integer division ● 0b0000000: 2 ● 0b0000001: 3 ● ..... ● 0b1111110: 128 ● 0b1111111: Setting prohibited **Note** The integer division coefficient DIV= DIVN+2, so when DIVN is set to 0x00, DIV=2, and so on. For example, with an input clock of 6MHz, DIVN set to 0x3E, and DIVF set to 0x0, the multiplied clock F_{vco}= 6 × (62 + 2) = 384MHz. The recommended selectable range for DIVN is 35 to 123.	RW
7	*Reserved*	-	-
6:4	WAITCYCLE	PLL stable time selection ● 0b000: 128 PLL cycles ● 0b001: 256 PLL cycles ● 0b010: 512 PLL cycles ● 0b011: 1024 PLL cycles ● 0b100: 2048 PLL cycles ● 0b101: 4096 PLL cycles ● 0b110: 8192 PLL cycles ● 0b111: 16384 PLL cycles	RW
3	*Reserved*	-	-
2	DIVR	Source clock prescaler ● 0b0: 1 ● 0b1: 2 **Note** As a frequency division of the input clock source, DIVR can be set to 0 or 1. If the source clock RCH is 12MHz, setting DIVR to 0 results in an input clock of 12MHz; setting DIVR to 1 results in an input clock of 6MHz. It should be ensured that the input clock frequency divided by (DIVR+1) is between 4MHz and 12MHz.	RW
1:0	SRC	PLL input clock source selection ● 0b00: RCH 12MHz clock ● 0b01: XTH clock ● 0b1X: Setting prohibited	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

4.6.11 RCH Trim Value Register (SYSCTRL_RCHTRIM)

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	0x00000XXX																															
Name	Reserved																					TRIMH					TRIML					
Access																						RW					RW					

Bit	Name	Functional Description	Access
31:11	*Reserved*	-	-
10:6	TRIMH	RCH frequency calibration value (coarse adjustment) Clock frequency adjustment, changing the value of this register can adjust the basic frequency of RCH. Each time the register value increases by 1, the RCH output frequency increases by approximately 3.5%. The calibration value of 12MHz has been saved in Flash. Read the calibration value in Flash and write it into RCHTRIM to obtain the accurate frequency.	RW
5:0	TRIML	RCH frequency calibration value (fine adjustment) Clock frequency adjustment, changing the value of this register can adjust the basic frequency of RCH. Each time the register value increases by 1, the RCH output frequency increases by approximately 0.08%. The calibration value of 12MHz has been saved in Flash. Read the calibration value in Flash and write it into RCHTRIM to obtain the accurate frequency. **Note** The calibration value for RCH 12M is stored at the Flash address range: 0x00101BE1-0x00101BE0. Envolves: ● TRIMH=RCH_TRIM[10:6] ● TRIML=RCH_TRIM[5:0] When writing, it is recommended to write the entire group to the SYSCTRL_RCHTRIM register.	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

4.6.12 RCL Trim Value Register (SYSCTRL_RCLTRIM)

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	0x00000XXX																															
Name	Reserved																						TRIM									
Access																							RW									

Bit	Name	Functional Description	Access
31:10	*Reserved*	-	-
9:0	TRIM	RCL frequency calibration value Internal low-speed clock frequency adjustment, 2 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 RCLTRIM. ● 38.4kHz calibration value address: 0x00101BE9-0x00101BE8 ● 32.768kHz calibration value address: 0x00101BEB-0x00101BEA	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

4.6.13 Clock Status Register (SYSCTRL_STATE)

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	0x00000001																															
Name	Reserved																				XTLFAULT	XTLFAIL	XTHFAULT	XTHFAIL	Reserved			PLL	XTL	RCL	XTH	RCH
Access																					RW	RW	RW	RW				RO	RO	RO	RO	RO

Bit	Name	Functional Description	Access
31:12	*Reserved*	-	-
11	XTLFAULT	XTL work flag Write 0 to clear flag, write 1 has no effect Read: ● 0b0: Works fine ● 0b1: Failure during operation	RW
10	XTLFAIL	XTL startup flag Write 0 to clear flag, write 1 has no effect Read: ● 0b0: Oscillation startup successful ● 0b1: Oscillation startup failure	RW
9	XTHFAULT	XTH work flag Write 0 to clear flag, write 1 has no effect Read: ● 0b0: Works fine ● 0b1: Failure during operation	RW
8	XTHFAIL	XTH startup flag Write 0 to clear flag, write 1 has no effect Read: ● 0b0: Oscillation startup successful ● 0b1: Oscillation startup failure	RW
7:5	*Reserved*	-	-
4	PLL	PLL stable state ● 0b0: Unstable ● 0b1: Stable	RO
3	XTL	XTL clock stable state ● 0b0: Unstable ● 0b1: Stable	RO
2	RCL	RCL clock stable state ● 0b0: Unstable ● 0b1: Stable	RO

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Bit	Name	Functional Description	Access
1	XTH	XTH clock stable state ● 0b0: Unstable ● 0b1: Stable	RO
0	RCH	RCH clock stable state ● 0b0: Unstable ● 0b1: Stable	RO

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

4.6.14 Peripheral Module Clock Control Register 0 (SYSCTRL_PERI_CLKEN0)

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	0x80800000																															
Name	FLASH	EAU	DMA	Reserved		CRC	Reserved				CTRIM	RTC	Reserved	I2CSLV	VCLVD	ADC	IWDT	WWDT	OPA	ATIM2	ATIM3	ATIM1	LPTIM	ATIM0	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	RW	RW

Bit	Name	Functional Description	Access
31	FLASH	FLASH controller module clock enable bit ● 0b0: Disable ● 0b1: Enable **Note** After closing, the FLASH configuration registers cannot be written, and the programs in the FLASH can still run.	RW
30	EAU	EAU module clock enable bit ● 0b0: Disable ● 0b1: Enable	RW
29	DMA	DMAC module clock enable bit ● 0b0: Disable ● 0b1: Enable	RW
28:27	*Reserved*	-	-
26	CRC	CRC module clock enable bit ● 0b0: Disable ● 0b1: Enable	RW
25:22	*Reserved*	-	-
21	CTRIM	CTRIM module clock enable bit ● 0b0: Disable ● 0b1: Enable	RW
20	RTC	RTC module clock enable bit ● 0b0: Disable ● 0b1: Enable	RW
19	*Reserved*	-	-
18	I2CSLV	I2CSLV module clock enable bit ● 0b0: Disable ● 0b1: Enable	RW
17	VCLVD	VC, LVD module clock enable bit ● 0b0: Disable ● 0b1: Enable	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Bit	Name	Functional Description	Access
16	ADC	ADC, BGR module clock enable bit ● 0b0: Disable ● 0b1: Enable	RW
15	IWDT	IWDT module clock enable bit ● 0b0: Disable ● 0b1: Enable	RW
14	WWDT	WWDT module clock enable bit ● 0b0: Disable ● 0b1: Enable	RW
13	OPA	OPA module clock enable bit ● 0b0: Disabled ● 0b1: Enable	RW
12	ATIM2	ATIM2 module clock enable bit ● 0b0: Disable ● 0b1: Enable	RW
11	ATIM3	ATIM3 module clock enable bit ● 0b0: Disable ● 0b1: Enable	RW
10	ATIM1	ATIM1 module clock enable bit ● 0b0: Disable ● 0b1: Enable	RW
9	LPTIM	LPTIM module clock enable bit ● 0b0: Disable ● 0b1: Enable	RW
8	ATIM0	ATIM0 module clock enable bit ● 0b0: Disable ● 0b1: Enable	RW
7	SPI1	SPI1 module clock enable bit ● 0b0: Disable ● 0b1: Enable	RW
6	SPI0	SPI0 module clock enable bit ● 0b0: Disable ● 0b1: Enable	RW
5	I2C1	I2C1 module clock enable bit ● 0b0: Disable ● 0b1: Enable	RW
4	I2C0	I2C0 module clock enable bit ● 0b0: Disable ● 0b1: Enable	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Bit	Name	Functional Description	Access
3	LPUART1	LPUART1 module clock enable bit ● 0b0: Disable ● 0b1: Enable	RW
2	LPUART0	LPUART0 module clock enable bit ● 0b0: Disable ● 0b1: Enable	RW
1	USART1	USART1 module clock enable bit ● 0b0: Disable ● 0b1: Enable	RW
0	USART0	USART0 module clock enable bit ● 0b0: Disable ● 0b1: Enable	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

4.6.15 Peripheral Module Clock Control Register 1 (SYSCTRL_PERI_CLKEN1)

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																						CTIM1	CTIM0	CAN	Reserved			GPIOD	GPIOC	GPIOB	GPIOA
Access																							RW	RW	RW				RW	RW	RW	RW

Bit	Name	Functional Description	Access
31:10	*Reserved*	-	-
9	CTIM1	CTIM1 module clock enable bit ● 0b0: Disable ● 0b1: Enable	RW
8	CTIM0	CTIM0 module clock enable bit ● 0b0: Disable ● 0b1: Enable	RW
7	CAN	CAN module clock enable bit ● 0b0: Disable ● 0b1: Enable	RW
6:4	*Reserved*	-	-
3	GPIOD	GPIOD module clock enable bit ● 0b0: Disable ● 0b1: Enable	RW
2	GPIOC	GPIOC module clock enable bit ● 0b0: Disable ● 0b1: Enable	RW
1	GPIOB	GPIOB module clock enable bit ● 0b0: Disable ● 0b1: Enable	RW
0	GPIOA	GPIOA module clock enable bit ● 0b0: Disable ● 0b1: Enable	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

4.6.16 Peripheral Module Reset Control Register 0 (SYSCTRL_PERI_RESET0)

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	0x6737FFFF																															
Name	Reserved	EAU	DMA	Reserved		CRC	Reserved				CTRIM	RTC	Reserved	I2CSLV	VCLVD	ADC	Reserved	WWDT	OPA	ATIM2	ATIM3	ATIM1	LPTIM	ATIM0	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	*Reserved*	-	-
30	EAU	EAU module reset enable bit ● 0b0: The module is in reset state ● 0b1: Operate normally	RW
29	DMA	DMAC module reset enable bit ● 0b0: The module is in reset state ● 0b1: Operate normally	RW
28:27	*Reserved*	-	-
26	CRC	CRC module reset enable bit ● 0b0: The module is in reset state ● 0b1: Operate normally	RW
25:22	*Reserved*	-	-
21	CTRIM	CTRIM module reset enable bit ● 0b0: The module is in reset state ● 0b1: Operate normally	RW
20	RTC	RTC module reset enable bit ● 0b0: The module is in reset state ● 0b1: Operate normally	RW
19	*Reserved*	-	-
18	I2CSLV	I2CSLV module reset enable bit ● 0b0: The module is in reset state ● 0b1: Operate normally	RW
17	VCLVD	VC module reset enable bit ● 0b0: The module is in reset state ● 0b1: Operate normally **Note** The LVD module cannot be reset.	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Bit	Name	Functional Description	Access
16	ADC	ADC, BGR module reset enable bit ● 0b0: The module is in reset state ● 0b1: Operate normally	RW
15	*Reserved*	-	-
14	WWDT	WWDT module reset enable bit ● 0b0: The module is in reset state ● 0b1: Operate normally	RW
13	OPA	OPA module reset enable bit ● 0b0: The module is in reset state ● 0b1: Operate normally	RW
12	ATIM2	ATIM2 module reset enable bit ● 0b0: The module is in reset state ● 0b1: Operate normally	RW
11	ATIM3	ATIM3 module reset enable bit ● 0b0: The module is in reset state ● 0b1: Operate normally	RW
10	ATIM1	ATIM1 module reset enable bit ● 0b0: The module is in reset state ● 0b1: Operate normally	RW
9	LPTIM	LPTIM module reset enable bit ● 0b0: The module is in reset state ● 0b1: Operate normally	RW
8	ATIM0	ATIM0 module reset enable bit ● 0b0: The module is in reset state ● 0b1: Operate normally	RW
7	SPI1	SPI1 module reset enable bit ● 0b0: The module is in reset state ● 0b1: Operate normally	RW
6	SPI0	SPI0 module reset enable bit ● 0b0: The module is in reset state ● 0b1: Operate normally	RW
5	I2C1	I2C1 module reset enable bit ● 0b0: The module is in reset state ● 0b1: Operate normally	RW
4	I2C0	I2C0 module reset enable bit ● 0b0: The module is in reset state ● 0b1: Operate normally	RW
3	LPUART1	LPUART1 module reset enable bit ● 0b0: The module is in reset state ● 0b1: Operate normally	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Bit	Name	Functional Description	Access
2	LPUART0	LPUART0 module reset enable bit ● 0b0: The module is in reset state ● 0b1: Operate normally	RW
1	USART1	USART1 module reset enable bit ● 0b0: The module is in reset state ● 0b1: Operate normally	RW
0	USART0	USART0 module reset enable bit ● 0b0: The module is in reset state ● 0b1: Operate normally	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

4.6.17 Peripheral Module Reset Control Register 1 (SYSCTRL_PERI_RESET1)

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	0x0000038F																															
Name	Reserved																						CTIM1	CTIM0	CAN	Reserved			GPIOD	GPIOC	GPIOB	GPIOA
Access																							RW	RW	RW				RW	RW	RW	RW

Bit	Name	Functional Description	Access
31:10	*Reserved*	-	-
9	CTIM1	CTIM1 module reset enable bit ● 0b0: The module is in reset state ● 0b1: Operate normally	RW
8	CTIM0	CTIM0 module reset enable bit ● 0b0: The module is in reset state ● 0b1: Operate normally	RW
7	CAN	CAN module reset enable bit ● 0b0: The module is in reset state ● 0b1: Operate normally	RW
6:4	*Reserved*	-	-
3	GPIOD	GPIOD module reset enable bit ● 0b0: The module is in reset state ● 0b1: Operate normally	RW
2	GPIOC	GPIOC module reset enable bit ● 0b0: The module is in reset state ● 0b1: Operate normally	RW
1	GPIOB	GPIOB module reset enable bit ● 0b0: The module is in reset state ● 0b1: Operate normally	RW
0	GPIOA	GPIOA module reset enable bit ● 0b0: The module is in reset state ● 0b1: Operate normally	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

4.6.18 Reset Flag Register (SYSCTRL_RSTFLAG)

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																													
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	0x00000003																															
Name	Reserved																								RSTB	SYSREQ	LOCKUP	WWDT	IWDT	LVD	POR12V	POR5V
Access																									RW	RW	RW	RW	RW	RW	RW	RW

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

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Bit	Name	Functional Description	Access
2	LVD	LVD reset flag Write 0 to clear flag, write 1 has no effect Read: ● 0b0: No reset occurred ● 0b1: Reset occurred	RW
1	POR12V	VCAP domain reset flag Write 0 to clear flag, write 1 has no effect Read: ● 0b0: No reset occurred ● 0b1: Reset occurred	RW
0	POR5V	VCC power domain reset flag Write 0 to clear flag, write 1 has no effect Read: ● 0b0: No reset occurred ● 0b1: Reset occurred	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

4.6.19 Peripheral Debug Function Enable Register (SYSCTRL_DBG_STOP)

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	0x00001FBF																															
Name	Reserved																			WWDT	ATIM3	Reserved	RTC	IWDT	CTRIM	Reserved	CTIM1	CTIM0	LPTIM	ATIM2	ATIM1	ATIM0
Access																				RW	RW		RW	RW	RW		RW	RW	RW	RW	RW	RW

Bit	Name	Functional Description	Access
31:13	*Reserved*	-	-
12	WWDT	WWDT counting function configuration during debug ● 0b0: In the SWD debugging interface, the WWDT counts normally ● 0b1: In the SWD debugging interface, suspend the WWDT counting function	RW
11	ATIM3	ATIM3 counting function configuration during debug ● 0b0: In the SWD debugging interface, the ATIM3 counts normally ● 0b1: In the SWD debugging interface, suspend the ATIM3 counting function	RW
10	*Reserved*	-	-
9	RTC	RTC counting function configuration during debug ● 0b0: In the SWD debugging interface, the RTC counts normally ● 0b1: In the SWD debugging interface, suspend the RTC counting function	RW
8	IWDT	IWDT counting function configuration during debug ● 0b0: In the SWD debugging interface, the IWDT counts normally ● 0b1: In the SWD debugging interface, suspend the IWDT counting function	RW
7	CTRIM	CTRIM counting function configuration during debug ● 0b0: In the SWD debugging interface, the CTRIM counts normally ● 0b1: In the SWD debugging interface, suspend the CTRIM counting function	RW
6	*Reserved*	-	-
5	CTIM1	CTIM1 counting function configuration during debug ● 0b0: In the SWD debugging interface, the CTIM1 counts normally ● 0b1: In the SWD debugging interface, suspend the CTIM1 counting function	RW
4	CTIM0	CTIM0 counting function configuration during debug ● 0b0: In the SWD debugging interface, the CTIM0 counts normally ● 0b1: In the SWD debugging interface, suspend the CTIM0 counting function	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Bit	Name	Functional Description	Access
3	LPTIM	LPTIM counting function configuration during debug ● 0b0: In the SWD debugging interface, the LPTIM counts normally ● 0b1: In the SWD debugging interface, suspend the LPTIM counting function	RW
2	ATIM2	ATIM2 counting function configuration during debug ● 0b0: In the SWD debugging interface, the ATIM2 counts normally ● 0b1: In the SWD debugging interface, suspend the ATIM2 counting function	RW
1	ATIM1	ATIM1 counting function configuration during debug ● 0b0: In the SWD debugging interface, the ATIM1 counts normally ● 0b1: In the SWD debugging interface, suspend the ATIM1 counting function	RW
0	ATIM0	ATIM0 counting function configuration during debug ● 0b0: In the SWD debugging interface, the ATIM0 counts normally ● 0b1: In the SWD debugging interface, suspend the ATIM0 counting function	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

4.6.20 GPIO Auxiliary Control Register 0 (SYSCTRL_IOAUX0)

Offset	Bit Position																													
0x070	31	30	29	28	27	26	25	24	23	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																	MCOSEL			Reserved	MCODIV			Reserved					PCLKDIV		
Access																		RW				RW								RW		

Bit	Name	Functional Description	Access
31:15	*Reserved*	-	-
14:12	MCOSEL	MCO pin clock selection ● 0b000: RCH analog output ● 0b001: XTH analog output ● 0b010: RCL analog output ● 0b011: XTL analog output ● 0b100: PLL analog output ● 0b101: RC150K analog output ● 0b110/0b111: Setting prohibited	RW
11	*Reserved*	-	-
10:8	MCODIV	MCO frequency division control ● 0b000: Output disabled ● 0b001: Output directly ● 0b010: 8 frequency division output ● 0b011: 64 frequency division output ● Others: 256 frequency division output	RW
7:3	*Reserved*	-	-
2:0	PCLKDIV	PCLK frequency division control ● 0b000: Output disabled ● 0b001: Output directly ● 0b010: 2 frequency division output ● 0b011: 4 frequency division output ● Others: 8 frequency division output	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

4.6.21 GPIO Auxiliary Control Register 1 (SYSCTRL_IOAUX1)

Offset	Bit Position																													
0x074	31	30	29	28	27	26	25	24	23	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																				ATIM3ETR			Reserved		CTIM1ETR			Reserved		CTIM0ETR			Reserved		ATIM0ETR		
Access																					RW					RW					RW					RW		

Bit	Name	Functional Description	Access
31:15	*Reserved*	-	-
14:12	ATIM3ETR	ATIM3ETR input signal Refer to the "Port Multiplexing Function" in General-Purpose I/O (GPIO) or the "Input Interconnection" section in Advanced Timer (ATIM).	RW
11	*Reserved*	-	-
10:8	CTIM1ETR	CTIM1ETR input signal Refer to the "Port Multiplexing Function" in General-Purpose I/O (GPIO) or the "On-Chip Peripheral Interconnection" section in Composite Timer (CTIM).	RW
7	*Reserved*	-	-
6:4	CTIM0ETR	CTIM0ETR input signal Refer to the "Port Multiplexing Function" in General-Purpose I/O (GPIO) or the "On-Chip Peripheral Interconnection" section in Composite Timer (CTIM).	RW
3	*Reserved*	-	-
2:0	ATIM0ETR	ATIM0ETR input signal Refer to the "Port Multiplexing Function" in General-Purpose I/O (GPIO) or the "Input Interconnection" section in Advanced Timer (ATIM).	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

4.6.22 GPIO Auxiliary Control Register 2 (SYSCTRL_IOAUX2)

Offset	Bit Position																													
0x078	31	30	29	28	27	26	25	24	23	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																	ATIM3CH2B			Reserved			CTIM1CAP3			Reserved			CTIM0CAP3			Reserved			ATIM0CHA		
Access																		RW				RW				RW				RW		

Bit	Name	Functional Description	Access
31:15	*Reserved*	-	-
14:12	ATIM3CH2B	ATIM3CH2B input signal Refer to the "Port Multiplexing Function" in General-Purpose I/O (GPIO) or the "Input Interconnection" section in Advanced Timer (ATIM).	RW
11	*Reserved*	-	-
10:8	CTIM1CAP3	CTIM1CAP3 input signal Refer to the "Port Multiplexing Function" in General-Purpose I/O (GPIO) or the "On-Chip Peripheral Interconnection" section in Composite Timer (CTIM).	RW
7	*Reserved*	-	-
6:4	CTIM0CAP3	CTIM0CAP3 input signal Refer to the "Port Multiplexing Function" in General-Purpose I/O (GPIO) or the "On-Chip Peripheral Interconnection" section in Composite Timer (CTIM).	RW
3	*Reserved*	-	-
2:0	ATIM0CHA	ATIM0CHA input signal Refer to the "Port Multiplexing Function" in General-Purpose I/O (GPIO) or the "Input Interconnection" section in Advanced Timer (ATIM).	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:

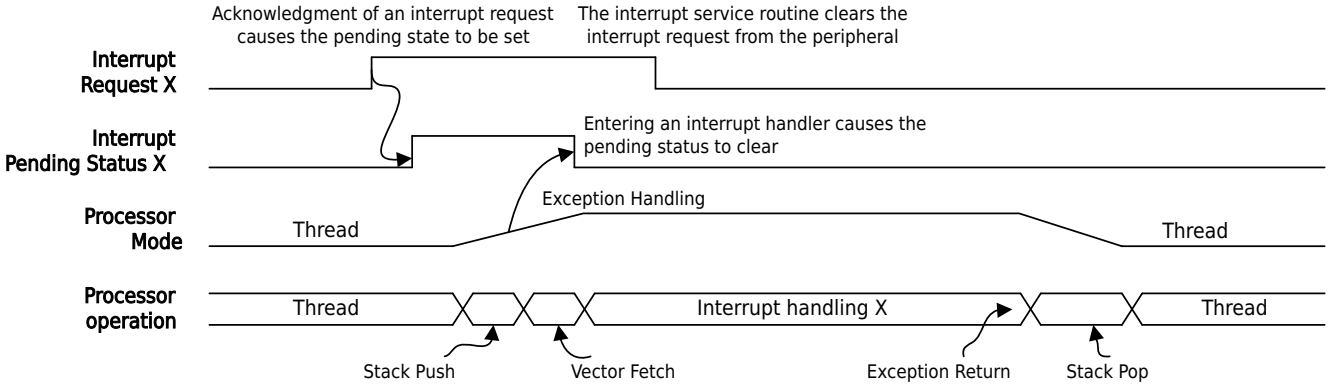


Figure 5-2 Interrupt Active and Pending States

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:

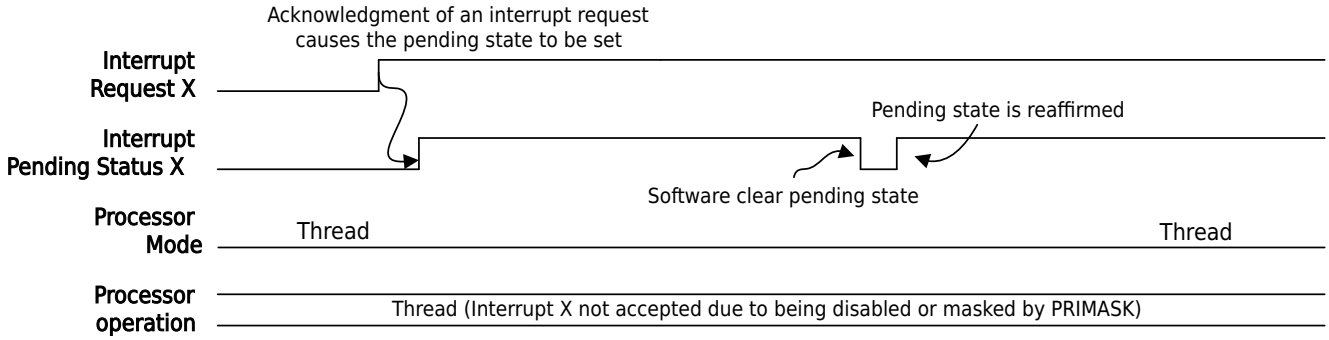


Figure 5-3 Interrupt Pending State is Cleared and Then Reasserted

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:

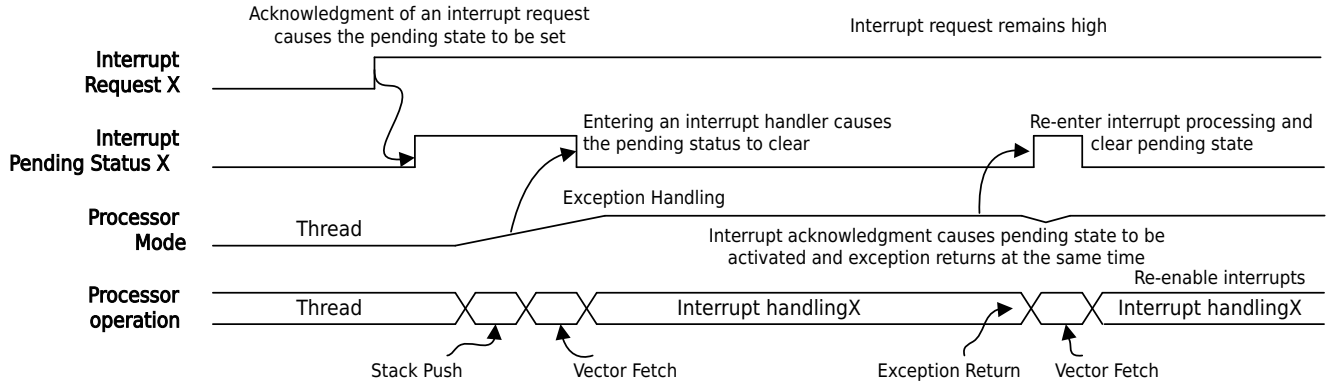


Figure 5-4 When the Interrupt Exits, If the Interrupt Request Remains High, It Will Cause the Interrupt Processing to Be Executed Again

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:

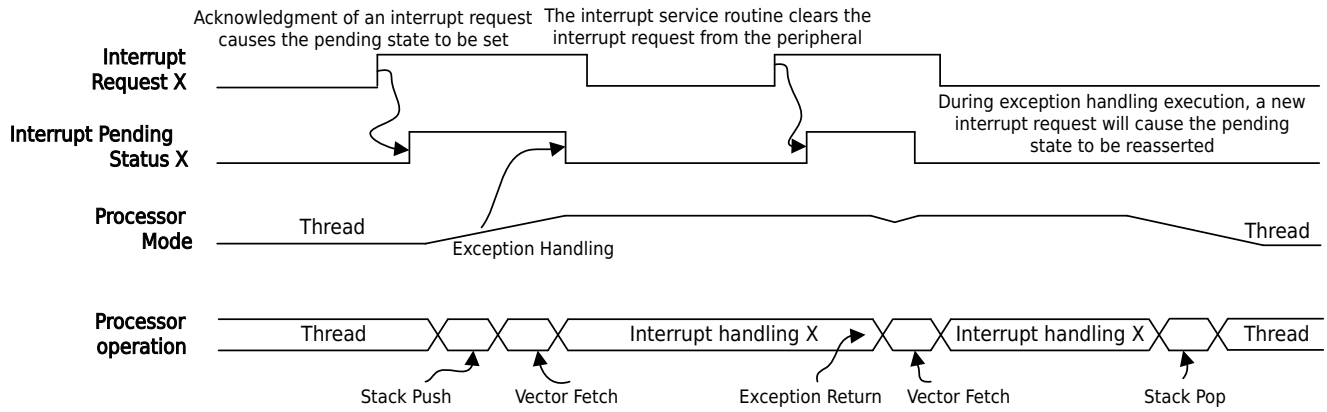


Figure 5-5 Interrupt Pending Interrupts Generated During Interrupt Processing Can Also Be Acknowledged

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	GPIOA	✓	✓	✓
Interrupt #1	GPIOB	✓	✓	✓
Interrupt #2	GPIOC	✓	✓	✓

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

NVIC Interrupt Input	External Interrupt Source	Active Mode	Sleep mode	DeepSleep mode
Interrupt #3	GPIOD	✓	✓	✓
Interrupt #4	DMAC	✓	✓	×
Interrupt #5	ATIM3	✓	✓	×
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	ATIM0	✓	✓	×
Interrupt #15	ATIM1	✓	✓	×
Interrupt #16	ATIM2	✓	✓	×
Interrupt #17	LPTIM	✓	✓	✓
Interrupt #18	GTIM0/BTIM012	✓	✓	×
Interrupt #19	GTIM1/BTIM345	✓	✓	×
Interrupt #20	I2CSLV	✓	✓	×
Interrupt #21	WWDT	✓	✓	×
Interrupt #22	IWDT	✓	✓	✓
Interrupt #23	RTC	✓	✓	✓
Interrupt #24	ADC	✓	✓	×
Interrupt #25	-	-	-	-
Interrupt #26	VC/LVD	✓	✓	✓
Interrupt #27	-	-	-	-
Interrupt #28	CAN	✓	✓	×
Interrupt #29	ATIM_BK	✓	✓	×
Interrupt #30	FLASH/RAM	✓	✓	×
Interrupt #31	CTRIM/FAULT	✓	✓	✓

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.

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.

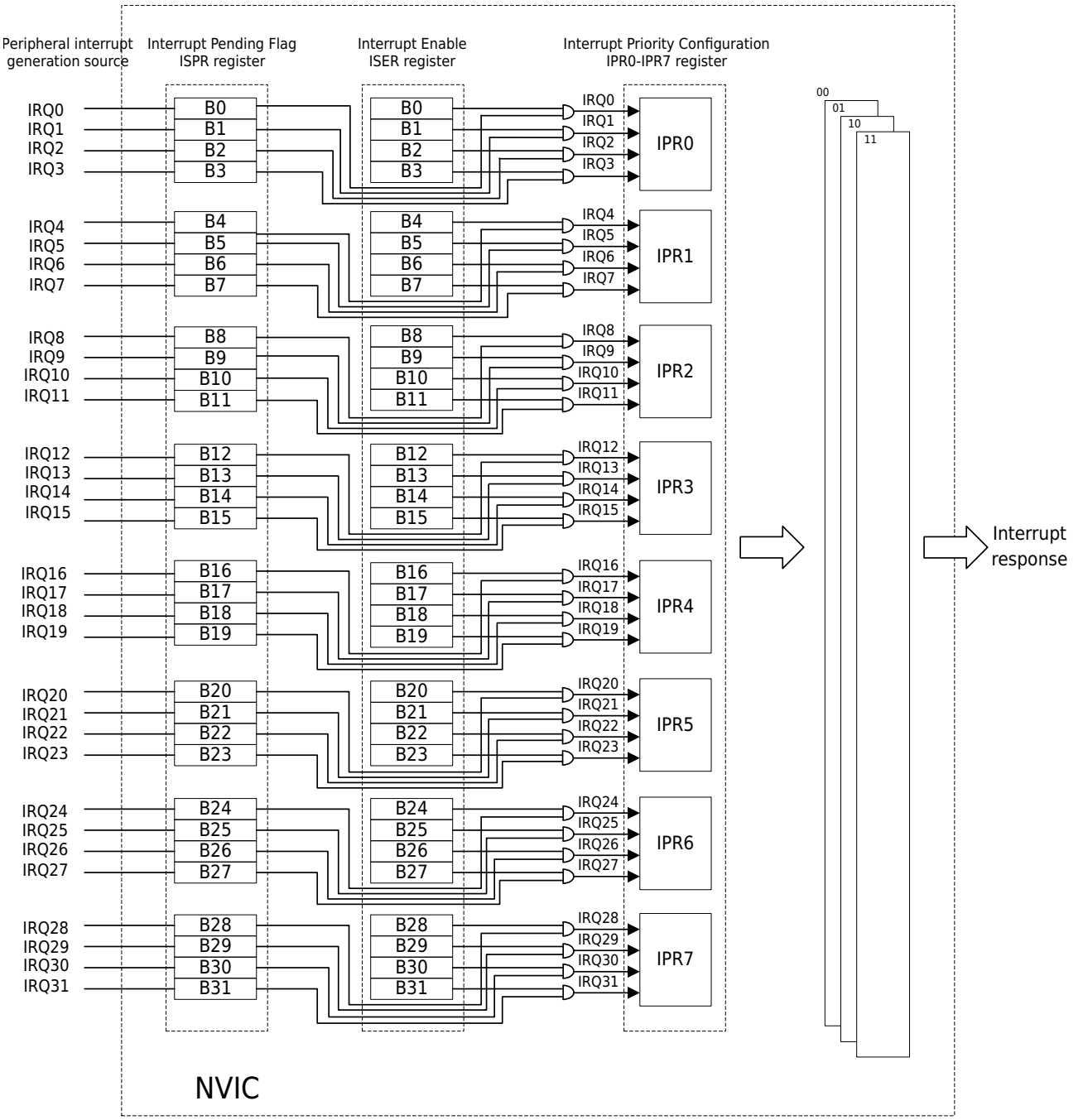


Figure 5-6 Interrupt Structure Diagram

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 Set enable interrupt #0 to interrupt #31 ● bit 0: IRQ0 ● bit 1: IRQ1 ● bit 2: IRQ2 ● ... ● bit 31: IRQ31 **Note** 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 Clear-enable interrupt #0 to interrupt #31 ● bit 0: IRQ0 ● bit 1: IRQ1 ● bit 2: IRQ2 ● ... ● bit 31: IRQ31 **Note** 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 Set the pending state of interrupt #0 to interrupt #31 ● bit 0: IRQ0 ● bit 1: IRQ1 ● bit 2: IRQ2 ● ... ● bit 31: IRQ31 **Note** 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 Clear the pending state of interrupt #0 to interrupt #31 ● bit 0: IRQ0 ● bit 1: IRQ1 ● bit 2: IRQ2 ● ... ● bit 31: IRQ31 **Note** 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 The corresponding bits for each interrupt priority are as follows: ● [31:30]: Priority of interrupt #3 ● [23:22]: Priority of interrupt #2 ● [15:14]: Priority of interrupt #1 ● [7:6]: Priority of interrupt #0 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 The corresponding bits for each interrupt priority are as follows: ● [31:30]: Priority of interrupt #7 ● [23:22]: Priority of interrupt #6 ● [15:14]: Priority of interrupt #5 ● [7:6]: Priority of interrupt #4 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 The corresponding bits for each interrupt priority are as follows: ● [31:30]: Priority of interrupt #11 ● [23:22]: Priority of interrupt #10 ● [15:14]: Priority of interrupt #9 ● [7:6]: Priority of interrupt #8 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 The corresponding bits for each interrupt priority are as follows: ● [31:30]: Priority of interrupt #15 ● [23:22]: Priority of interrupt #14 ● [15:14]: Priority of interrupt #13 ● [7:6]: Priority of interrupt #12 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 The corresponding bits for each interrupt priority are as follows: ● [31:30]: Priority of interrupt #19 ● [23:22]: Priority of interrupt #18 ● [15:14]: Priority of interrupt #17 ● [7:6]: Priority of interrupt #16 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 The corresponding bits for each interrupt priority are as follows: ● [31:30]: Priority of interrupt #23 ● [23:22]: Priority of interrupt #22 ● [15:14]: Priority of interrupt #21 ● [7:6]: Priority of interrupt #20 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 The corresponding bits for each interrupt priority are as follows: ● [31:30]: Priority of interrupt #27 ● [23:22]: Priority of interrupt #26 ● [15:14]: Priority of interrupt #25 ● [7:6]: Priority of interrupt #24 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 The corresponding bits for each interrupt priority are as follows: ● [31:30]: Priority of interrupt #31 ● [23:22]: Priority of interrupt #30 ● [15:14]: Priority of interrupt #29 ● [7:6]: Priority of interrupt #28 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 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. ● 0b0: All interrupts and exceptions are in the enabled state. ● 0b1: Only the NMI and hardware error exceptions are enabled.	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

6 GPIO

6.1 Introduction

GPIO is the most basic input and output channel of MCU. This product contains four sets of input and output ports, of which PA and PB are 16 bits, PC is 3 bits, and PD is 5 bits. For the actual number of available GPIO ports, please refer to the corresponding "Datasheet" for more information.

Each pin has associated control and configuration registers to meet specific application needs. By configuring the corresponding registers, the pins can be configured as the following three types of operating modes: analog peripheral input and output, digital peripheral input and output, and GPIO input and output.

6.2 Features

- Function multiplexing: each pin can be configured as multiple digital or analog functions
- Input mode: analog input, high-impedance input, pull-up input, pull-down input
- Output mode: analog output, push-pull output, open-drain output (pull-up can be enabled)
- Signal input: the input signal arrives at the data input register (GPIO_IN) and on-chip peripherals
- Signal output: the source of the output signal is the data output register (GPIO_OUT) or on-chip peripherals
- Atomic operations: bit operation registers (GPIO_BSRR, GPIO_BRR) realize atomic IO operations
- Interrupt type: rising edge interrupt, falling edge interrupt, high level interrupt, low level interrupt
- Auxiliary functions: output various internal clocks, redirect input signals of on-chip peripherals
- Input/output registers support FAST IO/AHB bus reading and writing



Note

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

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

6.3 Functional Description

6.3.1 Port Circuit Block Diagram

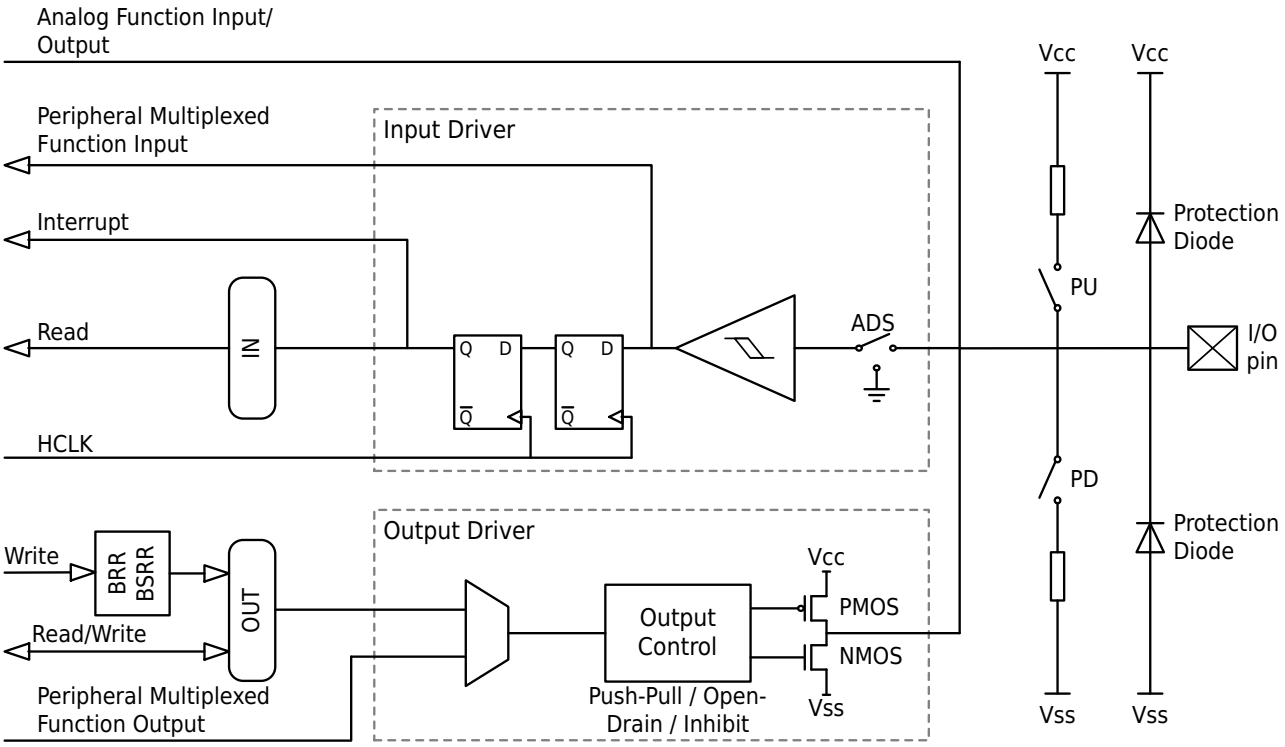


Figure 6-1 General Port Circuit Diagram

6.3.2 Port Mode Configuration

Through the control register, the port can be configured in various working modes, and its truth table is as follows:

Table 6-1 Port Function Configuration Table

Pin Mode	IN	ADS	AFRL/AFRH	DIR	OD	OUT	PU	PD
Analog	0	1	-	-	-	-	-	-
High-impedance input	X	0	0	1	-	-	0	0
Pull-up input	1	0	0	1	-	-	1	-
Pull-down input	0	0	0	1	-	-	0	1
Open-drain output	X	0	0	0	1	1	0	0
Output 0	0	0	0	0	0	0	-	-
Output 1	0	0	0	0	0	1	-	-
Peripheral output	X	0	>0	0	0	-	-	-
Peripheral pull-up input	X	0	>0	1	0	-	1	-
Peripheral pull-down input	X	0	>0	1	0	-	0	1

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Note

"-" stands for any value; "X" stands for indeterminate."

6.3.3 Port Reset Status

During reset and before the GPIO register is not configured after reset, most I/O ports are configured as analog input without pull-up and pull-down status; the status of other pins is as follows:

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

6.3.4 Port Analog Function

When you need to use the corresponding functions of the analog peripherals (ADC, LVD.....), you need to configure the port as an analog function. The implementation method is to set the corresponding bit of the GPIO_ADS register to 1, so that the on-chip analog peripherals are connected to this port.

Under this configuration, the port has the following characteristics:

- Analog peripherals are connected to ports
- The digital function of the port is closed: the level cannot be output; the read pin level is always 0; there is no built-in pull-up and pull-down

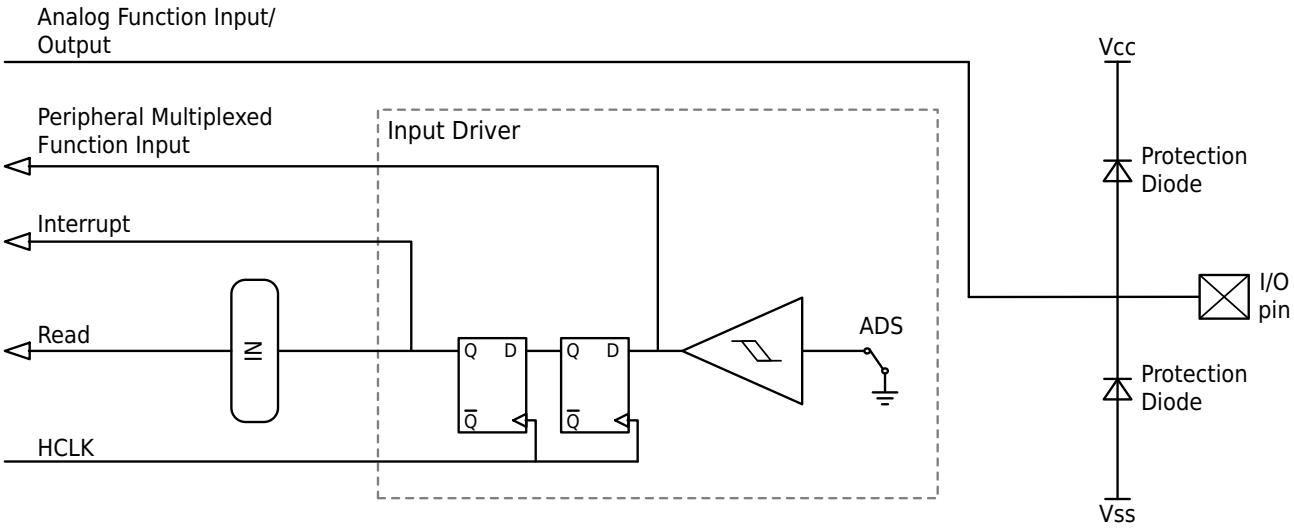


Figure 6-2 HiZ-Analog Configuration Diagram

6.3.5 Port General Input/Output Function

6.3.5.1 Functional Overview

When you need to use the general input and output function of the port, you need to configure the port as a digital function and set the value of the digital multiplexing function register to 0. The implementation method is to set the corresponding bit of the GPIO_ADS register to 0, and set the corresponding bit of the GPIO_AFRL and GPIO_AFRH register to 0. Under this configuration, the port has the following features:

- Input mode: pull-up or pull-down can be enabled
- Push-pull output: bit manipulation is supported
- Open-drain output: pull-up can be enabled; bit manipulation is supported

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Each GPIO has 5 control registers, whose functions are described as follows:

- GPIO_ADS register can configure the port as a digital port
- GPIO_DIR register can configure the port as input or output
- GPIO_PUPD.PU register can enable the pull-up of the port
- GPIO_PUPD.PD register can enable the pull-down of the port
- GPIO_OD register can configure the port output type as push-pull or open-drain
- GPIO_DR register output port driving capability configuration

Each GPIO has 2 data registers, and their functions are described as follows:

- GPIO_IN register stores data input via I/O and are read only
- GPIO_OUT register stores the data to be output and can be read/write accessed

Notice

The process of [reading-modifying-writing] for the GPIO_OUT register is not an atomic operation, if all ports used for output in a group are not operated at once, it is not recommended to use this method to modify port output level.

Each GPIO has 2 bit manipulation registers, and their functions are described as follows:

- The BS bit of GPIO_BSRR register can set the corresponding bit of GPIO_OUT to 1 and BR bit can clear the corresponding bit of GPIO_OUT to 0.
- GPIO_BRR register bit can clear the corresponding bit of GPIO_OUT to 0

Notice

The process of modifying the value of GPIO_OUT through the bit operation register is an atomic operation. It is recommended to modify the port output level.

6.3.5.2 FAST IO and AHB Bus Access

The port input/output register (GPIO_IN/GPIO_OUT) supports the reading and writing of the AHB bus and the FAST IO bus (controlled by the register SYSCTRL_CR1.FASTIOENB bit). For other registers, it only supports the reading and writing of the AHB bus. For these two different buses, the system clock (HCLK) has different processing cycles for these two buses. The figure below shows the fastest timing of two bus port flips:

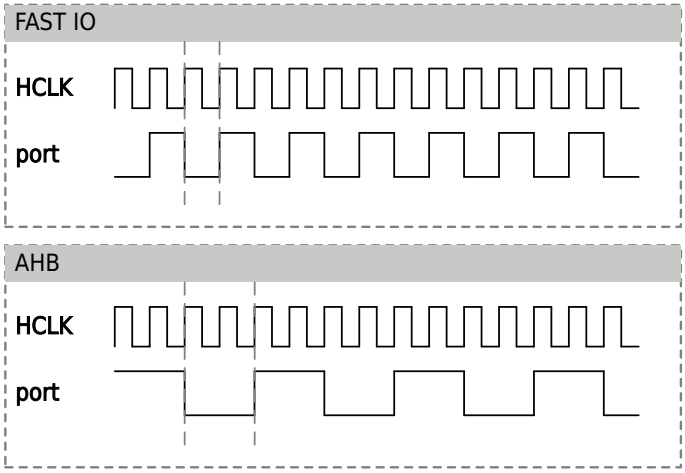


Figure 6-3 AHB/FASTIO Bus Port Changes with System Clock

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

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

6.3.5.3 Input Mode

When GPIO is configured as input mode, its simplified block diagram is as follows:

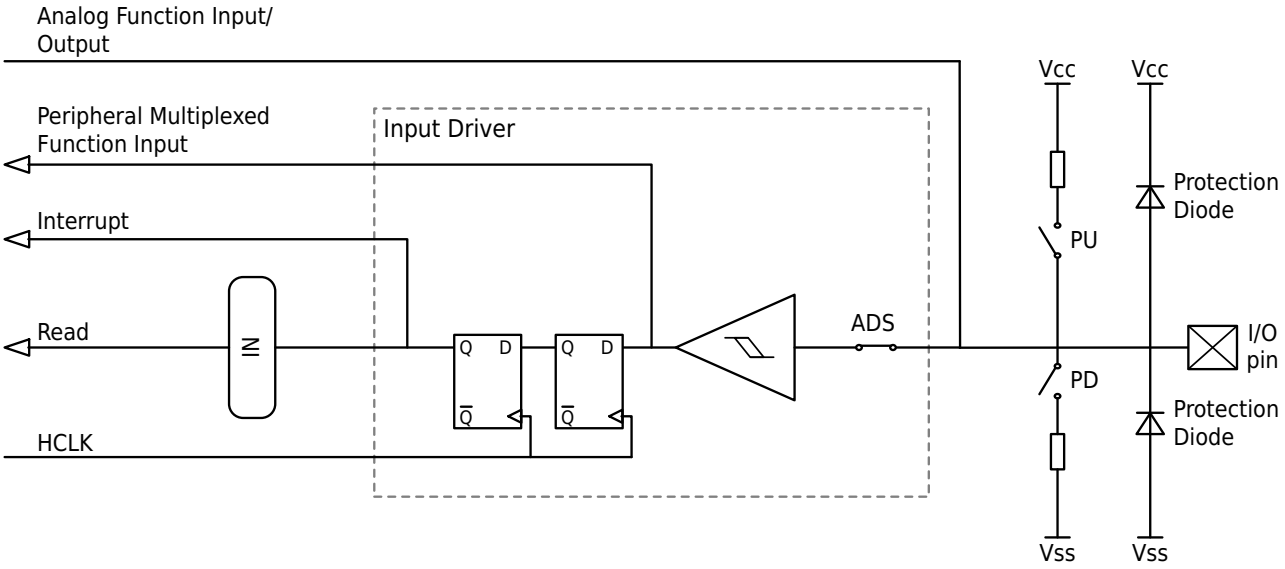


Figure 6-4 Input Floating/Pull-up/Pull-down Configuration Diagram

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

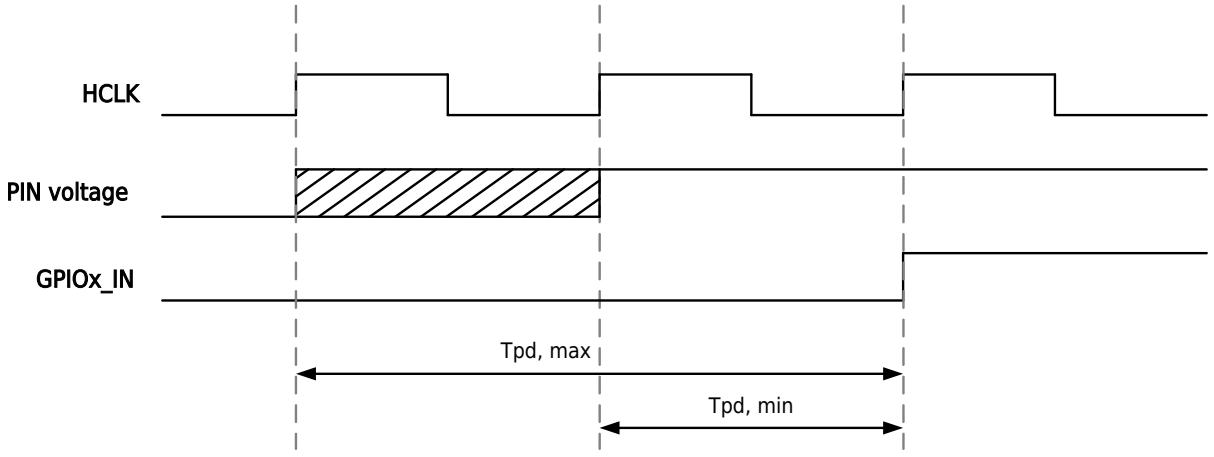


Figure 6-5 Read Port Pin Data Synchronization Diagram

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, at the rising edge of the system clock, the data is latched into the GPIO_IN register. The signal transition delay Tpd is 1-2 system clocks.

6.3.5.4 Output Mode

When GPIO is configured as output mode, its simplified block diagram is as follows:

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

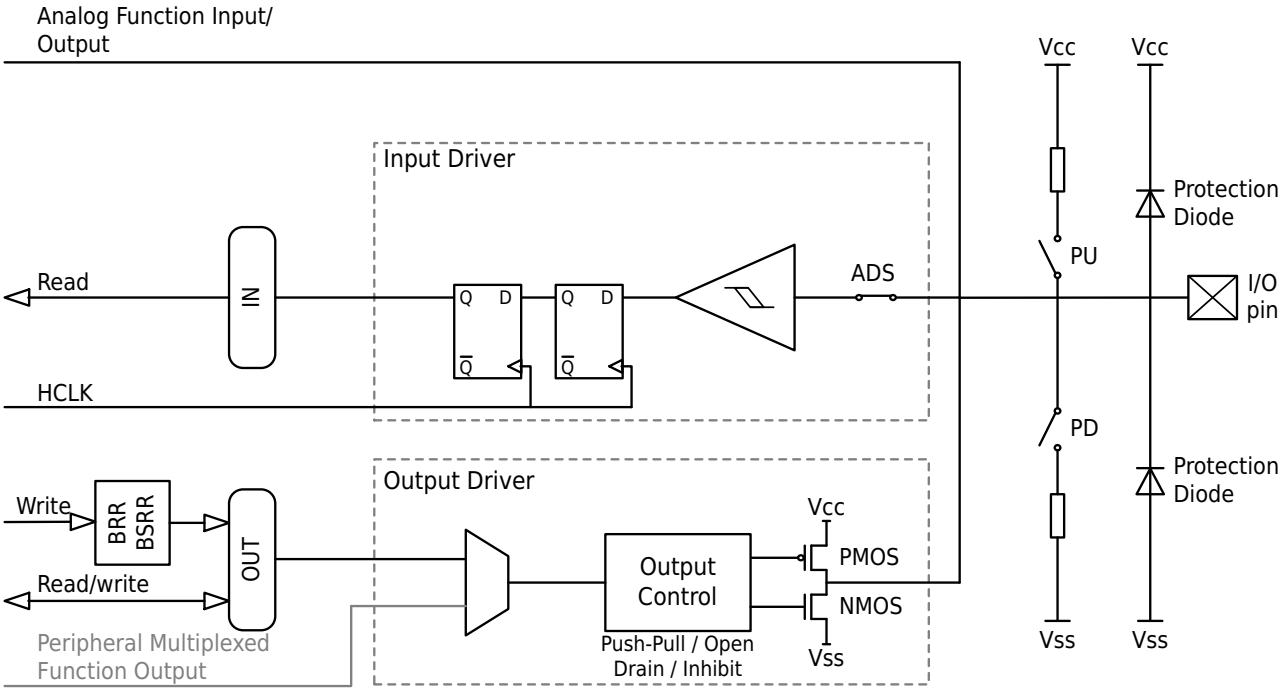


Figure 6-6 Output Configuration Diagram

Each port can control the output level of the IO port by writing data to GPIO_BRR, GPIO_BSRR or GPIO_OUT. GPIO_BRR and GPIO_BSRR are bit operation registers. The number of IO in the same group of ports can be modified from 1-16 each time and the unmodified IO in the same group of ports remain unchanged. The process of using the above two registers to modify IO is an atomic operation. The value written to the register GPIO_OUT is the output value of the output PIN corresponding to each bit.

6.3.6 Port Multiplexing Function

When utilizing the corresponding function of digital peripheral (GPIO, USART, SPI, etc.), the port must be configured for digital function. The operation method is to set the corresponding bit of GPIO_ADS to 0 and configure GPIO_AFRL or GPIO_AFRH to connect a specific on-chip peripheral to the port.

Under this configuration, the port has the following features:

- Input mode: pull-up or pull-down can be enabled; external signals are input to on-chip peripherals; GPIO_IN can read external signals and generate port interrupts
- Output mode: open-drain (pull-up can be enabled) or push-pull can be selected; on-chip peripherals output signals to ports; port interrupts cannot be generated

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

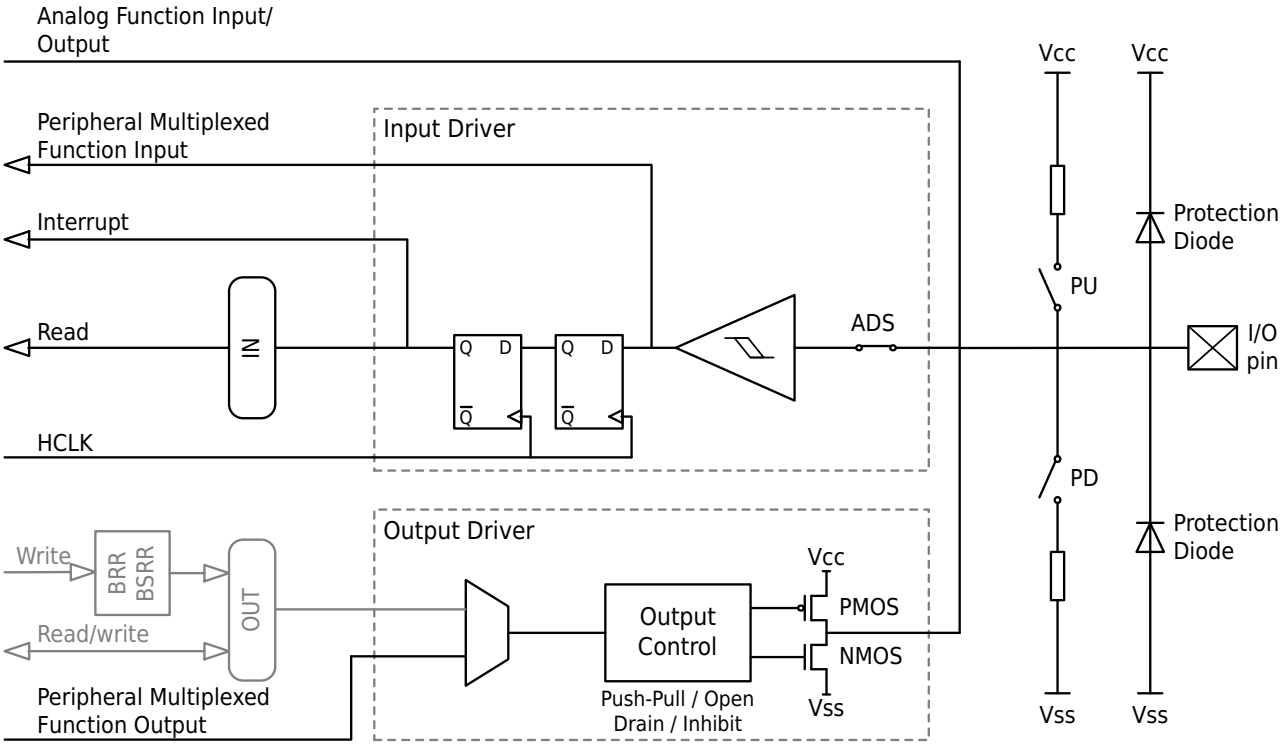


Figure 6-7 Multiplexed Function Configuration Diagram

The digital multiplexed function of each port of this chip is shown in the table below.

Table 6-2 Port Digital Multiplexed Function

Name	AF1	AF2	AF3	AF4	AF5	AF6	AF7
PC13	CTIM1_ETR	RTC_1HZ	ATIM3_CH1B	CTIM0_TOG	-	-	-
PC14	CTIM1_CH0	CTIM0_TOGN	LVD_OUT	-	-	-	-
PC15	CTIM1_CH0	-	-	-	-	-	-
PD00	I2C0_SDA	CTRIM_ETRTOG	USART1_TXD	CTIM1_CH2	I2CSLV_SCL	-	-
PD01	I2C0_SCL	ATIM3_CH0B	USART1_RXD	CTIM1_CH3	I2CSLV_SDA	-	-
PA00	USART1_CTS	LPUART1_TXD	ATIM0_ETR	VC0_OUT	ATIM1_CHA	ATIM3_ETR	ATIM0_CHA
PA01	USART1_RTS	LPUART1_RXD	ATIM0_CHB	ATIM1_ETR	ATIM1_CHB	HCLK_OUT	SPI1_MOSI
PA02	USART1_TXD	ATIM0_CHA	VC1_OUT	ATIM1_CHA	ATIM2_CHA	ATIM3_CH0A	SPI1_MISO
PA03	USART1_RXD	ATIM0_GATE	ATIM1_CHB	ATIM2_CHB	SPI1_NSS	ATIM3_CH0B	PCLK_OUT
PA04	SPI0_NSS	USART1_SCK	CTIM1_CH0	ATIM2_ETR	ATIM3_CH1A	USART1_TXD	I2CSLV_SCL
PA05	SPI0_SCK	ATIM0_ETR	CTIM0_ETR	ATIM0_CHA	ATIM3_CH1B	XTL_OUT	XTH_OUT
PA06	SPI0_MISO	CTIM0_CH0	ATIM3_BK	ATIM1_CHA	VC0_OUT	ATIM3_GATE	LPUART0_CTS
PA07	SPI0_MOSI	CTIM0_CH1	HCLK_OUT	ATIM3_CH2A	ATIM2_CHA	VC1_OUT	CTIM1_TOG
PB00	CTIM0_CH2	CTIM1_TOGN	LPUART0_TXD	ATIM3_CH2B	RCH_OUT	RCL_OUT	PLL_OUT
PB01	CTIM0_CH3	PCLK_OUT	I2CSLV_SDA	ATIM3_CH1B	LPUART0_RTS	LPTIM_TOGN	USART0_SCK
PB02	LPTIM_TOG	CTIM0_ETR	LPUART1_TXD	ATIM3_CH0B	ATIM1_BK	ATIM0_BK	ATIM2_BK
PB10	I2C1_SCL	SPI1_SCK	ATIM1_CHA	LPUART0_TXD	ATIM3_CH2A	LPUART1_RTS	USART1_RTS
PB11	I2C1_SDA	ATIM1_CHB	LPUART0_RXD	ATIM2_GATE	ATIM3_CH1A	LPUART1_CTS	USART1_CTS
PB12	SPI1_NSS	ATIM3_BK	LPUART0_TXD	ATIM0_BK	-	LPUART0_RTS	ATIM3_CH0A
PB13	SPI1_SCK	I2C1_SCL	ATIM3_CH0B	LPUART0_CTS	ATIM1_CHA	ATIM1_GATE	-
PB14	SPI1_MISO	I2C1_SDA	ATIM3_CH1B	ATIM0_CHA	ATIM3_CH2A	LPUART0_RTS	ATIM1_BK
PB15	SPI1_MOSI	ATIM3_CH2B	ATIM0_CHB	ATIM0_GATE	RTC_1HZ	CTIM0_TOG	LPUART1_RXD
PA08	USART0_SCK	ATIM3_CH0A	USART0_TXD	CAN_STBY	ATIM1_GATE	CTIM0_TOGN	ATIM3_BK
PA09	USART0_TXD	ATIM3_CH1A	ATIM0_BK	I2C0_SCL	MCO_OUT	HCLK_OUT	I2CSLV_SCL
PA10	USART0_RXD	ATIM3_CH2A	ATIM2_BK	I2C0_SDA	ATIM2_GATE	PCLK_OUT	I2CSLV_SDA
PA11	USART0_CTS	ATIM3_GATE	I2C1_SCL	CAN_TX	VC0_OUT	SPI0_MISO	ATIM3_CH1B
PA12	USART0_RTS	ATIM3_ETR	I2C1_SDA	CAN_RX	VC1_OUT	SPI0_MOSI	CTIM1_ETR
PA13	USART1_SCK	USART0_RXD	LVD_OUT	ATIM3_ETR	RTC_1HZ	CTIM1_CH1	CTRIM_ETRTOG
PD06	I2C1_SCL	LPUART1_CTS	USART0_CTS	RTC_OUT	ATIM3_BK	CTIM1_CH2	-
PD07	I2C1_SDA	LPUART1_RTS	USART0_RTS	RTC_TS	ATIM3_ETR	CTIM1_CH3	-
PA14	USART1_TXD	USART0_TXD	ATIM3_CH2A	LVD_OUT	RCH_OUT	RCL_OUT	PLL_OUT
PA15	SPI0_NSS	USART1_RXD	LPUART1_RTS	ATIM0_ETR	ATIM0_CHA	ATIM3_CH1A	CAN_STBY

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Name	AF1	AF2	AF3	AF4	AF5	AF6	AF7
PB03	SPI0_SCK	ATIM0_CHB	ATIM1_GATE	ATIM3_CH0A	LPTIM_GATE	XTL_OUT	XTH_OUT
PB04	SPI0_MISO	CTIM0_CH0	ATIM2_BK	USART0_CTS	ATIM2_GATE	ATIM3_CH0B	LPTIM_ETR
PB05	SPI0_MOSI	ATIM3_BK	ATIM1_BK	CTIM0_CH1	LPTIM_GATE	CTRIM_ETRTOG	USART0_RTS
PB06	I2C0_SCL	USART0_TXD	ATIM1_CHB	ATIM0_CHA	LPTIM_ETR	ATIM3_CH0A	LPTIM_TOG
PB07	I2C0_SDA	USART0_RXD	ATIM2_CHB	LPUART1_CTS	ATIM0_CHB	LPTIM_TOGN	ATIM3_ETR
PD03	-	-	-	-	-	-	-
PB08	I2C0_SCL	ATIM1_CHA	CAN_RX	ATIM2_CHA	ATIM0_GATE	ATIM3_CH2A	USART0_TXD
PB09	I2C0_SDA	USART0_SCK	SPI1_NSS	ATIM2_CHA	CAN_TX	ATIM2_CHB	USART0_RXD

Table 6-3 IOAUX1 Multiplexed Function

SYSCTRL->IOAUX1	0	1	2	3	4	5	6	7
ATIM3ETR[14:12]	ATIM3_ETR	LPUART1_RXD	VC0_OUT	LVD_OUT	VC1_OUT	PA07	PA15	PB07
CTIM1ETR[10:8]	CTIM1_ETR	LPUART0_RXD	VC0_OUT	LVD_OUT	VC1_OUT	PA06	PA14	PB06
CTIM0ETR[6:4]	CTIM0_ETR	USART1_RXD	VC0_OUT	LVD_OUT	VC1_OUT	PA05	PA13	PB05
ATIM0ETR[2:0]	ATIM0_ETR	USART0_RXD	VC0_OUT	LVD_OUT	VC1_OUT	PA04	PA12	PB04

Table 6-4 IOAUX2 Multiplexed Function

SYSCTRL->IOAUX2	0	1	2	3	4	5	6	7
ATIM3CH2B[14:12]	ATIM3_CH2B	LPUART1_RXD	VC0_OUT	LVD_OUT	VC1_OUT	PA03	PA11	PB03
CTIM1CAP3[10:8]	CTIM1_CH3	LPUART0_RXD	VC0_OUT	LVD_OUT	VC1_OUT	PA02	PA10	PB02
CTIM0CAP3[6:4]	CTIM0_CH3	USART1_RXD	VC0_OUT	LVD_OUTt	VC1_OUT	PA01	PA09	PB01
ATIM0CHA[2:0]	ATIM0_CHA	USART0_RXD	VC0_OUT	LVD_OUT	VC1_OUT	PA00	PA08	PB00

6.3.7 Port Clock Output

6.3.7.1 Output PCLK0

Configure the GPIO register to output the 2 frequency division output of PCLK. The following example outputs PCLK from PA10:

Step1. Set the SYSCTRL_IOAUX0.PCLKDIV of the system control module to 2, and select the frequency division by 2;

Step2. Set GPIOA_ADS.PIN[10] to 0, and configure PA10 as digital port;

Step3. Set GPIOA_DIR.PIN[10] to 0, and configure PA10 as output mode;

Step4. Set GPIOA_AFRH.AFSEL10 to 6, and configure the multiplexing function of PA10 as PCLK_OUT.

6.3.7.2 Output MCO

Configure the GPIO register to output RCH/RCL/XTH/XTL/PLL/RC150K, the example below outputs RCH/8 from PA09:

Step1. Set SYSCTRL_IOAUX0.MCOSEL of the system control module to 0, select RCH;

Step2. Set SYSCTRL_IOAUX0.MCODIV of the system control module to 2, select 8 frequency division;

Step3. Set GPIOA_ADS.PIN[9] to 0, and configure PA09 as digital port;

Step4. Set GPIOA_DIR.PIN[9] to 0, and configure PA09 as output mode;

Step5. Set GPIOA_AFRH.AFSEL9 to 5, and configure the multiplexing function of PA09 as MCO_OUT.

6.3.8 Port External Interrupt

When the port is configured as a digital function (GPIO_ADS is 0) and works in input mode (GPIO_DIR is 1), each port has an interrupt function. Each port supports four types of interrupts: rising-edge interrupt, falling-edge interrupt, high-level interrupt, and low-level interrupt. The corresponding interrupt enable registers are: GPIO_RIE, GPIO_FIE, GPIO_HIE, GPIO_LIE. When an interrupt is triggered, it can be

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

determined which port triggered the interrupt by querying the interrupt status register (GPIO_IFR), and the corresponding interrupt status flag bit can be cleared through the interrupt clear register (GPIO_ICR).

The simplified block diagram related to GPIO external interrupt is as follows:

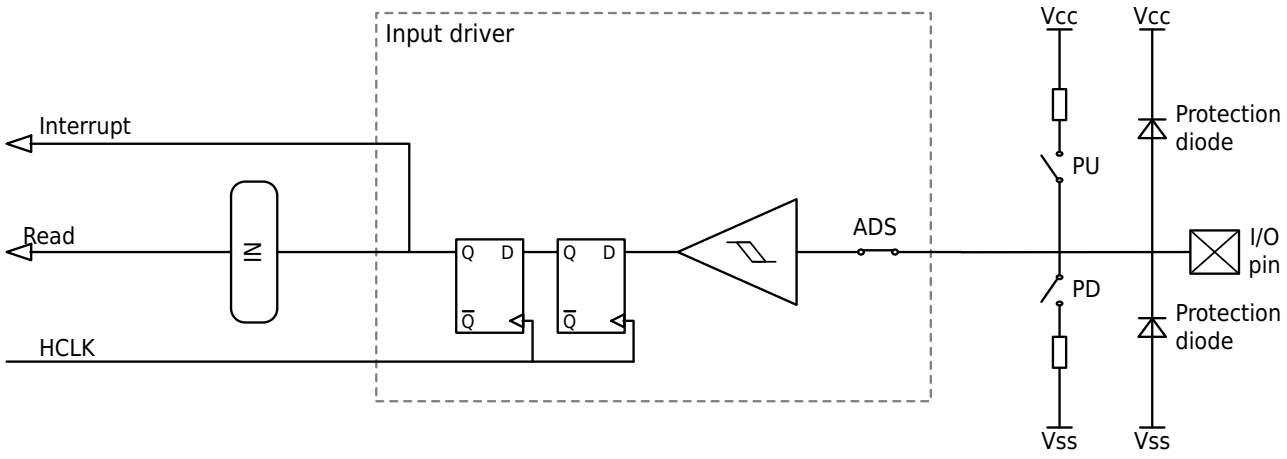


Figure 6-8 Port External Interrupt Diagram

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

6.4 Register Description

6.4.1 Register List

- GPIOA base address: 0x40020C00
- GPIOB base address: 0x40021C00
- GPIOC base address: 0x40022000
- GPIOD base address: 0x40022400

Note

GPIOA and GPIOB have 16 valid bits, GPIOC has 3 valid bits (PC[15:13]), and GPIOD has 5 valid bits (PD[7:6], PD3, PD[1:0]). Write operations to non-existent bits have no effect, and reads always return 0.

Table 6-5 General-Purpose I/O (GPIO) Register Offset

Offset	Register	Description	Width
0x00	GPIO_ADS	Port Digital-to-Analog Configuration Register	32
0x04	GPIO_DIR	Port Input/Output Direction Configuration Register	32
0x08	GPIO_OD	Port Output Type Register	32
0x10	GPIO_DR	Port Output Driving Capability Register	32
0x18	GPIO_PUPD	Port Pull-up Pull-down Register	32
0x1C	GPIO_IN	Port Data Input Register	32
0x20	GPIO_OUT	Port Data Output Register	32
0x24	GPIO_BRR	Port Reset Register	32
0x28	GPIO_BSRR	Port Set Reset Register	32
0x30	GPIO_AFRL	Port Multiplexing Function Low Register	32
0x34	GPIO_AFRH	Port Multiplexing Function High Register	32
0x38	GPIO_HIE	Port High Level Interrupt Enable Configuration Register	32
0x3C	GPIO_LIE	Port Low Level Interrupt Enable Configuration Register	32
0x40	GPIO_RIE	Port Rising Edge Interrupt Enable Configuration Register	32
0x44	GPIO_FIE	Port Falling Edge Interrupt Enable Configuration Register	32
0x48	GPIO_IFR	Port Interrupt Status Register	32
0x4C	GPIO_ICR	Port Interrupt Clear Register	32

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

6.4.2 Port Digital-to-Analog Configuration Register (GPIO_ADS)

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	0x00009BFF(PA); 0x0000FFFF(PB); 0x0000E000(PC); 0x000000CB(PD)																													
Name	Reserved																PIN													
Access																	RW													

Bit	Name	Functional Description	Access
31:16	*Reserved*	-	-
15:0	PIN	Port digital function or analog function control ● 0b0: Configure the port as a digital function ● 0b1: Configure the port as an analog function **Note** Each bit corresponds to one port, please refrain from configuring non-existent ports, for example: PIN[7] corresponds to port 7.	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

6.4.3 Port Input/Output Direction Configuration Register (GPIO_DIR)

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	0x0000FFFF(PA); 0x0000FFFF(PB); 0x0000E000(PC); 0x000000CB(PD)																													
Name	Reserved																PIN													
Access																	RW													

Bit	Name	Functional Description	Access
31:16	*Reserved*	-	-
15:0	PIN	Port input and output direction control ● 0b0: Configure the port as an output ● 0b1: Configure the port as an input **Note** Each bit corresponds to one port, please refrain from configuring non-existent ports.	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

6.4.4 Port Output Type Register (GPIO_OD)

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																PIN															
Access																	RW															

Bit	Name	Functional Description	Access
31:16	*Reserved*	-	-
15:0	PIN	Port output mode control bit ● 0b0: Push-pull output ● 0b1: Open-drain output **Note** Each bit corresponds to one port, please refrain from configuring non-existent ports.	RW

6.4.5 Port Output Driving Capability Register (GPIO_DR)

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																PIN															
Access																	RW															

Bit	Name	Functional Description	Access
31:16	*Reserved*	-	-
15:0	PIN	Port output driving capability control bit ● 0b0: High driving capability ● 0b1: Low driving capability **Note** Each bit corresponds to one port, please refrain from configuring non-existent ports.	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

6.4.6 Port Pull-Up Pull-Down Register (GPIO_PUPD)

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	0x04006000(PA); 0x00000000(PB); 0x00000000(PC); 0x00000000(PD)																													
Name	PD																PU													
Access	RW																RW													

Bit	Name	Functional Description	Access
31:16	PD	Port pull-down resistor enable control ● 0b0: Disable pull-down resistor ● 0b1: Enable pull-down resistor **Note** Each bit corresponds to one port, please refrain from configuring non-existent ports.	RW
15:0	PU	Port pull-up resistor enable control ● 0b0: Disable pull-up resistor ● 0b1: Enable pull-up resistor **Note** Each bit corresponds to one port, please refrain from configuring non-existent ports.	RW

Note

When the same bit of PD and PU is set to 1 simultaneously, then this port status is to enable the pull-up resistor.

6.4.7 Port Data Input Register (GPIO_IN)

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	0x0000XXXX																													
Name	Reserved																PIN													
Access																	RO													

Bit	Name	Functional Description	Access
31:16	*Reserved*	-	-

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Bit	Name	Functional Description	Access
15:0	PIN	Port input data **Note** Each bit corresponds to one port, please refrain from configuring non-existent ports.	RO

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

6.4.8 Port Data Output Register (GPIO_OUT)

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	0x0000XXXX																															
Name	Reserved																PIN															
Access																	RW															

Bit	Name	Functional Description	Access
31:16	*Reserved*	-	-
15:0	PIN	Port output data **Note** Each bit corresponds to one port, please refrain from configuring non-existent ports.	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

6.4.9 Port Reset Register (GPIO_BRR)

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																PIN															
Access																	WO															

Bit	Name	Functional Description	Access
31:16	*Reserved*	-	-
15:0	PIN	Port Bit Clear Control ● 0b0: Does not affect the corresponding bit of the GPIO_OUT register ● 0b1: Set the corresponding bit of the GPIO_OUT register to 0 **Note** Each bit corresponds to one port, please refrain from configuring non-existent ports.	WO

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

6.4.10 Port Set Reset Register (GPIO_BSRR)

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	BR																BS															
Access	WO																WO															

Bit	Name	Functional Description	Access
31:16	BR	Port Bit Clear Control ● 0b0: Does not affect the corresponding bit of the GPIO_OUT register ● 0b1: Set the corresponding bit of the GPIO_OUT register to 0 **Note** Each bit corresponds to one port, please refrain from configuring non-existent ports.	WO
15:0	BS	Port Bit Bit Control ● 0b0: Does not affect the corresponding bit of the GPIO_OUT register ● 0b1: Set the corresponding bit of the GPIO_OUT register to 1 **Note** Each bit corresponds to one port, please refrain from configuring non-existent ports.	WO

Note

When the same bit of BR and BS is set to 1 simultaneously, BS has high priority.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

6.4.11 Port Multiplexing Function Low Register (GPIO_AFRL)

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	AFSEL7				AFSEL6				AFSEL5				AFSEL4				AFSEL3				AFSEL2				AFSEL1				AFSEL0			
Access	RW				RW				RW				RW				RW				RW				RW				RW			

Bit	Name	Functional Description	Access
31:0	AFSELy (y = 7-0)	Port multiplexing function control ● 0b0000: GPIO ● 0b0001: AF1 ● 0b0010: AF2 ● 0b0011: AF3 ● 0b0100: AF4 ● 0b0101: AF5 ● 0b0110: AF6 ● 0b0111: AF7 ● 0b1000-0b1111: Setting prohibited **Note** The specific functions of AF1-AF7 are detailed in "Port Multiplexing Functionality".	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

6.4.12 Port Multiplexing Function High Register (GPIO_AFRH)

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																															
Name	AFSEL15				AFSEL14				AFSEL13				AFSEL12				AFSEL11				AFSEL10				AFSEL9				AFSEL8			
Access	RW				RW				RW				RW				RW				RW				RW				RW			

Bit	Name	Functional Description	Access
31:0	AFSELy (y = 15-8)	Port multiplexing function control ● 0b0000: GPIO ● 0b0001: AF1 ● 0b0010: AF2 ● 0b0011: AF3 ● 0b0100: AF4 ● 0b0101: AF5 ● 0b0110: AF6 ● 0b0111: AF7 ● 0b1000-0b1111: Setting prohibited **Note** The specific functions of AF1-AF7 are detailed in the "Port Multiplexing Function".	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

6.4.13 Port High Level Interrupt Enable Configuration Register (GPIO_HIE)

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																PIN															
Access																	RW															

Bit	Name	Functional Description	Access
31:16	*Reserved*	-	-
15:0	PIN	Port high level interrupt enable control ● 0b0: Disable the high level interrupt of the corresponding port ● 0b1: Enable the high level interrupt of the corresponding port **Note** Each bit corresponds to one port, please refrain from configuring non-existent ports.	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

6.4.14 Port Low Level Interrupt Enable Configuration Register (GPIO_LIE)

Offset	Bit Position																													
0x3C	31	30	29	28	27	26	25	24	23	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																PIN															
Access																	RW															

Bit	Name	Functional Description	Access
31:16	*Reserved*	-	-
15:0	PIN	Port low level interrupt enable control ● 0b0: Disable the low level interrupt of the corresponding port ● 0b1: Enable the low level interrupt of the corresponding port **Note** Each bit corresponds to one port, please refrain from configuring non-existent ports.	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

6.4.15 Port Rising Edge Interrupt Enable Configuration Register (GPIO_RIE)

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																PIN															
Access																	RW															

Bit	Name	Functional Description	Access
31:16	*Reserved*	-	-
15:0	PIN	Port rising edge interrupt enable control ● 0b0: Disable the rising edge interrupt of the corresponding port ● 0b1: Enable the rising edge interrupt of the corresponding port **Note** Each bit corresponds to one port, please refrain from configuring non-existent ports.	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

6.4.16 Port Falling Edge Interrupt Enable Configuration Register (GPIO_FIE)

Offset	Bit Position																													
0x44	31	30	29	28	27	26	25	24	23	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																PIN															
Access																	RW															

Bit	Name	Functional Description	Access
31:16	*Reserved*	-	-
15:0	PIN	Port falling edge interrupt enable control ● 0b0: Disable the falling edge interrupt of the corresponding port ● 0b1: Enable the falling edge interrupt of the corresponding port **Note** Each bit corresponds to one port, please refrain from configuring non-existent ports.	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

6.4.17 Port Interrupt Status Register (GPIO_IFR)

Offset	Bit Position																													
0x48	31	30	29	28	27	26	25	24	23	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																PIN															
Access																	RO															

Bit	Name	Functional Description	Access
31:16	*Reserved*	-	-
15:0	PIN	Port interrupt status flag ● 0b0: No qualifying interrupt occurred ● 0b1: A qualifying interrupt has occurred **Note** Each bit corresponds to one port, please refrain from configuring non-existent ports .	RO

6.4.18 Port Interrupt Clear Register (GPIO_ICR)

Offset	Bit Position																													
0x4C	31	30	29	28	27	26	25	24	23	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(PA); 0x0000FFFF(PB); 0x0000E000(PC); 0x000000CB(PD)																													
Name	Reserved																PIN															
Access																	WO															

Bit	Name	Functional Description	Access
31:16	*Reserved*	-	-
15:0	PIN	Port interrupt flag cleared Write 0 to clear flag, write 1 has no effect; read always returns 1. **Note** Each bit corresponds to one port, please refrain from configuring non-existent ports.	WO

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

7 I2C

7.1 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.

7.2 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)/High-speed (1Mbps)
- Support 7-bit addressing function
- Support noise filtering function
- Support 3 slave addresses
- Support broadcast address
- Support interrupt status query function

7.3 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.

7.3.1 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).

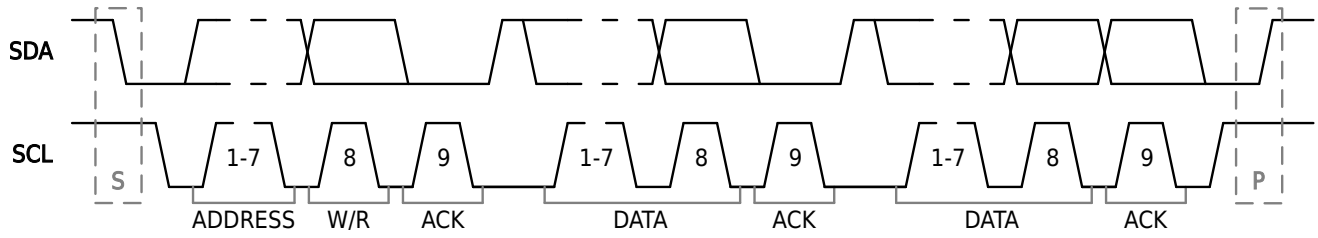


Figure 7-1 I2C Transmission Protocol

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.

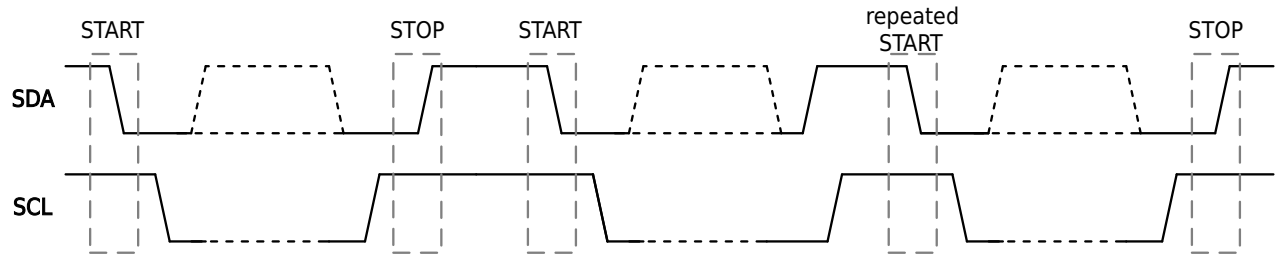


Figure 7-2 START, Repeated START and STOP

- 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.

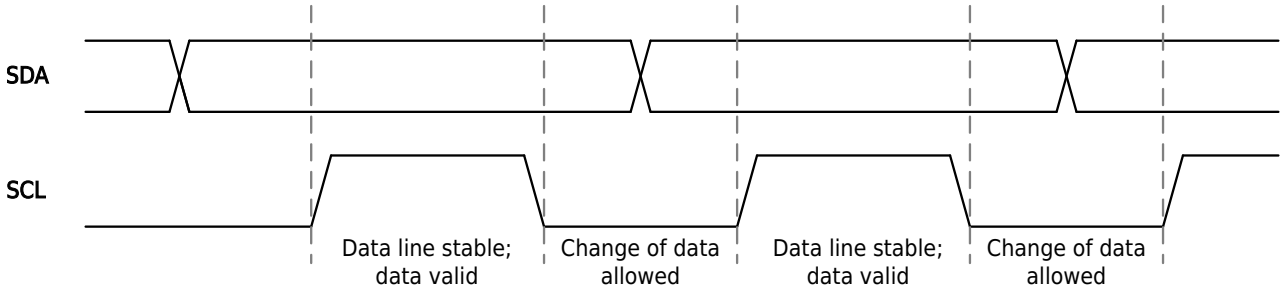


Figure 7-3 Bit Transfer on I2C Bus

7.3.2 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.

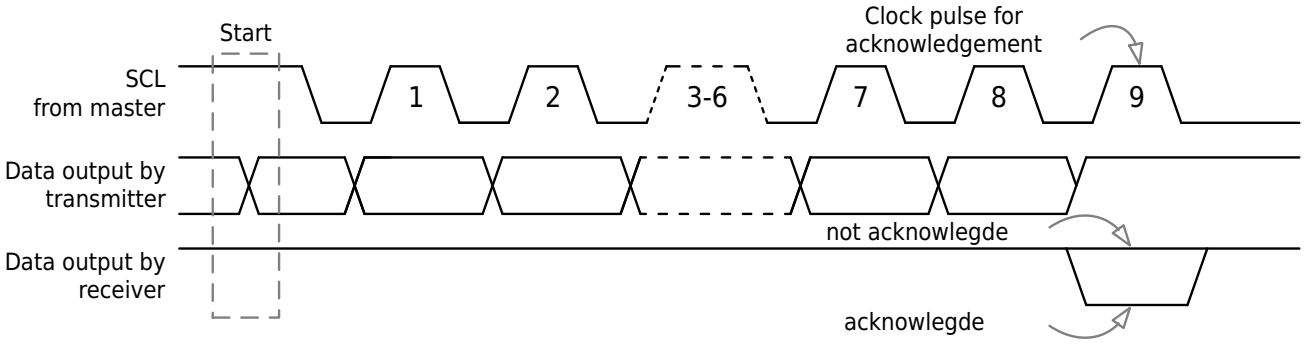


Figure 7-4 Acknowledgment Signal on I2C Bus

7.3.3 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.

  If the slave wants the master to reduce the transmission speed, it can notify the master by actively pulling SCL low to extend its low level time. When the master finds that the SCL level is pulled low when preparing for the next transmission, it waits until the slave completes the operation and release control of the SCL line.

- 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.

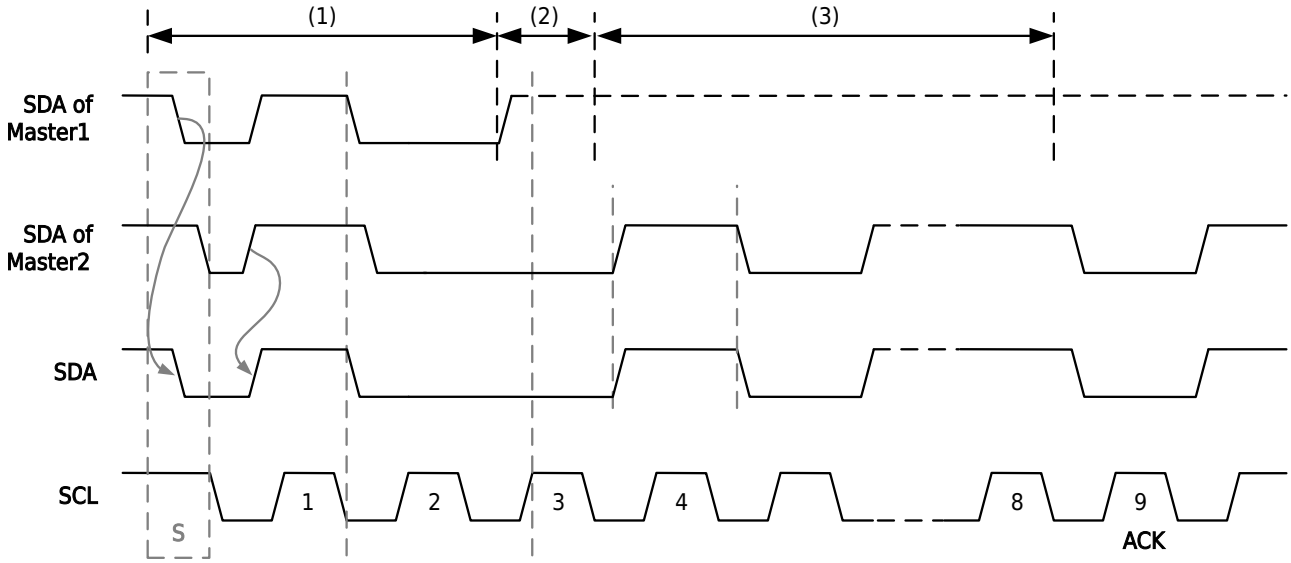


Figure 7-5 Arbitration on the I2C Bus

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.

7.4 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.

7.4.1 Function Block Diagram

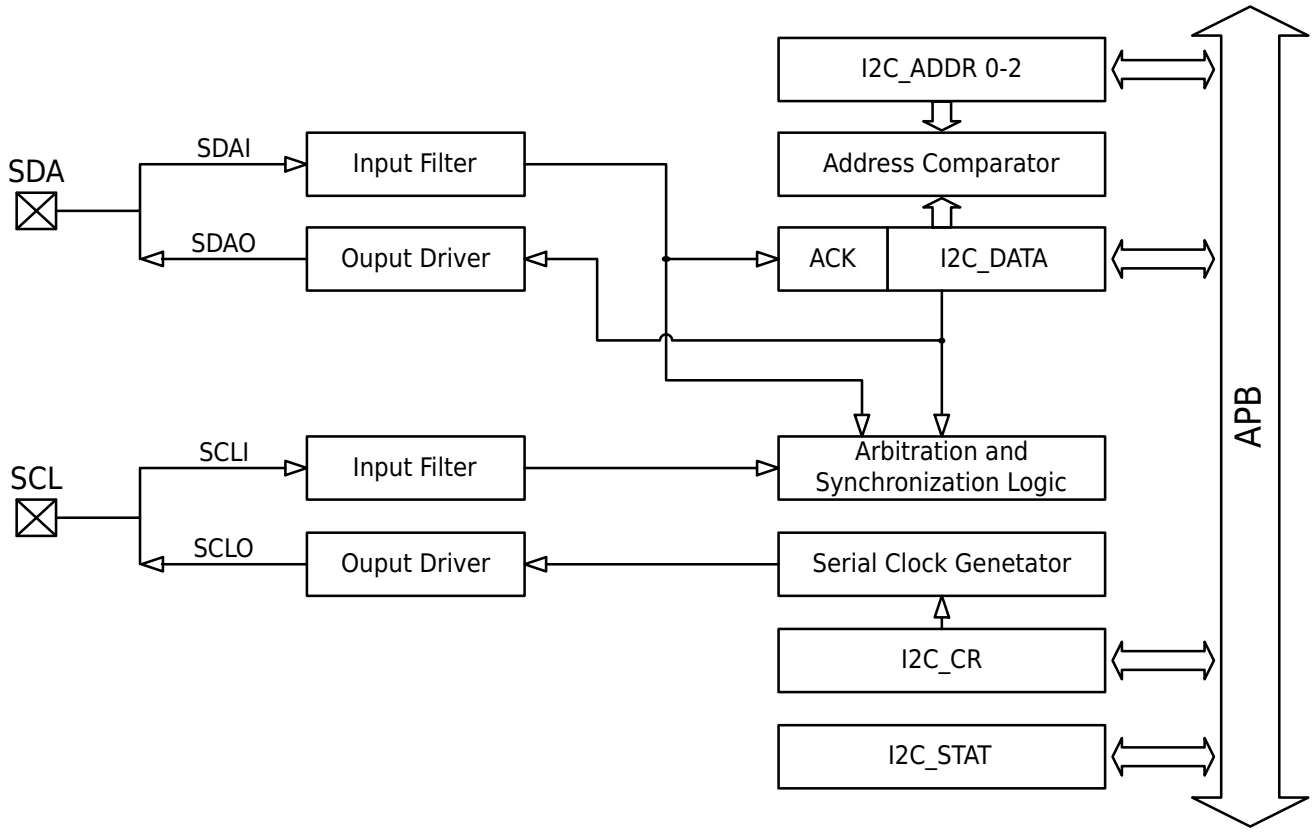


Figure 7-6 I2C Function Block Diagram

7.4.2 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 7-1 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.

7.4.3 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.

7.4.4 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.

7.4.5 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.

7.4.6 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.

7.4.7 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 high-speed 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.

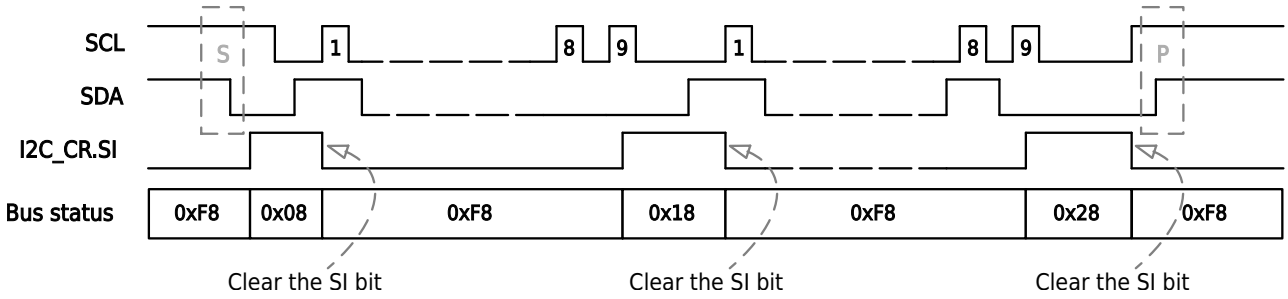


Figure 7-7 Data Synchronization Diagram of Master Sending Mode

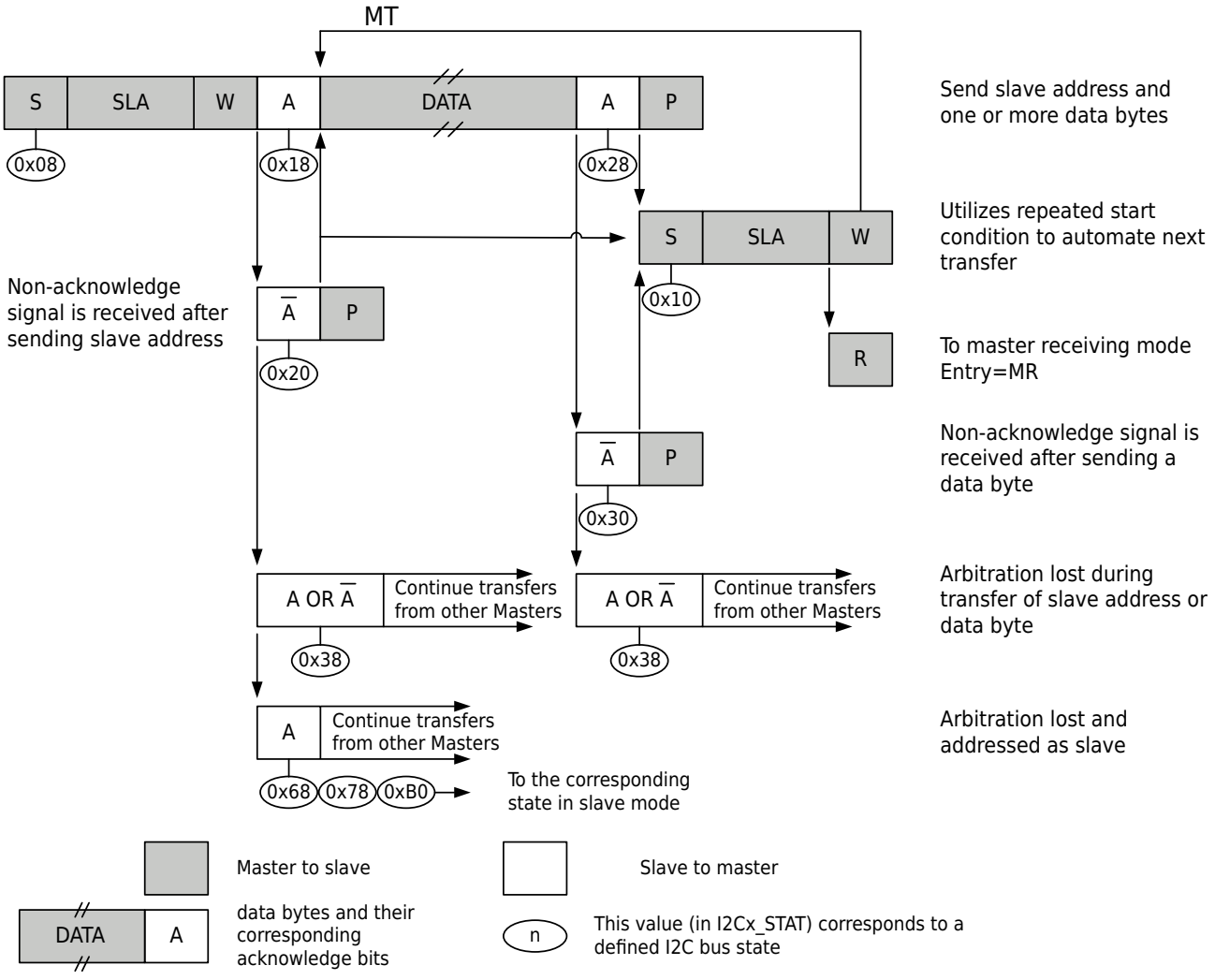


Figure 7-8 I2C Master Sending State Diagram

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.

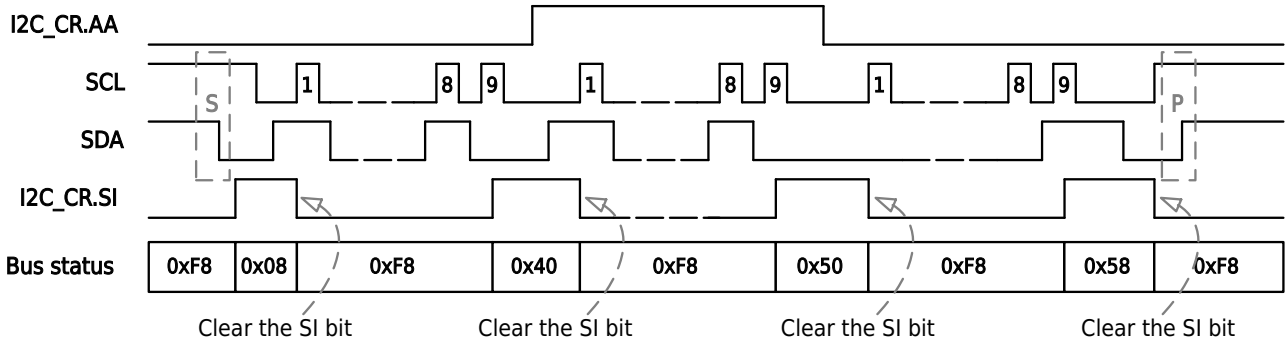


Figure 7-9 Data Synchronization Diagram of Master Receiving Mode

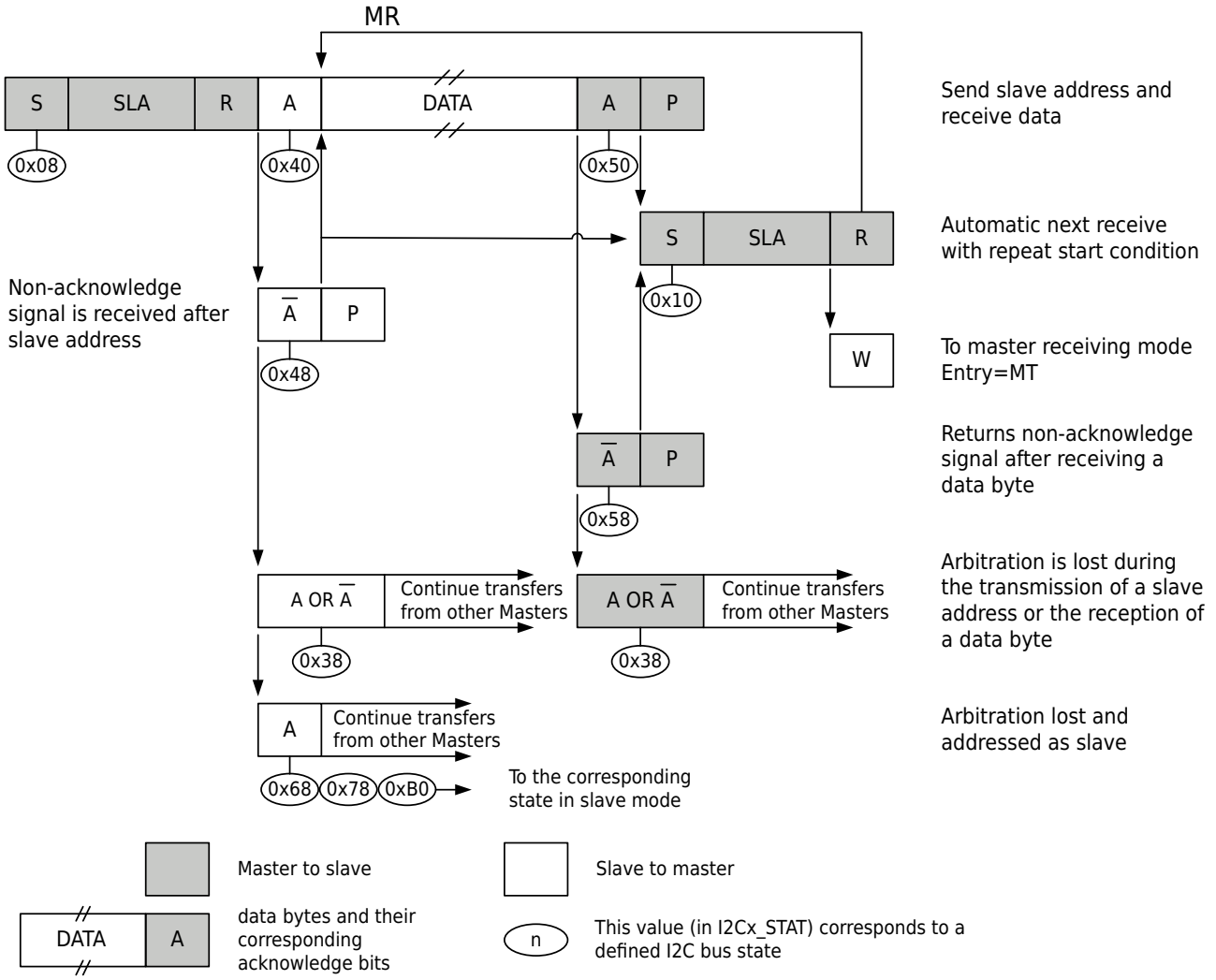


Figure 7-10 I2C Master Receiving State Diagram

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.

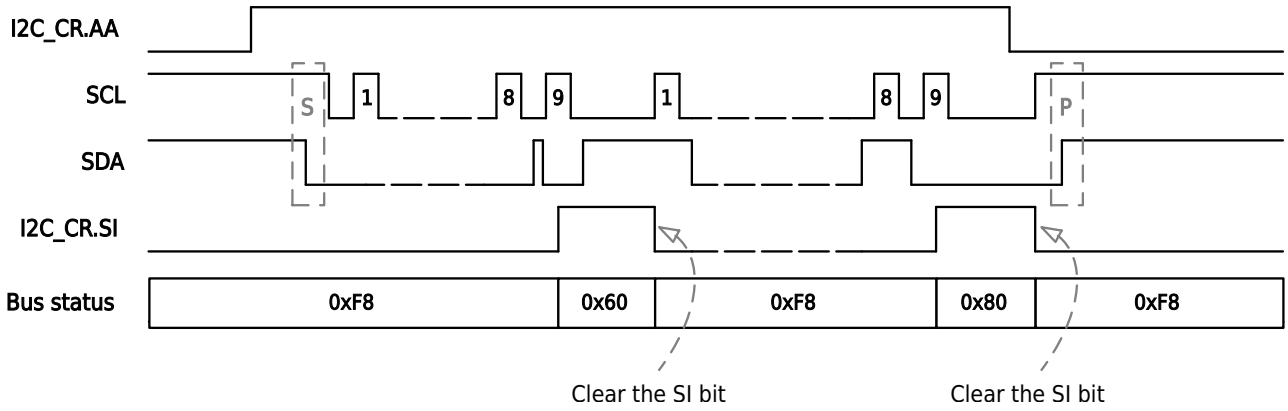


Figure 7-11 Slave Receiving Mode Data Synchronization Diagram

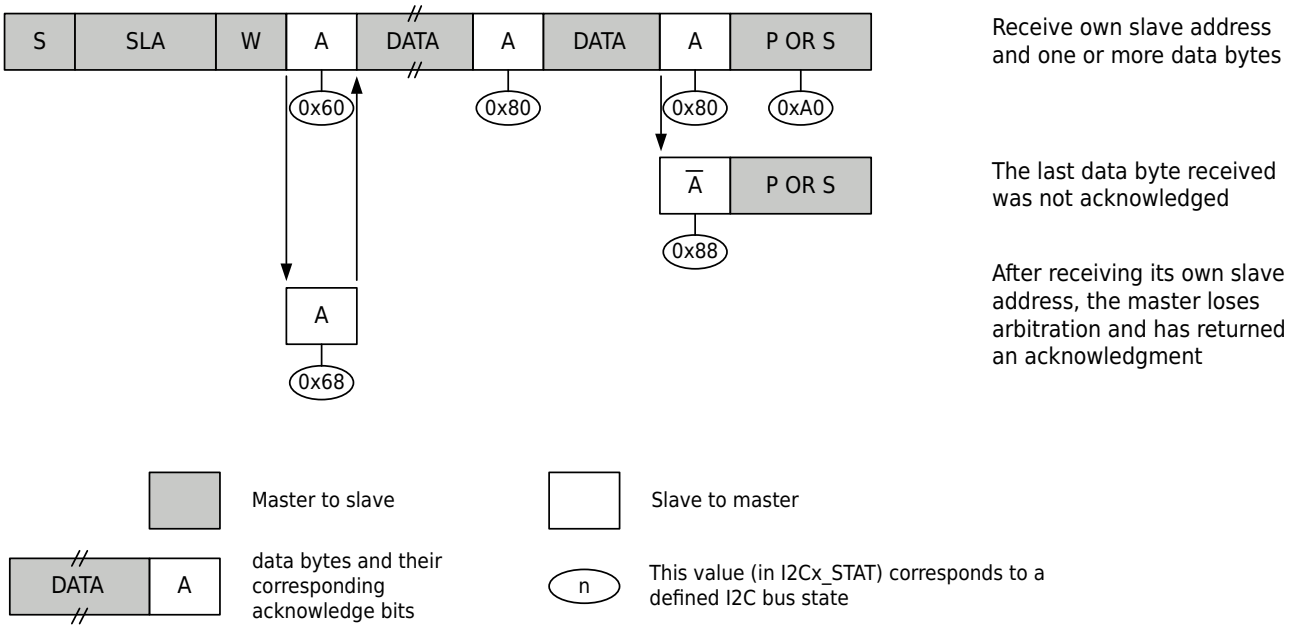


Figure 7-12 Slave Receiving State Diagram

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 then send the last byte of data and, in the subsequent transmission, send all 1 data, and will change itself to an unaddressed slave.

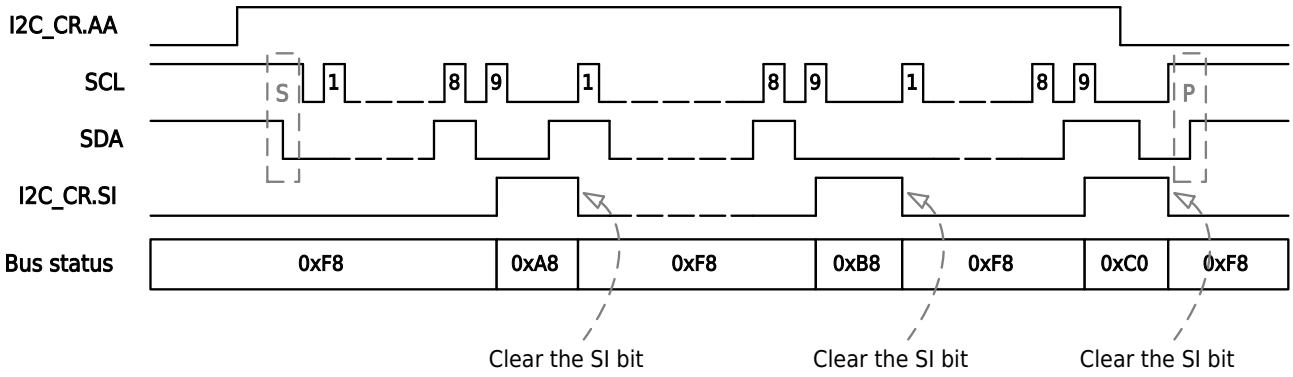


Figure 7-13 Slave Sending Mode Data Synchronization Diagram

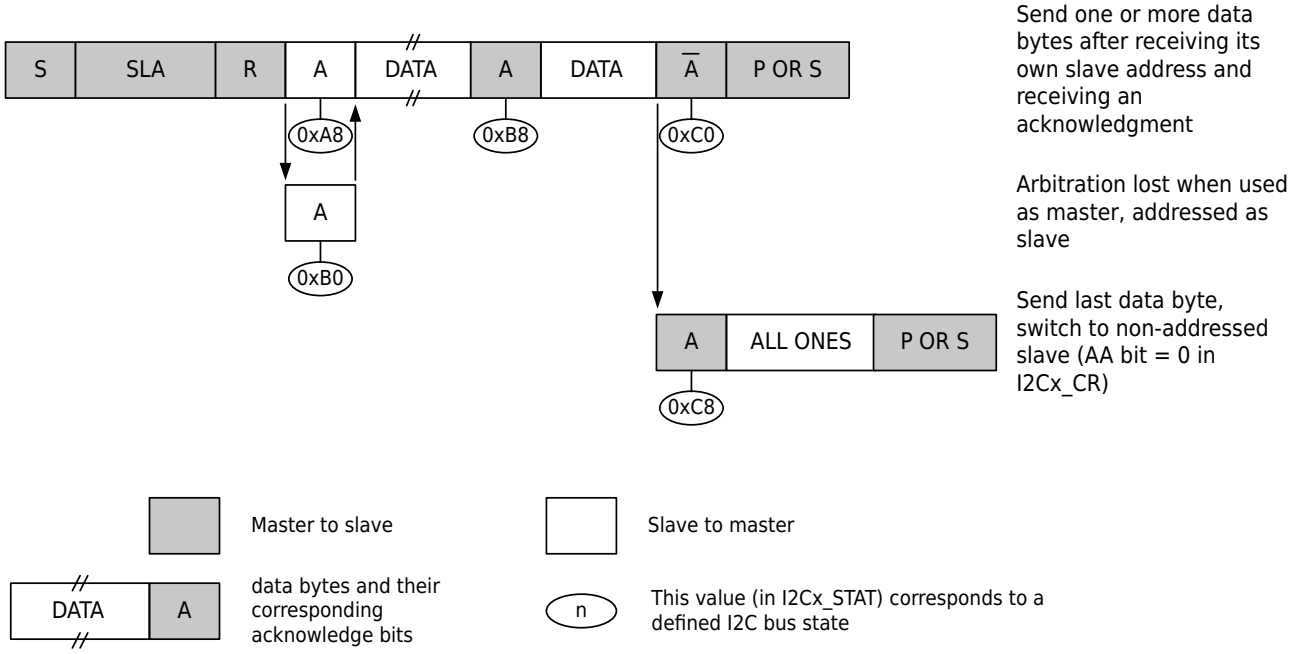


Figure 7-14 I2C Slave Sending State Diagram

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.

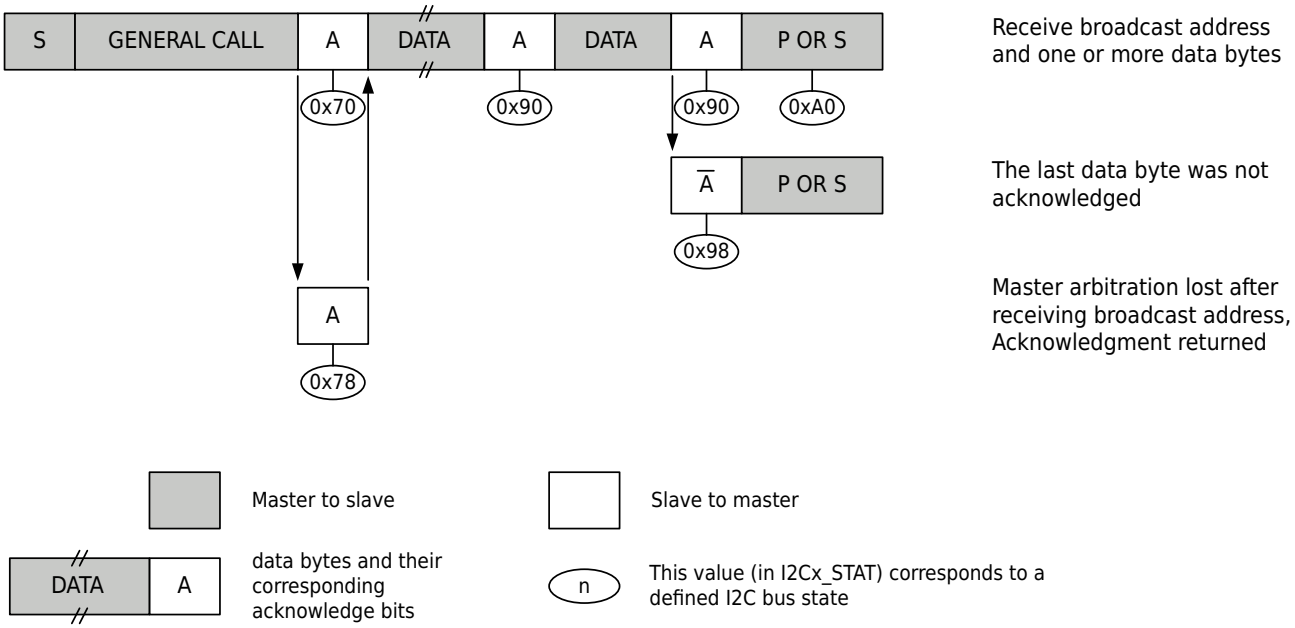


Figure 7-15 I2C Broadcast Call State Diagram

7.4.8 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 7-2 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	After receiving the stop condition or the repeated start condition, the device enters the non-addressable slave state and can still be addressed.
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	After receiving the stop condition or the repeated start condition, the device enters the non-addressable slave state and can still be addressed.
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

7.5 Operation Example

7.5.1 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.

> **Notice**
> SCL and SDA need to be externally connected with pull-up resistors according to the requirements of the I2C protocol.

Step2. Set SYSCTRL_PERI_CLKEN0.I2Cx to 1 to enable the I2C configuration clock and working clock.

Step3. Write 0 and 1 to SYSCTRL_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.

7.5.2 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.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

> **Notice**
> SCL and SDA need to be externally connected with pull-up resistors according to the requirements of the I2C protocol.

Step2. Set SYSCTRL_PERI_CLKEN0.I2Cx to 1 to enable the I2C configuration clock and working clock.
Step3. Write 0 and 1 to SYSCTRL_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+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.

7.5.3 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_PERI_CLKEN0.I2Cx to 1 to enable the I2C configuration clock and working clock.
Step3. Write 0 and 1 to SYSCTRL_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.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

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.

7.5.4 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.

Step2. Set SYSCTRL_PERI_CLKEN0.I2Cx to 1 to enable the I2C configuration clock and working clock.

Step3. Write 0 and 1 to SYSCTRL_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.

7.6 Register Description

7.6.1 Register List

- I2C0 base address: 0x40000400
- I2C1 base address: 0x40004400

Table 7-3 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.

7.6.2 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 ● 0b0: Disable ● 0b1: Enable	RW

7.6.3 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 $f_{SCL}= f_{PCLK}/8/(TM+1)$, and TM>0	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

7.6.4 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 ● 0b0: Disable ● 0b1: Enable	RW
5	STA	Bus state control ● 0b0: No function ● 0b1: Send START to the bus **Note** User program should set this bit to 0 after START transmission is completed.	RW
4	STO	Bus state control ● 0b0: No function ● 0b1: Send STOP to the bus **Note** The hardware automatically sets this bit to 0 after the STOP transmission is completed.	RW
3	SI	I2C interrupt flag 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. Read: ● 0b0: No I2C interrupt occurred ● 0b1: I2C interrupt has occurred	RW
2	AA	Acknowledge control ● 0b0: Send NAK during the acknowledge phase ● 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 ● 0b0: advanced filtering, higher anti-interference performance ● 0b1: simple filtering, faster communication rate **Note** For related functions, please refer to the input filter.	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

7.6.5 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 ● In send mode, write the data to be sent ● In receive mode, read out received data	RW

7.6.6 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 For the specific definitions of status codes, please refer to the Status Code Expression.	RO

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

7.6.7 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 ● 0b0: Disable ● 0b1: Enable	RW

7.6.8 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.

7.6.9 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.

7.6.10 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 ● 0b0: The device address received from the bus is not the same as ADDR2 ● 0b1: The device address received from the bus is the same as ADDR2	RO
1	AD1F	I2C slave mode address 1 match flag bit ● 0b0: The device address received from the bus is not the same as ADDR1 ● 0b1: The device address received from the bus is the same as ADDR1	RO
0	AD0F	I2C slave mode address 0 match flag bit ● 0b0: The device address received from the bus is not the same as ADDR0 ● 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.

8 SPI

8.1 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.

8.2 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

8.3 Functional Description

8.3.1 Function Block Diagram

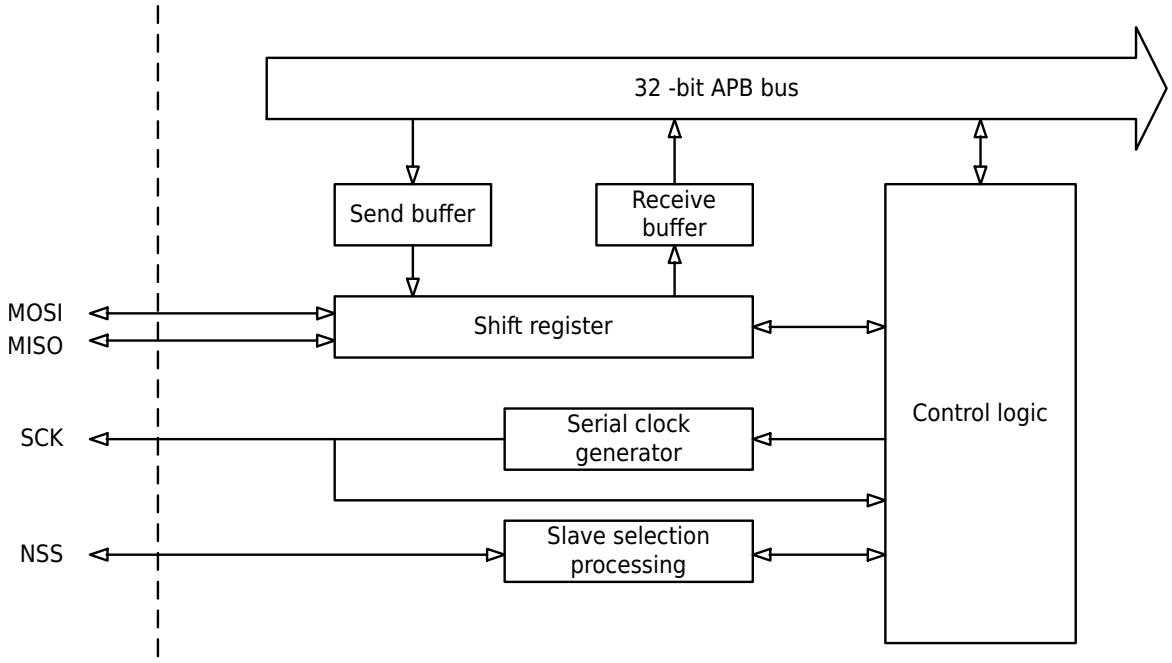


Figure 8-1 SPI Block Diagram

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

8.3.2 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.

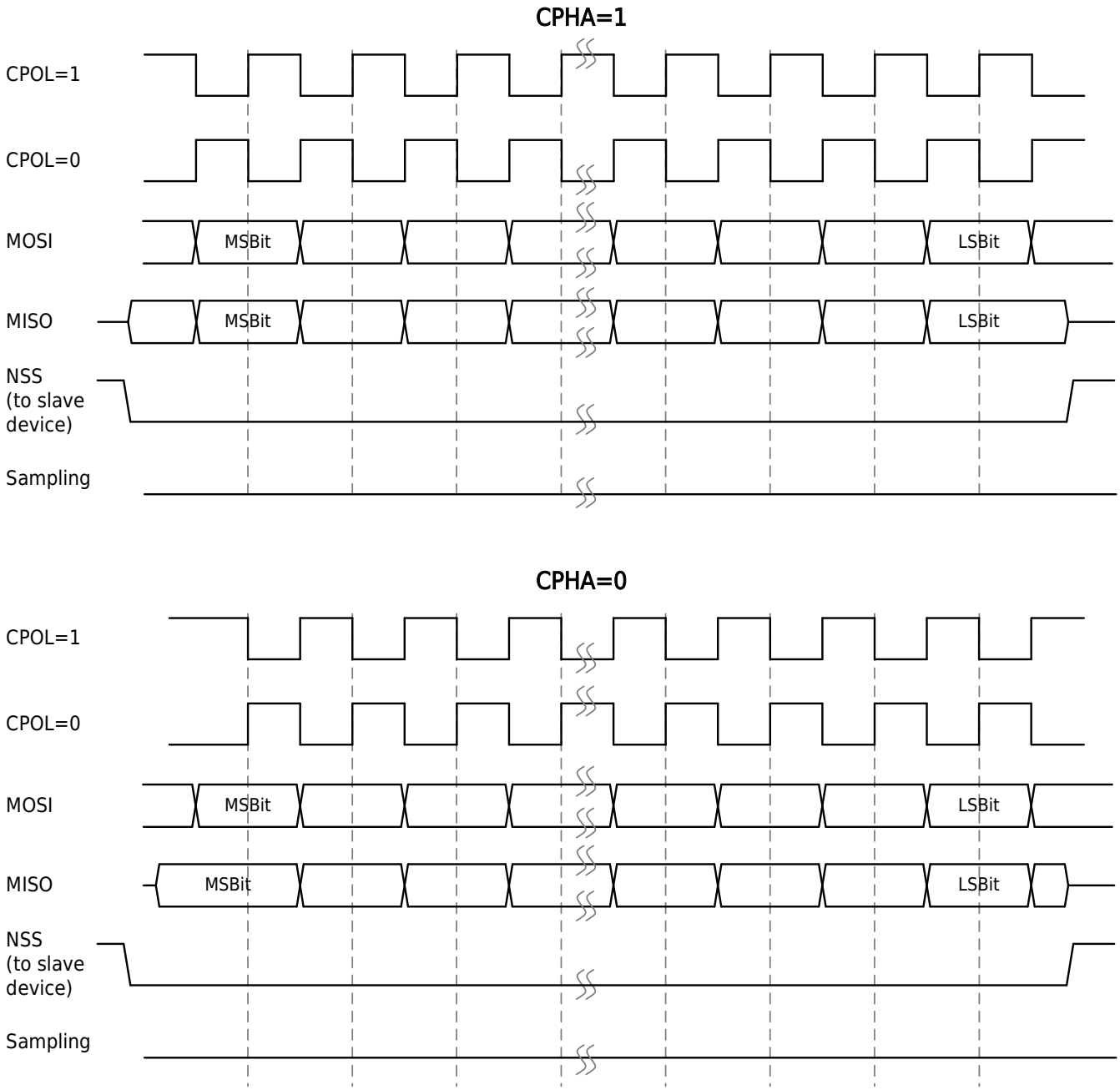


Figure 8-2 SPI Communication Format Timing Diagram

8.3.3 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.

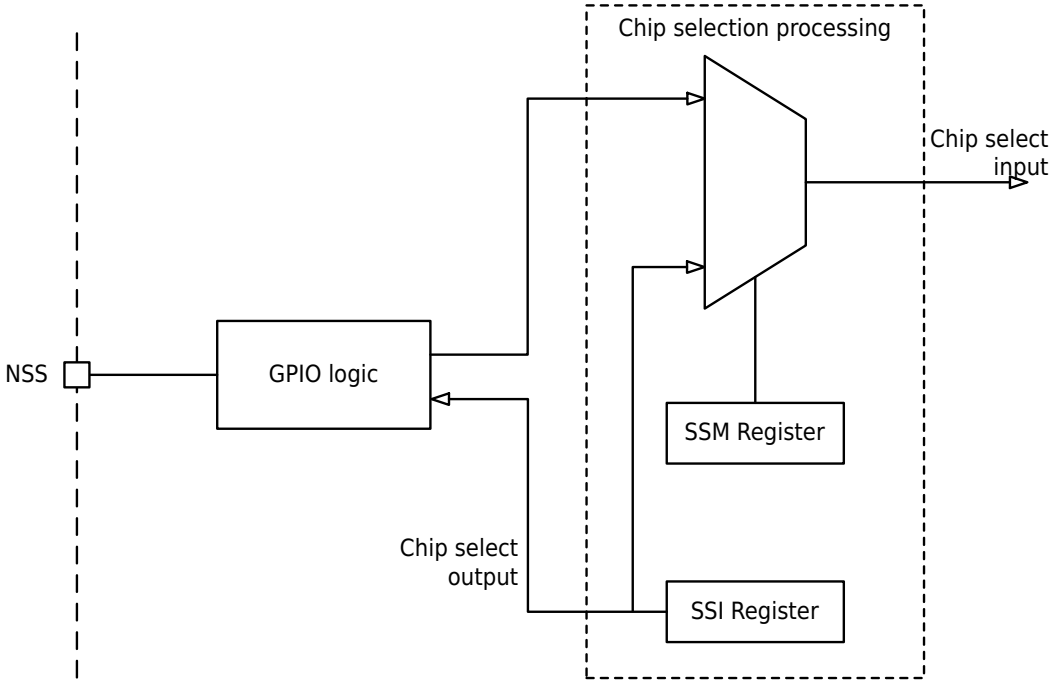


Figure 8-3 Slave Select NSS Pin Configuration

- 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.

8.3.4 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.

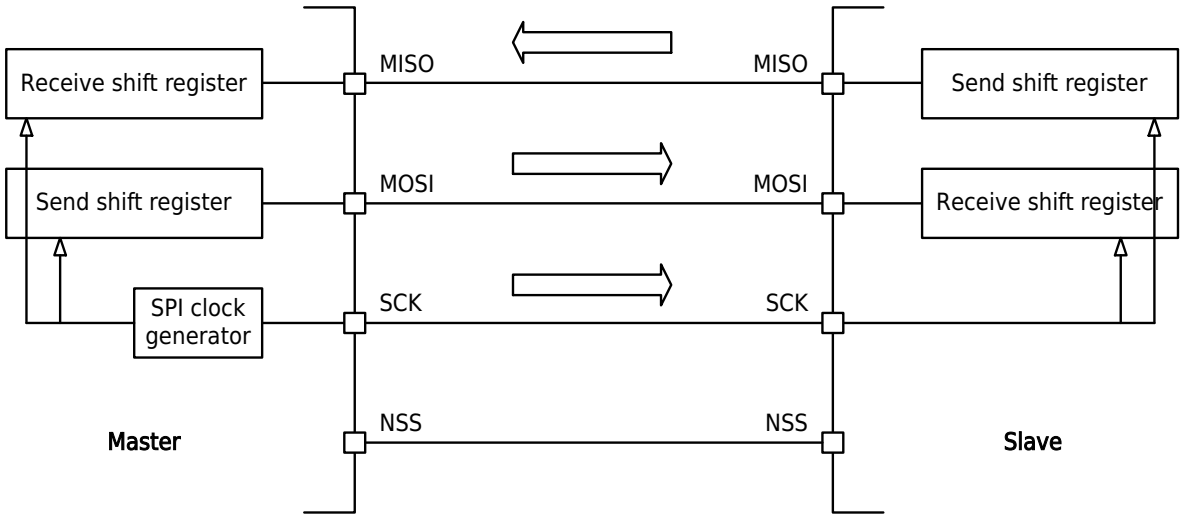


Figure 8-4 SPI Full-duplex Communication

8.3.5 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.

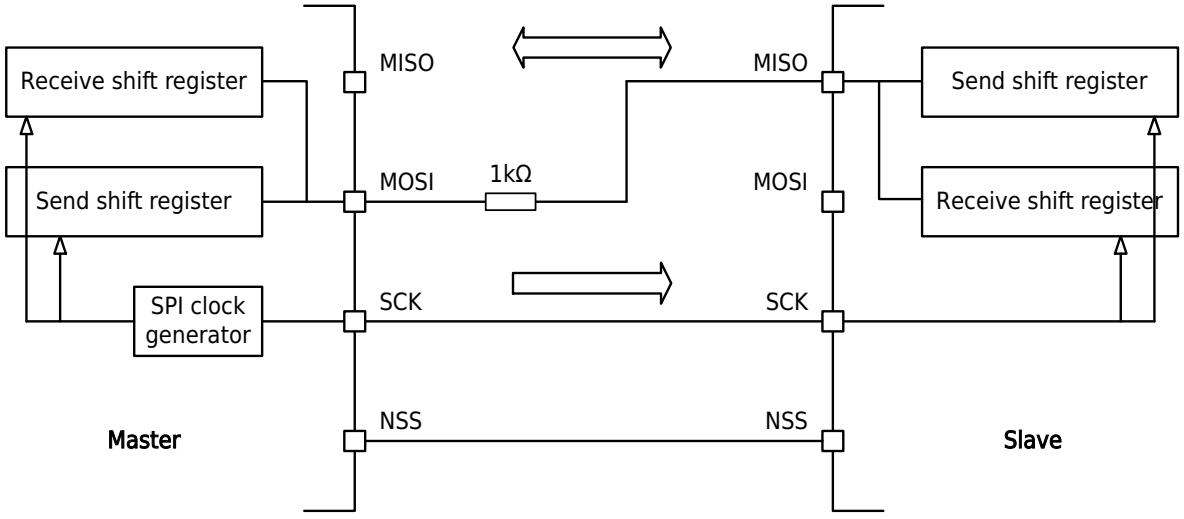


Figure 8-5 SPI Single-wire Half-duplex Communication

8.3.6 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.

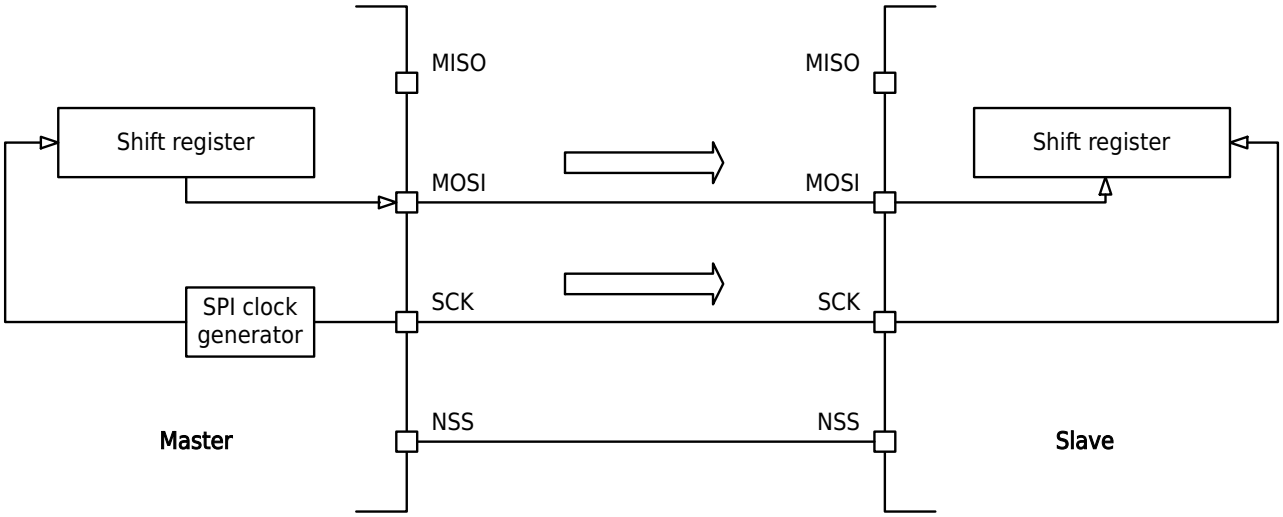


Figure 8-6 SPI Simplex Transmit Communication Diagram

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.

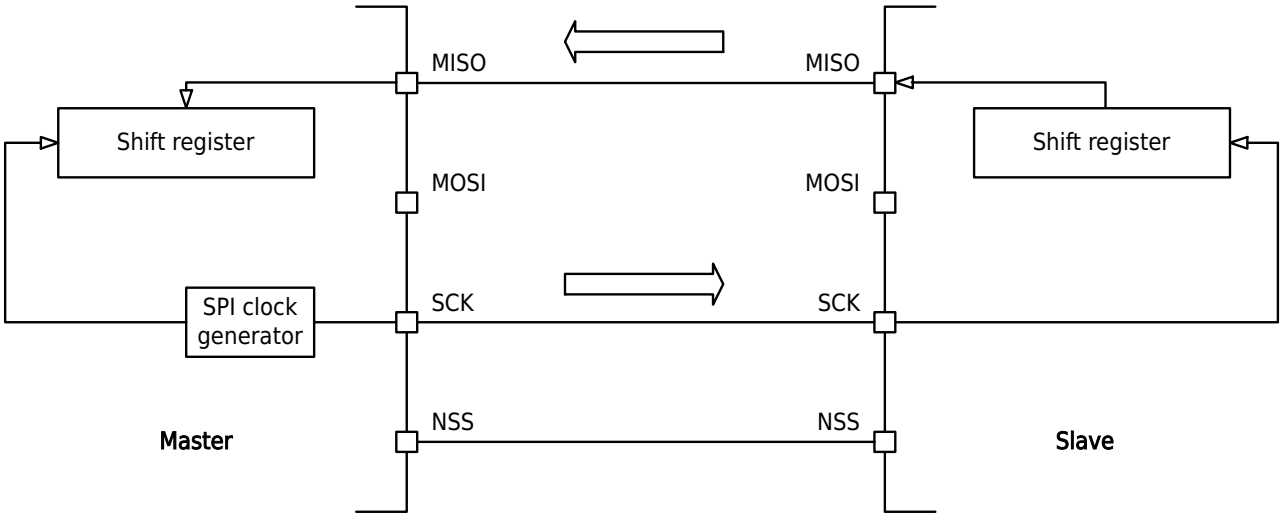


Figure 8-7 SPI Simplex Receive Communication Diagram

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.

8.3.7 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.

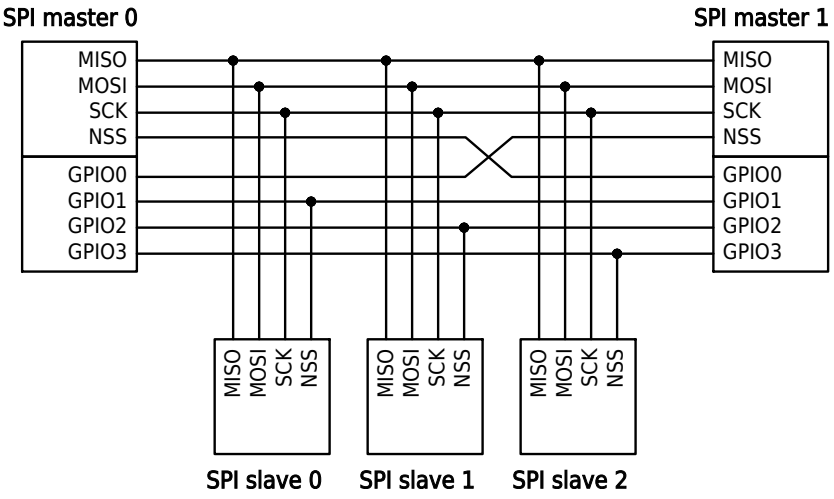


Figure 8-8 SPI Multi-machine System

8.3.8 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.

8.3.9 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

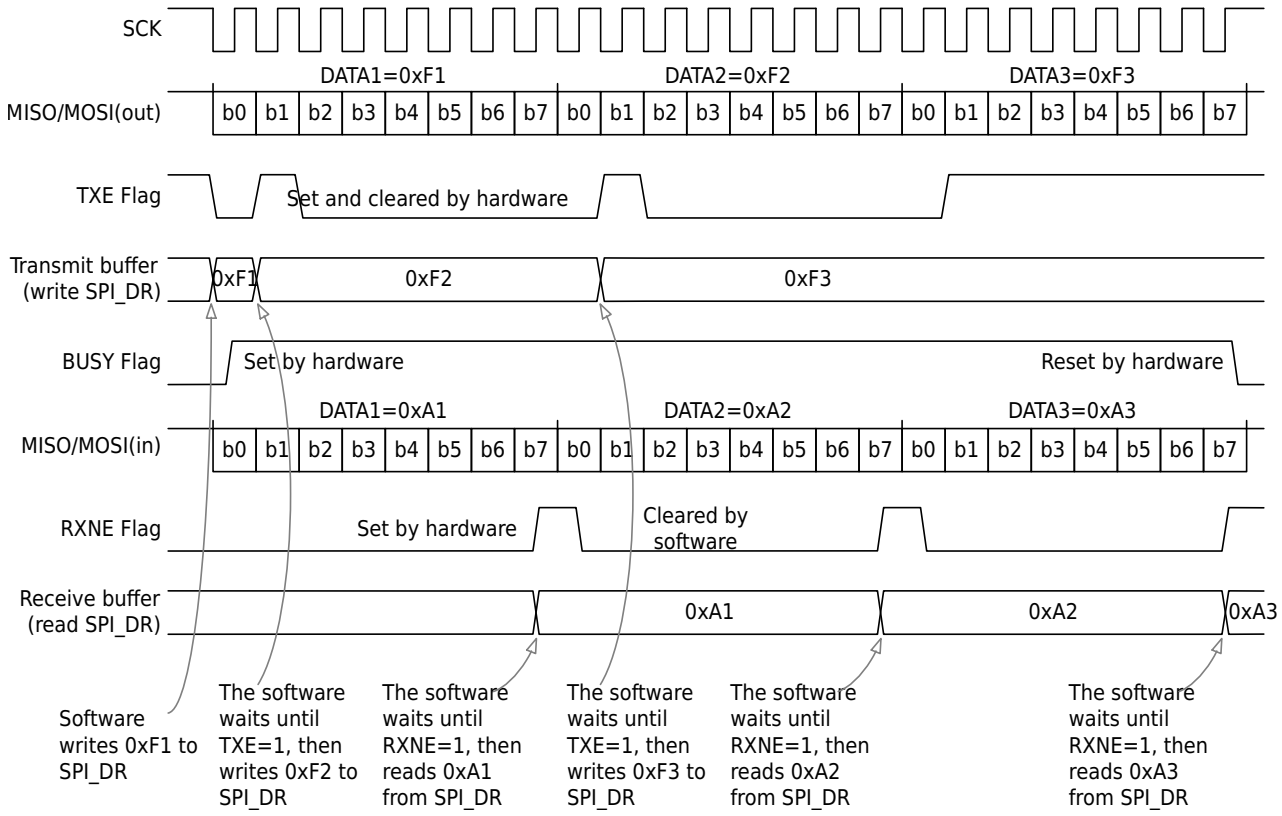


Figure 8-9 Explanation Diagram of General Flags in SPI Master Mode

Slave mode example with CPOL=1 and CPHA=1

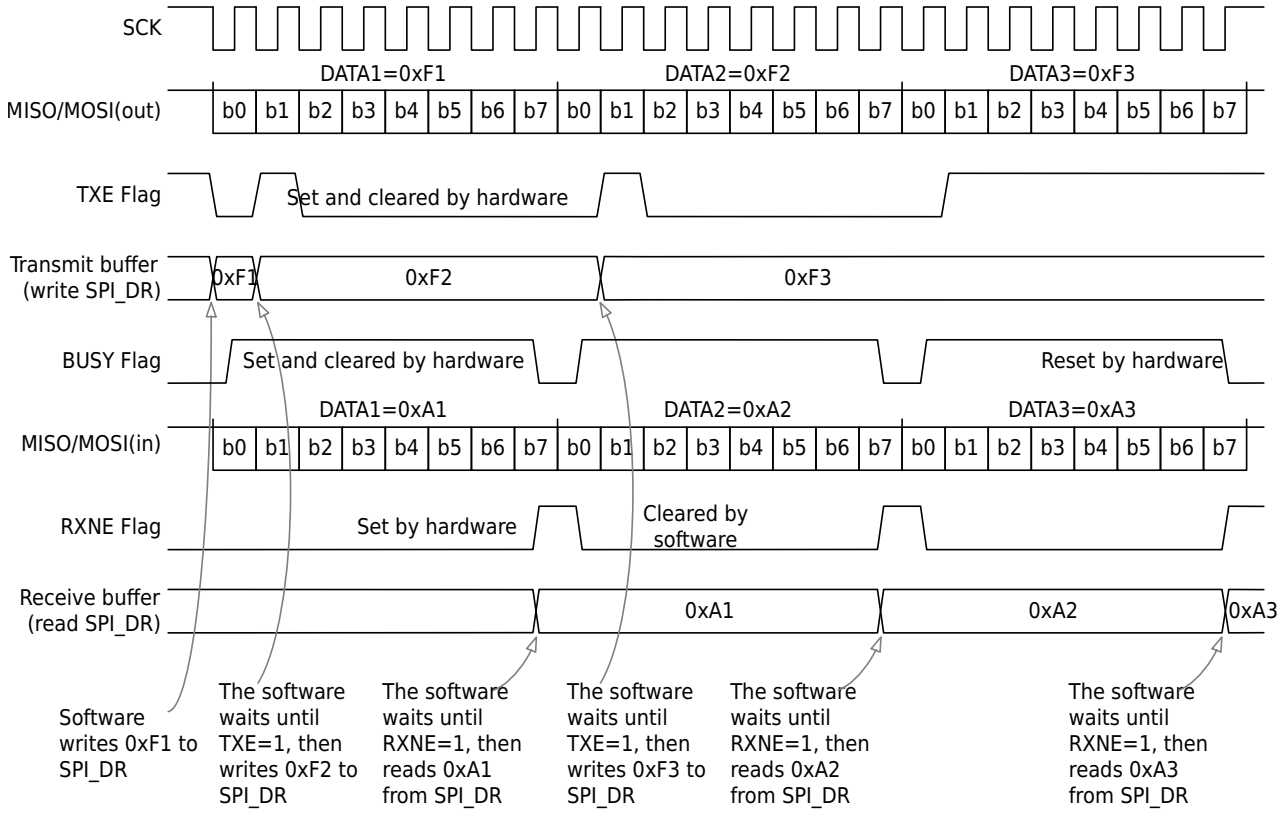


Figure 8-10 SPI Slave Mode General Flag Bit Description Diagram

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

8.3.10 Interrupt Generation and Clearing

SPI supports generating interrupt requests from different events, the specific information is as follows:

Table 8-1 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.

8.4 Operation Example

8.4.1 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_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.

8.4.2 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_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.

8.4.3 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_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.

8.5 Register Description

8.5.1 Register List

- SPI0 base address: 0x40000800
- SPI1 base address: 0x40004800

Table 8-2 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.

8.5.2 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 ● 0b0: In traditional SPI mode, the data received by the master is sampled according to the settings of CPOL and CPHA ● 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 ● 0b00: full-duplex bidirectional ● 0b01: simplex transmit only ● 0b10: simplex receive only ● 0b11: single-wire half-duplex	RW
12	SSM	Slave selection configuration Master mode: ● 0b0: NSS pin determines the slave select input value ● 0b1: SSI register determines the slave select output value Slave mode: ● 0b0: NSS pin determines the slave select input value ● 0b1: SSI register determines the slave select input value	RW
11:8	WIDTH	Frame data width configuration ● 0b0011: 4-bit ● 0b0100: 5-bit ● ... ● 0b1110: 15-bit ● 0b1111: 16-bit ● Others: Setting prohibited	RW
7	LSBF	Data frame high and low order selection ● 0b0: Most Significant Bit (MSB) transmitted and received first ● 0b1: Least Significant Bit (LSB) transmitted and received first	RW
6	MSTR	Operating mode configuration ● 0b0: Slave mode ● 0b1: Master mode	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Bit	Name	Functional Description	Access
5	CPOL	Serial clock polarity configuration ● 0b0: Low level during standby ● 0b1: High level during standby	RW
4	CPHA	Serial clock phase configuration ● 0b0: Front edge sampling/back edge shift ● 0b1: Front edge shift/back edge sampling	RW
3:1	BRR	Master mode baud rate configuration ● 0b000: PCLK/2 ● 0b001: PCLK/4 ● 0b010: PCLK/8 ● 0b011: PCLK/16 ● 0b100: PCLK/32 ● 0b101: PCLK/64 ● 0b110: PCLK/128 ● 0b111: Setting prohibited	RW
0	EN	SPI module enable control ● 0b0: SPI module disabled ● 0b1: SPI module enabled	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

8.5.3 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 ● 0b0: Disable ● 0b1: Enable	RW
8	DMARXEN	Receive DMA enable ● 0b0: Disable ● 0b1: Enable	RW
7	MODFIE	Mode error interrupt enable control ● 0b0: Disable ● 0b1: Enable	RW
6	SSERRIE	Slave select error interrupt enable control in slave mode ● 0b0: Disable ● 0b1: Enable	RW
5	OVFIE	Receive buffer overflow error interrupt enable control ● 0b0: Disable ● 0b1: Enable	RW
4	UDFIE	Transmit buffer underflow error interrupt enable control in slave mode ● 0b0: Disable ● 0b1: Enable	RW
3	SSRIE	Slave select input rising edge interrupt enable control ● 0b0: Disable ● 0b1: Enable	RW
2	SSFIE	Slave select input falling edge interrupt enable control ● 0b0: Disable ● 0b1: Enable	RW
1	RXNEIE	Receive buffer non-empty interrupt enable control ● 0b0: Disable ● 0b1: Enable	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Bit	Name	Functional Description	Access
0	TXEIE	Transmit buffer empty interrupt enable control ● 0b0: Disable ● 0b1: Enable	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

8.5.4 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 ● 0b0: Receive only ● 0b1: Transmit only **Note** Only valid in single-wire half-duplex communication.	RW

8.5.5 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 Slave select output value is determined by the register in master mode. Slave select input value is determined by the register in slave mode. **Note** Valid when the SPI_CR0.SSM register is 1.	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

8.5.6 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 ● 0b0: Slave select input is low level ● 0b1: Slave select input is high level **Note** This bit setting is invalid when in master mode and SPI_CR0.SSM=1.	RO
8	BUSY	Bus busy flag bit ● 0b0: Bus idle ● 0b1: Bus busy	RO
7	MODF	Mode error flag bit ● 0b0: Normal ● 0b1: Error **Note** MODF is triggered when the slave select input is low in master mode.	RO
6	SSERR	Slave selection error flag in slave mode ● 0b0: Normal ● 0b1: Error	RO
5	OVF	Receive buffer overflow error flag bit ● 0b0: Normal ● 0b1: Error	RO
4	UDF	Transmit buffer underflow error flag in slave mode ● 0b0: Normal ● 0b1: Error	RO
3	SSR	Slave select input rising edge flag ● 0b0: No rising edge detected ● 0b1: Rising edge detected **Note** 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 ● 0b0: No falling edge detected ● 0b1: Falling edge detected **Note** This bit setting is invalid when in master mode and SPI_CR0.SSM=1.	RO
1	RXNE	Receive buffer not empty flag ● 0b0: Receive buffer empty ● 0b1: Receive buffer not empty	RO
0	TXE	Send buffer empty flag ● 0b0: Transmit buffer not empty ● 0b1: Transmit buffer empty	RO

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

8.5.7 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 Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
6	SSERR	SR.SSERR flag clear bit Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
5	OVF	SR.OVF flag clear bit Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
4	UDF	SR.UDF flag clear bit Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
3	SSR	SR.SSR flag clear bit Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
2	SSF	SR.SSF flag clear bit Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
1	RXNE	SR.RXNE flag clear bit Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
0	TXE	SR.TXE clear bit 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.

8.5.8 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 When sending data, write the data to be sent into this register When receiving data, read the received data from this register	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

9 CTRIM

9.1 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.

9.2 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 with a period range of 60μs to 65536s (at 32.768kHz frequency)
- Timing mode: 7 clock signals available for selection
- Timing mode: Toggle output

9.3 Functional Description

9.3.1 RCH Clock Calibration Mode

RCH Coarse and Fine One-Step Calibration Mode

When CTRIM_CR1.MD is set to 0b00 and CTRIM_CR1.MDRCH is set to 0b00, the module operates in RCH coarse/fine one-step calibration mode. This mode can automatically calibrate the output frequency of the RCH in real time, so that the accuracy of the RCH output frequency is no longer affected by environmental changes.

The operational block diagram of RCH clock calibration is shown below.

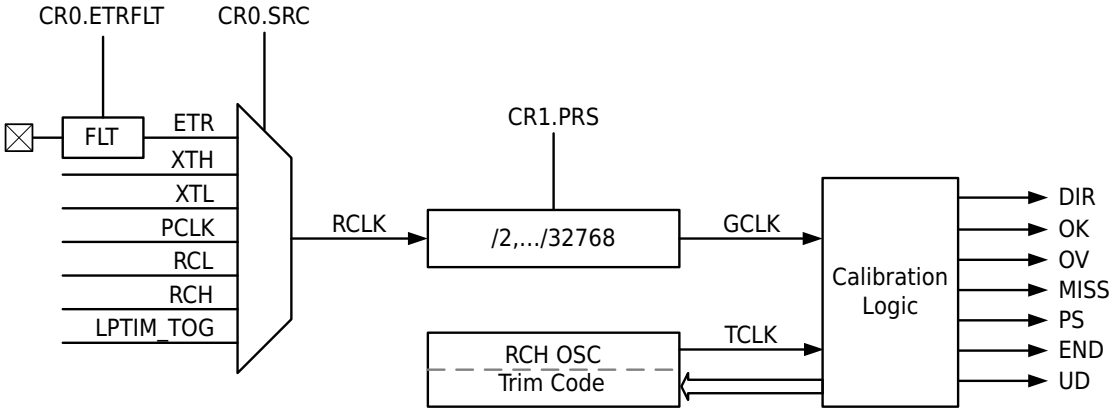


Figure 9-1 RCH Clock Calibration Operational Block Diagram

When calibrating RCH, it is necessary to provide an accurate low-frequency clock signal to the module, and the clock source can be selected through CTRIM_CR0.SRC. Note: The frequency of the low-frequency clock signal should be less than 1/2 of the RCH frequency. The low-frequency clock signal generates GCLK after the frequency division output set by CTRIM_CR1.PRS, and GCLK is used as the input of the calibration logic unit.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

When automatic calibration begins, the error counter performs down-counting on the TCLK signal (RCH oscillation frequency) starting from CTRIM_ARR with an initial step size set by CTRIM_CR0.STEP at each GCLK cycle, then switches to up-counting upon reaching zero.

If the error counter value within a GCLK cycle is less than or equal to the value of the CTRIM_FLIM register, CTRIM_ISR.OK is set to flag that the RCH output frequency meets expectations, and CTRIM_ISR.END is asserted. If the error counter value within a GCLK cycle is greater than the value of the CTRIM_FLIM register, the RCH Trim Code is automatically adjusted for the next counting cycle. The automatic adjustment rules are: At the end of each GCLK cycle, if the counting direction is down counting, then increase the step amount for the Trim Code of RCH; If the counting mode is up counting, then decrease the step amount for the Trim Code of RCH.

The calibration logic will automatically adjust the value of the step to achieve the frequency of the RCH output approaching the target value successively. When the step size is adjusted to 1, set CTRIM_ISR.END to mark the end of automatic calibration.

UD (underflow) event is generated and CTRIM_ISR.UD is set when the error counter count decrements to zero; halting further counting, MISS event occurs and CTRIM_ISR.MISS is set when the error counter count increments to CTRIM_ARR. Whenever the rising edge of GCLK arrives, the error counter starts counting again.

After single calibration completes, the user must write CTRIM_TVAL[10:0] to the SYSCTRL_RCHTRIM register and disable the calibration module if needed.

RCH Coarse Calibration Mode

When CTRIM_CR1.MD is set to 0b00 and CTRIM_CR1.MDRCH is set to 0b10, the module operates in RCH coarse calibration mode. The method is the same as above. After the calibration is completed, the user needs to write the CTRIM_TVAL[10:6] value to the SYSCTRL_RCHTRIM.TRIMH register and disable the calibration module as needed.

RCH Fine Calibration Mode

When CTRIM_CR1.MD is set to 0b00 and CTRIM_CR1.MDRCH is set to 0b11, the module operates in RCH fine calibration mode. The method is the same as above. After the calibration is completed, the user needs to write the CTRIM_TVAL[5:0] value to the SYSCTRL_RCHTRIM.TRIML register and disable the calibration module as needed.

9.3.2 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.

The operational block diagram of RCL clock calibration is shown below.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

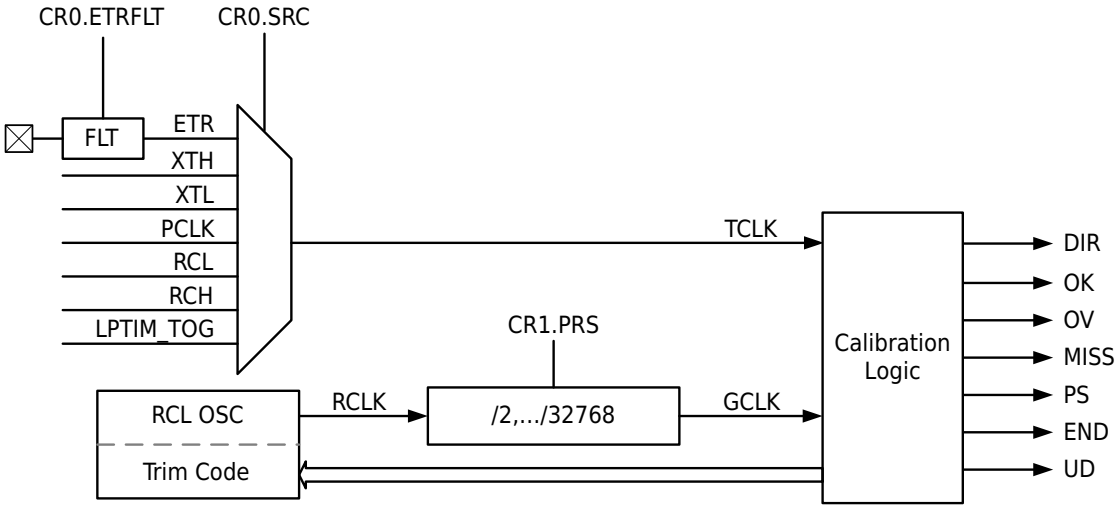


Figure 9-2 RCL Clock Calibration Operational Block Diagram

When calibrating RCL, it is necessary to provide an accurate high-frequency clock signal to the module, and the clock source can be selected through CTRIM_CR0.SRC. This clock source is input to the calibration logic as TCLK.

When automatic calibration begins, the error counter performs down-counting on the TCLK clock signal starting from CTRIM_ARR with an initial step size set by CTRIM_CR0.STEP 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 or equal to the value of CTRIM_FLIM register, CTRIM_ISR.OK is set to flag that the RCL output frequency meets expectations, and CTRIM_ISR.END is asserted. If the error counter value within a GCLK cycle is greater than the value of CTRIM_FLIM register, the Trim Code of RCL is automatically adjusted for the next counting cycle. The automatic adjustment rules are: At the end of each GCLK cycle, if the counting direction is down counting, the step amount is reduced for the Trim Code of RCL; if the counting mode is up counting, the step amount is increased for the Trim Code of RCL.

The calibration logic will automatically adjust the value of the step to achieve the frequency of the RCL output approaching the target value successively. When the step size is adjusted to 1, set CTRIM_ISR.END to mark the end of automatic calibration.

UD (underflow) event is generated and CTRIM_ISR.UD is set when the error counter value decrements to zero; halting further counting, MISS event occurs and CTRIM_ISR.MISS is set when the error counter value increments to CTRIM_ARR. Whenever the rising edge of GCLK arrives, the error counter starts counting again.

After a single calibration is complete, write CTRIM_TVAL[8:0] to the SYSCTRL_RCLTRIM[8:0] register and turn off the calibration module as needed.

Note

SYSCTRL_RCLTRIM[9] is the frequency selection bit and should remain unchanged.

9.3.3 Low Power Timer Mode

When CTRIM_CR1.MD is set to 0b10 or 0b11, the module operates in general low power timer 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 a period range of 60μs-65536s.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

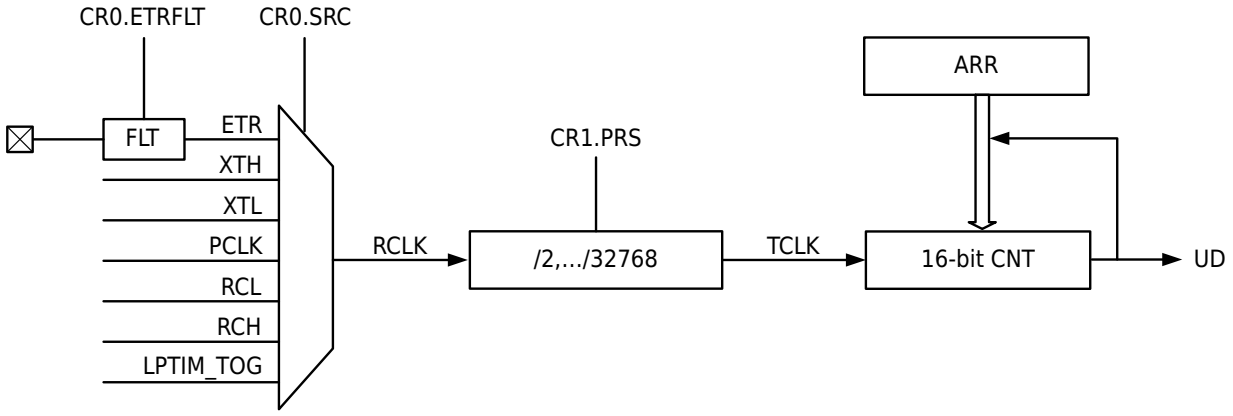


Figure 9-3 General Low Power Timer Scenario Operational Block Diagram

The timer counts down from ARR, and when it reaches 0, UD event is generated with CTRIM_ISR.UD set. If the interrupt(CTRIM_IER.UD) is enabled, UD interrupt is triggered, 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.

Note

The TOG output is unavailable when using the external ETR input.

9.4 Operation Example

9.4.1 RCH Clock Automatic Calibration Example

Procedure

Step1. According to the port multiplexing function, configure the function of the corresponding port as CTRIM_ETRTOG, and provide it with a 500Hz 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 250Hz.

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

Note

The RCH target calibration is 12MHz, the frequency of GCLK is 250Hz, the theoretical count value of the error counter for RCH output frequency within 4ms(250Hz) is 24M/500=48000, thus CTRIM_ARR=48000-1=47999.

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

Step7. Set CTRIM_CR0.STEP to 4, 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.

Note

Query CTRIM_ISR.OK to determine whether the accuracy meets the setting requirements when Trim is completed.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

9.4.2 RCL Clock Automatic Calibration Example

Procedure

Step1. According to the port multiplexing function, configure the function of the corresponding port as CTRIM_ETRTOG, and provide 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 32.768kHz), 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 RCL target calibration is 32.768kHz, the frequency of GCLK is 32Hz, the theoretical count value of the error counter for TCLK (1MHz) within 31.25ms (32Hz) is 1M/32=31250, thus CTRIM_ARR=312501=31249.

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

Step7. Set CTRIM_CR0.STEP to 4, 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.

> **Note**
> Query CTRIM_ISR.OK to determine whether the accuracy meets the setting requirements when Trim is completed.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

9.5 Register Description

9.5.1 Register List

Base address: 0x40005000

Table 9-1 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.

9.5.2 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

9.5.3 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.

9.5.4 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 ● 0b000: ETR ● 0b001: XTH ● 0b010: XTL ● 0b011: PCLK ● 0b100: RCL ● 0b101: RCH ● 0b110: LPTIM_TOG ● 0b111: Setting prohibited	RW
7	*Reserved*	-	-
6:4	ETRFLT	External input pin (ETR) filter configuration ● 0b000: No filter ● 0b001: F_{sample}=PCLK, N=2 ● 0b010: F_{sample}=PCLK, N=4 ● 0b011: F_{sample}=PCLK, N=6 ● 0b100: F_{sample}=PCLK/4, N=4 ● 0b101: F_{sample}=PCLK/4, N=6 ● 0b110: F_{sample}=PCLK/8, N=4 ● 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 ● 0b000: 0x10 ● 0b001: 0x02 ● 0b010: 0x04 ● 0b011: 0x08 ● 0b100: 0x10 ● 0b101: 0x20 ● 0b110: 0x40 ● 0b111: 0x80	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

9.5.5 Control Register 1 (CTRIM_CR1)

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																					MDRCH		OST	PRS				AUTO	MD		EN
Access																						RW		RW	RW				RW	RW		RW

Bit	Name	Functional Description	Access
31:11	*Reserved*	-	-
10:9	MDRCH	RCH calibration configuration ● 0b00: RCH coarse and fine one-step calibration 11bits ● 0b01: Setting prohibited ● 0b10: RCH coarse calibration 5bits ● 0b11: RCH fine tuning 6bits	RW
8	OST	Calibration mode configuration ● 0b0: Real-time calibration mode, the TRIM process continues ● 0b1: Single calibration mode, end when STEP becomes 1	RW
7:4	PRS	Reference clock prescaler configuration ● 0b0000: Setting prohibited ● 0b0001: Divide by 2 ● 0b0010: Divide by 4 ● 0b0011: Divide by 8 ● 0b0100: Divide by 16 ● 0b0101: Divide by 32 ● 0b0110: Divide by 64 ● 0b0111: Divide by 128 ● 0b1000: Divide by 256 ● 0b1001: Divide by 512 ● 0b1010: Divide by 1024 ● 0b1011: Divide by 2048 ● 0b1100: Divide by 4096 ● 0b1101: Divide by 8192 ● 0b1110: Divide by 16384 ● 0b1111: Divide by 32768	RW
3	AUTO	RCH/RCL automatic TRIM enable configuration ● 0b0: Disable ● 0b1: Enable	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Bit	Name	Functional Description	Access
2:1	MD	Operating mode configuration ● 0b00: RCH calibration mode, CR0.SRC needs to select a low-speed precision clock ● 0b01: RCL calibration mode, CR0.SRC needs to select a high-speed precision clock ● 0b10: Timer mode ● 0b11: Timer mode	RW
0	EN	Timer enable control ● 0b0: Disable ● 0b1: Enable	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

9.5.6 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 ● 0b0: Disable ● 0b1: Enable	RW
4	OV	TrimCode overflow interrupt enable ● 0b0: Disable ● 0b1: Enable	RW
3	MISS	Count failure interrupt enable ● 0b0: Disable ● 0b1: Enable	RW
2	PS	GCLK reference clock rising edgeinterrupt enable ● 0b0: Disable ● 0b1: Enable	RW
1	END	Auto Trim end interrupt enable ● 0b0: Disable ● 0b1: Enable	RW
0	UD	Error counter underflow interrupt enable ● 0b0: Disable ● 0b1: Enable	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

9.5.7 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 ● 0b0: Counting up, RCH output frequency is higher than the target frequency, RCL output frequency is lower than the target frequency ● 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 ● 0b0: Accuracy does not meet FLIM requirements when Trim is completed ● 0b1: Accuracy meets FLIM requirements when Trim is completed	RO
4	OV	TrimCode overflow flag ● 0b0: The TrimCode of RCH/RCL is not 0x000 or maximum ● 0b1: The TrimCode of RCH/RCL is adjusted to 0x000 or maximum	RO
3	MISS	Counting failure flag ● 0b0: The error counter does not count up to ARR during one cycle of GCLK ● 0b0: The error counter counts up to ARR during one cycle of GCLK	RO
2	PS	GCLK reference clock rising edge flag ● 0b0: No GCLK reference clock rising edge is detected ● 0b0: GCLK reference clock rising edge is detected	RO
1	END	Auto Trim end flag ● 0b0: Auto Trim is in progress ● 0b1: Auto Trim is completed	RO
0	UD	Error counter underflow flag ● 0b0: The error counter has not counted to 0 ● 0b1: The error counter has counted to 0	RO

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

9.5.8 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 Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
4	OV	Clear TrimCode overflow flag Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
3	MISS	Clear count failure flag Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
2	PS	Clear CLK reference clock rising edge flag Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
1	END	Clear auto Trim end flag Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
0	UD	Clear error counter underflow flag Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

9.5.9 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 **Note** At the end of each GCLK cycle, the count value of the error counter is automatically captured and stored in the FCAP register.	RO

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

9.5.10 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 **Note** 1. After auto TRIM completes, write the TVAL value to the corresponding TRIM register. 2. If reading during TRIM operation, two identical reads are required for validity.	RO

9.5.11 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.

10 FLASH

10.1 Introduction

This system includes a Flash memory with a capacity of 128KB/64KB, which is divided into 256/128 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.

10.2 Functional Description

10.2.1 Capacity Division

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

Table 10-1 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

Note

- 64KByte Flash consists of Sector0-127.
- 128KByte Flash consists of Sector0-255.

10.2.2 Read Wait Period

The fastest instruction fetch frequency supported by the built-in Flash of this device is 24MHz.When the CPU frequency (HCLK) is greater than 24MHz, the CPU must insert a wait cycle (set FLASH_WAIT greater than 0) to fetch instructions from the Flash. When FLASH_WAIT is equal to n (n = 1-2), every n + 1 HCLK can complete a Flash instruction fetch operation.

10.2.3 Flash Security Protection

Sector Erase/Write Protection

The entire 128KB/64KB Flash memory is divided into 256/128 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

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

is protected, the whole-chip erasing of the Flash is invalid, and an alarm flag and an interrupt signal are generated.

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. Based on this feature, only when the code is executed from RAM, can chip erase operation be performed on Flash.

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, FLASH_WAIT
- **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 and written through ISP and SWD interfaces
Level1	Data cannot be read Date can be downgraded to Level 0 via ISP and SWD interfaces. After downgrading, the Flash content is all FF
Level2	Data cannot be read Data can be downgraded to Level0 through ISP. After downgrading, the Flash content is all FF The SWD interfaces is disabled and the data cannot be read out

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Protection Level	Chip Function
Level3	Data cannot be read The functions of the ISP and SWD interfaces being unreadable, that is, the chip can only be programmed once

10.3 Operation Example

10.3.1 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 executed from FLASH, the CPU will stop fetching instructions, and the hardware will automatically wait for the operation to complete (FLASH_IFR.BUSY becomes 0); if the erase operation is executed from RAM, the CPU will not stop fetching instructions, user software should wait for the operation to complete (FLASH_IFR.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, and 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_SLOCK0/1 to 1 to remove the sector 's erasure protection.

Step6. Check whether the corresponding bit of FLASH_SLOCK0/1 is 1, if it is not 1, jump to Step 4.

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

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

Step8. Wait for FLASH_IFR.BUSY to becomes 0, and the Sector erase operation is completed.

Step9. To erase other Sectors, repeat Step4-Step8. If the erase operation is completed, proceed to the next step.

Step10. Write 0x5A5A and 0xA5A5 to the FLASH_BYPASS register in sequence to enable register rewriting.

Step11. Configure FLASH_CR.OP to 0, set the Flash operation mode to read.

Step12. Check if FLASH_CR.OP is 0, if it is not 0, jump to Step10, if it is 0, end the process.

10.3.2 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 FLASH, the operation will be prohibited and the corresponding error flag will be set; if the erase operation is performed from RAM, the CPU will not stop fetching instructions, and the user software should wait the operation completes (FLASH_IFR.BUSY becomes 0).

Procedure

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

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Step2. Configure FLASH_CR.OP to 3 and set the Flash operating 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 to 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 int *)0x00000000) = 0x00
```

Step11. Wait for FLASH_IFR.BUSY to become 0, and the chip erase operation is completed.

Step12. Write 0x5A5A and 0xA5A5 to the FLASH_BYPASS register in sequence to enable register rewriting.

Step13. Configure FLASH_CR.OP to 0, set the Flash operation mode to read.

Step14. Check if FLASH_CR.OP is 0, if it is not 0, jump to Step12, if it is 0, end the process.

10.3.3 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 writing 3 data 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 (FLASH_IFR.BUSY becomes 0); if the program is executed from RAM, the CPU will not stop fetching instructions, user software should wait for the operation to complete (FLASH_IFR.BUSY becomes 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_SLOCK0/1 to 1 to remove the sector 's erasure protection.

  Step6. Check whether the corresponding bit of FLASH_SLOCK0/1 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) = 0xAB
  ```

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

  Step9. To write Byte to other addresses, repeat Step7-Step8. If the operation is completed, proceed to the next step.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Step10. Write 0x5A5A and 0xA5A5 to the FLASH_BYPASS register in sequence to enable register rewriting.

Step11. Configure FLASH_CR.OP to 0, set the Flash operation mode to read.

Step12. Check if FLASH_CR.OP is 0, if it is not 0, jump to Step10, if it is 0, end the process.

- **Half-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_SLOCK0/1 to 1 to remove the sector 's erasure protection.

Step6. Check whether the corresponding bit of FLASH_SLOCK0/1 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 FLASH_IFR.BUSY to become 0, and the program is completed.

Step9. To write Half-word to other addresses, repeat Step7-Step8. If the operation is completed, proceed to the next step.

Step10. Write 0x5A5A and 0xA5A5 to the FLASH_BYPASS register in sequence to enable register rewriting.

Step11. Configure FLASH_CR.OP to 0, set the Flash operation mode to read.

Step12. Check if FLASH_CR.OP is 0, if it is not 0, jump to Step10, if it is 0, end the process.

- **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_SLOCK0/1 to 1 to remove the sector 's erasure protection.

Step6. Check whether the corresponding bit of FLASH_SLOCK0/1 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 FLASH_IFR.BUSY to become 0, and the program is completed.

Step9. To write Word to other addressed, repeat Step7-Step8. If the operation is completed, proceed to the next step.

Step10. Write 0x5A5A and 0xA5A5 to the FLASH_BYPASS register in sequence to enable register rewriting.

Step11. Configure FLASH_CR.OP to 0, set the Flash operation mode to read.

Step12. Check if FLASH_CR.OP is 0, if it is not 0, jump to Step10, if it is 0, end the process.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

10.3.4 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 writing 3 data 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 (FLASH_IFR.BUSY becomes 0).

It is recommended that the system clock should not be lower than 1 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 to 4 and set the FLASH operation mode to write.
  - **Step3.** Check if FLASH_CR.OP is 4, 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_SLOCK0/1 to 1 to remove the sector 's erasure protection.
  - **Step6.** Check whether the corresponding bit of FLASH_SLOCK0/1 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) = 0xAB
    ```

  - **Step8.** Wait for FLASH_IFR.BUSY to become 0, and the write operation is completed.
  - **Step9.** To write Byte to other addresses, repeat Step7-Step8.
  - **Step10.** If all the writing operation is completed, proceed to the next step.
  - **Step11.** Write 0x5A5A and 0xA5A5 to the FLASH_BYPASS register in sequence to enable register rewriting.
  - **Step12.** Configure FLASH_CR.OP to 1, set the Flash operation mode to write.
  - **Step13.** Check if FLASH_CR.OP is 1, if not, jump to Step11.
  - **Step14.** Check whether the FLASH_IFR.BUSY is 0. If it is 1, loop the query in this step, otherwise go to the next step.
  - **Step15.** Write 0x5A5A and 0xA5A5 to the FLASH_BYPASS register in sequence to enable register rewriting.
  - **Step16.** Configure FLASH_CR.OP to 0, set the Flash operation mode to read.
  - **Step17.** Check if FLASH_CR.OP is 0, if it is not 0, jump to Step15, if it is 0, end the process.
- **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 to 4 and set the FLASH operation mode to write.
  - **Step3.** Check if FLASH_CR.OP is 4, if not, jump to Step1.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

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

Step5. Set the corresponding bit of FLASH_SLOCK0/1 to 1 to remove the sector 's erasure protection.

Step6. Check whether the corresponding bit of FLASH_SLOCK0/1 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 FLASH_IFR.BUSY to become 0, and the program is completed.

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

Step10. If all the writing operation is completed, proceed to the next step.

Step11. Write 0x5A5A and 0xA5A5 to the FLASH_BYPASS register in sequence to enable register rewriting.

Step12. Configure FLASH_CR.OP to 1, set the Flash operation mode to write.

Step13. Check if FLASH_CR.OP is 1, if not, jump to Step11.

Step14. Check whether the FLASH_IFR.BUSY is 0. If it is 1, loop the query in this step, otherwise go to the next step.

Step15. Starting from this step, FLASH_CR.OP can be changed to the value of other modes to complete other functional operations.

Step16. Write 0x5A5A and 0xA5A5 to the FLASH_BYPASS register in sequence to enable register rewriting.

Step17. Configure FLASH_CR.OP to 0, set the Flash operation mode to read.

Step18. Check if FLASH_CR.OP is 0, if it is not 0, jump to Step16, if it is 0, end the process.

- **Word Program**

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

Step2. Configure FLASH_CR.OP Value to 4 and set the FLASH operation mode to write.

Step3. Check if FLASH_CR.OP is 4, 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_SLOCK0/1 to 1 to remove the sector 's erasure protection.

Step6. Check whether the corresponding bit of FLASH_SLOCK40/1 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 FLASH_IFR.BUSY to become 0, and the write operation is completed.

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

Step10. If all the writing operation is completed, proceed to the next step.

Step11. Write 0x5A5A and 0xA5A5 to the FLASH_BYPASS register in sequence to enable register rewriting.

Step12. Configure FLASH_CR.OP to 1, set the Flash operation mode to write.

Step13. Check if FLASH_CR.OP is 1, if not, jump to Step11.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Step14. Check whether the FLASH_IFR.BUSY is 0. If it is 1, loop the query in this step, otherwise go to the next step.

Step15. Starting from this step, FLASH_CR.OP can be changed to the value of other modes to complete other functional operations.

Step16. Write 0x5A5A and 0xA5A5 to the FLASH_BYPASS register in sequence to enable register rewriting.

Step17. Configure FLASH_CR.OP to 0, set the Flash operation mode to read.

Step18. Check if FLASH_CR.OP is 0, if it is not 0, jump to Step16, if it is 0, end the process.

10.3.5 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.

10.4 Register Description

10.4.1 Register List

Base address: 0x40020000

Table 10-3 Flash Controller (FLASH) Register Offset

Offset	Register	Description	Width
0x00	FLASH_CR	Control Register	32
0x04	FLASH_WAIT	Waiting Period Setting Register	32
0x08	FLASH_LOCK_ST	LOCK Status Register	32
0x0C	FLASH_IER	Interruption Enable Register	32
0x10	FLASH_IFR	Interrupt Flag Register	32
0x14	FLASH_ICR	Interrupt Flag Clear Register	32
0x18	FLASH_BYPASS	BYPASS Sequence Register	32
0x30	FLASH_SLOCK0	Sector Erase/Write Protection Register0	32
0x34	FLASH_SLOCK1	Sector Erase/Write Protection Register1	32

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

10.4.2 Control Register (FLASH_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	0x00000080																															
Name	Reserved																								DPSTB_EN	Reserved	RO	FWUP	Reserved	OP		
Access																									RW		RW	RW		RW		

Bit	Name	Functional Description	Access
31:8	*Reserved*	-	-
7	DPSTB_EN	FLASH dpstb enable Mask bit ● 0b0: When the system enters DeepSleep mode, FLASH does not enter low power consumption mode; ● 0b1: When the system enters DeepSleep mode, FLASH enters low power consumption mode; **Note** Before entering low-power mode, writing 1 to this bit can reduce the power consumption of the chip in Sleep or DeepSleep, but will increase the wake-up time.	RW
6	*Reserved*	-	-
5	RO	Read-only enable bit ● 0b0: Readable, programmable, erasable ● 0b1: Read only, not programmable or erasable	RW
4	FWUP	Fast wake-up enable bit ● 0b0: Normal wake up ● 0b1: Quick wake up	RW
3	*Reserved*	-	-
2:0	OP	FLASH operation ● 0b000: read ● 0b001: program ● 0b010: sector erase ● 0b011: chip erase ● 0b1xx: continuous writing **Note** After erase/write operations, the OP field must be restored to 0b00.	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

10.4.3 Waiting Period Setting Register (FLASH_WAIT)

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																														WAIT	
Access																															RW	

Bit	Name	Functional Description	Access
31:3	*Reserved*	-	-
2:0	WAIT	Read FLASH cycle ● 0b000: 1 cycles (applicable to 0-24 MHz) ● 0b001: 2 cycles (applicable to 24-48 MHz) ● 0b010: 3 cycles (applicable to 48-64 MHz) ● Others: Setting prohibited **Note** The highest clock of this series of products is 64MHz.	RW

10.4.4 LOCK Status Register (FLASH_LOCK_ST)

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																														STATE	
Access																															RO	

Bit	Name	Functional Description	Access
31:3	*Reserved*	-	-
2:0	STATE	FLASH Protection Status ● 0b000: Level0, ISP can read and write, SWD can read and write ● 0b001: Level1, ISP can be downgraded, SWD can be downgraded; data cannot be read out ● 0b010: Level2, ISP can be downgraded, SWD has no function; data cannot be read out ● 0b011: Level3, ISP has no function, SWD has no function; data cannot be read out ● Other: Reserved	RO

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

10.4.5 Interrupt Enable Register (FLASH_IER)

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																														PROT	PC
Access																															RW	RW

Bit	Name	Functional Description	Access
31:2	*Reserved*	-	-
1	PROT	FLASH erasing protection alarm interrupt enable ● 0b0: Disable ● 0b1: Enable	RW
0	PC	FLASH erase PC address alarm interrupt enable ● 0b0: Disable ● 0b1: Enable	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

10.4.6 Interrupt Flag Register (FLASH_IFR)

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	0x00000020																															
Name	Reserved																											BUSY	Reserved		PROT	PC
Access																												RO			RO	RO

Bit	Name	Functional Description	Access
31:5	*Reserved*	-	-
4	BUSY	Idle/Busy Flag Bit ● 0b0: Non-program status ● 0b1: Program in progress	RO
3:2	*Reserved*	-	-
1	PROT	FLASH erasing protection alarm interrupt flag bit ● 0b0: No erasing operation has occurred on the protected sector ● 0b1: An over-erase operation occurred on the protected sector	RO
0	PC	FLASH erase PC address alarm interrupt flag bit ● 0b0: The sector where the PC address is located has not been erased ● 0b1: The sector where the PC address is located has been erased	RO

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

10.4.7 Interrupt Flag Clear Register (FLASH_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	0x0000000F																															
Name	Reserved																														PROT	PC
Access																															WO	WO

Bit	Name	Functional Description	Access
31:2	*Reserved*	-	-
1	PROT	FLASH erasing protection alarm interrupt flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
0	PC	FLASH erase PC address alarm interrupt flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO

10.4.8 BYPASS Sequence Register (FLASH_BYPASS)

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																BYSEQ															
Access																	RW															

Bit	Name	Functional Description	Access
31:16	*Reserved*	-	-
15:0	BYSEQ	Bypass sequence bit Write 0x5A5A first , then wirte 0xA5A5 to FLASH_BYPASS to enable write access to FLASH module registers (specific registers described in "Erase/Write Protection"). Once a write to any FLASH module register is performed, this protection bit automatically reverts to the protected state. Subsequent writes to FLASH module registers require repeating the previous procedures.	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

10.4.9 Sector Erase/Write Protection Register 0 (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 ● 0b0: Disable erase/write ● 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.

10.4.10 Sector Erase/Write Protection Register 1 (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 ● 0b0: Disable erase/write ● 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.

11 RAM

11.1 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.

11.2 Functional Description

11.2.1 RAM Address Range

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

Table 11-1 RAM Address Mapping

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

11.2.2 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.

11.2.3 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.

11.3 Register Description

11.3.1 Register List

Base address: 0x40020400

Table 11-2 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.

11.3.2 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	0x00000000																															
Name	Reserved																													IE	Reserved	
Access																														RW		

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

11.3.3 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 After the interrupt flag is cleared, the corresponding address is also cleared.	RO

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

11.3.4 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 ● 0b0: when reading RAM, the parity check is correct ● 0b1: parity check error when reading RAM	RO

11.3.5 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 Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

12 DMAC

12.1 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 5 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.

12.2 Features

DMA controller supports the following features:

- 5 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.

12.3 Functional Description

12.3.1 Function Block Diagram

- DMAC

  DMAC has 5 DMA channels, and when conflicts occur between channels, arbitration is performed by the priority controller.

- 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.

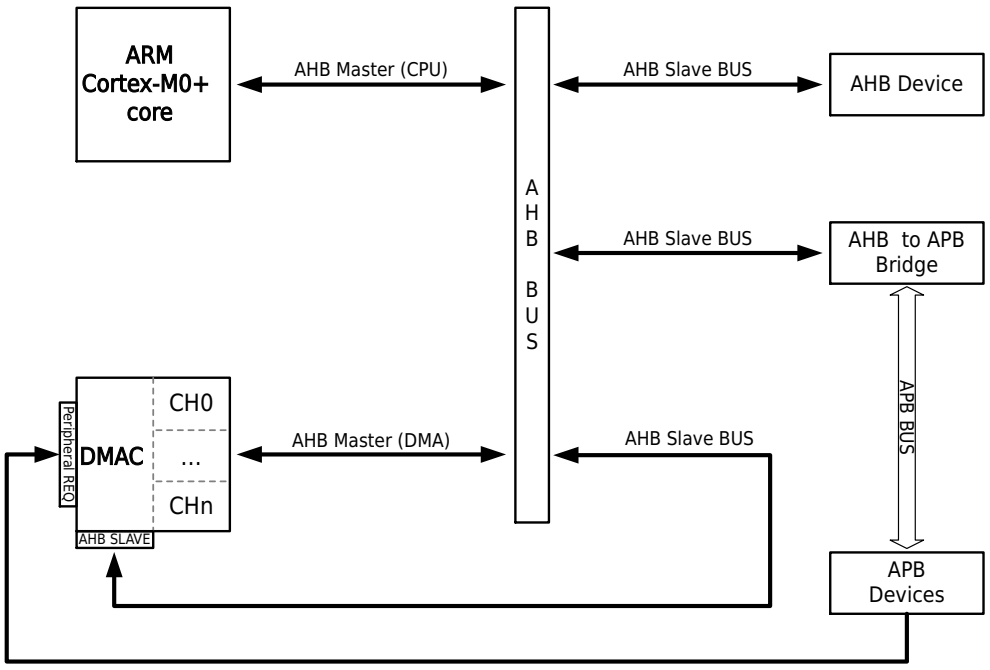


Figure 12-1 DMA Controller Functional Block Diagram

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

12.3.2 DMAC Transfer Mode

12.3.2.1 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 12-1 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 ● Memory to peripherals without DMA requests		● Memory to peripherals with DMA requests ● Peripherals with DMA requests to memory	● Memory to memory ● Memory to peripherals without DMA requests

12.3.2.2 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.

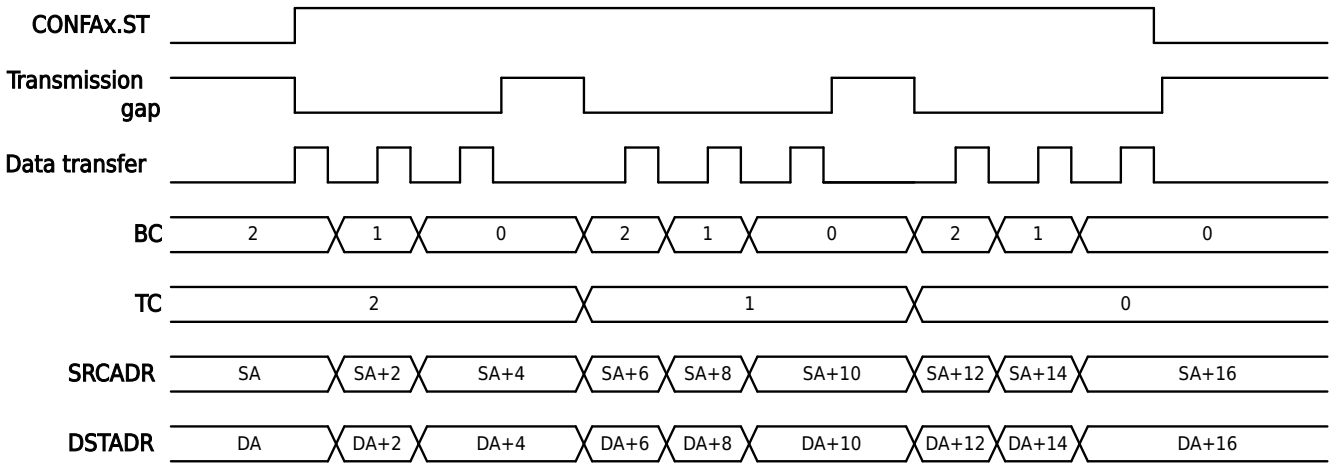


Figure 12-2 DMA Software Block Transfer Mode

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

12.3.2.3 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.

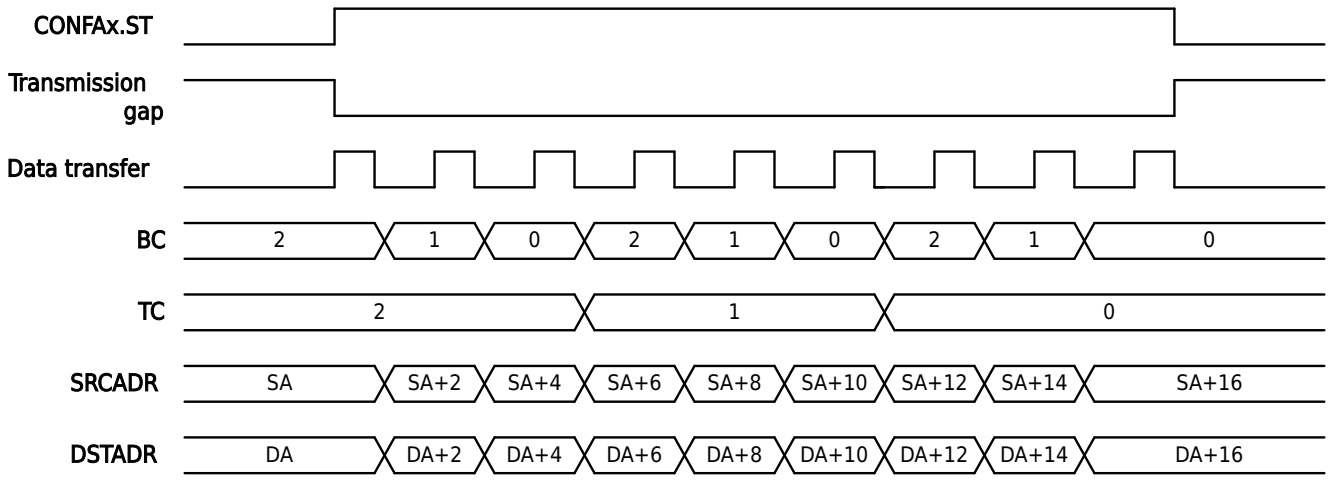


Figure 12-3 DMA Software Burst Transfer Mode

12.3.2.4 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.

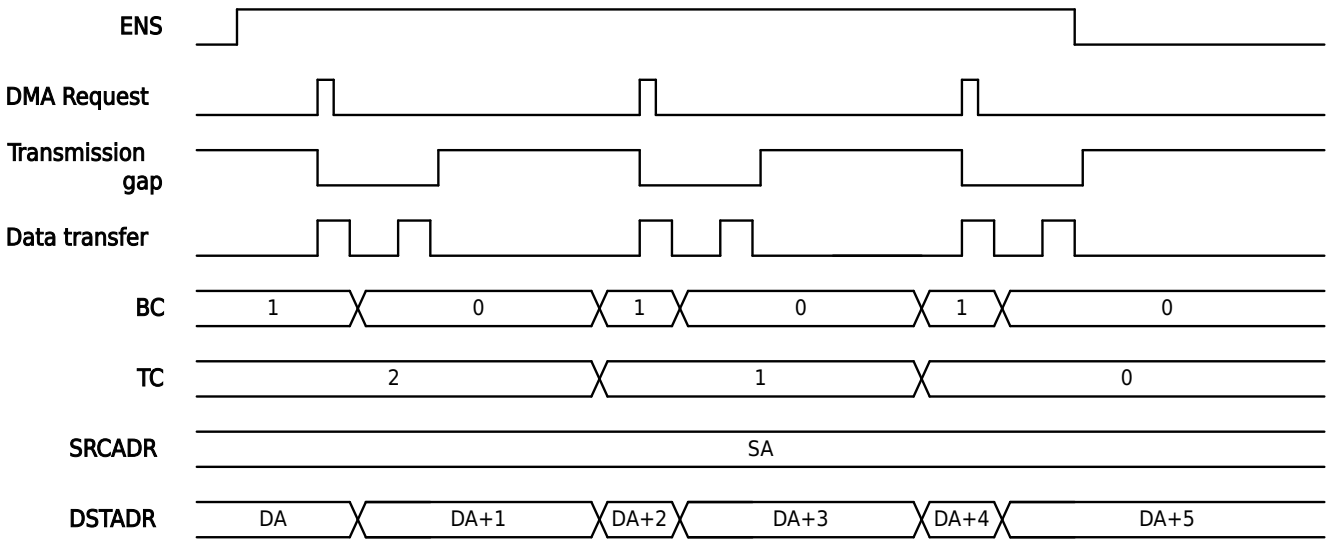


Figure 12-4 DMA Hardware Block Transfer Mode

12.3.2.5 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.

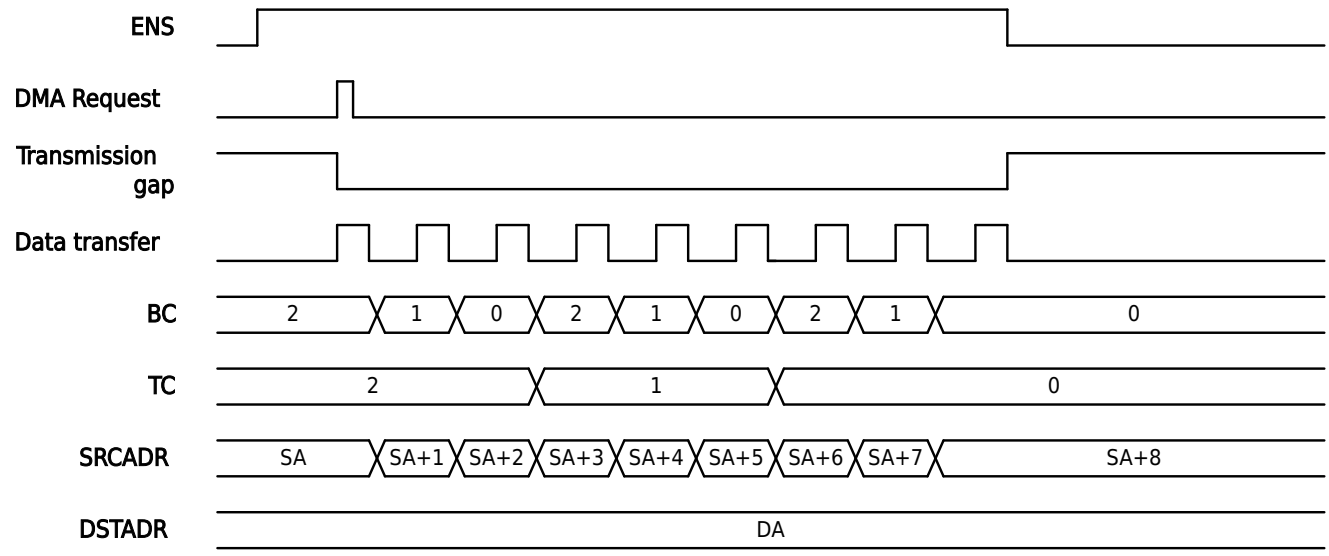


Figure 12-5 DMA Hardware Burst Transfer Mode

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

12.3.3 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.

12.3.4 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 thhis time, the priority order is: CH0 > CH1 > CH2 > CH3 > CH4. 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, CH1, CH2, CH3 and CH4 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.

DMA Operation in FLASH Continuous Writing Mode

- The DMA operation in FLASH continuous writing mode only supports the program running in RAM, and is aimed at the DMA source address being a non-FLASH address segment, FLASH cont-program scenario where the destination address is the FLASH address segment.
- DMA source addresses are prohibited from using the FLASH address range from 0x00000000 to 0x0001FFFF. Other address ranges are permissible as source addresses for continuous programming.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

- Operation process after DMA continuous programming:

  After DMA data transfer completes, the cont-program mode (FLASH_CR.OP=4) must be switched to write mode (FLASH_CR.OP=1), wait until FLASH_IFR.BUSY=0 in FLASH, and finally switch to read mode (FLASH_CR.OP=0), then the data written to the FLASH area can be correctly read.
- For the detailed FLASH continuous write operation configuration procedure, refer to the "Operation Example-Cont-Program" in the "FLASH Controller (FLASH)" chapter.

12.3.5 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.

Support conditions that trigger the DMA interrupt are as follows:

- Transport address out of addressing range
- Peripheral request to stop DMA
- Access transfer source address error
- Access transfer destination address error
- Transmission completed

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

12.4 Register Description

12.4.1 Register List

Base address: 0x40021000

Table 12-2 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=0-4)	32
0x14+0x10*x	DMAC_CONFBx	Channel x configuration B register(x=0-4)	32
0x18+0x10*x	DMAC_SRCADRx	Channel x transfer source address register (x=0-4)	32
0x1C+0x10*x	DMAC_DSTADRx	Channel x transfer destination address register (x=0-4)	32

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

12.4.2 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 ● 0b0: Disable ● 0b1: Enable **Notice** 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 ● 0b0: Fixed priority mode, CH0>CH1>CH2>CH3>CH4 priority ● 0b1: Cycle priority mode, CH0/CH1/CH2/CH3/CH4 take turns to obtain bus access	RW
27:24	HALT	DMA all channel transfer pause control ● 0b0000: Resume all channel data transmission ● 0b0001-0b1111: Suspend all channel data transmission	RW
23:0	*Reserved*	-	-

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

12.4.3 Channel x Configuration A Register (DMAC_CONFAx) (x=0-4)

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																
Access	RW	RW	RW	RW												RW																

Bit	Name	Functional Description	Access
31	ENS	DMA channel enable control ● 0b0: Disable the current DMA channel ● 0b1: Enable the current DMA channel; if CONFBx.MSK is 0, this bit is automatically cleared when the transfer is completed **Notice** ● When the DMA controller is disabled, ongoing transfers will be forcibly terminated, resulting in unpredictable consequences. ● 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 ● 0b0: Resume DMA channel data transfer ● 0b1: Suspend DMA channel data transfer	RW
29	ST	DMA channel software trigger control ● 0b0: Stop DMA channel software Block/Burst transfer ● 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 ● 0x00: Software trigger ● 0x40: USART0 sending Buf is empty ● 0x41: USART0 receiving Buf is not empty ● 0x42: USART1 sending Buf is empty ● 0x43: USART1 receiving Buf is not empty ● 0x44: LPUART0 sending Buf is empty ● 0x45: LPUART0 receiving Buf is not empty ● 0x46: LPUART1 sending Buf is empty ● 0x47: LPUART1 receiving Buf is not empty ● 0x48: GTIM0 occurrs capture event or compare event ● 0x49: GTIM1 occurrs capture event or compare event ● 0x4A: Setting prohibited ● 0x4B: Setting prohibited ● 0x4C: ADC JQR scan conversion completed ● 0x4D: ADC SQR scan singel conversion completed ● 0x4E: Setting prohibited	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Bit	Name	Functional Description	Access
28:22	TRI_SEL	● 0x4F: Setting prohibited ● 0x50: SPI0 sending Buf is empty ● 0x51: SPI0 receiving Buf is not empty ● 0x52: SPI1 sending Buf is empty ● 0x53: SPI1 receiving Buf is not empty ● 0x54: I2CSLV receiving Buf is not empty ● 0x55: I2CSLV sending Buf is empty ● 0x56: Setting prohibited ● 0x57: Setting prohibited ● 0x58: ATIM0 channel A, occurrs capture event or compare event ● 0x59: ATIM0 channel B, occurrs capture event or compare event ● 0x5A: ATIM1 channel A, occurrs capture event or compare event ● 0x5B: ATIM1 channel B, occurrs capture event or compare event ● 0x5C: ATIM2 channel A, occurrs capture event or compare event ● 0x5D: ATIM2 channel B, occurrs capture event or compare event ● 0x5E: ATIM3 channel A, occurrs capture event or compare event ● 0x5F: ATIM3 channel B, occurrs capture event or compare event ● Others: Setting prohibited	RW
21:20	*Reserved*	-	-
19:16	BC	DMA channel transfer data block size Configure the size of the data block to be transmitted by the DMA channel as BC+1 **Note** 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 Configure the number of data blocks to be transmitted by the DMA channel as TC+1 During transmission, every time a data block is transmitted, the value of this register will be decremented by 1 **Note** The total amount of data transferred by DMA is (TC +1)*(BC +1).	RW

Note

The trigger source of ATIM0/1/2/3:

- ATIM0/1/2/3 channel A: Compare capture A, external trigger
- ATIM0/1/2/3 channel B: Compare capture B, event update

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

12.4.4 Channel x Configuration B Register (DMAC_CONFBx) (x=0-4)

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 ● 0b00: Block transfer ● 0b01: Burst transfer ● 0b10: Setting prohibited ● 0b11: Setting prohibited	RW
27:26	WIDTH	DMA channel transfer data width configuration ● 0b00: 8-bit ● 0b01: 16-bit ● 0b10: 32-bit ● 0b11: Setting prohibited	RW
25	FS	DMA channel source address auto-increment enable ● 0b0: The source address will be incremented automatically after each data transmission is completed ● 0b1: The source address remains unchanged during transmission **Note** 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 ● 0b0: The target address will be incremented automatically after each data transmission is completed ● 0b1: The destination address remains unchanged during transmission **Note** 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 ● 0b0: Disable BC/TC reloading, and the values of BC/TC are all 0 after the transmission is completed ● 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 ● 0b0: Disable source address overloading, the value of the source address is uncertain after the transmission is completed ● 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 ● 0b0: Disable destination address overloading, the value of the destination address is uncertain after the transmission is completed ● 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 ● 0b0: Disable ● 0b1: Enable **Note** 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 ● 0b0: Disable ● 0b1: Enable **Note** 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 ● 0b000: Initial value ● 0b001: Transmission error, the transmission address exceeds the addressing range ● 0b010: Transmission error, peripheral request to stop DMA ● 0b011: Transmission error, access transmission source address error ● 0b100: Transmission error, access transmission destination address error ● 0b101: Transmission completed ● 0b110: Setting prohibited ● 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 ● 0b0: CONFAx.ENS is automatically cleared when the DMA channel transfer completes ● 0b1: CONFAx.ENS remains unchanged when DMA channel transfer completes	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

12.4.5 Channel x Transfer Source Address Register (DMAC_SRCADRx) (x=0-4)

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 Auto-reload can be configured via CONFBx.RS **Notice** The address alignment should match the data width, otherwise, it will cause address access errors.	RW

12.4.6 Channel x Transfer Destination Address Register (DMAC_DSTADRx) (x=0-4)

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 Auto-reload can be configured via CONFBx.RD **Notice** The address alignment should match the data width, otherwise, it will cause address access errors.	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

13 ATIM

13.1 Introduction

ATIM0/1/2 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, with support for 2 capture inputs.

ATIM3 is a timer composed of 1 counting unit and 7 comparison units, supporting 6 independent PWM outputs or 3 pairs of complementary PWM outputs, with support for 6 capture inputs.

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 ATIM0/1/2 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
- Dead-time control
- Braking control
- Cycle-by-cycle current limiting
- Edge alignment, symmetric center alignment and asymmetric center alignment PWM output

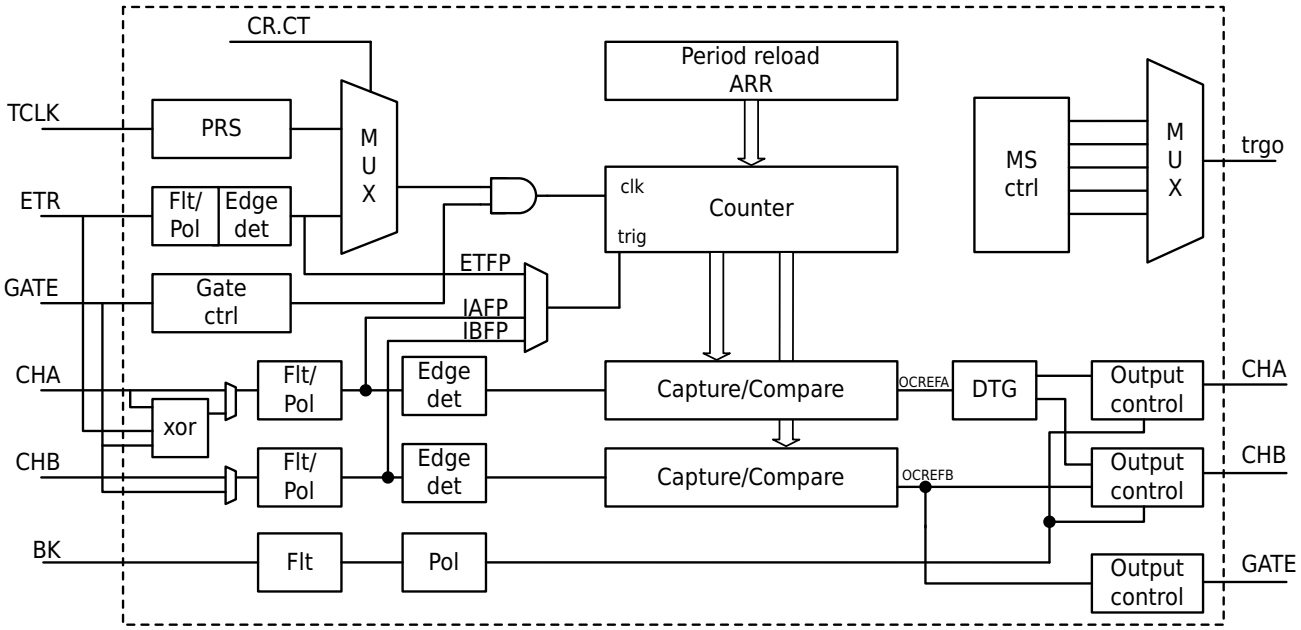


Figure 13-1 ATIM0/1/2 Block Diagram

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

13.2.2 ATIM3 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)
- Up to 6 capture inputs
- Pulse width measurement
- Quadrature encoder counting function
- Single pulse mode
- External counting function
- Dead-time control
- Braking control
- Cycle-by-cycle current limiting
- Edge alignment, symmetric center alignment and asymmetric center alignment PWM output

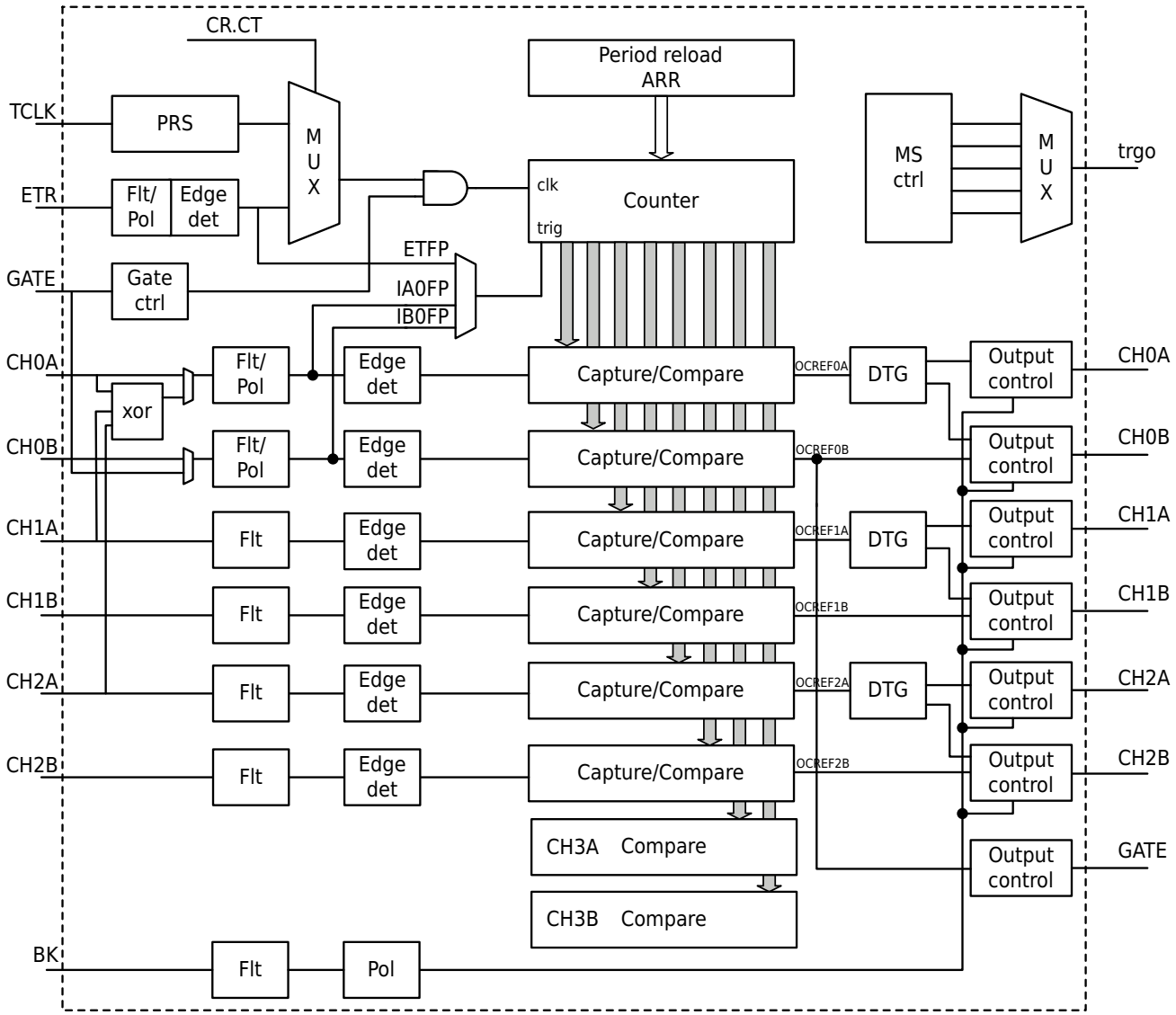


Figure 13-2 ATIM3 Block Diagram

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

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:

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.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Function Block Diagram

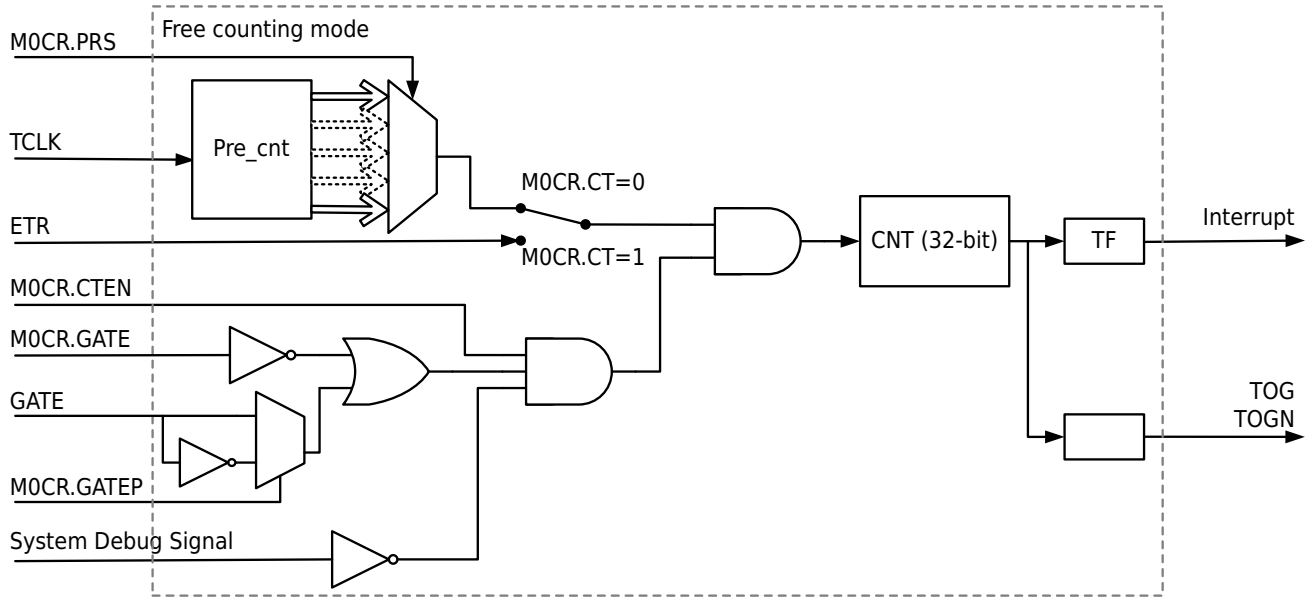


Figure 13-3 Mode0 Count Timer Free Counting Block Diagram

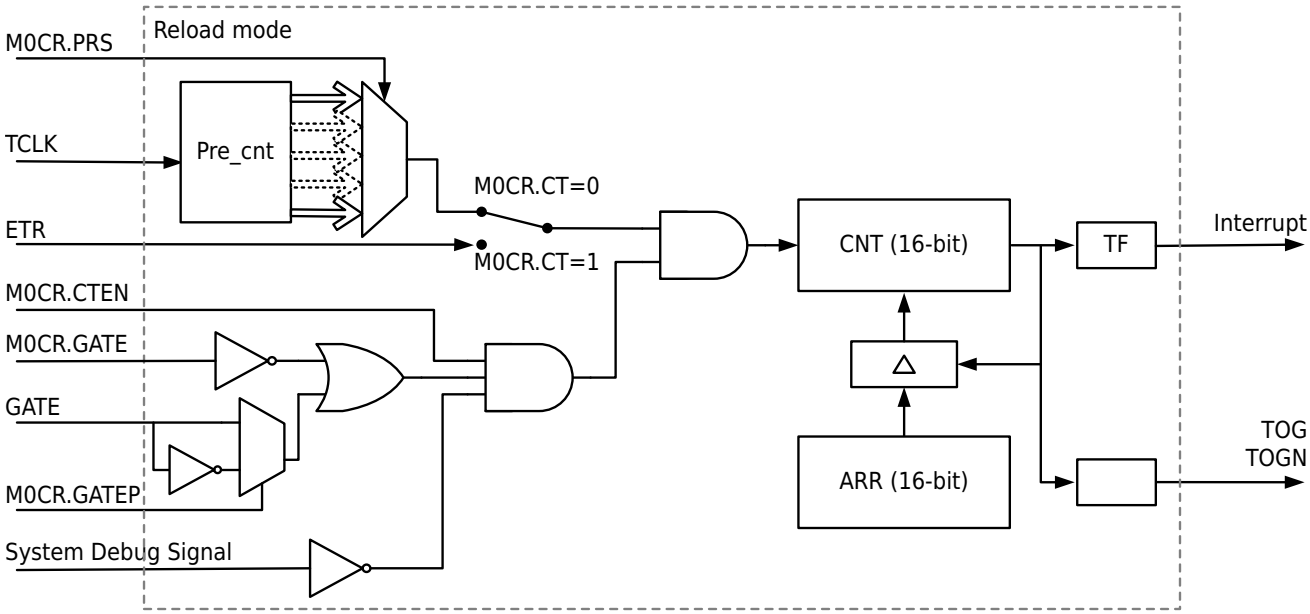


Figure 13-4 Mode0 Count Timer Reload Counting Block Diagram

Note

The "System Debug Signal" in the block diagram originates from SYSCTRL_DBG_STOP.ATIMx.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Count Waveform

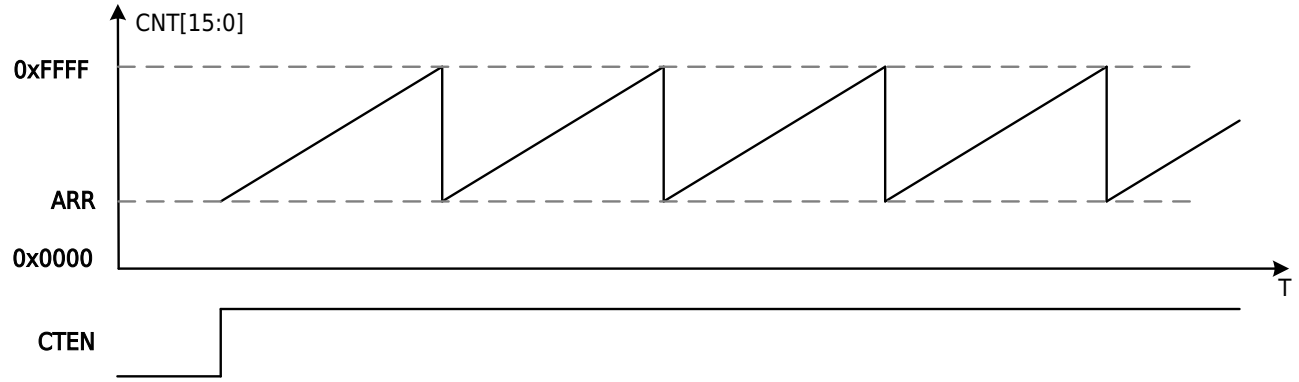


Figure 13-5 Mode0 Count Timer 16-Bit Reload Counting Waveform

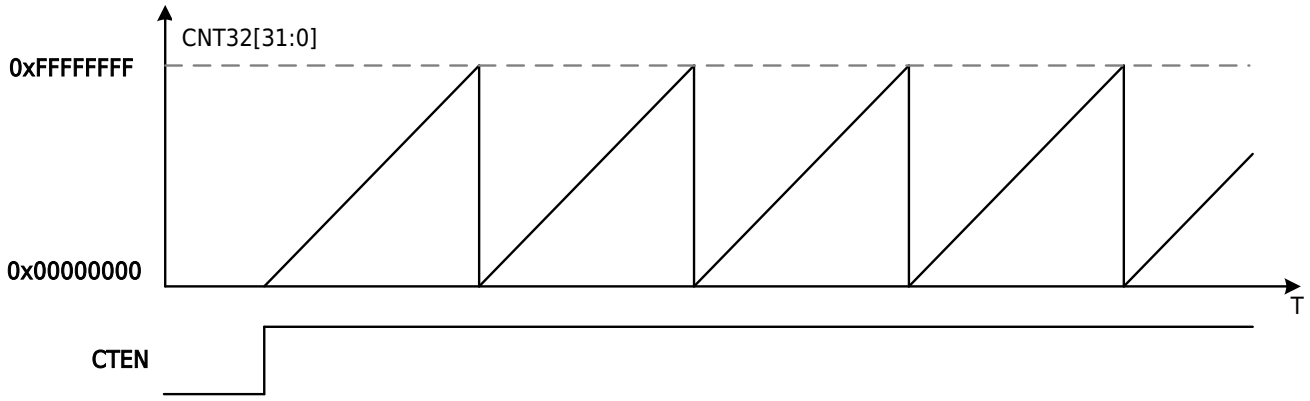


Figure 13-6 Mode0 Count Timer 32-Bit Free Counting Waveform

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.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Timing Diagram

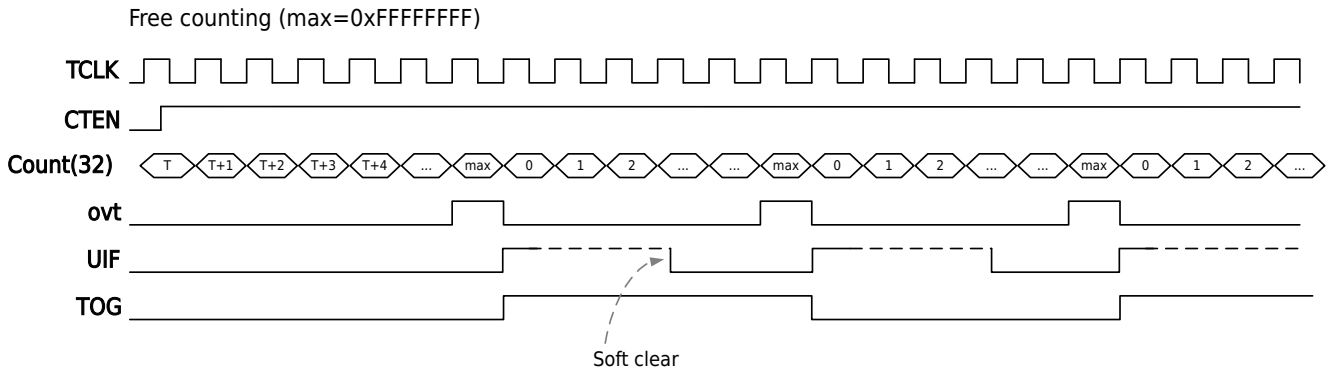


Figure 13-7 Mode0 Count Timer Free Counting Timing Diagram

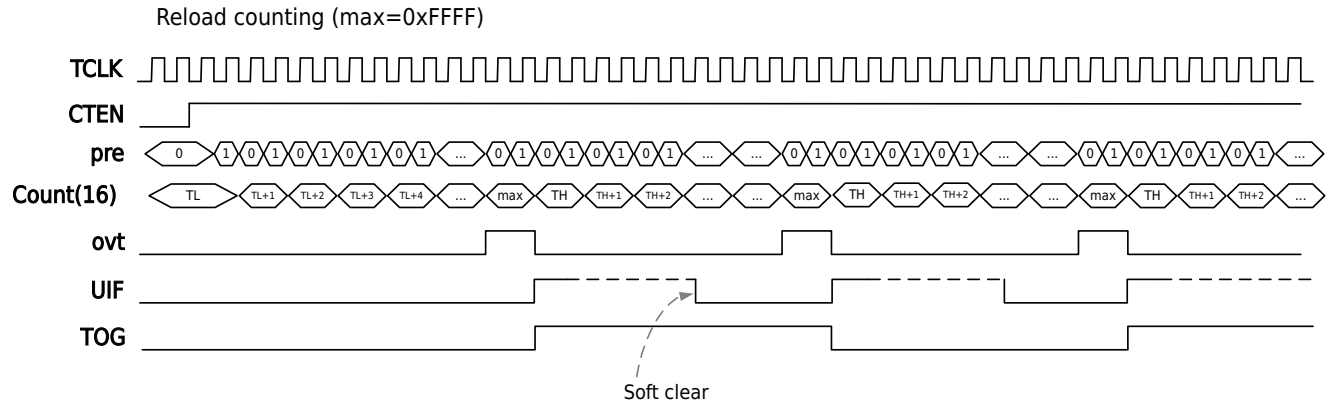


Figure 13-8 Mode0 Count Timer Reload Counting Timing Diagram (Prescaler=2)

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:

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 PWC, 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

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Pulse width measurement	Upper edge-upper edge Cycle width	Upper edge-lower edge High level width	Lower edge-upper edge Low level width	Lower edge-lower edge Cycle width

During period measurement, a period is measured at intervals of one period.

PWC Function Block Diagram

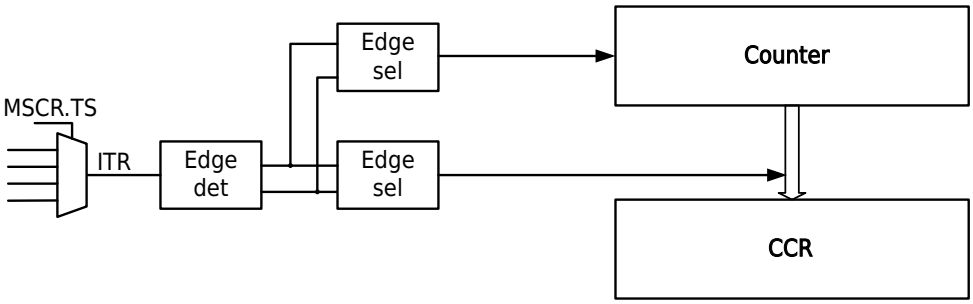


Figure 13-9 Mode1 Pulse Width Measurement PWC Function Block Diagram

Select the measurement signal source by registering MSCR.TS.

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)

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

PWC Waveform Measurement Timing Diagram

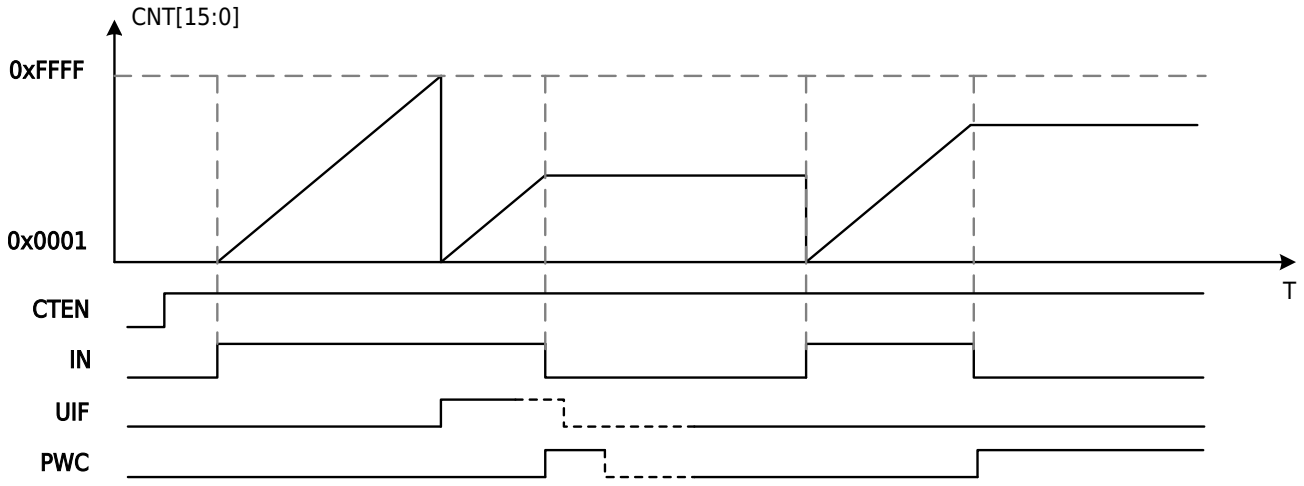


Figure 13-10 Mode1: Pulse Width Measurement PWC High-Level Pulse Width Measurement

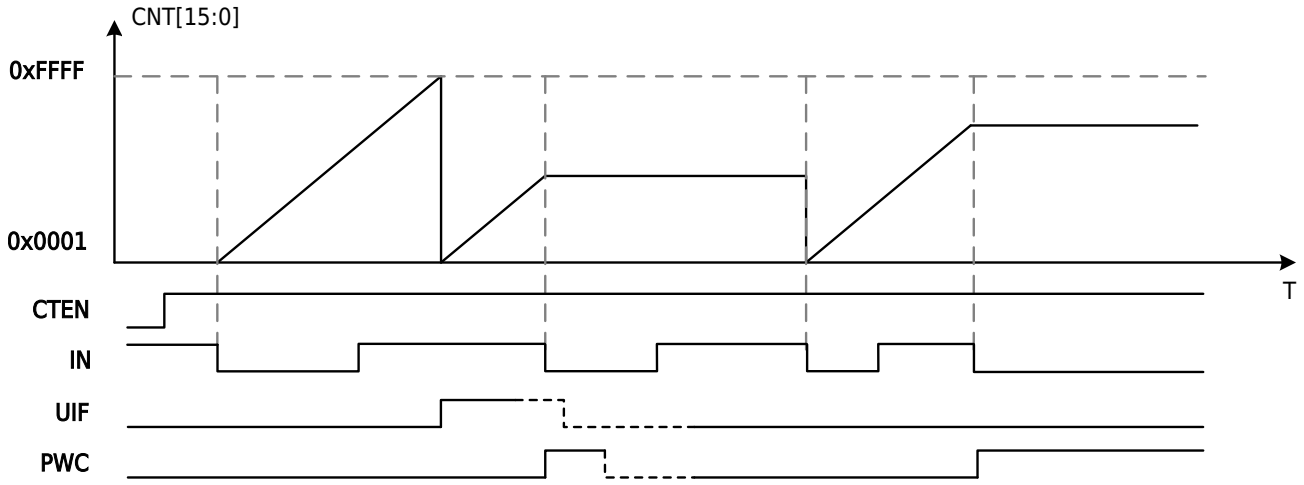


Figure 13-11 Mode1: Pulse Width Measurement PWC Falling Edge to Falling Edge Period Measurement

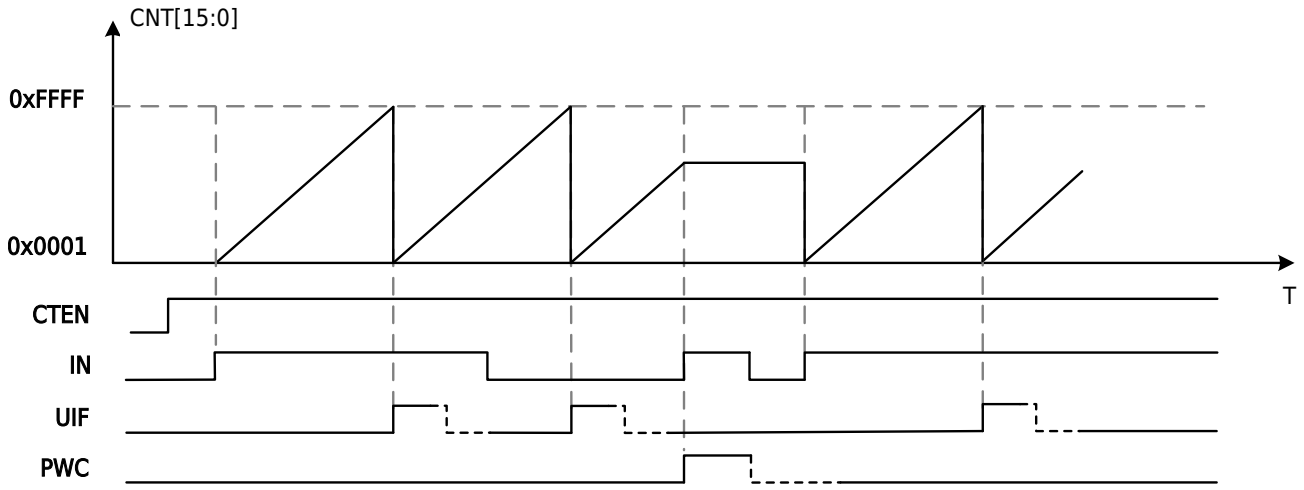


Figure 13-12 Mode1: Pulse Width Measurement PWC Rising Edge to Rising Edge Period Measurement

PWC Single trigger Mode

Set M1CR.ONESHOT=1 to set PWC single measurement, and CTEN will be cleared after the measurement is completed.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

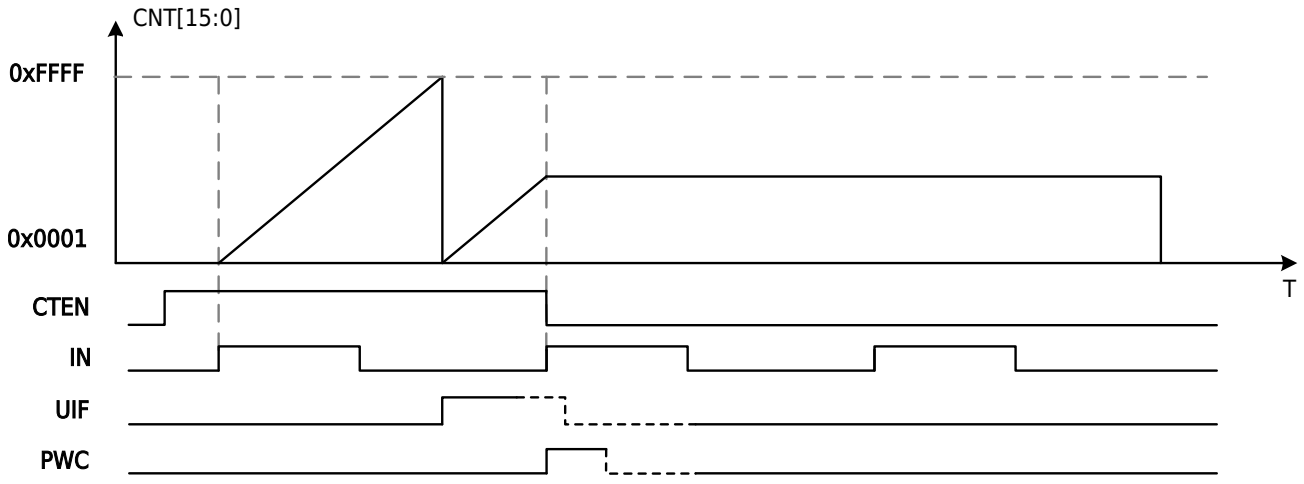


Figure 13-13 Mode1: Pulse Width Measurement PWC Rising Edge to Rising Edge Period Measurement Single Mode

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.

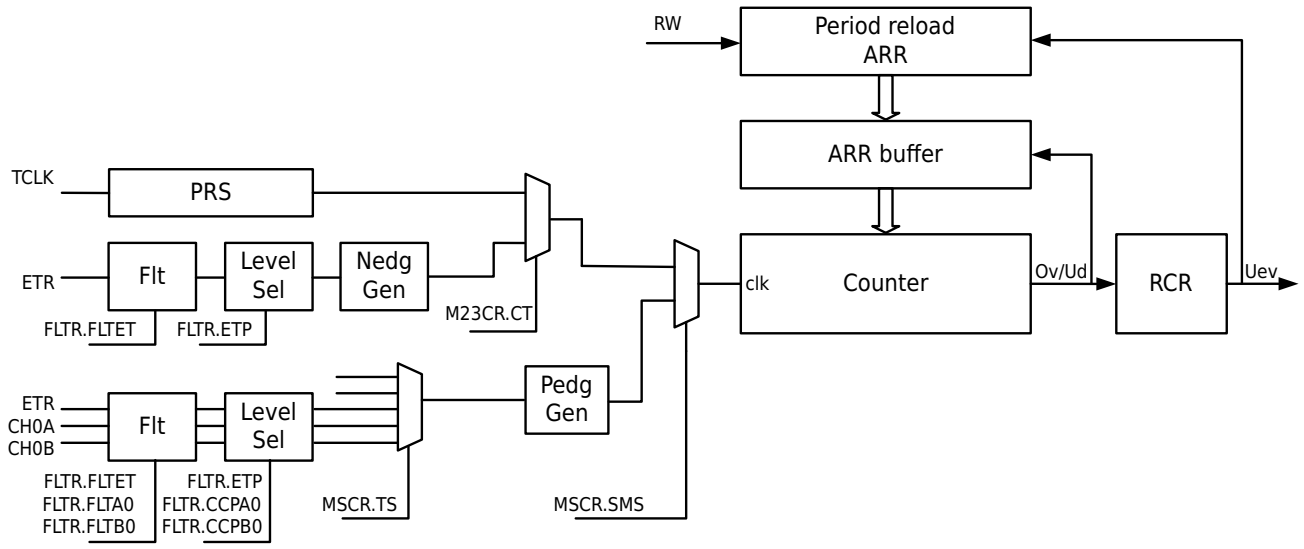


Figure 13-14 Mode 2/3 Compare Capture Mode Counter Block Diagram

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

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

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 (ATIMx_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 number of up-counting repetitions reaches the programmed value in the repeat counter register plus one (ATIMx_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 (ATIMx_IFR.UI 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.

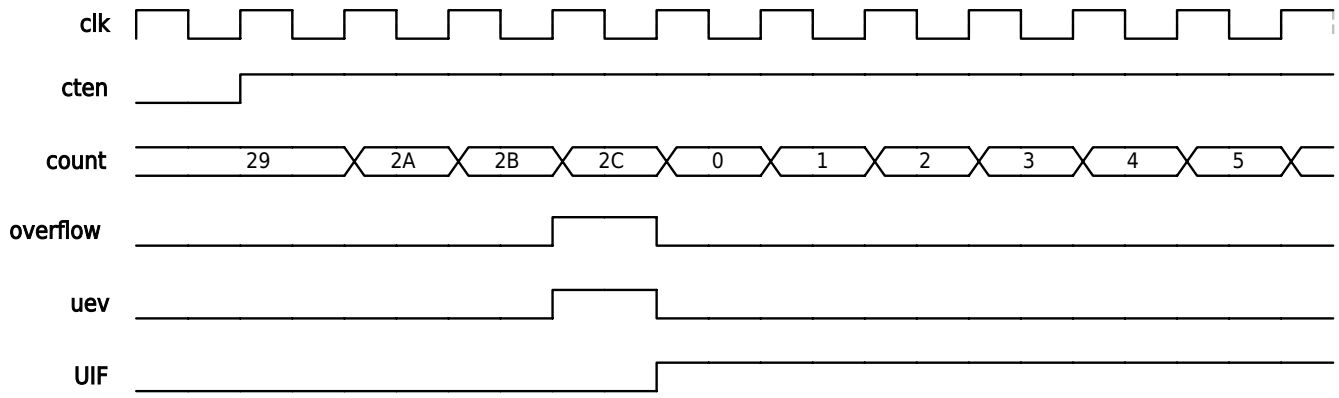


Figure 13-15 Mode 2 Compare Capture Mode Up-Counting Without Prescaler

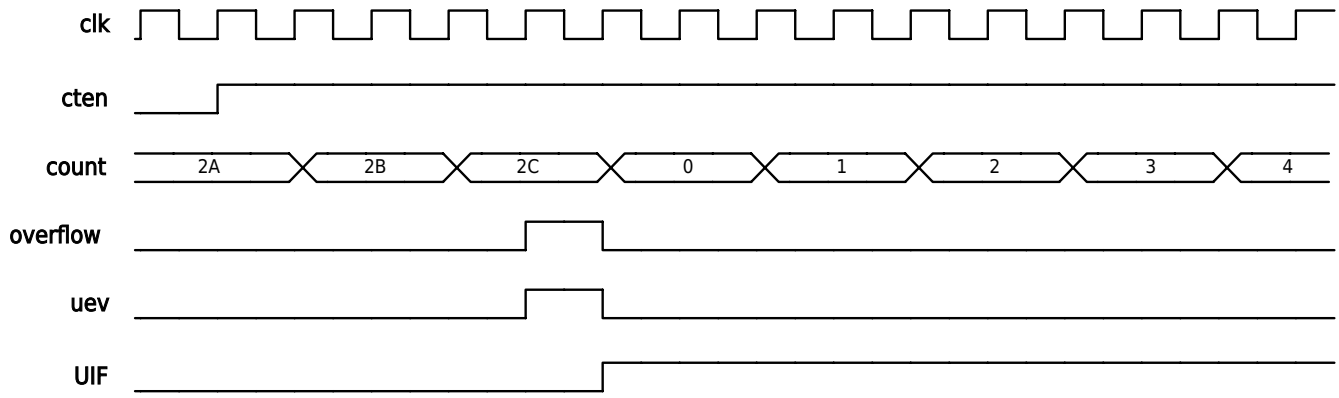


Figure 13-16 Mode 2 Compare Capture Mode Up-Counting With Prescaler

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

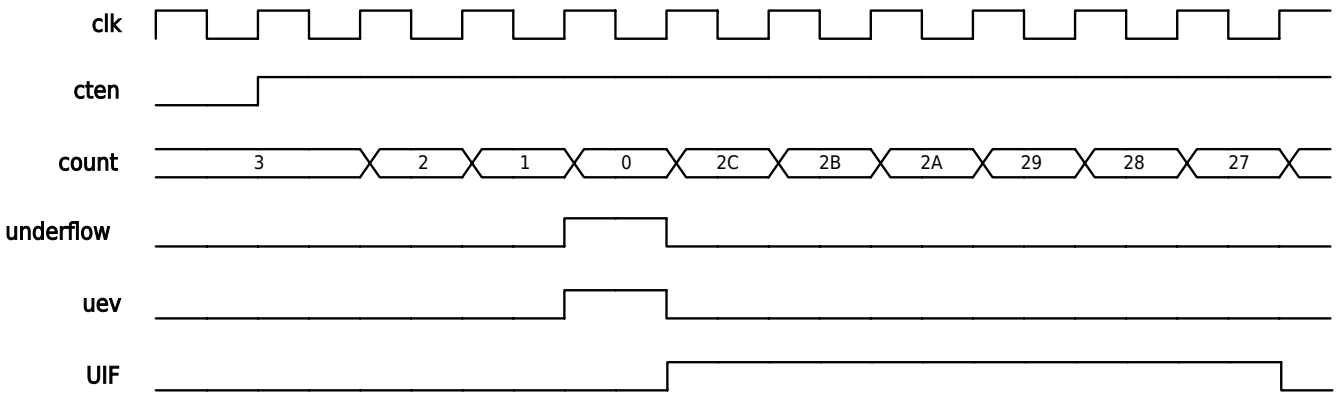


Figure 13-17 Mode 2 Compare Capture Mode Down-Counting Without Prescaler

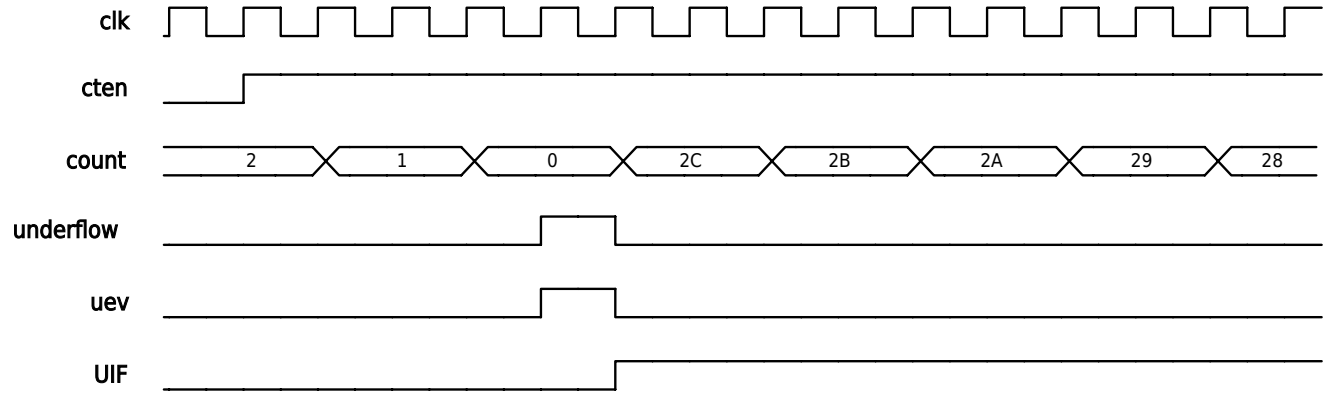


Figure 13-18 Mode 2 Compare Capture Mode Down-Counting With Prescaler

The counting waveforms for mode 2 are shown in the following diagrams:

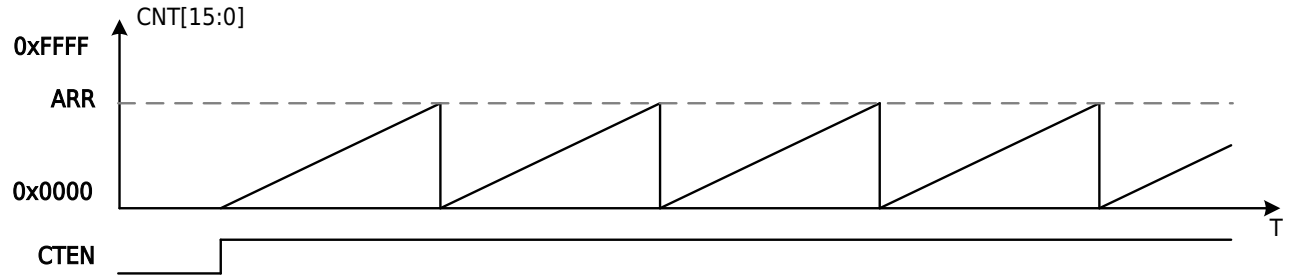


Figure 13-19 Mode2 Edge-Aligned Timer Waveform (M23CR.DIR=0)

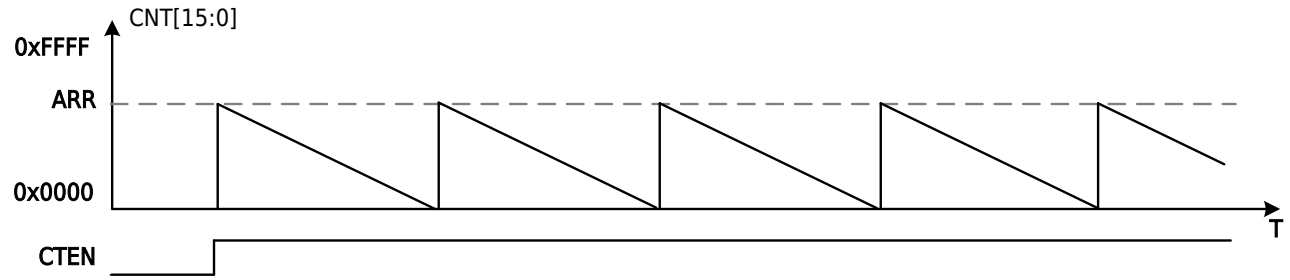


Figure 13-20 Mode2 Edge-Aligned Timer Waveform (M23CR.DIR=1)

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.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

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.

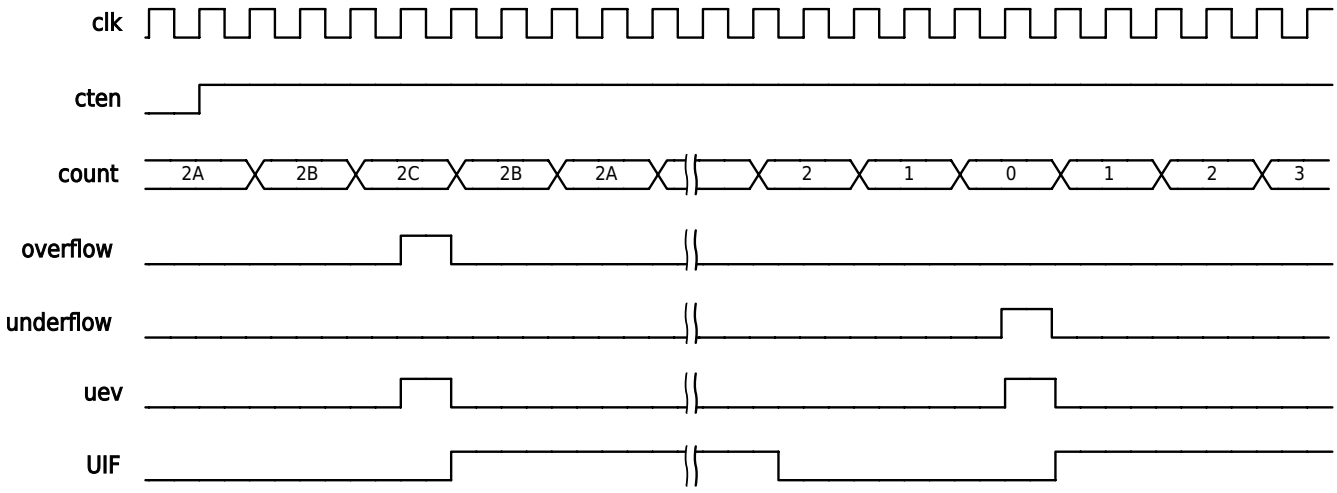


Figure 13-21 Mode3 Compare Capture Mode Up/Down-Counting with Prescaler

The counting waveforms for mode 3 are shown in the following diagrams:

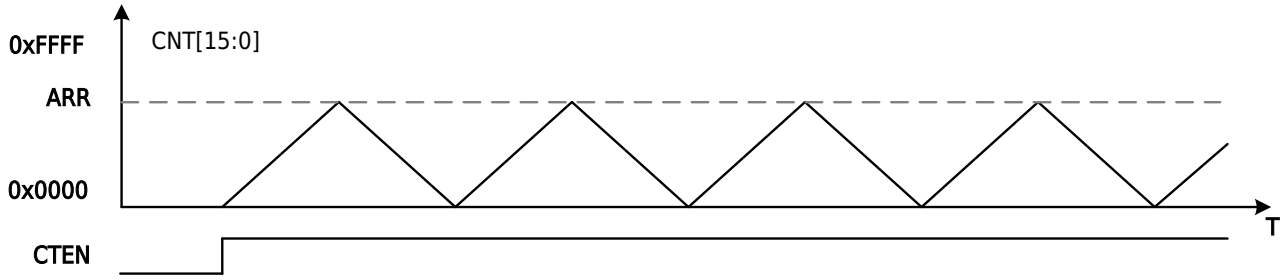


Figure 13-22 Mode3 Center-Aligned Counter Waveform

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.UD=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.

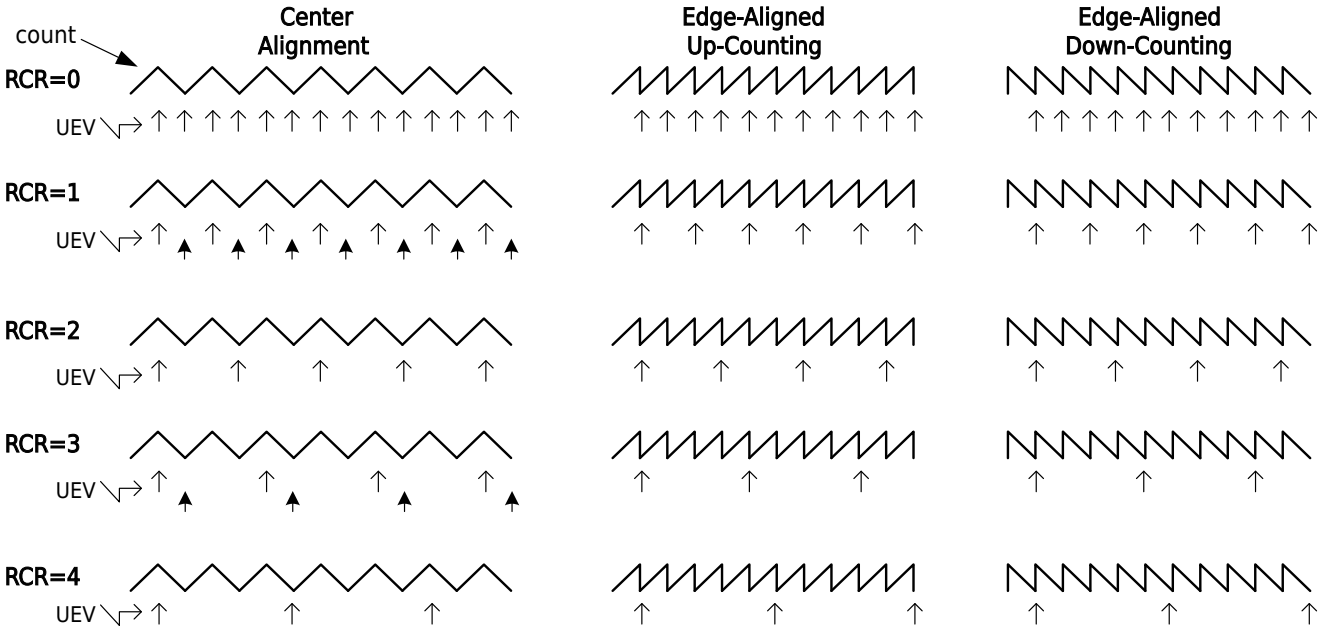


Figure 13-23 Repetition Counter Generates Update Timing (RCR.OV=0, RCR.UD=0)

RCR.OV is the overflow mask control bit for the repeat counter. When RCR.OV=1, overflow events are ignored during repeat counting.

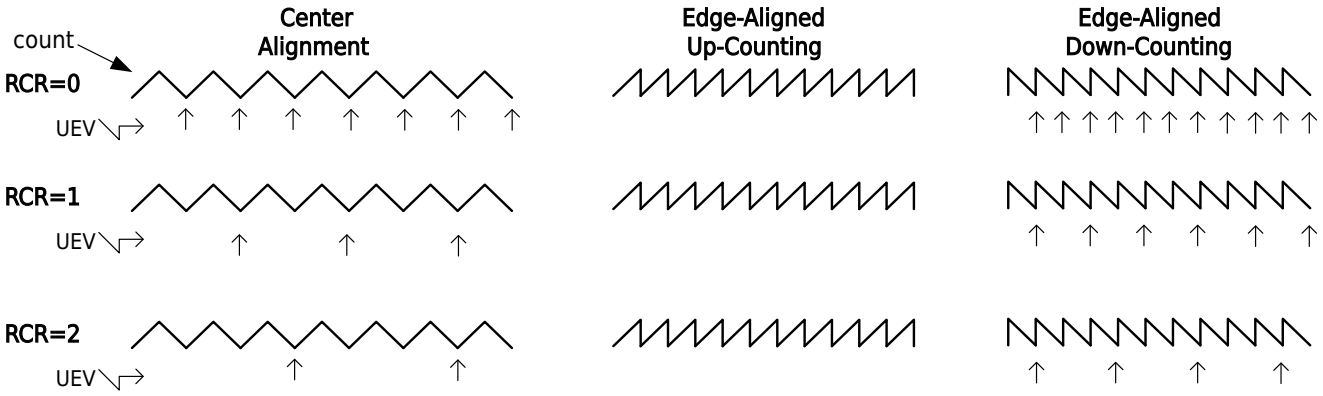


Figure 13-24 Update Timing Generated by Repetition Counter when Overflow Is Masked (RCR.OV=1, RCR.UD=0)

RCR.UD is the underflow mask control bit for the repetition counter. If RCR.UD is 1, underflow events will be ignored during repetition counting.

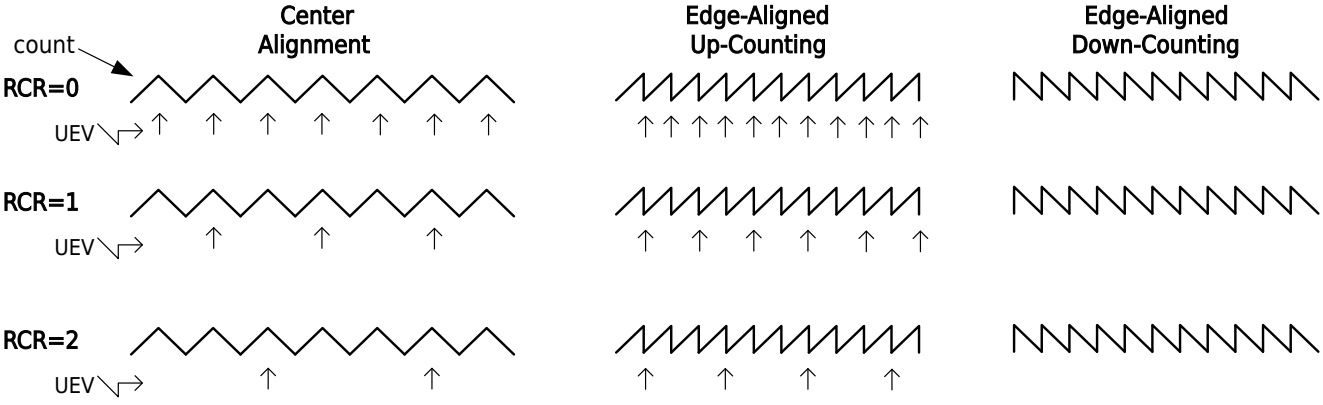


Figure 13-25 Update Timing Generated by Repetition Counter when Underflow Is Masked (RCR.OV=0, RCR.UD=1)

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:

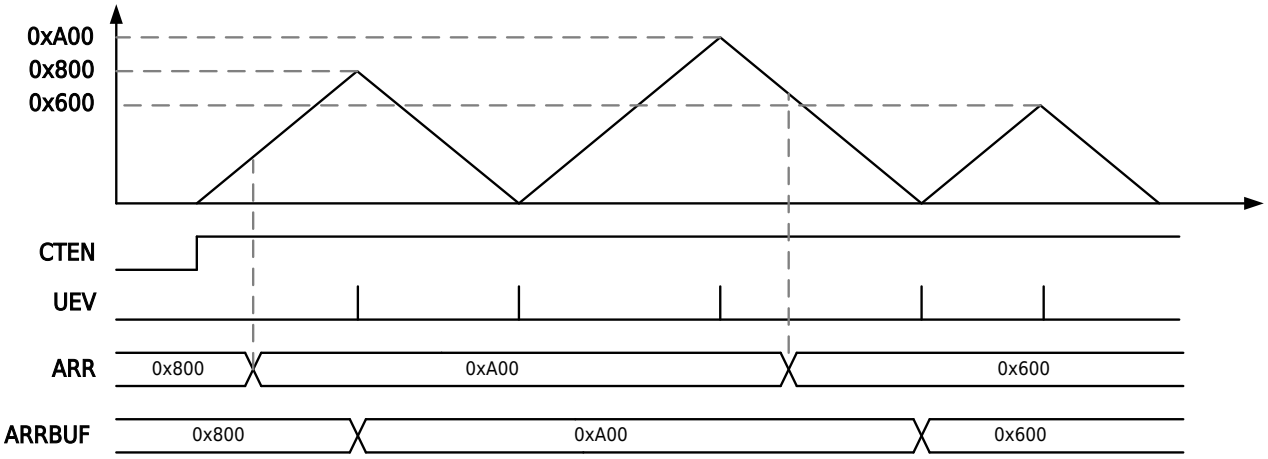


Figure 13-26 Mode 3 Triangular Wave Mode with Cache Enable

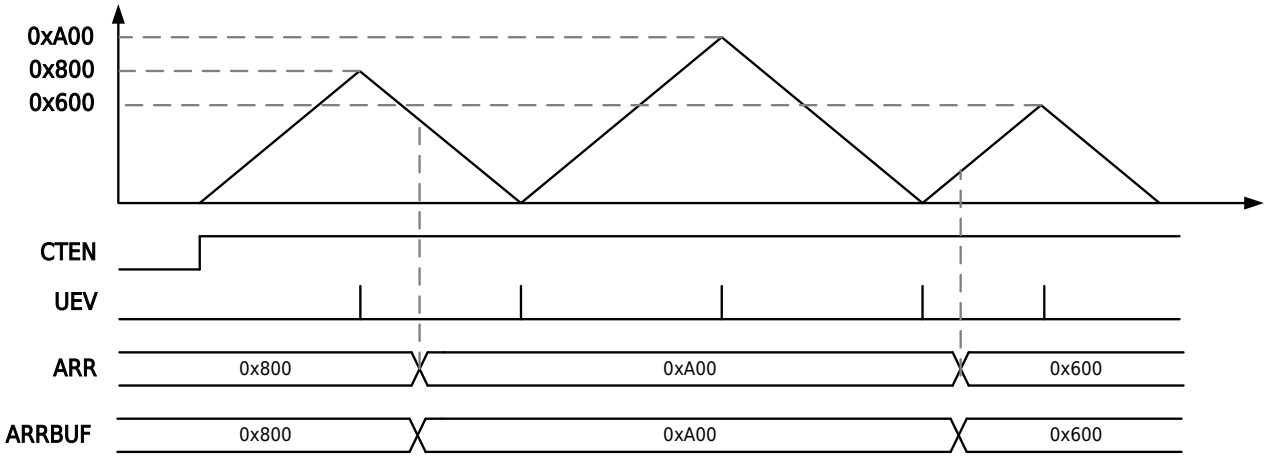


Figure 13-27 Mode 3 Triangular Wave Mode with Cache Invalid

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

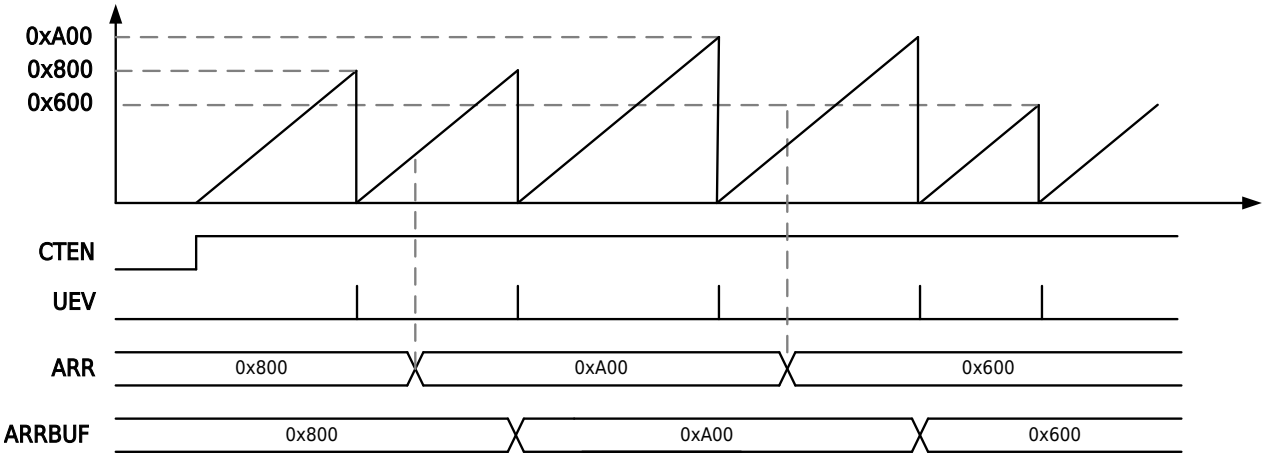


Figure 13-28 Mode 2 Sawtooth Wave Mode Up-Counting with Cache Enable

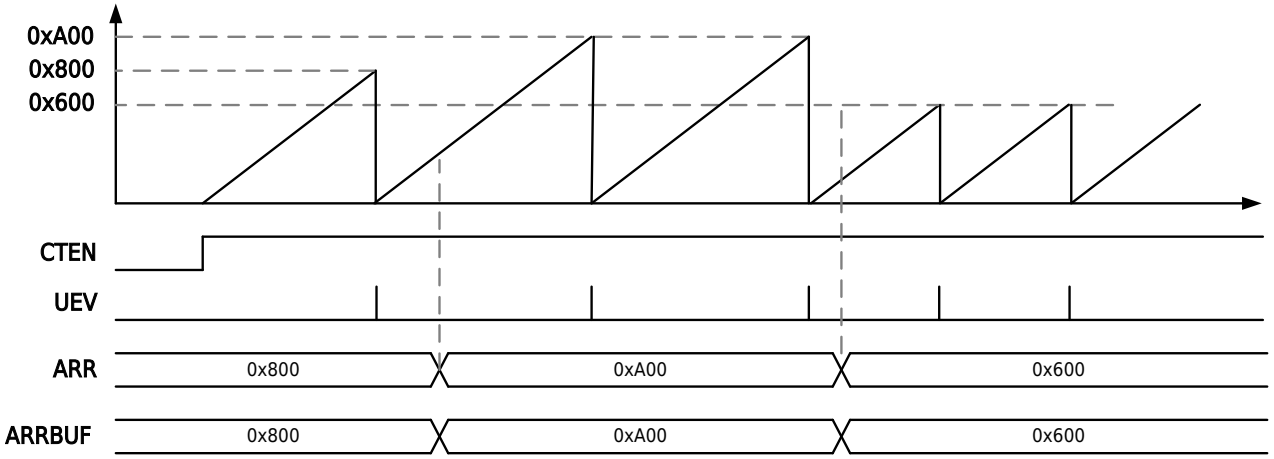


Figure 13-29 Mode 2 Sawtooth Wave Mode Up-Counting with Cache Invalid

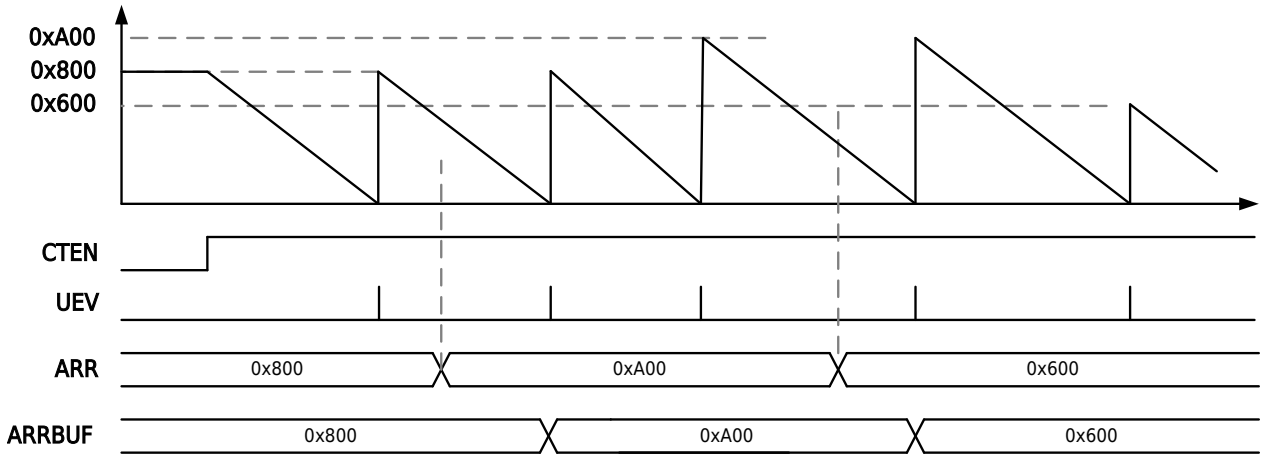


Figure 13-30 Mode 2 Sawtooth Wave Mode Down-Counting with Cache Enable

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

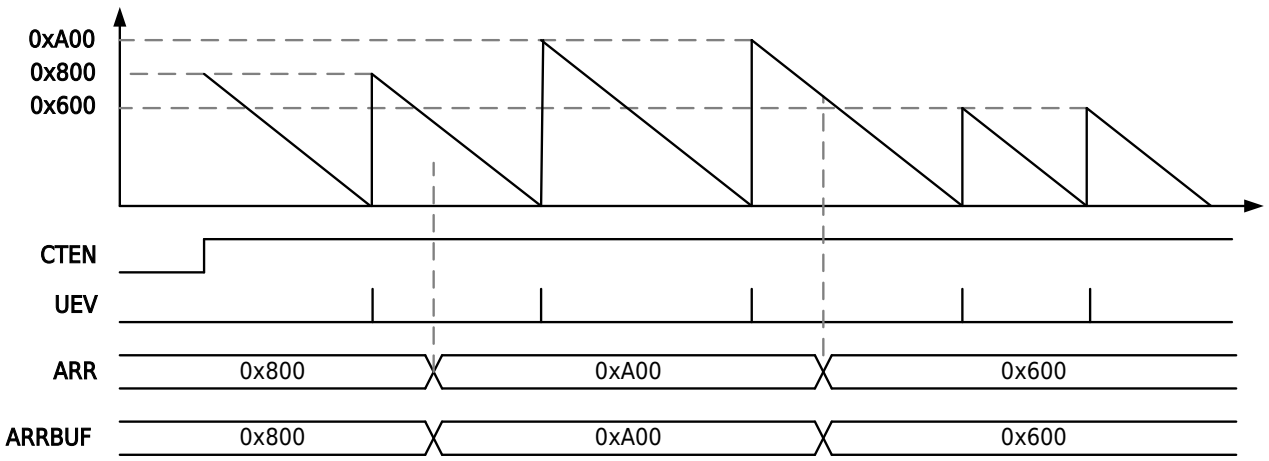


Figure 13-31 Mode 2 Sawtooth Wave Mode Down-Counting with Cache Invalid

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.

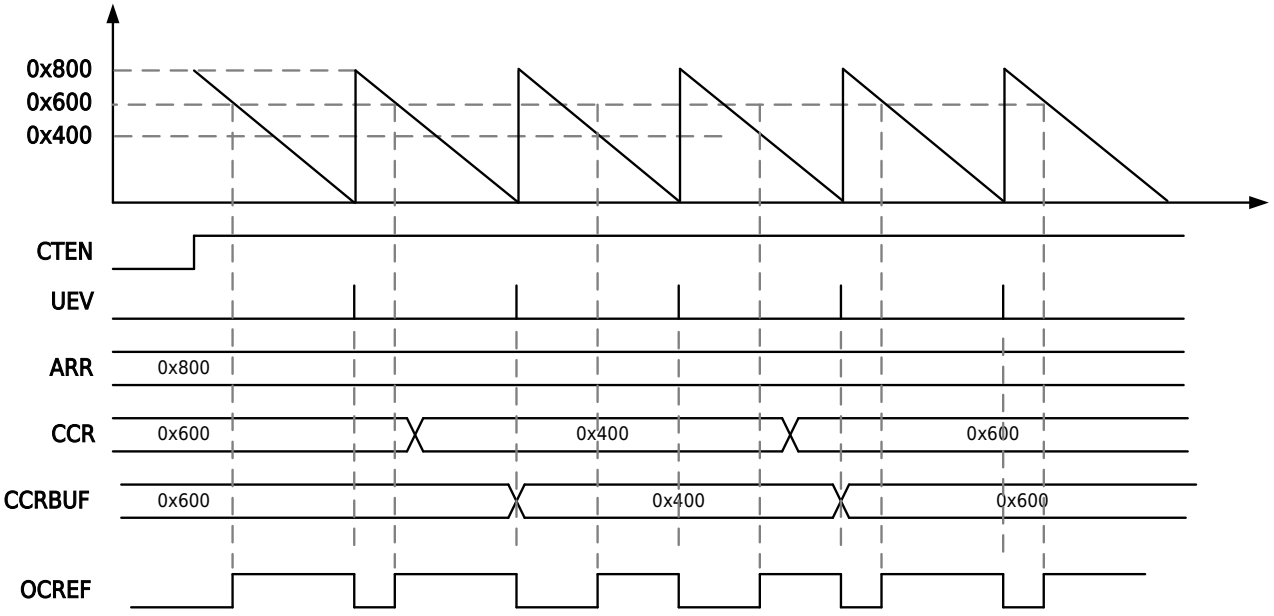


Figure 13-32 Mode2 Sawtooth Wave Down-Counting Mode with Compare Cache Enable

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.

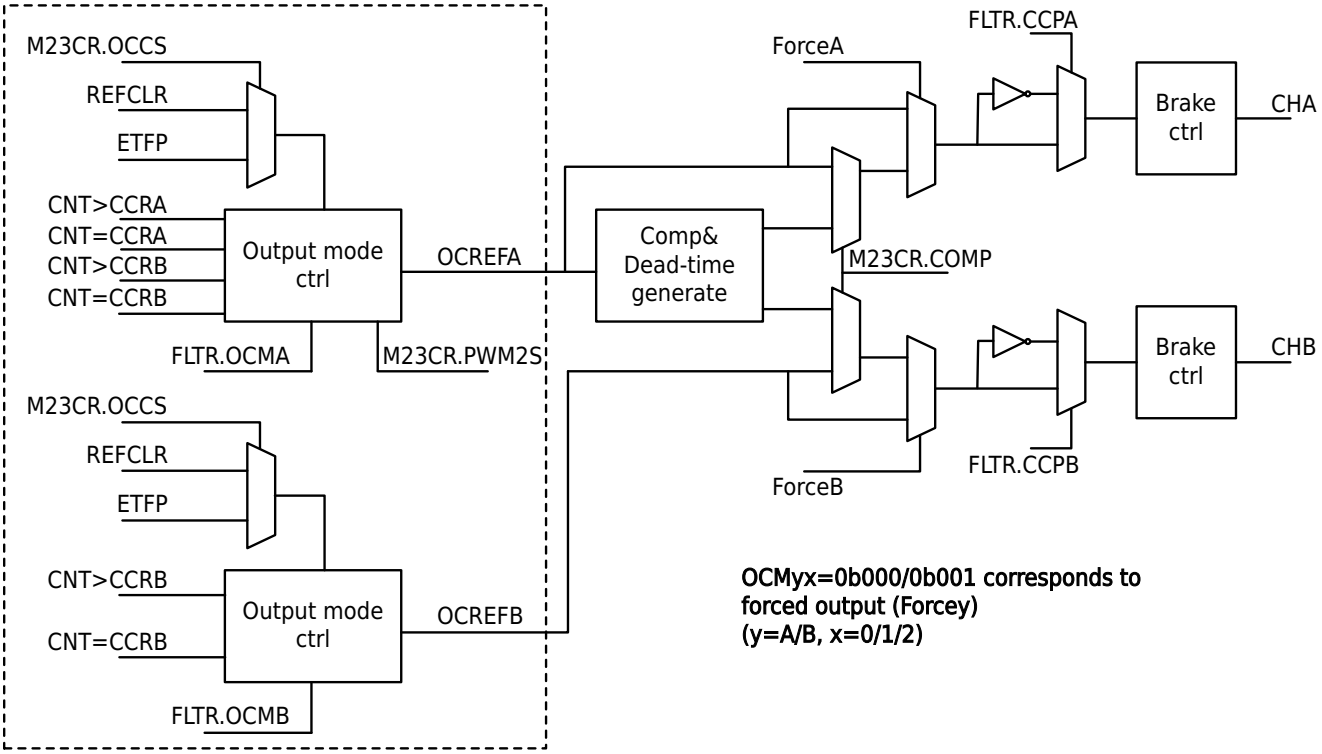


Figure 13-33 OCREF Output Block Diagram

OCREF output uses ATIMx_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.

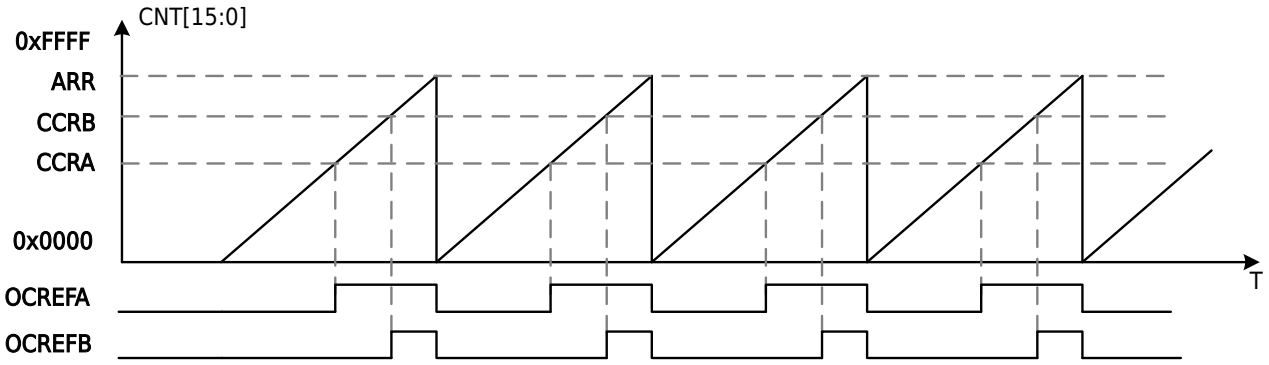


Figure 13-34 Sawtooth Counting Single Point Comparison OCREF Output Waveform (OCMyx=0b111)

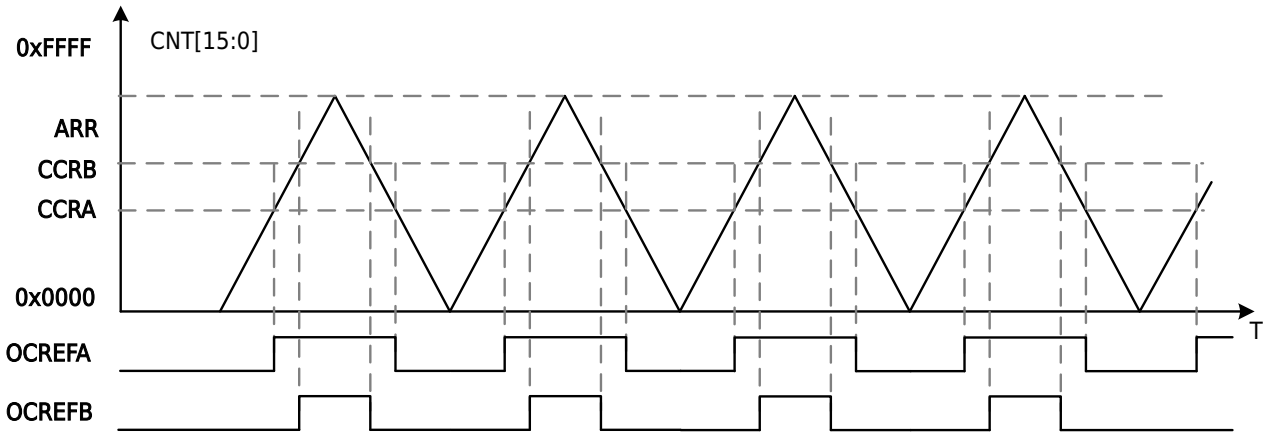


Figure 13-35 Triangular Counting Single Point Comparison OCREF Output Waveform (OCMyx=0b111)

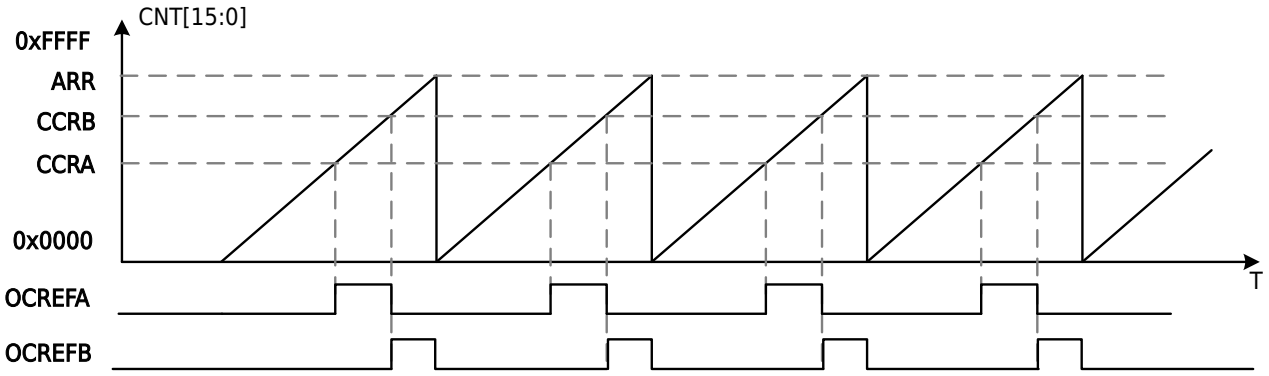


Figure 13-36 Sawtooth Counting Dual Point Comparison OCREF Output Waveform (OCMyx=0b111)

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

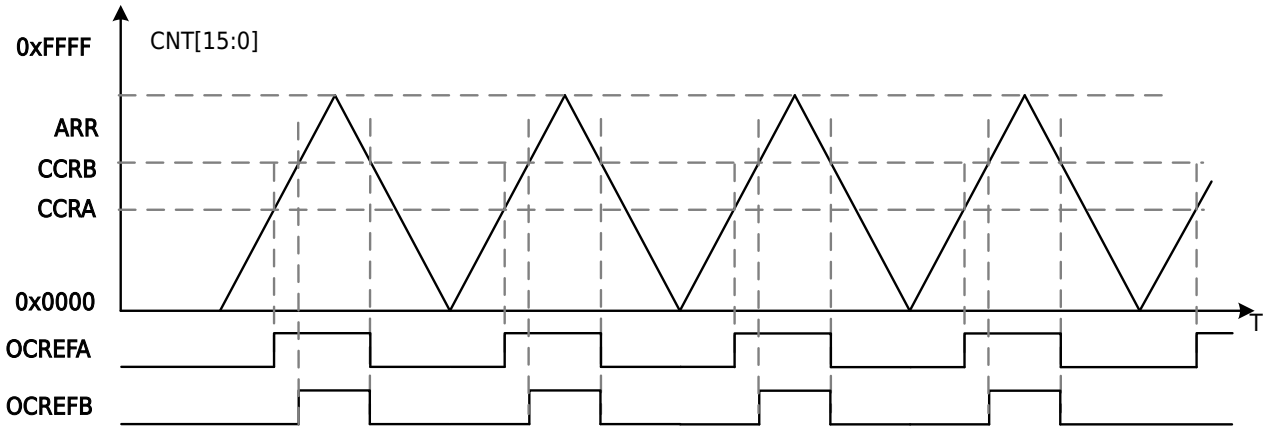


Figure 13-37 Triangular Counting Dual Point Comparison OCREF Output Waveform (OCMyx=0b111)

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.

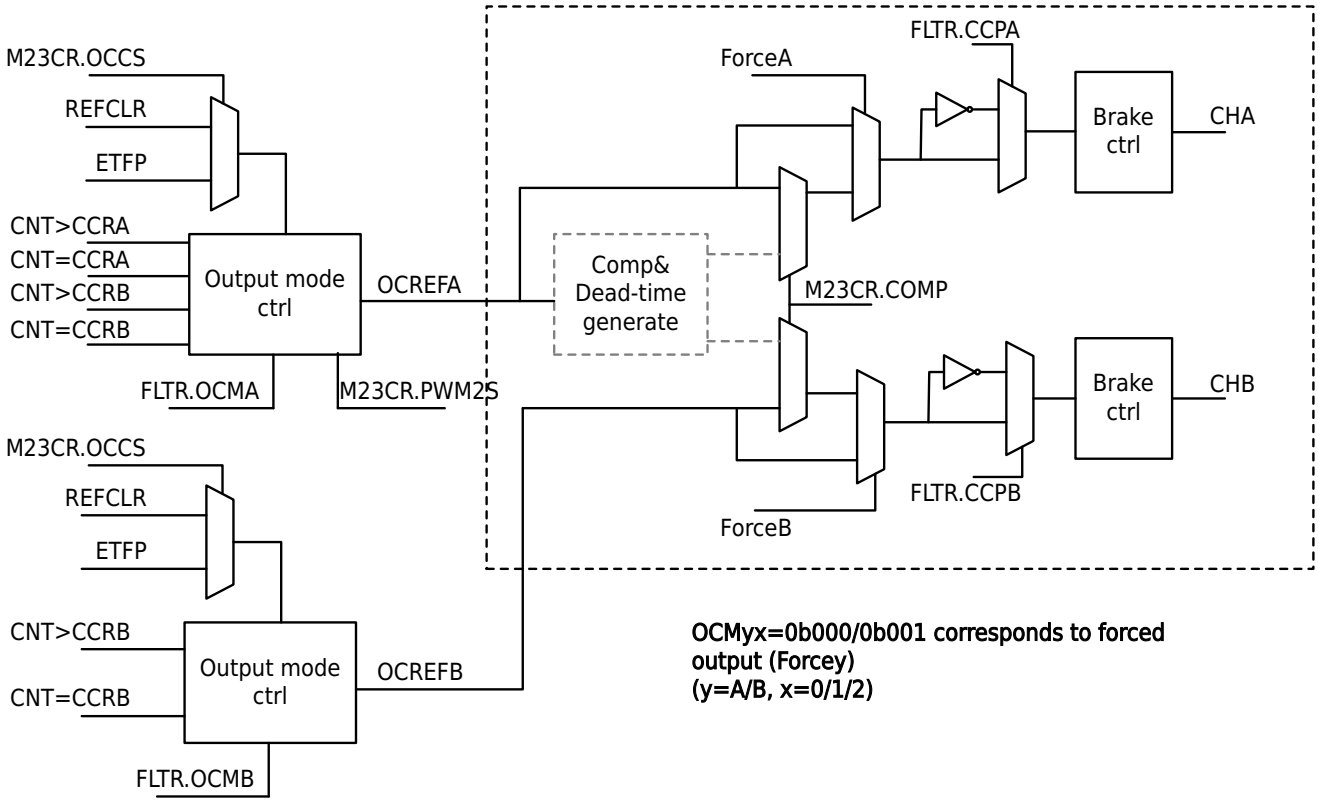


Figure 13-38 Independent PWM Output Block Diagram

PWM output and OCREF relationship, as shown in the following diagram:

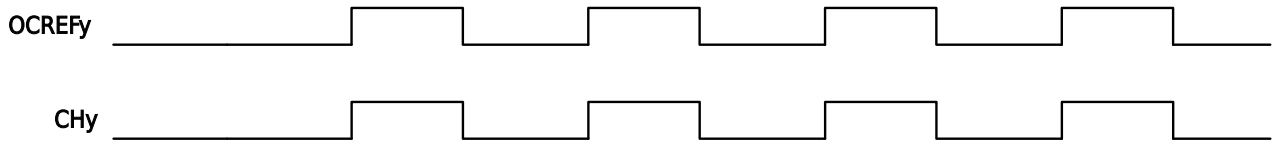


Figure 13-39 PWM Output Waveform When CCPy=0

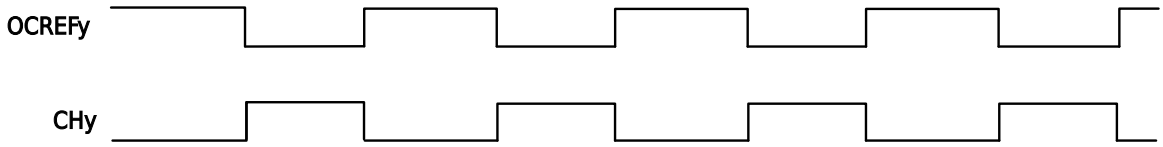


Figure 13-40 PWM Output Waveform When CCPy=1

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.

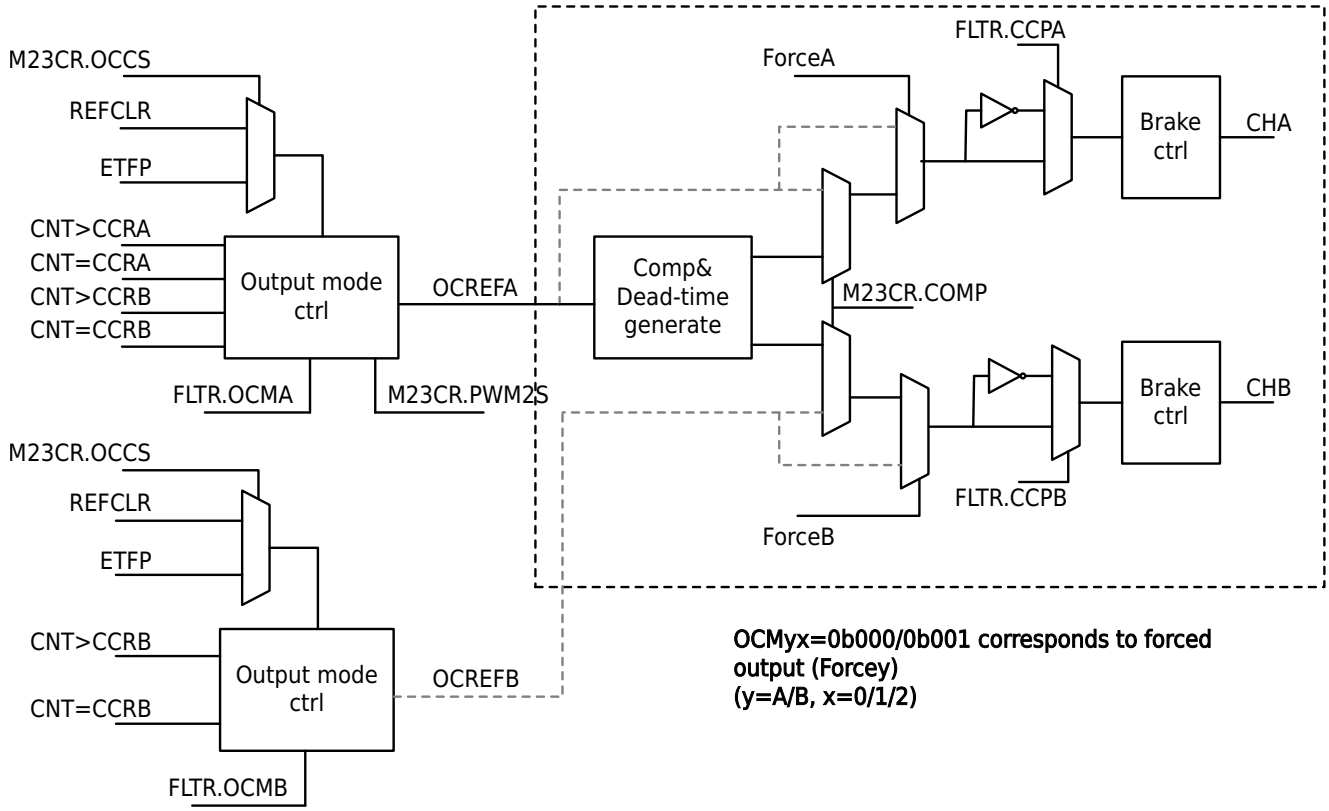


Figure 13-41 Complementary PWM Output Block Diagram

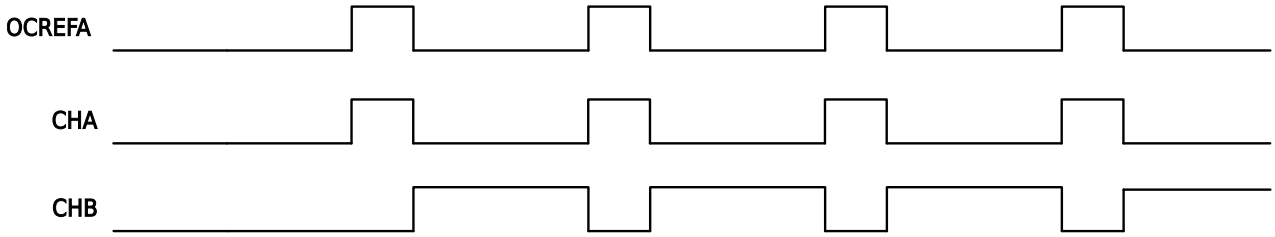


Figure 13-42 Complementary PWM Output Waveform

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.

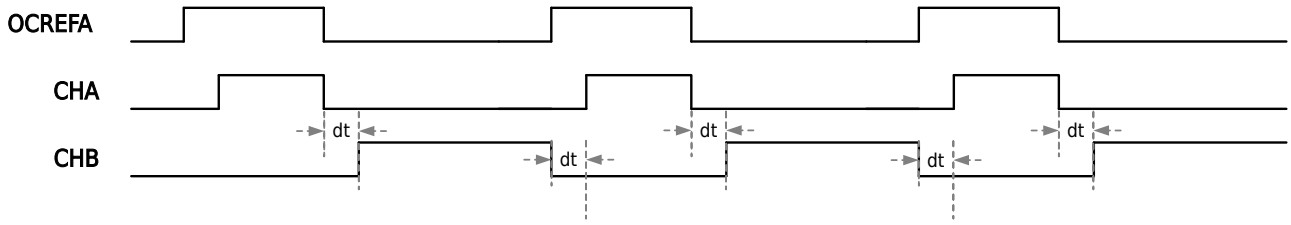


Figure 13-43 Complementary PWM Output Waveform (with Dead Time)

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

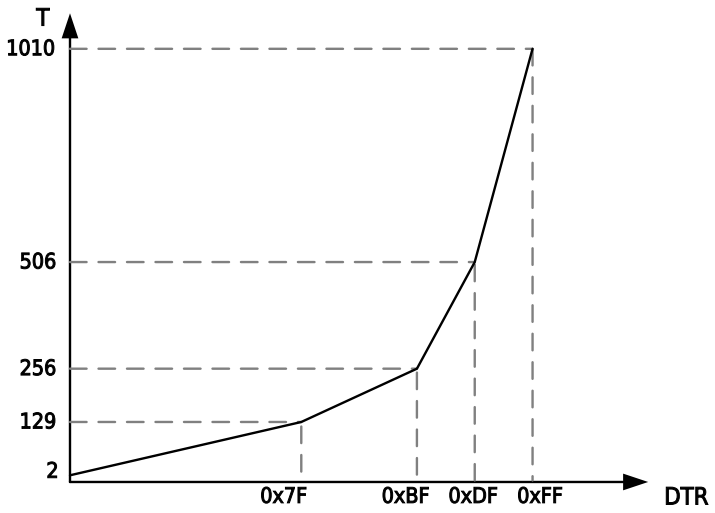


Figure 13-44 Dead Time

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 ATIMx_M23CR.ONESHOT register to 1 enables single pulse mode. Thus, when the next update event (UEV) occurs, the counter will automatically stop.

Because the counter will stop when the next update event UEV occurs, so when the initial value of the RCR setting is not 0, the timer will stop after the RCR value + 1 overflow event. If the overflow or underflow mask function of the RCR register is enabled, the corresponding mask overflow event needs to be ignored when RCR is counted.

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

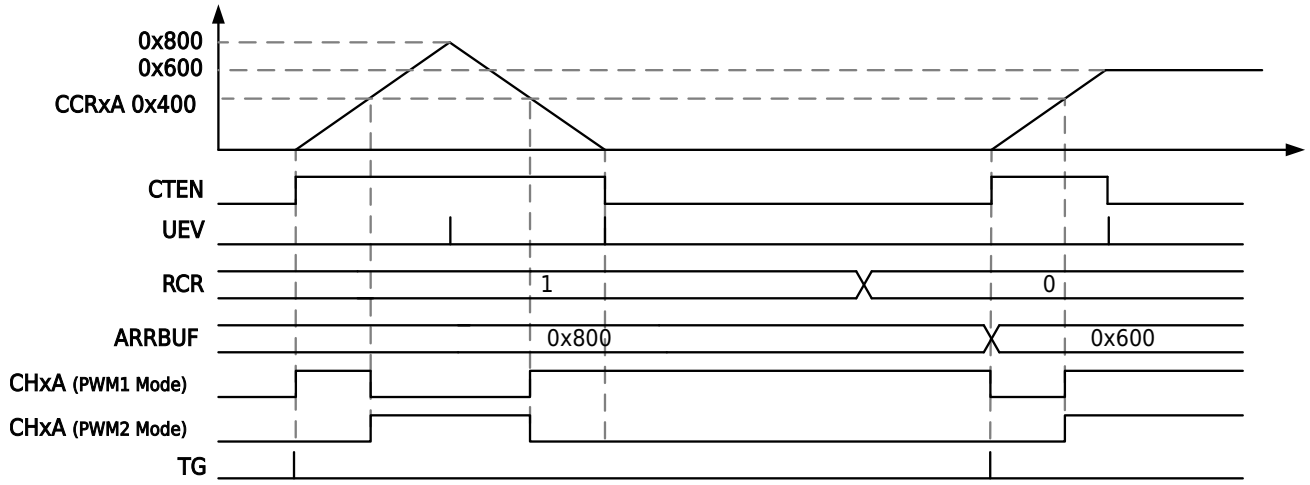


Figure 13-45 Triangular Wave Mode Single Pulse Counting

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

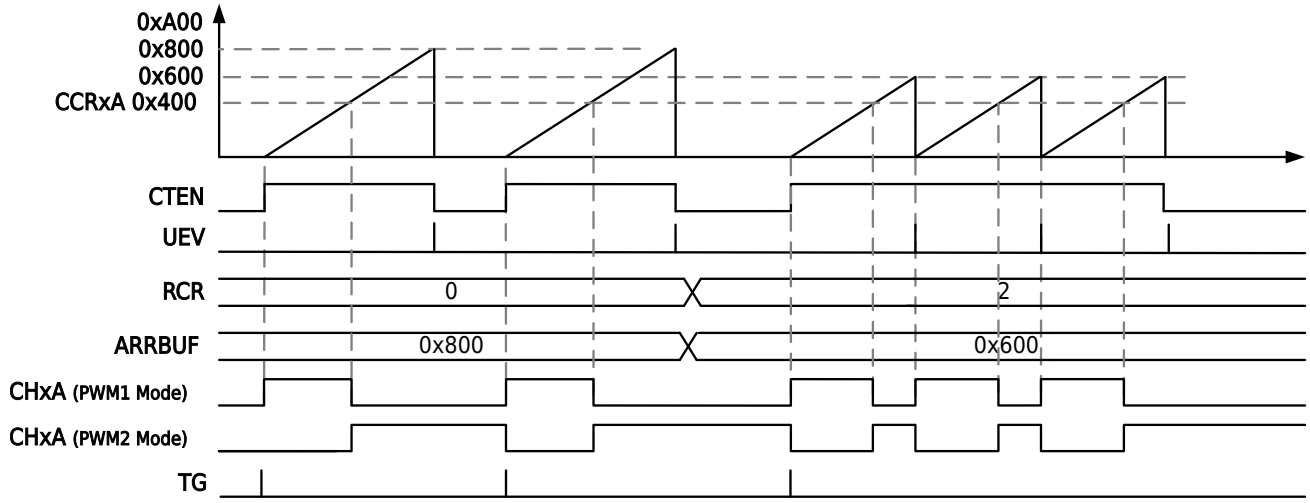


Figure 13-46 Sawtooth Up-Counting Single Pulse Mode

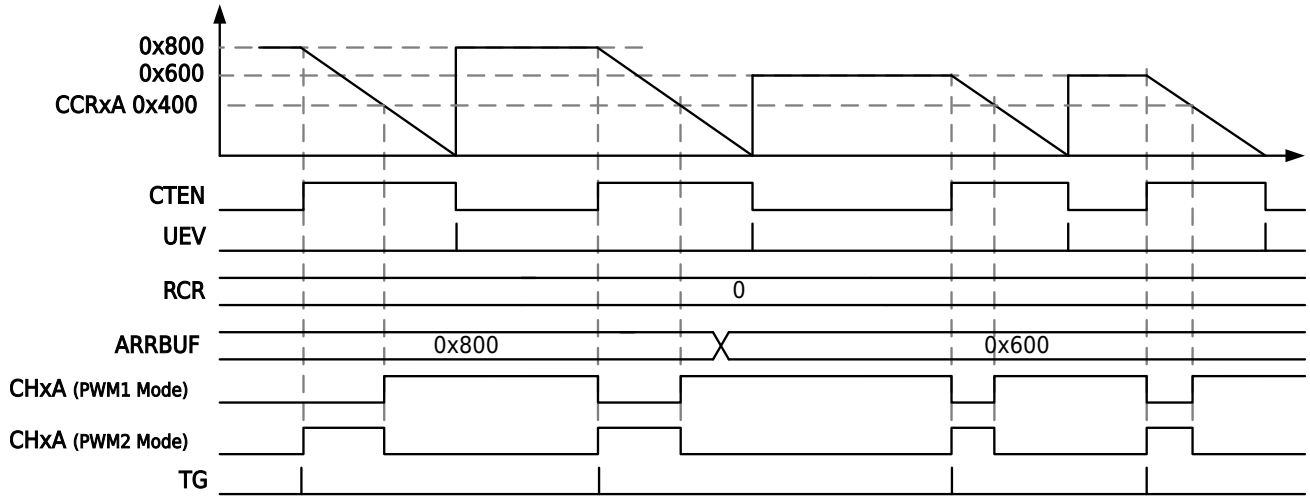


Figure 13-47 Sawtooth Down-Counting Single Pulse Mode

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

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

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.

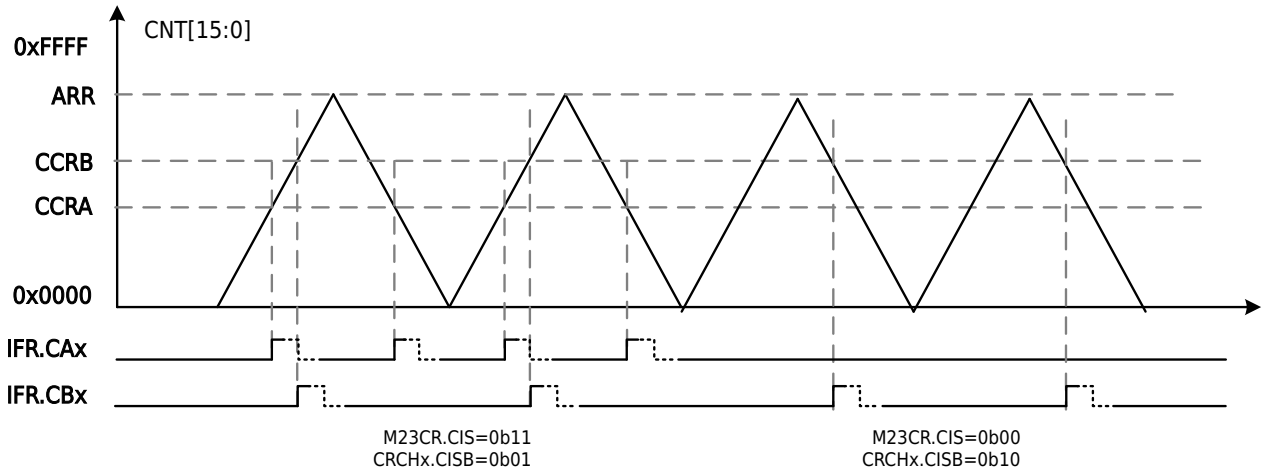


Figure 13-48 Compare Interrupt Diagram

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.

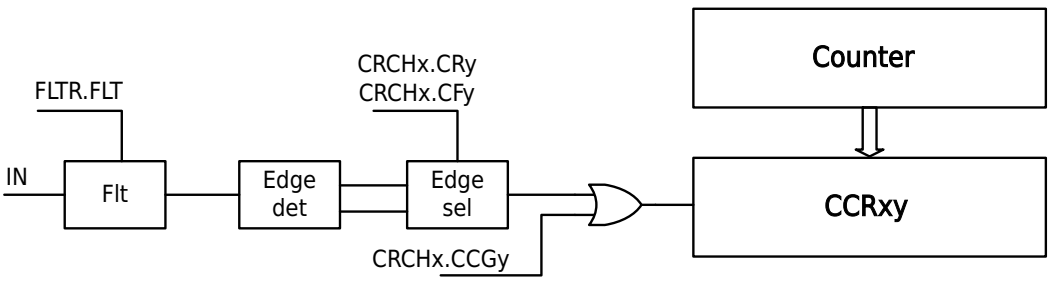


Figure 13-49 Capture Function Block Diagram

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

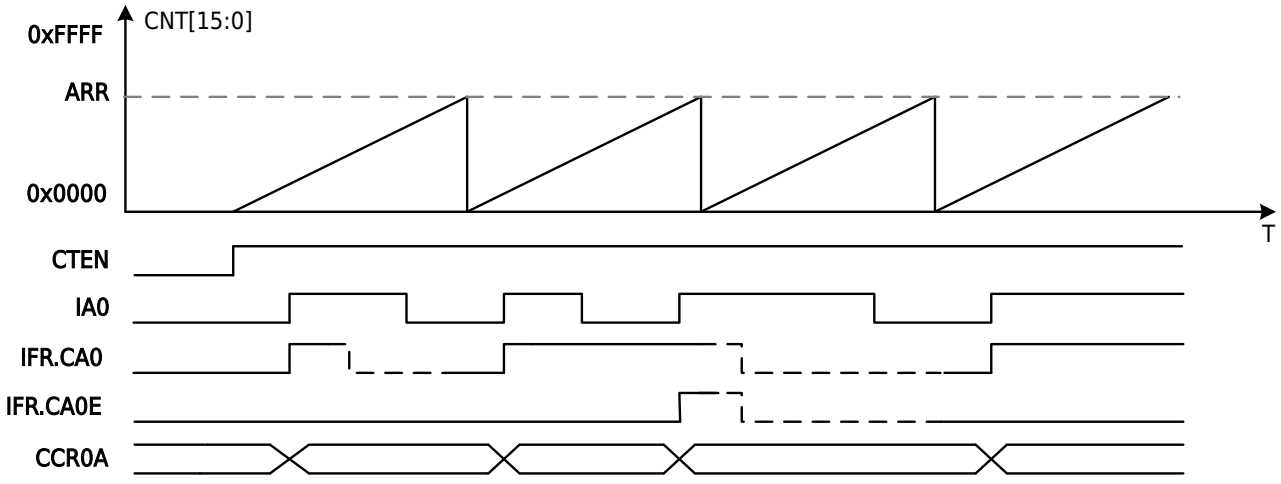


Figure 13-50 Capture Timing Diagram

CH0A Capture Input

The capture input signal for CH0A is as shown in the table below.

Table 13-6 CH0A Capture Input Signal

MSCR.IA0S	ATIM0/1/2	ATIM3
0b0	CH0A	CH0A
0b1	CH0A ETR GATE XOR Input	CH0A CH1A CH2A XOR Input

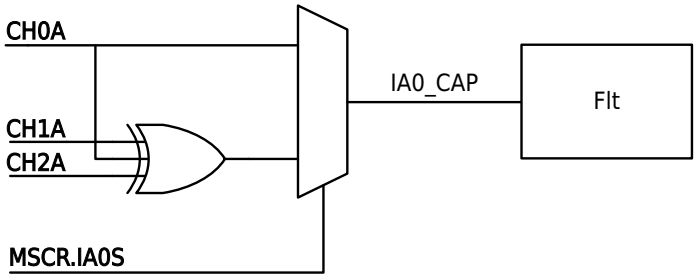


Figure 13-51 ATIM3 CH0A Capture Input Port Selection

CH0B Capture Input

The capture input signal for CH0B is as shown in the table below.

MSCR.IB0S	ATIM0/1/2/3
0b0	CH0B

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

<table>
<tr><th>MSCR.IB0S</th><th>ATIM0/1/2/3</th></tr>
<tr><td>0b1</td><td>Internal trigger MSCR.TS select signal
<table>
<tr><th>MSCR.TS Value</th><th>Channel B Capture Input</th></tr>
<tr><td>0b000</td><td>ETR filtered phase selection output signal, filter phase selection configurable</td></tr>
<tr><td>0b001</td><td>Internal interconnection ITR0</td></tr>
<tr><td>0b010</td><td>Internal interconnection ITR1</td></tr>
<tr><td>0b011</td><td>Internal interconnection ITR2</td></tr>
<tr><td>0b100</td><td>Internal interconnection ITR3</td></tr>
<tr><td>0b101</td><td>CH0A edge</td></tr>
<tr><td>0b110</td><td>CH0A filtered output signal, filter function configurable</td></tr>
<tr><td>0b111</td><td>Reserved</td></tr>
</table></td></tr>
</table>

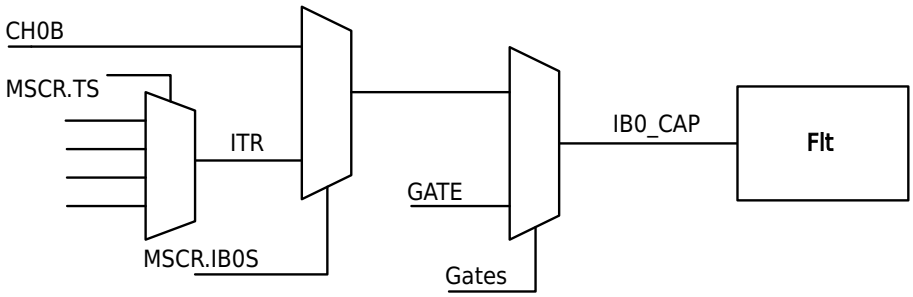


Figure 13-52 CH0B Capture Input Port Selection

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.

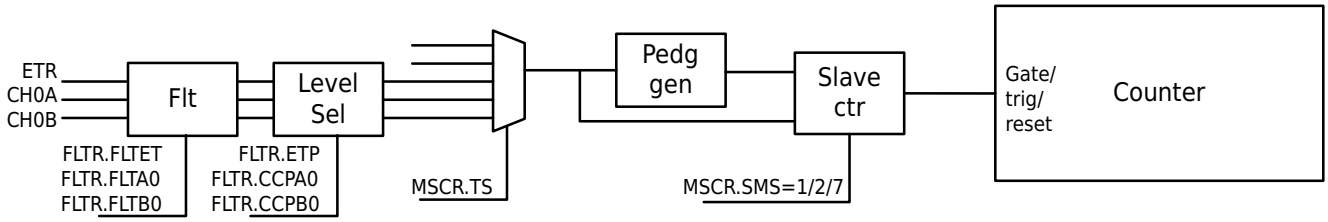


Figure 13-53 Mode2/3 Slave Mode Diagram

The slave mode function is selected via MSCR.SMS:

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

	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).
- 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.

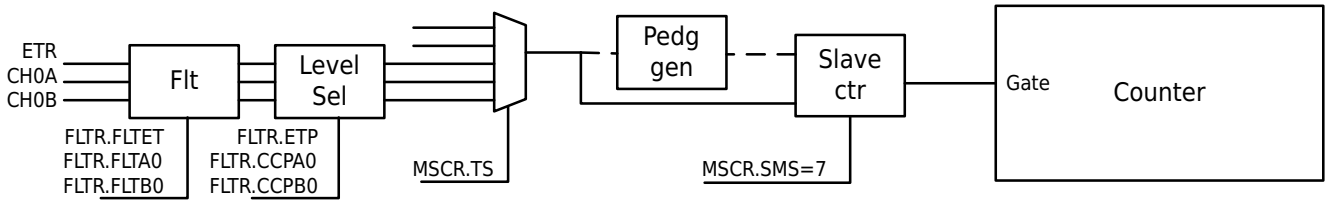


Figure 13-54 Mode2/3 Slave Mode Gating Function

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. 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:

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

- 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.

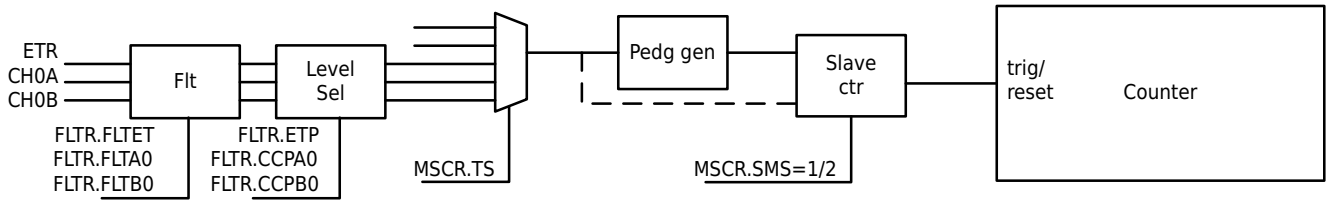


Figure 13-55 Mode2/3 Slave Mode Trigger and Reset Function

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.

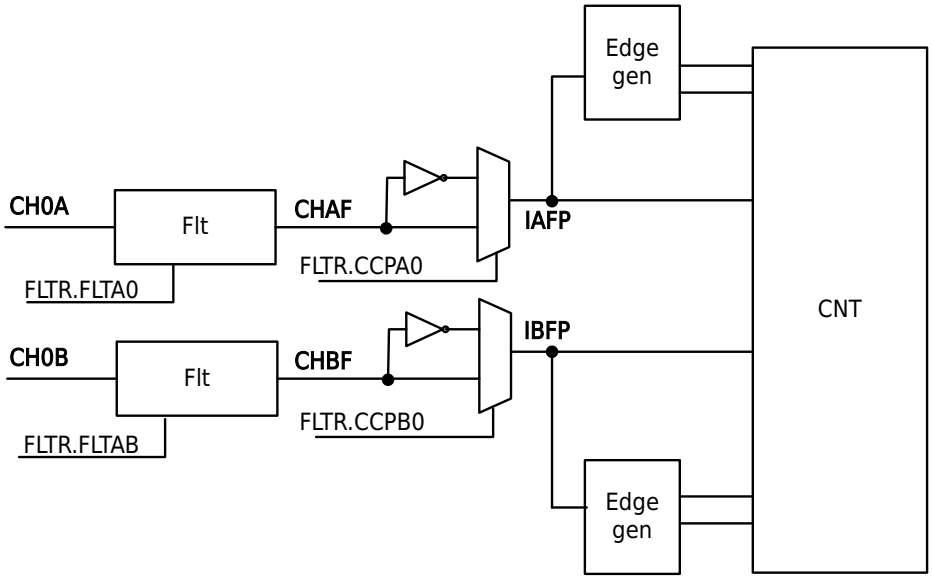
As shown in the table below, the edge condition of IAFP corresponds to the level signal of IBFP, while the edge condition of IBFP corresponds to the level signal of IAFP. For example, in MOD1, when IBFP is at a high level and a rising edge is detected on IAFP, the counter counts down; when IBFP is at 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	-	-

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

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
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.

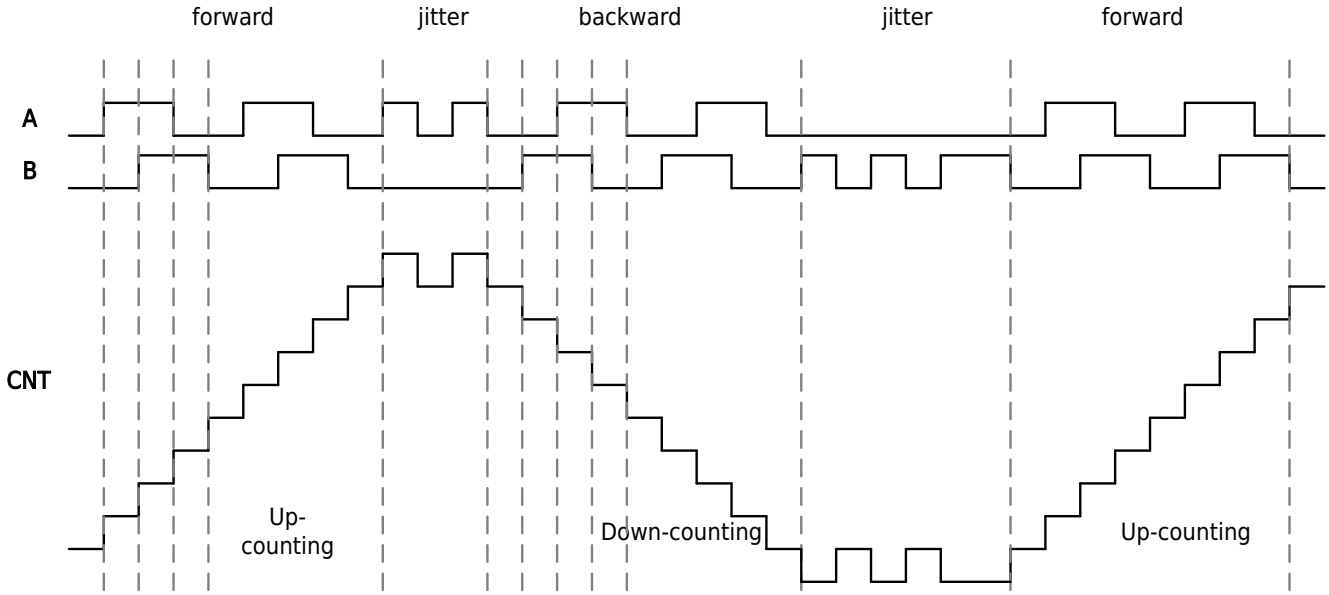


Figure 13-57 Quadrature Encoder Counting Mode3 Counting Diagram

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.
- Mode2: No configuration of CRCHx.CISB is required.

The comparison matches of CCR3A and CCR3B can also be configured to trigger the ADC (If using channel 3A or channel 3B to trigger the ADC, no separate channel trigger enable needs to be set in the ADTR register; only enabling ADTR.ADTE is required).

- 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.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

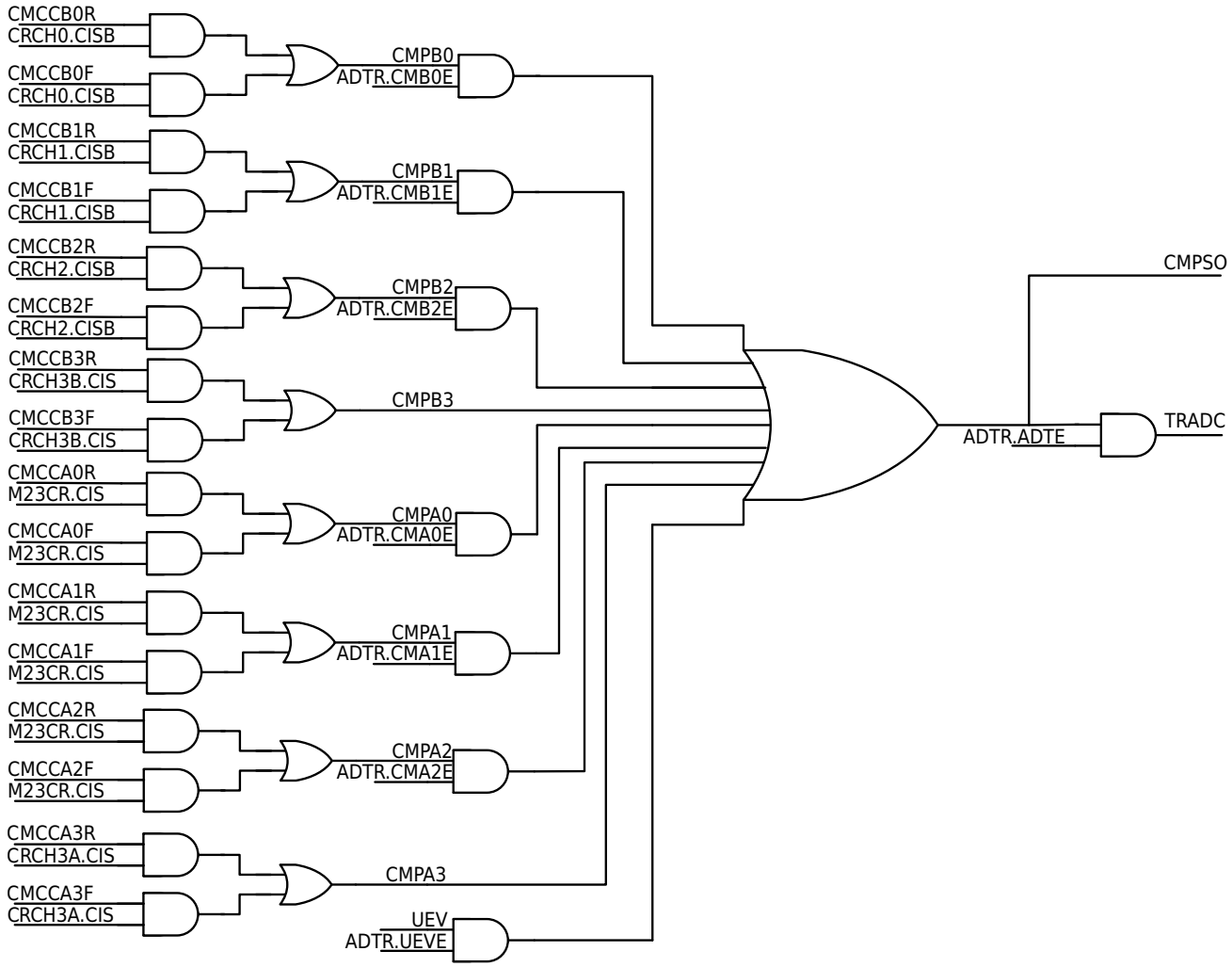


Figure 13-58 Timer Triggers ADC

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.ATIMxCLR (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 (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.ATIMBK, and

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

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 ATIMx_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 (osc fail, brown down, 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 ATIM0/1/2 outputs of the two channels can either use ATIM0 brake port to control the braking function of ATIM1/2 or use their respective brake inputs to control their own braking functions. When DTR.BKSEL is set to 1, ATIM1/2 share the brake input of ATIM0.

13.3.11 Interrupt

Table 13-11 ATIM0/1/2/3 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
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 13-12 Timer Interconnection Signal List

	ITR0	ITR1	ITR2	ITR3
ATIM0	-	ATIM1 TRGO	ATIM2 TRGO	ATIM3 TRGO
ATIM1	ATIM0 TRGO	-	ATIM2 TRGO	ATIM2 TRGO

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

	ITR0	ITR1	ITR2	ITR3
ATIM2	ATIM0 TRGO	ATIM1 TRGO	-	ATIM3 TRGO
ATIM3	ATIM0 TRGO	ATIM1 TRGO	ATIM2 TRGO	-

13.3.13 Input Interconnection

The ETR input of ATIM0 can be directly input from the port, and can also be connected to other modules or ports through the selection of the port function register SYSCTRL->IOAUX1.

When ATIM0ETR=0x0, the capture input is the port input selected by AFRH/AFRL. When ATIM0ETR=0x1-0x7, it can also be connected to the input or output of other modules through the selection of the port function register SYSCTRL->IOAUX1.

Table 13-13 ATIM0 ETR Signal Source List

SYSCTRL->IOAUX1	0	1	2	3	4	5	6	7
ATIM0ETR[2:0]	ATIM0_ETR	USART0_RXD	VC0_OUT	LVD_OUT	VC1_OUT	PA04	PA12	PB04

The CHA input of ATIM0 can be input from the port selected by AFRH/AFRL, or can be connected to other modules or ports through the selection of the port function register SYSCTRL->IOAUX2.

When ATIM0CHA=0x0, the capture input is the port input selected by AFRH/AFRL. When ATIM0CHA=0x1-0x7, it is connected to the input or output of other modules.

Table 13-14 ATIM0 CHA Input Signal Source List

SYSCTRL->IOAUX2	0	1	2	3	4	5	6	7
ATIM0CHA[2:0]	ATIM0_CHA	USART0_RXD	VC0_OUT	LVD_OUT	VC1_OUT	PA00	PA08	PB00

The input interconnection between ETR and CH2B of ATIM3 is as follows:

Table 13-15 ATIM3 ETR and CH2B Input Signal Source List

SYSCTRL->IOAUX1	0	1	2	3	4	5	6	7
ATIM3ETR[14:12]	ATIM3_ETR	LPUART1_RXD	VC0_OUT	LVD_OUT	VC1_OUT	PA07	PA15	PB07
SYSCTRL->IOAUX2	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
ATIM3CH2B[14:12]	ATIM3_CH2B	LPUART1_RXD	VC0_OUT	LVD_OUT	VC1_OUT	PA03	PA11	PB03

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.
- **BUZZER Output Control**
  - **Step1.** Set appropriate ARR value based on output frequency;

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

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;

Step10. Enable output DTR.MOE;

Step11. Enable timer M23CR.CTEN.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

- **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.



Note

When the timer is not started, if there is a valid capture edge, the capture flag and capture action will also be generated.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

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

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

13.5 Register Description

13.5.1 Register List

- ATIM0 base address: 0x40003000
- ATIM1 base address: 0x40003400
- ATIM2 base address: 0x40003800
- ATIM3 base address: 0x40005800

Table 13-16 Mode0 Timer Register Offset

Offset	Register	Description	Width
0x000	ATIM_ARR	16-Bit Mode Reload Register/Period	32
0x004	ATIM_CNT	16-Bit Mode Count Register	32
0x008	ATIM_CNT32	32-Bit Mode Count Register **Note** ATIM_CNT32 is the remapping register for ATIM_ARR and ATIM_CNT. In order to facilitate the 32-bit timer write operation.	32
0x00C	ATIM_M0CR	Mode0 Control Register	32
0x010	ATIM_IFR	Interrupt Flag Register	32
0x014	ATIM_ICLR	Interrupt Flag Clear Register	32
0x030	ATIM_DTR	Dead Time Register	32

Table 13-17 Mode1 Pulse Width Measurement (PWM) Register Offset

Offset	Register	Description	Width
0x004	ATIM_CNT	16-Bit Mode Count Register	32
0x00C	ATIM_M1CR	Mode1 Control Register	32
0x010	ATIM_IFR	Interrupt Flag Register	32
0x014	ATIM_ICLR	Interrupt Flag Clear Register	32
0x018	ATIM_MSCR	Master-Slave Mode Control Register	32
0x01C	ATIM_FLTR	Input Filter Register	32
0x024	ATIM_CR0	Control Register	32
0x03C	ATIM_CCR0A	Compare Capture Register	32

Table 13-18 Mode2/3 Compare-Capture Mode Register Offset

Offset	Register	Description	Width
0x000	ATIM_ARR	16-Bit Mode Reload Register/Period	32

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Offset	Register	Description	Width
0x004	ATIM_CNT	16-Bit Mode Count Register	32
0x00C	ATIM_M23CR	Mode2/3 Control Register	32
0x010	ATIM_IFR	Interrupt Flag Register	32
0x014	ATIM_ICLR	Interrupt Flag Clear Register	32
0x018	ATIM_MSCR	Master-Slave Mode Control Register	32
0x01C	ATIM_FLTR	Output Control/Input Filter Register	32
0x020	ATIM_ADTR	ADC Trigger Control Register	32
0x024	ATIM_CRCH0	Channel 0 Control Register	32
0x028	ATIM_CRCH1	Channel 1 Control Register (only ATIM3 exists)	32
0x02C	ATIM_CRCH2	Channel 2 Control Register (only ATIM3 exists)	32
0x030	ATIM_DTR	Dead Time Register	32
0x034	ATIM_RCR	Repeat Cycle Setting Value Register	32
0x038	ATIM_ARRDM	Reload Register/Period Mapped Address	32
0x03C	ATIM_CCR0A	Channel 0A Compare/Capture Register	32
0x040	ATIM_CCR0B	Channel 0B Compare/Capture Register	32
0x044	ATIM_CCR1A	Channel 1A Compare/Capture Register (only ATIM3 exists)	32
0x048	ATIM_CCR1B	Channel 1B Compare/Capture Register (only ATIM3 exists)	32
0x04C	ATIM_CCR2A	Channel 2A Compare/Capture Register (only ATIM3 exists)	32
0x050	ATIM_CCR2B	Channel 2B Compare/Capture Register (only ATIM3 exists)	32
0x054	ATIM_CCR3A	Channel 3A Compare Register (only ATIM3 exists)	32
0x058	ATIM_CRCH3A	Channel 3A Control Register (only ATIM3 exists)	32
0x05C	ATIM_CCR3B	Channel 3B Compare Register (only ATIM3 exists)	32
0x060	ATIM_CRCH3B	Channel 3B Control Register (only ATIM3 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 (ATIM_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 (ATIM_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 (ATIM_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 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 (ATIM_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 ● 0b00: Timer mode 0 ● 0b01: PWC mode ● 0b10: Sawtooth wave mode ● 0b11: Triangle wave mode	RW
11	*Reserved*	-	-
10	UIE	Interrupt enable control ● 0b0: Disable ● 0b1: Enable	RW
9	GATEP	Port GATE polarity control ● 0b0: Port GATE high level is valid ● 0b1: Port GATE low level is valid	RW
8	GATE	Timer gating enable ● 0b0: No gating, timer works when CTEN=1 ● 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 ● 0b000: 1 ● 0b001: 2 ● 0b010: 4 ● 0b011: 8 ● 0b100: 16 ● 0b101: 32 ● 0b110: 64 ● 0b111: 256	RW
3	TOGEN	Toggle output enable in mode 0 ● 0b0: Toggle output disabled, CHA and CHB outputs remain low level ● 0b1: Toggle output enabled	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Bit	Name	Functional Description	Access
2	CT	Count clock selection ● 0b0: Internal count clock TCLK ● 0b1: External count clock ETR	RW
1	MD	Mode selection 32 timing/16 timing mode selection ● 0b0: 32-bit free counting ● 0b1: 16-bit reload counting	RW
0	CTEN	Timer enable ● 0b0: Disable ● 0b1: Enable	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

13.5.2.5 Interrupt Flag Register (ATIM_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																														UI	
Access																															RO	

Bit	Name	Functional Description	Access
31:1	*Reserved*	-	-
0	UI	Overflow interrupt ● 0b0: Counter is not overflowed ● 0b1: Counter overflowed	RO

13.5.2.6 Interrupt Flag Clear Register (ATIM_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	0x0000FFFF																															
Name	Reserved																														UI	
Access																															WO	

Bit	Name	Functional Description	Access
31:1	*Reserved*	-	-
0	UI	Overflow interrupt clear 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 (ATIM_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 ● 0b0: Toggle output in input status ● 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 (ATIM_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 (ATIM_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

Bit	Name	Functional Description	Access
31:15	*Reserved*	-	-
14	ONESHOT	Single trigger mode selection ● 0b0: Cycle measurement ● 0b1: Automatically terminates after completing one pulse measurement, re-enable CTEN is required for subsequent measurements	RW
13:12	MODE	Working mode ● 0b00: Timer mode 0 ● 0b01: PWC mode ● 0b10: Sawtooth wave mode ● 0b11: Triangle wave mode	RW
11	*Reserved*	-	-
10	UIE	Interrupt enable control Counting to 0xFFFF will overflow and generate an overflow flag ● 0b0: Disable ● 0b1: Enable	RW
9	EDG2ND	Pulse width measurement end edge selection Refer to the EDG1ST function description for details	RW
8	EDG1ST	Pulse width measurement start edge selection **EDG2ND / EDG1ST / Measurement** 0b0 / 0b0 / Up edge-up edge period 0b0 / 0b1 / Low level width 0b1 / 0b0 / High level width 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 ● 0b000: 1 ● 0b001: 2 ● 0b010: 4 ● 0b011: 8 ● 0b100: 16 ● 0b101: 32 ● 0b110: 64 ● 0b111: 256	RW
3	*Reserved*	-	-
2	CT	Count clock selection ● 0b0: Internal count clock TCLK ● 0b1: External count clock ETR	RW
1	*Reserved*	-	-
0	CTEN	Pulse width measurement enable ● 0b0: Disable ● 0b1: Enable	RW

13.5.3.3 Interrupt Flag Register (ATIM_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																													PWC	Reserved	UI
Access																														RO		RO

Bit	Name	Functional Description	Access
31:3	*Reserved*	-	-
2	PWC	Pulse width measurement interrupt flag ● 0b0: Pulse width measurement is not completed ● 0b1: Pulse width measurement completed	RO
1	*Reserved*	-	-
0	UI	Overflow interrupt flag ● 0b0: Counter is not overflowed ● 0b1: Counter overflowed	RO

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

13.5.3.4 Interrupt Flag Clear Register (ATIM_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	0x0000FFFF																															
Name	Reserved																													PWC	Reserved	UI
Access																														WO		WO

Bit	Name	Functional Description	Access
31:3	*Reserved*	-	-
2	PWC	Pulse width measurement interrupt flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
1	*Reserved*	-	-
0	UI	Overflow interrupt clear 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 (ATIM_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 ● 0b0: CH0B ● 0b1: Internal trigger TS selection signal	RW
11	IA0S	IA0 input selection ● 0b0: CH0A ● 0b1: ▶ CH0A CH1A CH2A XOR (ATIM3) ▶ CH0A ETR GATE XOR (ATIM0/1/2) **Note** Setting to 1 will cause input changes upon any port transition.	RW
10:8	*Reserved*	-	-
7:5	TS	Trigger selection ● 0b000: Signal ETFP after port ETR filtered phase selection ● 0b001: Internal interconnect signal ITR0 ● 0b010: Internal interconnect signal ITR1 ● 0b011: Internal interconnect signal ITR2 ● 0b100: Internal interconnect signal ITR3 ● 0b101: Setting prohibited ● 0b110: Signal IAFP after port CH0A filtered phase selection ● 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 (ATIM_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 ● 0b0: Same phase ● 0b1: Reverse input	RW
30:28	FLTET	ETR filter control ● 0b000-0b011: Filter is invalid ● 0b100: PCLK, three consecutive valid samples ● 0b101: PCLK/4, three consecutive valid samples ● 0b110: PCLK/16, three consecutive valid samples ● 0b111: PCLK/64, three consecutive valid samples	RW
27:7	*Reserved*	-	-
6:4	FLTB0	CHB input filter control ● 0b000-0b011: Filter is invalid ● 0b100: PCLK, three consecutive valid samples ● 0b101: PCLK/4, three consecutive valid samples ● 0b110: PCLK/16, three consecutive valid samples ● 0b111: PCLK/64, three consecutive valid samples **Note** CR0.CSB needs to be configured as 1.	RW
3	*Reserved*	-	-
2:0	FLTA0	CHA input filter control ● 0b000-0b011: Filter is invalid ● 0b100: PCLK, three consecutive valid samples ● 0b101: PCLK/4, three consecutive valid samples ● 0b110: PCLK/16, three consecutive valid samples ● 0b111: PCLK/64, three consecutive valid samples **Note** CR0.CSA needs to be configured as 1.	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

13.5.3.7 Control Register (ATIM_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 ● 0b0: Disable ● 0b1: Enable	RW
7:6	*Reserved*	-	-
5	CSB	B channel capture/compare function selection When pulse width measurement uses channel B filtering, CSB must be set to 1 ● 0b0: Compare mode ● 0b1: Capture mode	RW
4	CSA	A channel capture/compare function selection When pulse width measurement uses channel A filtering, CSA must be set to 1 ● 0b0: Compare mode ● 0b1: Capture mode	RW
3:0	*Reserved*	-	-

13.5.3.8 Compare Capture Register (ATIM_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 (ATIM_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 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 (ATIM_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 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 (ATIM_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 ● 0b0: Disable ● 0b1: Enable	RW
28	OVE	Overflow interrupt enable ● 0b0: Disable ● 0b1: Enable	RW
27	DIR	Count direction Write only in sawtooth wave mode. Read only in other modes, write invalid ● 0b0: Up-counting ● 0b1: Down-counting **Note** 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 Auto-clear. Write 1 to generate a software brake, write 0 has no effect.	WO
25	UG	Software update Auto-clear. Write 1 to generate a software update, write 0 has no effect. 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 Auto-clear. Write 1 to generate a software trigger, write 0 has no effect. 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 ● 0b0: OCREF output unaffected by OCREF_CLR ● 0b1: OCREF_CLR signal can clear OCREF output	RW
22:21	CIS	Channel A compare match mode ● 0b00: No match ● 0b01: Compare match during up-counting ● 0b10: Compare match during down-counting ● 0b11: Compare match during both up/down counting **Note** Channel B compare match mode is controlled by the CISB bit in the CRCHx register.	RW
20	BIE	Brake interrupt enable ● 0b0: Disable ● 0b1: Enable	RW
19	TIE	Trigger interrupt enable ● 0b0: Disable ● 0b1: Enable	RW
18	TDE	Trigger start DMA enbale ● 0b0: Disable ● 0b1: Enable	RW
17	URS	Update source ● 0b0: Overflow/underflow/software update UG/slave mode reset ● 0b1: Overflow/underflow	RW
16	OCCS	OCREF_CLR signal source selection ● 0b0: Voltage comparator VC output, VC selection is configured in the VCx_CR2 register. ● 0b1: Signal after ETR port filtered phase selection 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 Valid only during PWM complementary output, using CCR0B as the GATE compare or capture channel ● 0b0: Compare ● 0b1: Capture	RW
14	ONESHOT	Single trigger mode selection ● 0b0: Cycle count ● 0b1: Timer stops after an event update occurs	RW
13:12	MODE	Working mode ● 0b00: Timer mode 0 ● 0b01: PWC mode ● 0b10: Sawtooth wave mode ● 0b11: Triangle wave mode	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Bit	Name	Functional Description	Access
11	UDE	Update trigger DMA enable ● 0b0: Disable ● 0b1: Enable	RW
10	UIE	Update interrupt enable Counting to 0xFFFF will overflow and generate an overflow flag ● 0b0: Disable ● 0b1: Enable	RW
9	CFG	Falling edge capture valid control when GATE is used as capture input Only valid when PWM complementary output ● 0b0: Lower edge capture is invalid ● 0b1: Lower edge capture is valid	RW
8	CRG	Rising edge capture valid control when GATE is used as capture input Only valid when PWM complementary output ● 0b0: Upper edge capture is invalid ● 0b1: Upper edge capture is valid	RW
7	BUFPEN	Auto-reload ARR cache enable ● 0b0: Period cache disabled, written value immediately affects the period value ● 0b1: Period cache enabled, written value affects the period value after the next event update	RW
6:4	PRS	Internal clock frequency division selection ● 0b000: 1 ● 0b001: 2 ● 0b010: 4 ● 0b011: 8 ● 0b100: 16 ● 0b101: 32 ● 0b110: 64 ● 0b111: 256	RW
3	PWM2S	OCREFA dual-point compare selection ● 0b0: Dual-point compare enabled, uses CCRA and CCRB compares to control OCREFA output ● 0b1: Single-point compare enabled, only uses CCRA compare to control OCREFA output **Note** OCREFB remains unaffected and still uses CCRB to control OCREFB output.	RW
2	CT	Count clock selection ● 0b0: Internal count clock TCLK ● 0b1: External count clock ETR	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Bit	Name	Functional Description	Access
1	COMP	PWM complementary output mode selection ● 0b0: Independent PWM output ● 0b1: Complementary PWM output	RW
0	CTEN	Timer enable ● 0b0: Disable ● 0b1: Enable 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 (ATIM_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												CA3B	CA3A	UND	OV	TI	BI	CB2E	CB1E	CB0E	CA2E	CA1E	CA0E	CB2	CB1	CB0	CA2	CA1	CA0	Reserved	UI
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	CA3B	Channel CH3B compare match flag Only ATIM3 exists ● 0b0: Not occurred ● 0b1: Occurred	RO
18	CA3A	Channel CH3A compare match flag Only ATIM3 exists ● 0b0: Not occurred ● 0b1: Occurred	RO
17	UND	Underflow interrupt flag ● 0b0: Not occurred ● 0b1: Occurred	RO
16	OV	Overflow interrupt flag ● 0b0: Not occurred ● 0b1: Occurred	RO
15	TI	Trigger interrupt flag ● 0b0: Not occurred ● 0b1: Occurred	RO
14	BI	Brake interrupt flag ● 0b0: Not occurred ● 0b1: Occurred	RO
13	CB2E	Channel CH2B capture data loss flag Only ATIM3 exists ● 0b0: No data loss ● 0b1: Data loss	RO
12	CB1E	Channel CH1B capture data loss flag Only ATIM3 exists ● 0b0: No data loss ● 0b1: Data loss	RO
11	CB0E	Channel CH0B capture data loss flag ● 0b0: No data loss ● 0b1: Data loss	RO

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Bit	Name	Functional Description	Access
10	CA2E	Channel CH2A capture data loss flag Only ATIM3 exists ● 0b0: No data loss ● 0b1: Data loss	RO
9	CA1E	Channel CH1A capture data loss flag Only ATIM3 exists ● 0b0: No data loss ● 0b1: Data loss	RO
8	CA0E	Channel CH0A capture data loss flag ● 0b0: No data loss ● 0b1: Data loss	RO
7	CB2	Channel CH2B capture/compare match flag Only ATIM3 exists ● 0b0: Not occurred ● 0b1: Occurred	RO
6	CB1	Channel CH1B capture/compare match flag Only ATIM3 exists ● 0b0: Not occurred ● 0b1: Occurred	RO
5	CB0	Channel CH0B capture/compare match flag ● 0b0: Not occurred ● 0b1: Occurred	RO
4	CA2	Channel CH2A capture/compare match flag Only ATIM3 exists ● 0b0: Not occurred ● 0b1: Occurred	RO
3	CA1	Channel CH1A capture/compare match flag Only ATIM3 exists ● 0b0: Not occurred ● 0b1: Occurred	RO
2	CA0	Channel CH0A capture/compare match flag ● 0b0: Not occurred ● 0b1: Occurred	RO
1	*Reserved*	-	-
0	UI	Event update interrupt flag ● 0b0: Not occurred ● 0b1: Occurred	RO

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

13.5.4.5 Interrupt Flag Clear Register (ATIM_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	0x0000FFFF																															
Name	Reserved												CA3B	CA3A	UND	OV	TI	BI	CB2E	CB1E	CB0E	CA2E	CA1E	CA0E	CB2	CB1	CB0	CA2	CA1	CA0	Reserved	UI
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	CA3B	Channel CH3B compare match flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1. Only ATIM3 exists	WO
18	CA3A	Channel CH3A compare match flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1. Only ATIM3 exists	WO
17	UND	Underflow interrupt flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
16	OV	Overflow interrupt flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
15	TI	Trigger interrupt flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
14	BI	Brake interrupt flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
13	CB2E	Channel CH2B capture data loss flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1. Only ATIM3 exists	WO
12	CB1E	Channel CH1B capture data loss flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1. Only ATIM3 exists	WO
11	CB0E	Channel CH0B capture data loss flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
10	CA2E	Channel CH2A capture data loss flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1. Only ATIM3 exists	WO
9	CA1E	Channel CH1A capture data loss flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1. Only ATIM3 exists	WO
8	CA0E	Channel CH0A capture data loss flag clear 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	CB2	Channel CH2B capture/compare match flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1. Only ATIM3 exists	WO
6	CB1	Channel CH1B capture/compare match flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1. Only ATIM3 exists	WO
5	CB0	Channel CH0B capture/compare match flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
4	CA2	Channel CH2A capture/compare match flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1. Only ATIM3 exists	WO
3	CA1	Channel CH1A capture/compare match flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1. Only ATIM3 exists	WO
2	CA0	Channel CH0A capture/compare match flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
1	*Reserved*	-	-
0	UI	Event update interrupt flag clear 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 (ATIM_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 ● 0b0: CH0B ● 0b1: Internal trigger TS selection signal	RW
11	IA0S	IA0 input selection ● 0b0: CH0A ● 0b1: ▶ CH0A CH1A CH2A XOR (ATIM3) ▶ CH0A ETR GATE XOR (ATIM0/1/2) **Note** Setting to 1 will cause input changes upon any port transition.	RW
10:8	SMS	Slave mode function selection ● 0b000: Use internal clock ● 0b001: Reset function ● 0b010: Trigger mode ● 0b011: External clock mode ● 0b100: Quadrature encoder counting mode1 ● 0b101: Quadrature encoder counting mode2 ● 0b110: Quadrature encoder counting mode3 ● 0b111: Gating function	RW
7:5	TS	Trigger selection ● 0b000: Signal ETFP after port ETR filtered phase selection ● 0b001: Internal interconnect signal ITR0 ● 0b010: Internal interconnect signal ITR1 ● 0b011: Internal interconnect signal ITR2 ● 0b100: Internal interconnect signal ITR3 ● 0b101: Edge signal of port CH0A ● 0b110: Signal IAFP after port CH0A filtered phase selection ● 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
● 0b0: No delay
● 0b1: Delay enabled, synchronously starts both master and slave counters
Note
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
● 0b0: Compare match trigger DMA
● 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 selection
For internal interconnection, connected to ITRx of other timers
<table>
<tr><th></th><th>ATIM0/1/2</th><th>ATIM3</th></tr>
<tr><td>0b000</td><td>Software update UG, write M23CR.UG</td><td>Software update UG, write M23CR.UG
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></td><td>RW</td></tr>
</table>

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

13.5.4.7 Output Control/Input Filter Register (ATIM_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 ● 0b0: Same phase ● 0b1: Inverted input	RW
30:28	FLTET	ETR filter control ● 0b000-0b011: Filter is invalid ● 0b100: PCLK, three consecutive valid samples ● 0b101: PCLK/4, three consecutive valid samples ● 0b110: PCLK/16, three consecutive valid samples ● 0b111: PCLK/64, three consecutive valid samples	RW
27	BKP	Brake BK input phase selection ● 0b0: Same phase ● 0b1: Inverted input	RW
26:24	FLTBK	Brake input filter control ● 0b000-0b011: Filter is invalid ● 0b100: PCLK, three consecutive valid samples ● 0b101: PCLK/4, three consecutive valid samples ● 0b110: PCLK/16, three consecutive valid samples ● 0b111: PCLK/64, three consecutive valid samples	RW
23	CCPB2	Compare function: CH2B channel compare output phase control ● 0b0: Normal output ● 0b1: Inverted output	RW
22:20	OCMB2	Compare function: CH2B channel compare control Refer to OCMB0 setting	RW
	FLTB2	Capture function: CH2B input channel filter setup Refer to FLTBK setting	RW
19	CCPA2	Compare function: CH2A channel compare output phase control ● 0b0: Normal output ● 0b1: Inverted output	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Bit	Name	Functional Description	Access
18:16	OCMA2	Compare function: CH2A channel compare control Refer to OCMB0 setting	RW
	FLTA2	Capture function: CH2A input channel filter setup Refer to FLTBK setting	RW
15	CCPB1	Compare function: CH1B channel compare output phase control ● 0b0: Normal output ● 0b1: Inverted output	RW
14:12	OCMB1	Compare function: CH1B channel compare control Refer to OCMB0 setting	RW
	FLTB1	Capture function: CH1B input channel filter setup Refer to FLTBK setting	RW
11	CCPA1	Compare function: CH1A channel compare output phase control ● 0b0: Normal output ● 0b1: Inverted output	RW
10:8	OCMA1	Compare function: CH1A channel compare control Refer to OCMB0 setting	RW
	FLTA1	Capture function: CH1A input channel filter setup Refer to FLTBK setting	RW
7	CCPB0	Compare function: output compare mode CCPBx compare output CHBx port polarity control ● 0b0: Normal output ● 0b1: Inverted output	RW
	CCPB0	Encode count and slave mode gating function: input phase control CCPB0 slave mode gating, reset, external trigger, external clock using CH0B port input polarity control ● 0b0: Normal input ● 0b1: Inverted input	RW
6:4	OCMB0	Compare function: CH0B channel compare control ● 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	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Bit	Name	Functional Description	Access
6:4	OCMB0	◆ 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	
	FLTB0	Capture function: CH0B input channel filter setup Refer to FLTBK setting **Note** 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 CCPAx compare output CHAx port polarity control ● 0b0: Normal output ● 0b1: Inverted output	RW
	CCPA0	Encode count and slave mode gating function: input phase control CCPA0 slave mode gating, reset, external trigger, external clock using CH0A port input polarity control ● 0b0: Normal input ● 0b1: Inverted input	RW
2:0	OCMA0	Compare function: channel A compare control Refer to OCMB0 setting	RW
	FLTA0	Capture function: CH0A input channel filter setup Refer to FLTBK setting	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

13.5.4.8 ADC Trigger Control Register (ATIM_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 ● 0b0: Disable ● 0b1: Enable	RW
6	CMB2E	Channel 2B compare match trigger ADC enable Only ATIM3 exists ● 0b0: Disable ● 0b1: Enable	RW
5	CMB1E	Channel 1B compare match trigger ADC enable Only ATIM3 exists ● 0b0: Disable ● 0b1: Enable	RW
4	CMB0E	Channel 0B compare match trigger ADC enable ● 0b0: Disable ● 0b1: Enable	RW
3	CMA2E	Channel 2A compare match trigger ADC enable Only ATIM3 exists ● 0b0: Disable ● 0b1: Enable	RW
2	CMA1E	Channel 1A compare match trigger ADC enable Only ATIM3 exists ● 0b0: Disable ● 0b1: Enable	RW
1	CMA0E	Channel 0A compare match trigger ADC enable ● 0b0: Disable ● 0b1: Enable	RW
0	UEVE	Event update trigger ADC enable ● 0b0: Disable ● 0b1: Enable	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

13.5.4.9 Channel x Control Register (ATIM_CRCHx) (x=0/1/2)

Offset: 0x024 (CRCH0), 0x028 (CRCH1), 0x02C (CRCH2)

Note

CRCH1/CRCH2 only ATIM3 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 Hardware auto-clear, writing 1 triggers the following actions, writing 0 has no effect: ● In compare mode, generates interrupt only ● In capture mode, generates interrupt and captures counter value to capture register	WO
14	CCGA	Capture compare A software trigger Hardware auto-clear, writing 1 triggers the following actions, writing 0 has no effect: ● In compare mode, generates interrupt only ● In capture mode, generates interrupt and captures counter value to capture register	WO
13:12	CISB	B channel compare match setting ● 0b00: No match ● 0b01: Compare match during up-counting ● 0b10: Compare match during down-counting ● 0b11: Compare match during both up/down counting	RW
11	CDEB	B capture compare trigger DMA enable ● 0b0: Disable ● 0b1: Enable	RW
10	CDEA	A capture compare trigger DMA enable ● 0b0: Disable ● 0b1: Enable	RW
9	CIEB	B capture compare trigger interrupt enable ● 0b0: Disable ● 0b1: Enable	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Bit	Name	Functional Description	Access
8	CIEA	A capture compare trigger interrupt enable ● 0b0: Disable ● 0b1: Enable	RW
7	BUFEB	Compare function: B compare cache enable control ● 0b0: Disable ● 0b1: Enable	RW
6	BUFEA	Compare function: A compare cache enable control ● 0b0: Disable ● 0b1: Enable	RW
5	CSB	B channel capture/compare function selection ● 0b0: Compare mode ● 0b1: Capture mode	RW
4	CSA	A channel capture/compare function selection ● 0b0: Compare mode ● 0b1: Capture mode	RW
3:2	CFB_CRB	Capture function: channel B rising falling edge capture enable ● 0b00: Disable ● 0b01: Rising edge capture ● 0b10: Falling edge capture ● 0b11: Both rising and falling edge capture	RW
	BKSB	Compare function: channel B output brake level control ● 0b00: High-impedance output ● 0b01: No effect on output ● 0b10: Force output low level ● 0b11: Force output high level **Note** No DEBUG_STOP mode.	RW
1:0	CFA_CRA	Capture function: channel A rising falling edge capture enable ● 0b00: Disable ● 0b01: Rising edge capture ● 0b10: Falling edge capture ● 0b11: Both rising and falling edge capture	RW
	BKSA	Compare function: channel A output brake level control ● 0b00: High-impedance output ● 0b01: No effect on output ● 0b10: Force output low level ● 0b11: Force output high level **Note** 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 (ATIM_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 ● 0b0: Disable ● 0b1: Enable	RW
13	SAFEEN	Safety brake enbale (osc fail,brown down,lockup) ● 0b0: Disable ● 0b1: Enable	RW
12	MOE	PWM output enable ● 0b0: Disable ● 0b1: Enable	RW
11	AOE	PWM output automatically enable ● 0b0: Disable ● 0b1: Enable	RW
10	BKE	Port brake enable ● 0b0: Disable ● 0b1: Enable	RW
9	DTEN	Dead time control enable ● 0b0: Disable ● 0b1: Enable	RW
8	BKSEL	Brake selection ● 0b0: Use own brake control ● 0b1: ATIM1/2 uses brake control of ATIM0 **Note** ATIM0 and ATIM3 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 (ATIM_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																						UD	OV	RCR							
Access																							RW	RW	RW							

Bit	Name	Functional Description	Access
31:10	*Reserved*	-	-
9	UD	Underflow mask control for repeating counting ● 0b0: Down-counting overflow enable ● 0b1: Down-count overflow mask	RW
8	OV	Overflow mask control for repetiting counting ● 0b0: Up-counting overflow enable ● 0b1: Up-counting overflow mask	RW
7:0	RCR	Repetition period count value 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 (ATIM_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 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 (ATIM_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 ATIM3 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 Comparison with cache function	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

13.5.4.14 Channel 3A Compare Register (ATIM_CCR3A) (Only ATIM3 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 Comparison with cache function	RW

13.5.4.15 Channel 3A Control Register (ATIM_CRCH3A) (Only ATIM3 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																										C3AEN	CIS		CD	CI	BUF
Access																											RW	RW		RW	RW	RW

Bit	Name	Functional Description	Access
31:6	*Reserved*	-	-
5	C3AEN	Channel compare enable ● 0b0: Disable ● 0b1: Enable	RW
4:3	CIS	Channel compare match setting ● 0b00: No match ● 0b01: Compare match during up-counting ● 0b10: Compare match during down-counting ● 0b11: Compare match during both up/down counting	RW
2	CD	Compare trigger DMA enable ● 0b0: Disable ● 0b1: Enable	RW
1	CI	Compare trigger interrupt enable ● 0b0: Disable ● 0b1: Enable	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Bit	Name	Functional Description	Access
0	BUF	Compare cache enable ● 0b0: Disable ● 0b1: Enable	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

13.5.4.16 Channel 3B Compare Register (ATIM_CCR3B) (Only ATIM3 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 Comparison with cache function	RW

13.5.4.17 Channel 3B Control Register (ATIM_CRCH3B) (Only ATIM3 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																										C3BEN	CIS		CD	CI	BUF
Access																											RW	RW		RW	RW	RW

Bit	Name	Functional Description	Access
31:6	*Reserved*	-	-
5	C3BEN	Channel compare enable ● 0b0: Disable ● 0b1: Enable	RW
4:3	CIS	Channel compare match setting ● 0b00: No match ● 0b01: Compare match during up-counting ● 0b10: Compare match during down-counting ● 0b11: Compare match during both up/down counting	RW
2	CD	Compare trigger DMA enable ● 0b0: Disable ● 0b1: Enable	RW
1	CI	Compare trigger interrupt enable ● 0b0: Disable ● 0b1: Enable	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Bit	Name	Functional Description	Access
0	BUF	Compare cache enable ● 0b0: Disable ● 0b1: Enable	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

14 CTIM

14.1 Introduction

CTIM is a composite timer, which can be configured as a general timer GTIM with four groups of comparison capture functions, or as a group (three) of basic timers BTIM with timing functions only. Controlled by SYSCTRL_CR1.GTIMxCFG. When this bit is 1, BTIM is active; when 0, GTIM is active. Each CTIM in this product can be independently and flexibly configured as either GTIM or BTIM.

- BTIM is a group of timing counters with basic timing functions, each group contains three 16-bit timers with reload function. This product can be configured with up to six basic timers: BTIM0-5.
- GTIM is a 16-bit timer with 4-way compare capture function. This product can be configured with up to two basic timers: BTIM0-1.

Table 14-1 Comparison Table of Pin Function and Timer Function

CTIM (x=0, 1)	GTIM (x=0, 1)	BTIM (y=0, 3)
CTIM*x*_ETR	GTIM*x*_ETR	BTIM*y*_ETR
CTIM*x*_TOG	GTIM*x*_TOG	BTIM*y+0*_TOG
CTIM*x*_TOGN	GTIM*x*_TOGN	BTIM*y+0*_TOGN
CTIM*x*_CH0	GTIM*x*_CH0	BTIM*y+1*_TOG
CTIM*x*_CH1	GTIM*x*_CH1	BTIM*y+1*_TOGN
CTIM*x*_CH2	GTIM*x*_CH2	BTIM*y+2*_TOG
CTIM*x*_CH3	GTIM*x*_CH3	BTIM*y+2*_TOGN

Note

- BTIM0/1/2 share CTIM0_ETR; BTIM3/4/5 share CTIM1_ETR.
- When x=0, y=0; when x=1, y=3.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

14.2 Features

14.2.1 BTIM Features

- Three 16-bit timers
- 2 toggle outputs per timer
- Single pulse mode
- External input counting function
- Gating function
- External trigger function
- Cascade function

14.2.2 GTIM Features

- 16-bit timer
- Two toggle outputs
- Single pulse mode
- External input counting function
- Gating function
- External trigger function
- Four independent PWM outputs, adjustment range 0%-100%
- 4-way independent capture
- Cascade function

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

14.3 Functional Description

14.3.1 BTIM Functional Description

14.3.1.1 Function Block Diagram

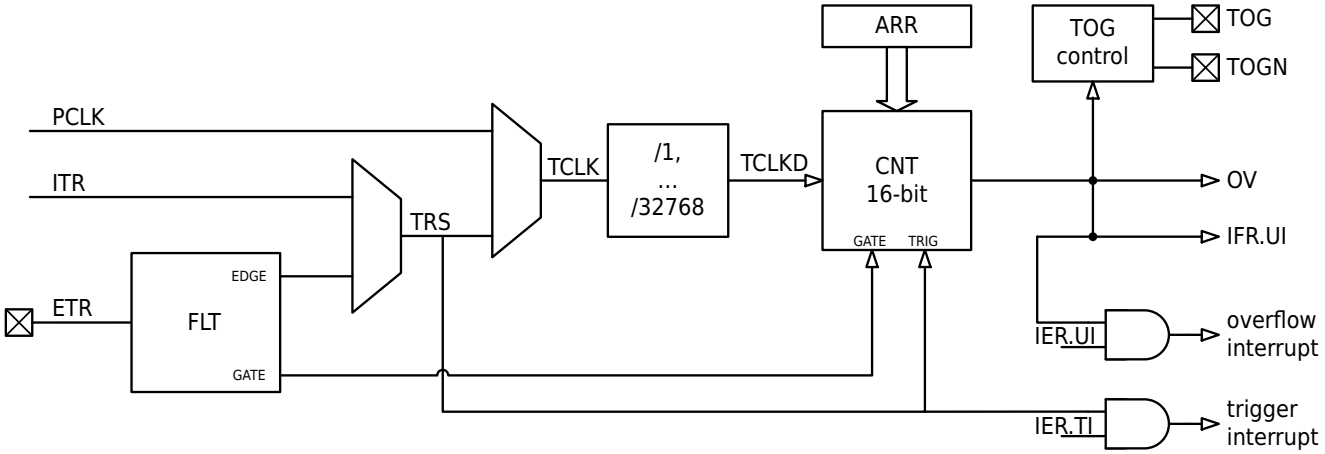


Figure 14-1 BTIM Block Diagram

Note

- A group of 3 BTIM share one ETR.
- For more information on the ITR signal sources, see the section "Timer Cascade".

14.3.1.2 Filter Unit

ETR signal input by the external pin is filtered, polarity selected, and edge detected, the gating signal and edge signal are output for use by the counting unit and the triggering unit. Since the ETR signal uses PCLK for synchronous filtering, the frequency of the input ETR signal should be lower than 1/2 of the frequency of PCLK.

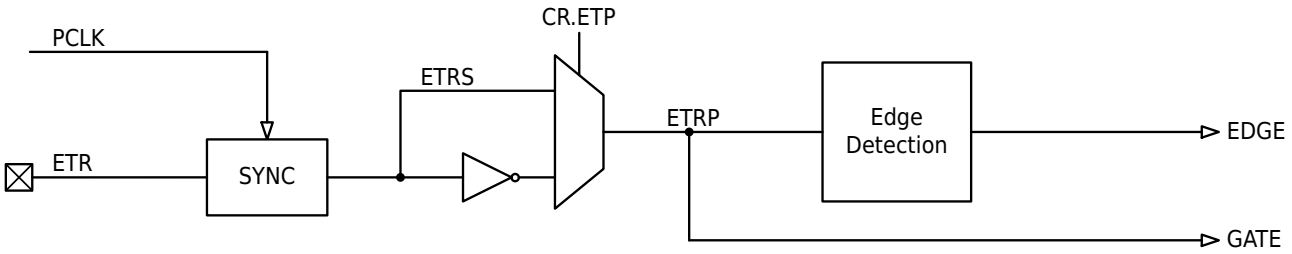


Figure 14-2 BTIM Filter Unit Diagram

14.3.1.3 Counting Unit

On each rising edge of the TCLKD signal, the internal counter CNT is automatically incremented by 1. Counting to ARR, an overflow signal OV (width of one PCLK) is generated, and the internal count value is reset to 0 and counting up starts again. The counting range of the counter is 0 to ARR, and the counting period of the counting unit is ARR+1.

The timer supports 2 counting modes, One-Shot mode and continuous counting mode.

- When BTIMx_CR.OST is set to 1, the timer works in One-Shot mode; the timer stops counting immediately after an overflow occurs, and the hardware automatically clears CEN.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

- When BTIMx_CR.OST is set to 0, the timer works in continuous counting mode; as long as BTIMx_CR.CEN is 1, the timer will keep counting.

Every time there is an overflow, BTIMx_IFR.UI will be set to 1 by hardware. If the corresponding interrupt is enabled, an overflow interrupt can be generated. The user should clear the overflow flag in the interrupt service routine.

If BTIMx_CR.TOGEN is set to 1, the TOG/TOGN output toggles once on each overflow. The TOG/TOGN signal can be output to a pin via GPIO multiplexing function to drive external circuits.

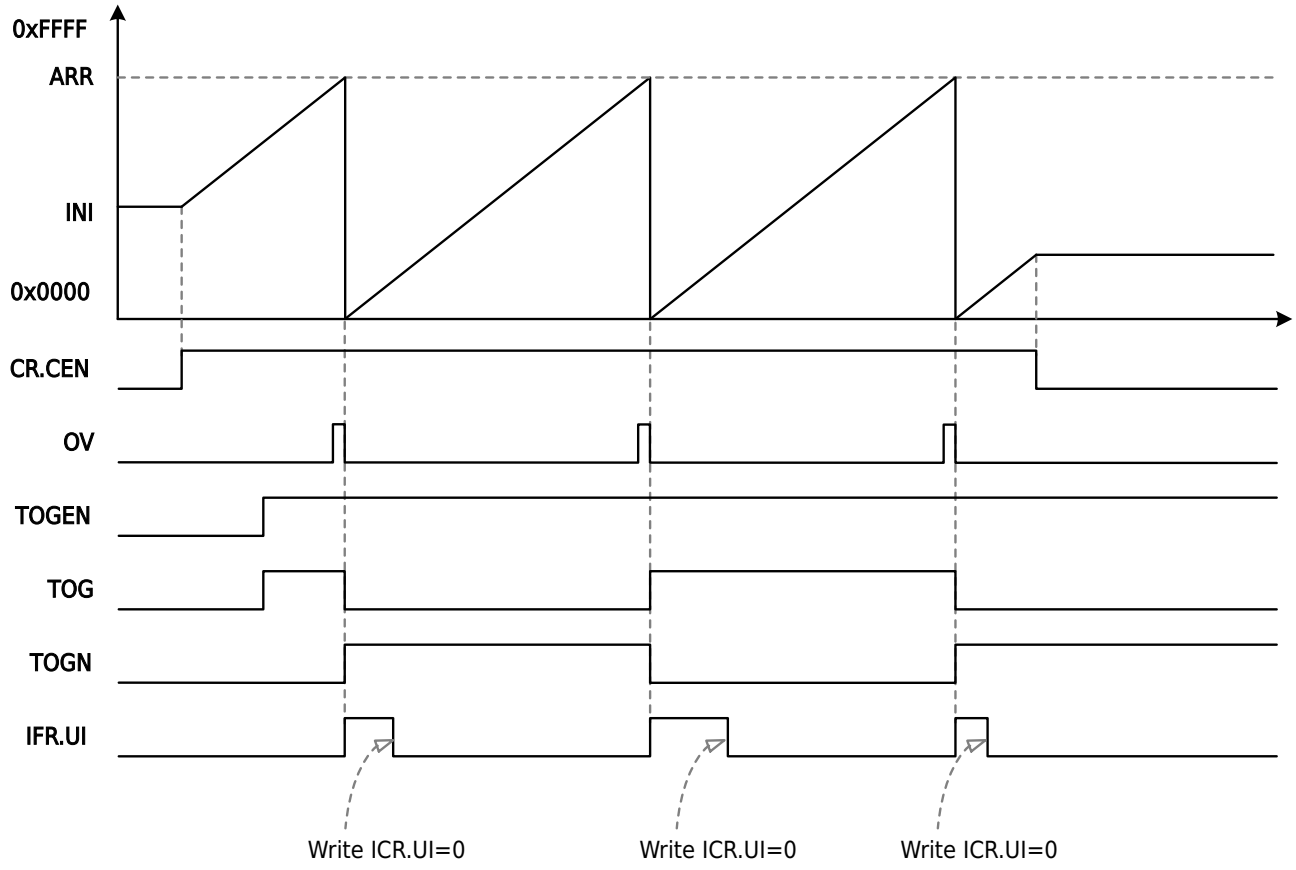


Figure 14-3 BTIM Counter Waveform

14.3.1.4 Timer Mode

When setting BTIMx_CR.MD to 0x0, BTIMx works in timer mode. In this mode, the counting clock is the clock TCLKD after PCLK frequency division. When BTIMx_CR.CEN is set to 1, the timer starts counting up from the initial value, and overflows after counting to ARR and restarts counting from 0.

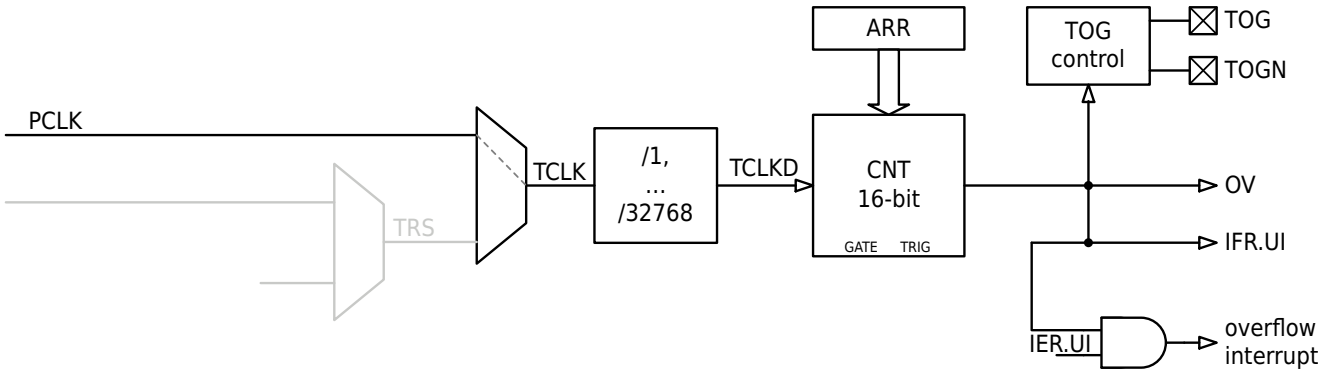


Figure 14-4 BTIM Timing Mode Block Diagram

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

14.3.1.5 Counter Mode

When setting BTIMx_CR.MD to 0x1, BTIMx works in counter mode. In this mode, the counting clock is the signal input by the ITR or ETR pin after frequency division. When BTIMx_CR.CEN is set to 1, the counter starts counting up from the initial value, and overflows after counting to ARR, and then starts counting from 0 again.

The counting clock source can be selected as ITR or ETR through BTIMx_CR.TRS; through BTIMx_CR.ETP, the rising edge or falling edge of the ETR pin input signal can be selected to be counted.

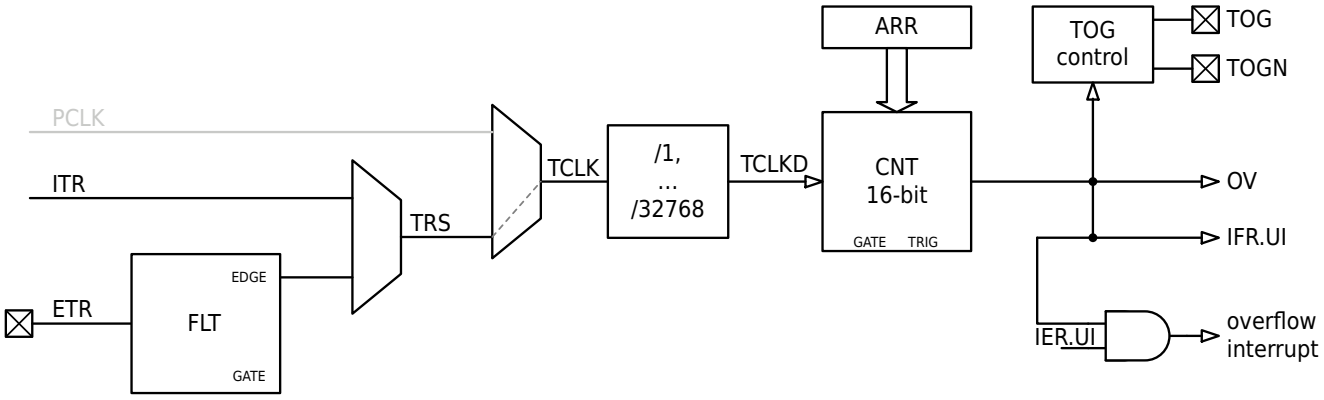


Figure 14-5 BTIM Counter Mode Block Diagram

14.3.1.6 Trigger Start Mode

When BTIMx_CR.MD is set to 0x2, BTIMx works in the trigger start mode, and the counting clock is the clock TCLKD after PCLK frequency division. In this mode, the timer has two start methods: set BTIMx_CR.CEN to 1 or there is an expected level change on the TRS signal.

After starting, the counter starts counting up from the initial value, overflows after counting to ARR and restarts counting from 0; set BTIMx_CR.CEN to 0 at any time, you can stop the timer.

Select TRS signal source as ETR or ITR through BTIMx_CR.TRS.

The rising edge or falling edge of the ETR pin input signal can be selected to start the timer through BTIMx_CR.ETP.

Every time an expected trigger signal is detected, BTIMx_IFR.TI will be set to 1 by hardware. If the corresponding interrupt is enabled, a trigger interrupt can be generated. The user should clear the trigger flag in the interrupt service routine.

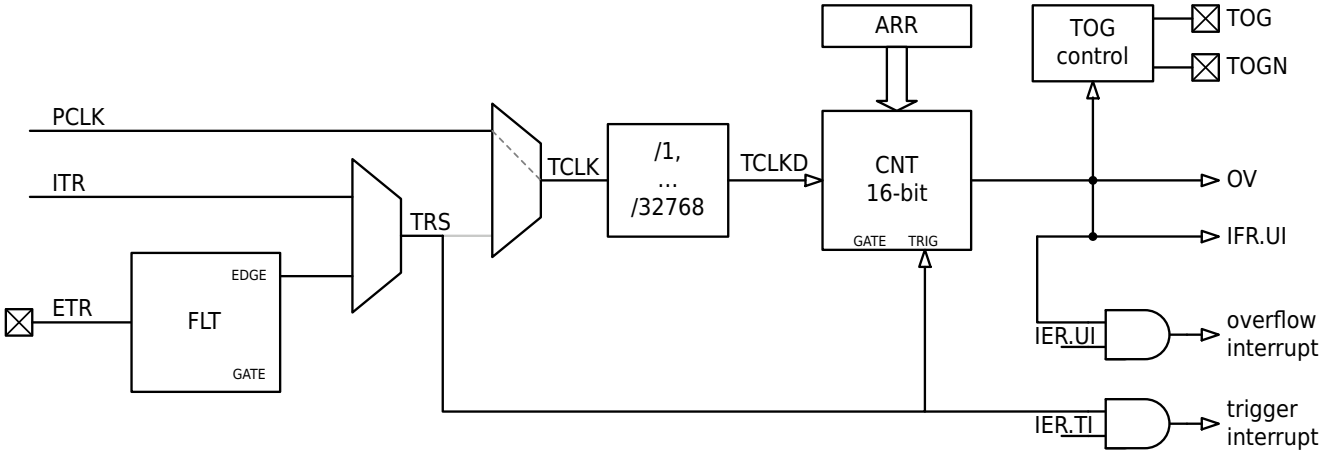


Figure 14-6 BTIM Trigger Start Mode Block Diagram

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

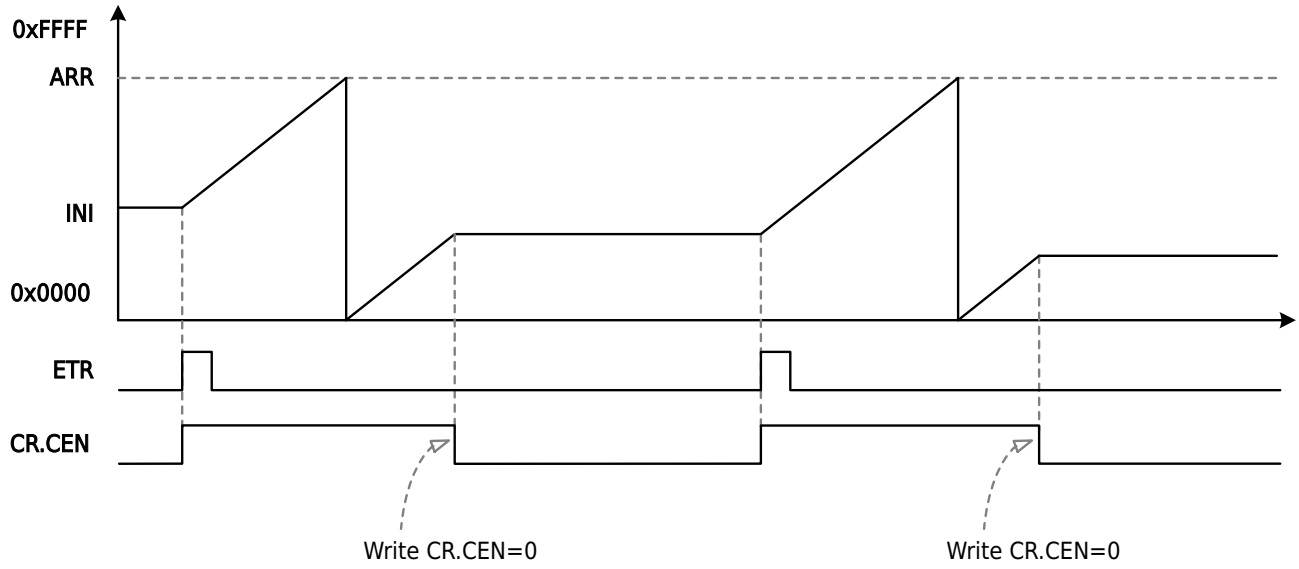


Figure 14-7 BTIM Trigger Start Mode Counting Diagram

14.3.1.7 Gating Mode

When BTIMx_CR.MD is set to 0x3, BTIMx works in gating mode, and the counting clock is the clock TCLKD after PCLK frequency division. In this mode, the timer needs to meet two conditions before it can count: BTIMx_CR.CEN is 1, and the level on the ETR pin meets expectations. When the timer is running, the counter starts counting up from the initial value, overflows after counting to ARR and starts counting from 0 again.

The high or low level of the ETR pin input signal can be selected as the counting level through BTIMx_CR.ETP.

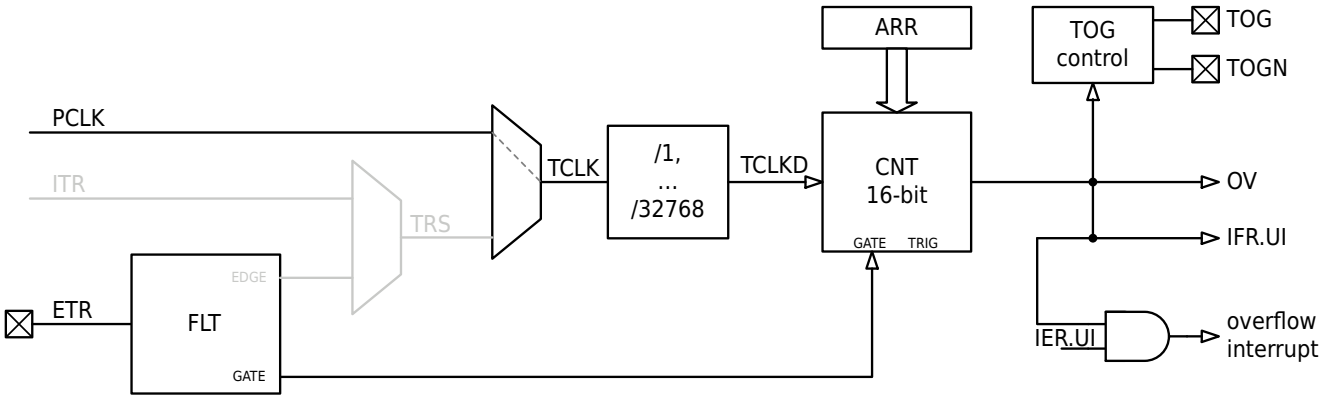


Figure 14-8 BTIM Gating Mode Block Diagram

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

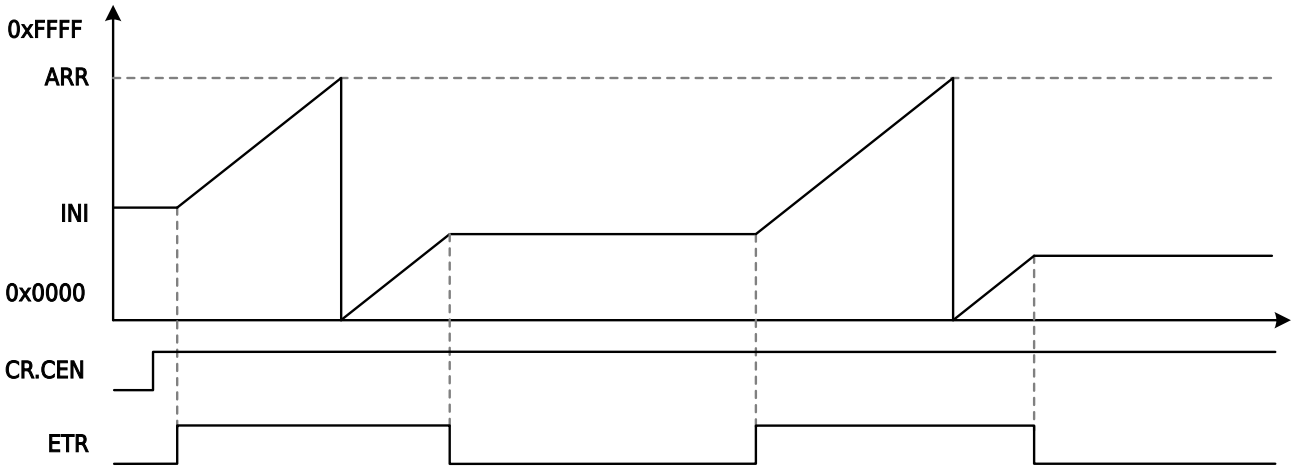


Figure 14-9 BTIM Gating Mode Counting Diagram

14.3.1.8 Timer Cascade

BTIM can be used in cascade through appropriate configuration, such as combining two 16-bit timers into a 32-bit timer. When used in cascade, the count clock of the first stage timer can be PCLK or the input signal from the ETR pin, and the count clock of the subsequent stage timer is the overflow signal of the previous stage timer. The signal cascading has been solidified in the chip, and its cascading method is as follows.

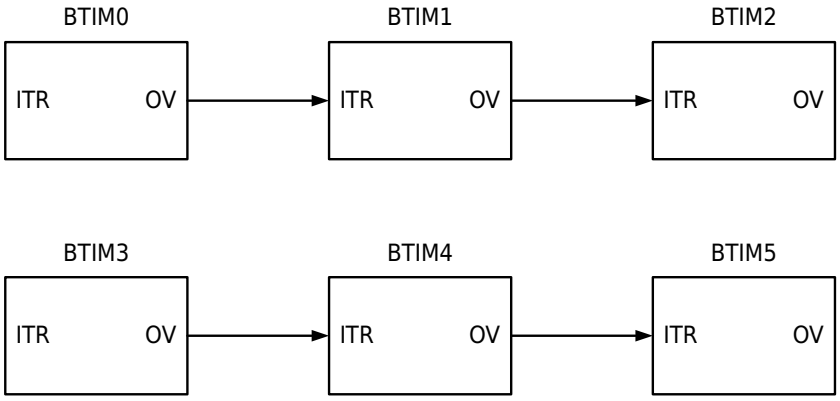


Figure 14-10 BTIM Timer Cascade Diagram

14.3.1.9 Interrupt

Table 14-2 BTIM Timer Interrupt

Interrupt	Description
BTIMn	Overflow/trigger interrupt
BTIMn+1	Overflow/trigger interrupt
BTIMn+2	Overflow/trigger interrupt

Note

The value of n can be 0 and 3.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

14.3.2 GTIM Functional Description

14.3.2.1 Function Block Diagram

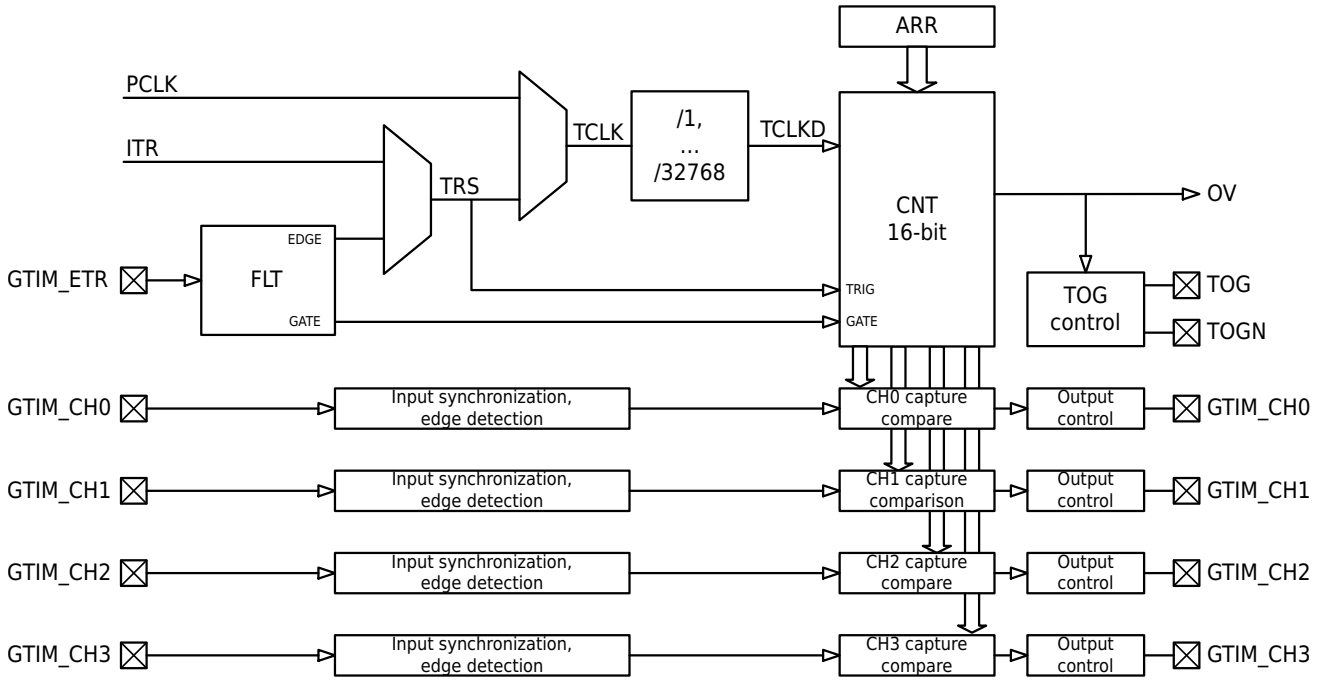


Figure 14-11 GTIM Block Diagram

Note

The source of the ITR signal is detailed in the "On-Chip Peripheral Interconnection" section.

14.3.2.2 Filter Unit

After synchronous filtering, polarity selection, and edge detection of the ETR signal input by the external pin, the gating signal and edge signal are output for use by the counting unit and the triggering unit. Since the ETR signal uses a filter clock for synchronous filtering, the frequency of the input ETR signal should be lower than one-half of the frequency of the filter clock.

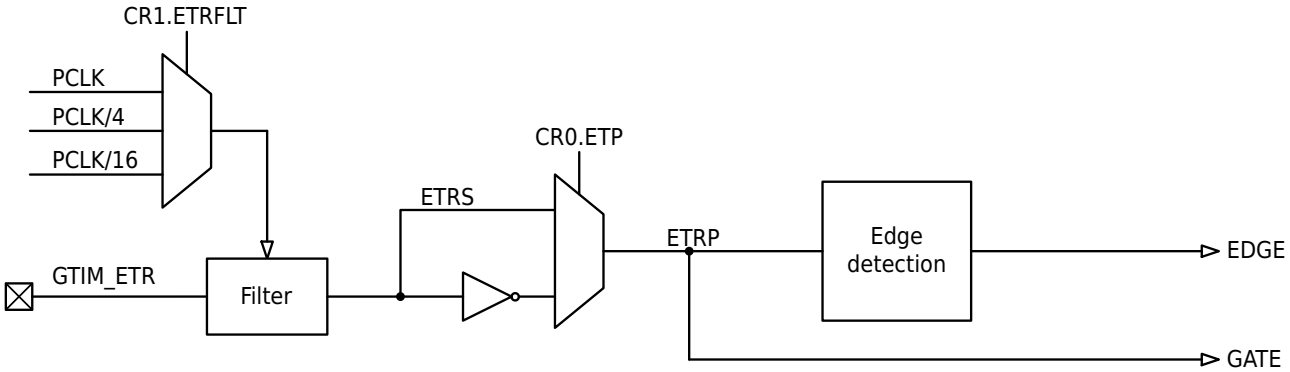


Figure 14-12 GTIM Filter Unit Diagram

14.3.2.3 Counting Unit

On each rising edge of the TCLKD signal, the internal counter CNT is automatically incremented by 1. counting to ARR, an overflow signal OV (width of one PCLK) is generated, and the internal count value is

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

reset to 0 and counting up starts again. The counting range of the counter is 0 to ARR, and the counting period of the counting unit is ARR+1.

The timer supports two counting modes: One-Shot mode and continuous counting mode.

- When GTIM_CR0.OST is set to 1, the timer works in One-Shot mode; the timer stops counting immediately after an overflow occurs, and the hardware automatically clears CEN.
- When setting GTIM_CR0.OST to 0, the timer works in continuous counting mode; as long as GTIM_CR0.CEN is 1, the timer will keep counting.

Every time there is an overflow, GTIM_IFR.UI will be set to 1 by hardware. If the corresponding interrupt is enabled, an overflow interrupt can be generated. The user should clear the overflow flag in the interrupt service routine.

If GTIM_CR0.TOGEN is 1, the TOG/TOGN output level toggles once per overflow. Users can output the TOG/TOGN signal to a pin through GPIO multiplexing function to drive external circuits.

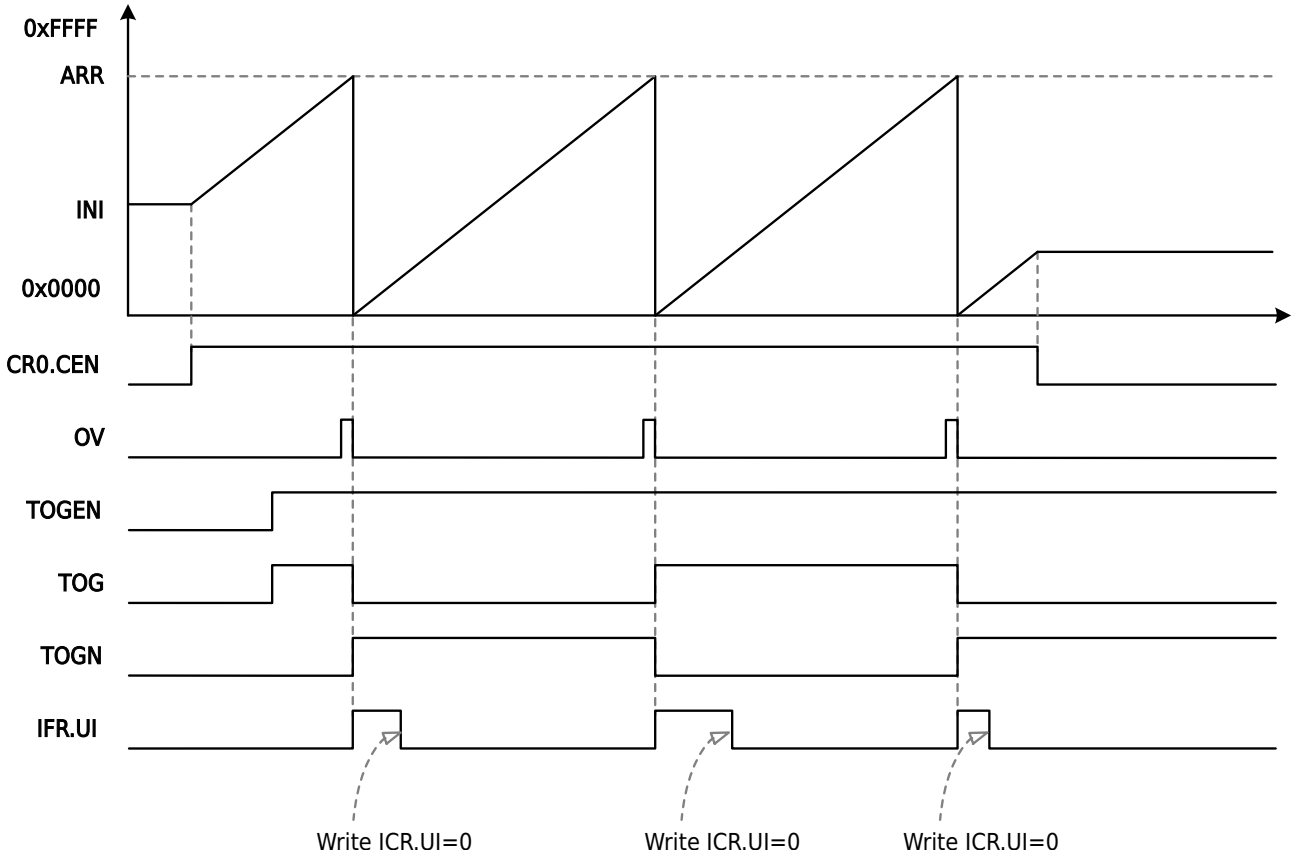


Figure 14-13 GTIM Counter Waveform

14.3.2.4 Timer Mode

GTIM_CR0.MD is set to 0x0, GTIM works in timer mode. In this mode, the counting clock is TCLKD derived from the divided PCLK. When GTIM_CR0.CEN is set to 1, the timer starts incrementing from the initial value. Upon reaching the ARR value, an overflow occurs and the counter restarts from 0.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

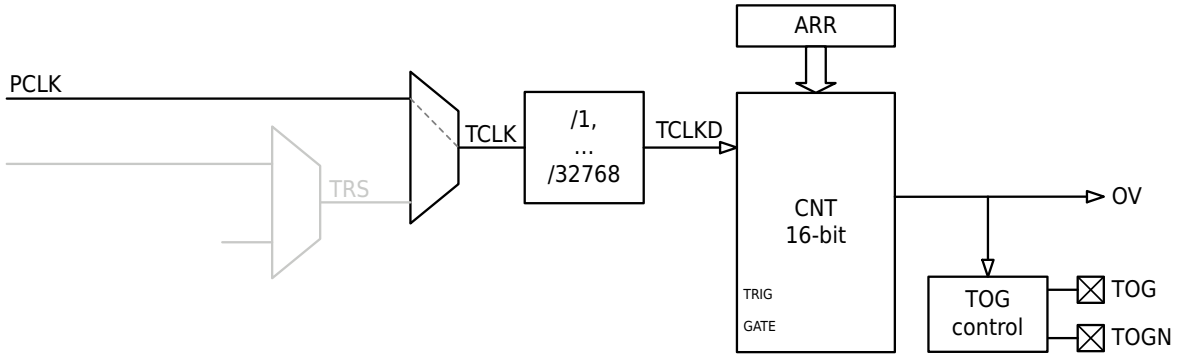


Figure 14-14 GTIM Timing Mode Block Diagram

14.3.2.5 Counter Mode

When GTIM_CR0.MD is set to 0x1, GTIM works in counter mode. In this mode, the counting clock is the signal input by the ITR or ETR pin after frequency division. When GTIM_CR0.CEN is set to 1, the counter begins incrementing from the initial value. Upon reaching the ARR value, an overflow occurs and the counting restarts from 0.

Through GTIM_CR0.TRS, you can select the counting clock source as ITR or ETR; through GTIM_CR0.ETP, you can choose to count the rising edge or falling edge of the ETR pin input signal.

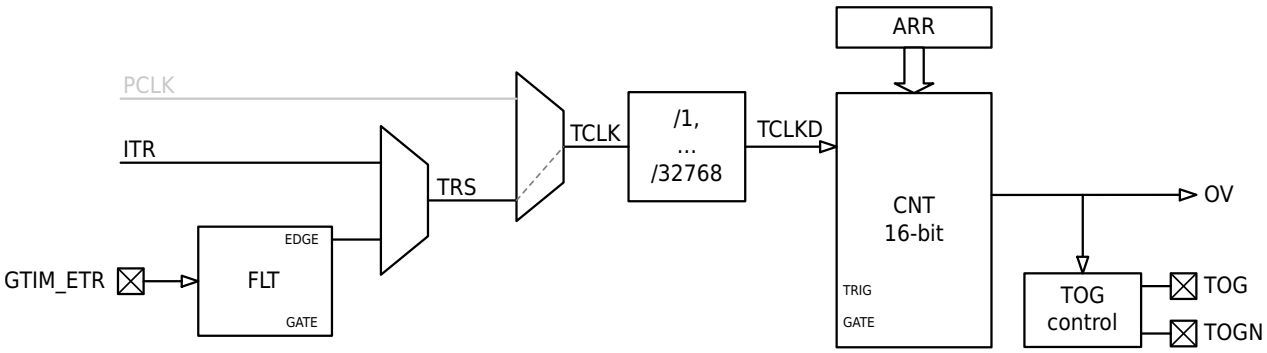


Figure 14-15 GTIM Counter Mode Block Diagram

14.3.2.6 Trigger Start Mode

When GTIM_CR0.MD is set to 0x2, GTIM works in trigger start mode, and the count clock is TCLKD after PCLK frequency division. In this mode, the timer has two startup methods: setting GTIM_CR0.CEN to 1 or detecting the expected level change on the TRS signal.

After starting, the counter starts counting up from the initial value, overflows after counting to ARR, and restarts counting from 0; set GTIM_CR0.CEN to 0 at any time, you can stop the timer.

Select the TRS signal source as ETR or ITR through GTIM_CR0.TRS.

The rising edge or falling edge of the ETR pin input signal can be selected to start the timer through GTIM_CR0.ETP.

Every time an expected trigger signal is detected, GTIM_IFR.TI will be set to 1 by hardware. If the corresponding interrupt is enabled, a trigger interrupt can be generated. The user should clear the trigger flag in the interrupt service routine.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

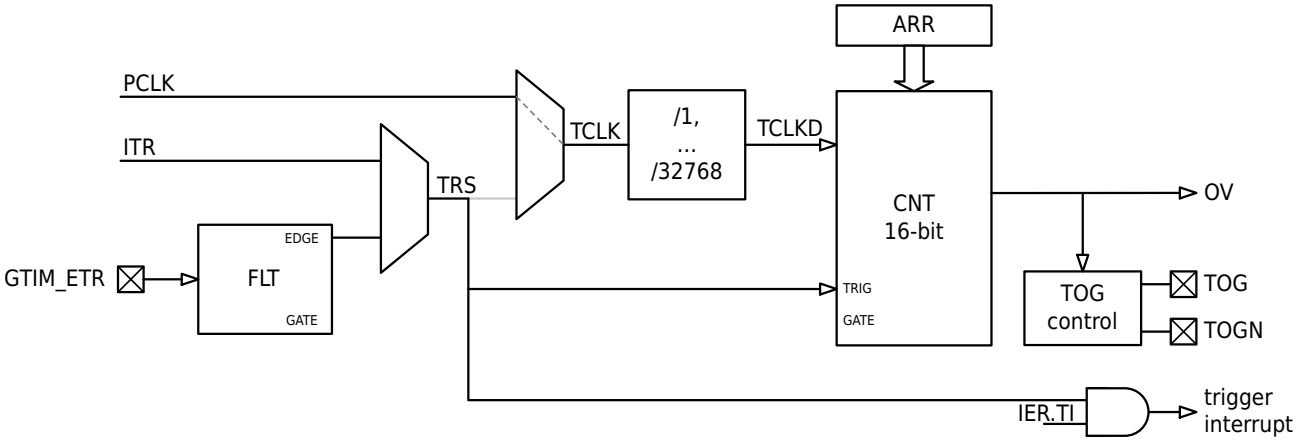


Figure 14-16 GTIM Trigger Start Mode Block Diagram

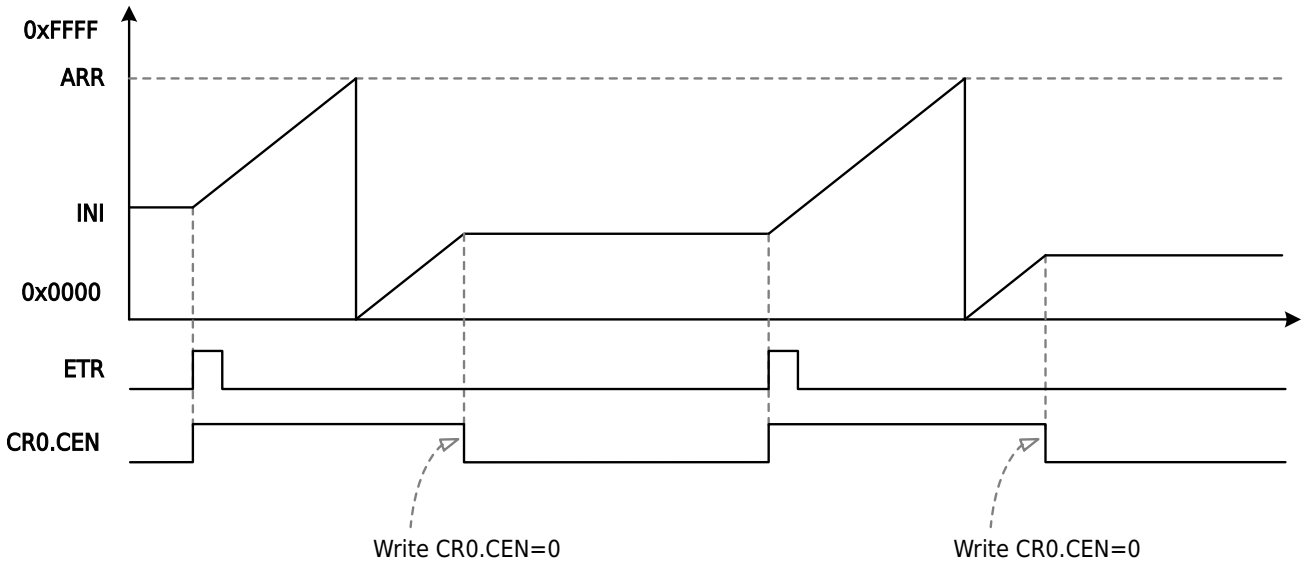


Figure 14-17 GTIM Trigger Start Mode Counting Diagram

14.3.2.7 Gating Mode

When GTIM_CR0.MD is set to 0x3, GTIM works in gating mode, and the count clock is TCLKD after PCLK frequency division. In this mode, the timer needs to meet two conditions before it can count: GTIM_CR0.CEN is 1, and the level on the ETR pin meets expectations. When the timer is running, the counter starts counting up from the initial value, overflows after counting to ARR and starts counting from 0 again.

The high or low level for the ETR pin input signal can be selected as the counting level via GTIM_CR0.ETP.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

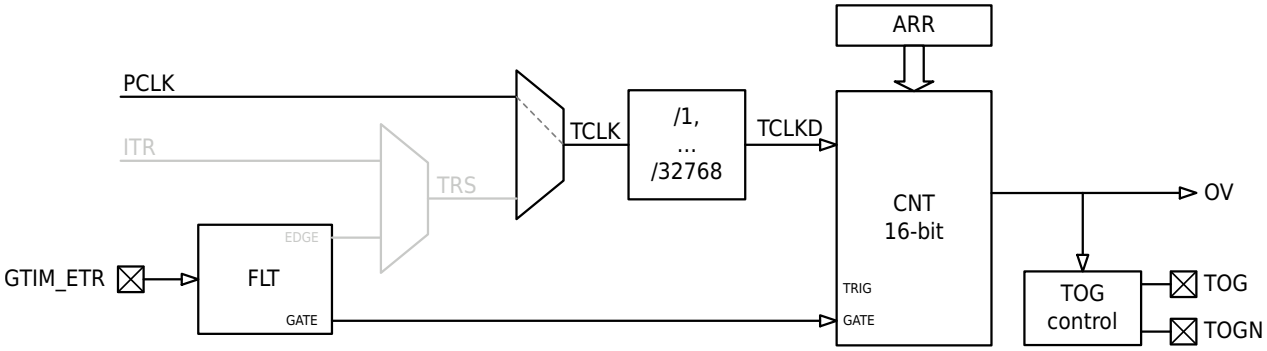


Figure 14-18 GTIM Gating Mode Block Diagram

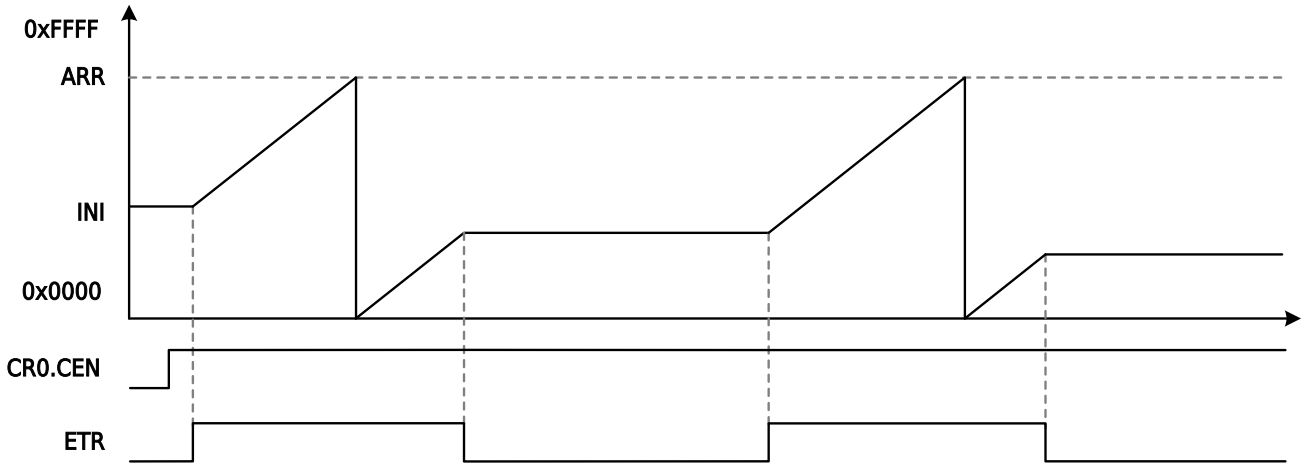


Figure 14-19 GTIM Gating Mode Counting Diagram

14.3.2.8 Compare Capture Function

The GTIM features four compare capture channels: CH0-CH3.

14.3.2.8.1 Capture Function

When CMMR.CCyM (y=0-3) is configured to 1-3, CHy operates in capture mode and supports three capture methods:

- 0b001: Rising edge capture
- 0b010: Falling edge capture
- 0b011: Capture on both rising and falling edges

When the desired pulse edge is detected on CHy channel, the hardware automatically writes the current counter value into the CCRy register and sets IFR.CCy. If the corresponding channel interrupt is enabled, a compare capture interrupt will be generated. The user should clear this interrupt flag in the interrupt service routine.

The figure below shows the schematic of two channels for input capture. Among them, CH0 is captured on the rising edge; CH1 is captured on the rising and falling edges at the same time.

When using the capture mode, the prescaler CR0.PRS must be less than 4, that is, the prescaler can only use 1, 2, 4, and 8; other prescaler values are prohibited from being configured. When using external ETR counting, the capture function cannot be used.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

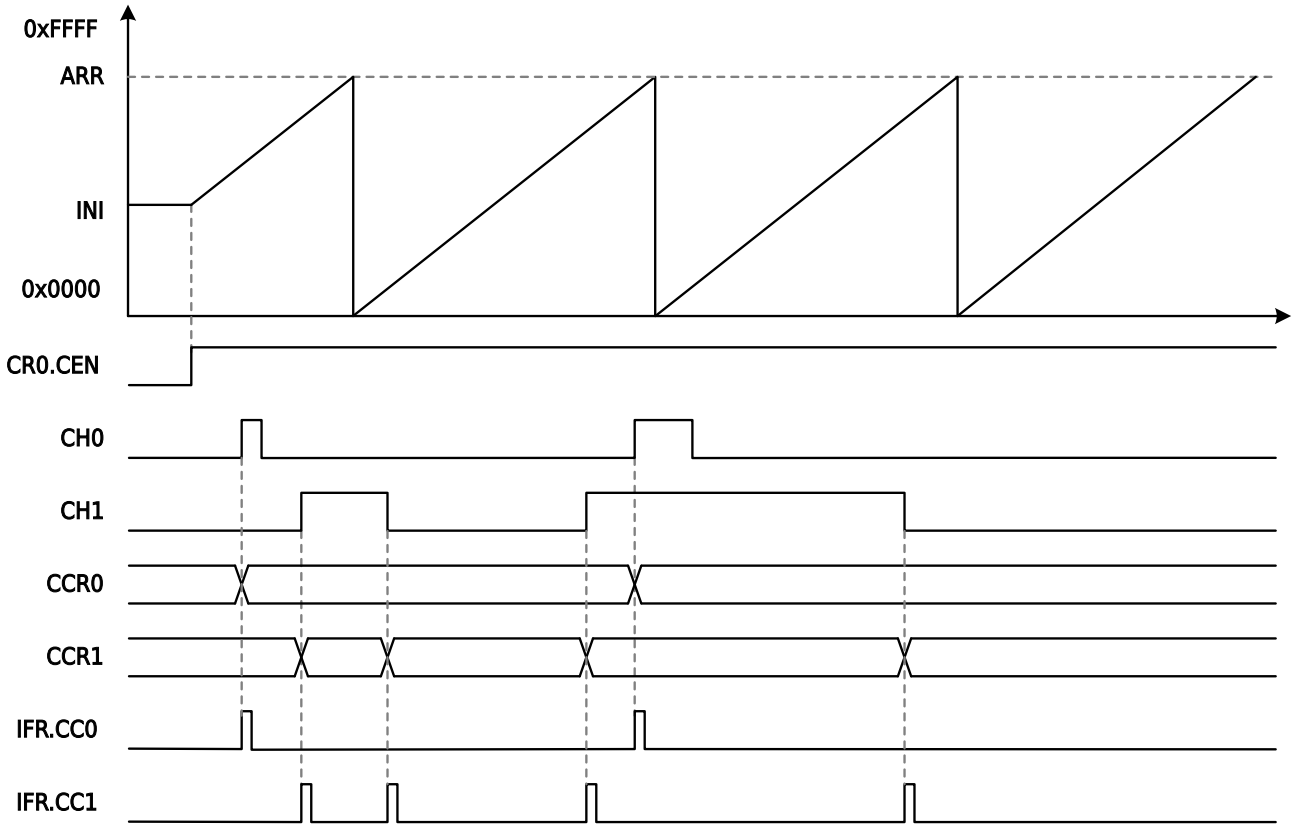


Figure 14-20 GTIM Capture Function Diagram

14.3.2.8.2 Compare Function

When CMMR.CCyM (y=0-3) is configured to 4-7, CHy operates in compare mode and can generate four types of waveforms:

- 0b100: Force output low
- 0b101: Force output high
- 0b110: PWM non-inverted output (outputs high level when CNT≥CCRy)
- 0b111: PWM inverted output (outputs high level when CNT<CCRy)

Note

When the ARR value is not 0xFFFF, the PWM adjustment range can be achieved from 0% to 100%.

For example, in non-inverted output mode when ARR is 99, the duty cycle corresponding to CCR values from 0-100 is 100%-0%.

The figure below shows the diagram of the comparison output of two channels. Among them, CH0 adopts PWM forward output; CH1 adopts PWM reverse output.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

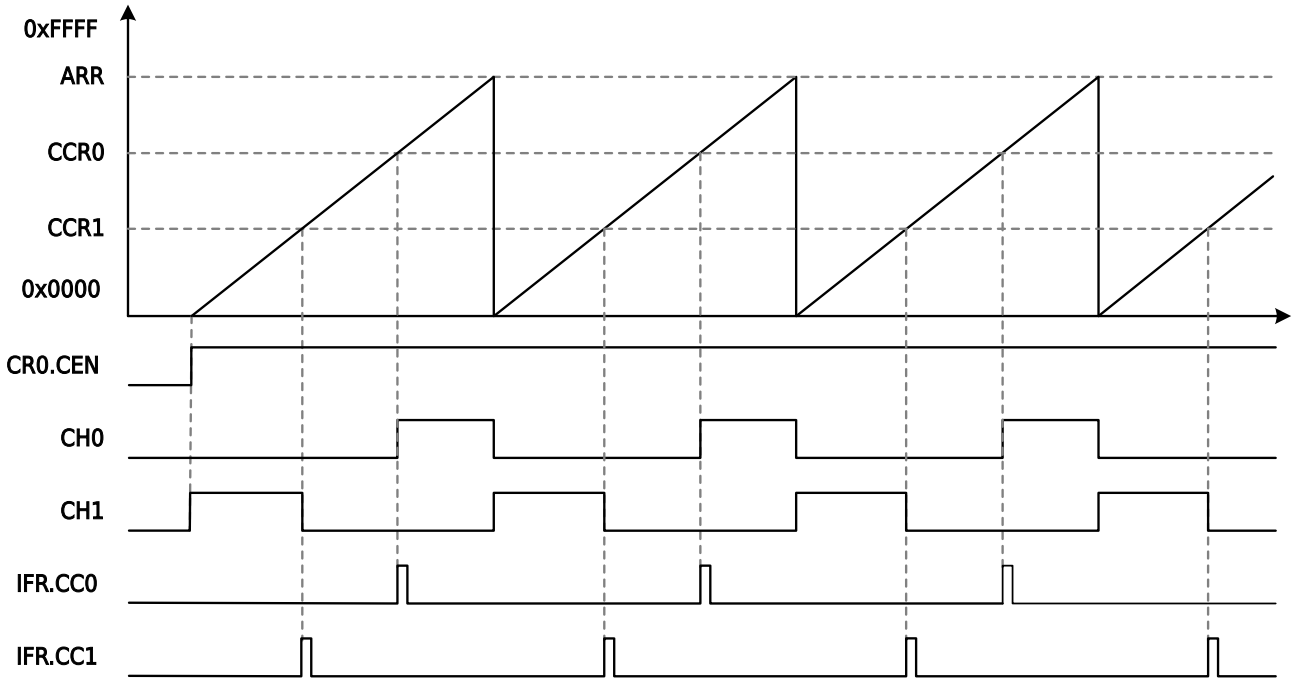


Figure 14-21 GTIM Compare Function Diagram

14.3.2.9 DMA Trigger

Compare capture can generate trigger signal to trigger DMA. Enable control via GTIM_IER.DCCy (y=0-3).

14.3.2.10 On-Chip Peripheral Interconnection

Trigger interconnection information as below:

Table 14-3 GTIM TRS Trigger Source List

CR0.TRS	0b00	0b01	0b10	0b11
GTIM0	ETR	ATIM0_TRGO	ADC0_EXTTRIGGER0.TRIG	ADC0_EXTTRIGGER1.TRIG
GTIM1	ETR	ATIM1_TRGO	GTIM0 overflow	ATIM3_TRGO

When SYSCTRL.IOAUX1.ETRx is 0, the source of GTIM_ETR signal is determined by GPIO_AFRH/ GPIO_AFRL; when SYSCTRL.IOAUX1.ETRx is 1-7, the source of GTIM_ETR signal is determined by the following table.

Table 14-4 GTIM ETR Source List

SYSCTRL->IOAUX1	0	1	2	3	4	5	6	7
CTIM1ETR[10:8]	CTIM1_ETR	LPUART0_RXD	VC0_OUT	LVD_OUT	VC1_OUT	PA06	PA14	PB06
CTIM0ETR[6:4]	CTIM0_ETR	USART1_RXD	VC0_OUT	LVD_OUT	VC1_OUT	PA05	PA13	PB05

When SYSCTRL.IOAUX2.CAPx is 0, the capture signal source of GTIM_CH3 is determined by GPIO_AFRH/ GPIO_AFRL; when SYSCTRL.IOAUX2.CAPx is 1-7, the source of GTIM_CH3 signal is determined by the following table.

Table 14-5 GTIM CH3 Signal Source List

SYSCTRL->IOAUX2	0	1	2	3	4	5	6	7
CTIM1CAP3[10:8]	CTIM1_CH3	LPUART0_RXD	VC0_OUT	LVD_OUT	VC1_OUT	PA02	PA10	PB02
CTIM0CAP3[6:4]	CTIM0_CH3	USART1_RXD	VC0_OUT	LVD_OUTt	VC1_OUT	PA01	PA09	PB01

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

14.3.2.11 Interrupt

Count overflow, external trigger, and channel capture comparison can all generate interrupts, and the corresponding interrupt can be selected through the corresponding interrupt enable.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

14.4 Register Description

14.4.1 Register List

- BTIM0 base address: 0x40001800
- BTIM1 base address: 0x40001900
- BTIM2 base address: 0x40001A00
- BTIM3 base address: 0x40001C00
- BTIM4 base address: 0x40001D00
- BTIM5 base address: 0x40001E00

Note

BTIM is enabled when SYSCTRL_CR1.GTIMxCFG is set to 1.

Table 14-6 BTIM Register Offset

Offset	Register	Description	Width
0x00	BTIM_ARR	Reload Register	32
0x04	BTIM_CNT	Count Register	32
0x10	BTIM_CR	Control Register	32
0x14	BTIM_IER	Interrupt Enable Control Register	32
0x18	BTIM_IFR	Interrupt Flag Register	32
0x1C	BTIM_ICR	Interrupt Flag Clear Register	32
0x48	BTIM_AIFR	Composite Interrupt Flag Register	32
0x4C	BTIM_AICR	Composite Interrupt Flag Clear Register	32

- GTIM0 base address: 0x40001800
- GTIM1 base address: 0x40001C00

Note

GTIM is enabled when SYSCTRL_CR1.GTIMxCFG is set to 0.

Table 14-7 GTIM Register Offset

Offset	Register	Description	Width
0x300	GTIM_ARR	Reload Register	32
0x304	GTIM_CNT	Count Register	32
0x308	GTIM_CMMR	Compare Capture Control Register	32
0x30C	GTIM_CR1	Control Register 1	32
0x310	GTIM_CR0	Control Register 0	32
0x314	GTIM_IER	Interrupt Enable Control Register	32

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Offset	Register	Description	Width
0x318	GTIM_IFR	Interrupt Flag Register	32
0x31C	GTIM_ICR	Interrupt Flag Clear Register	32
0x320+0x4*y	GTIM_CCRy	CHy Compare Capture Register (y=0-3)	32

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

14.4.2 BTIM Register

14.4.2.1 Reload Register (BTIM_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	Timer reload value	RW

14.4.2.2 Count Register (BTIM_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																	RW															

Bit	Name	Functional Description	Access
31:16	*Reserved*	-	-
15:0	CNT	Timer count value	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

14.4.2.3 Control Register (BTIM_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	0x00000000																															
Name	Reserved																				ETP	TRS		OST	PRS				TOGEN	MD	CEN	
Access																					RW	RW		RW	RW				RW	RW	RW	

Bit	Name	Functional Description	Access
31:12	*Reserved*	-	-
11	ETP	ETR signal polarity selection for external pin input ● 0b0: ETR forward (rising edge active in trigger mode, high level active in gating mode) ● 0b1: ETR reverse (falling edge active in trigger mode, low level active in gating mode)	RW
10:9	TRS	Trigger source selection ● 0b00: Signal from ETR pin input ● 0b01: ITR, see the section "Timer Cascade" ● Others: Setting prohibited **Note** BTIM0-2 share one ETR pin; BTIM3-5 share another ETR pin.	RW
8	OST	Single/continuous counting mode control ● 0b0: Continuous counting mode ● 0b1: Single counting mode	RW
7:4	PRS	Prescaler division ratio configuration ● 0b0000: DIV1 ● 0b0001: DIV2 ● 0b0010: DIV4 ● 0b0011: DIV8 ● 0b0100: DIV16 ● 0b0101: DIV32 ● 0b0110: DIV64 ● 0b0111: DIV128 ● 0b1000: DIV256 ● 0b1001: DIV512 ● 0b1010: DIV1024 ● 0b1011: DIV2048 ● 0b1100: DIV4096 ● 0b1101: DIV8192 ● 0b1110: DIV16384 ● 0b1111: DIV32768	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Bit	Name	Functional Description	Access
3	TOGEN	TOG pin output enable control ● 0b0: TOG, TOGN output levels are both 0 ● 0b1: TOG, TOGN output signals with opposite levels	RW
2:1	MD	Timer working mode configuration ● 0b00: Timer mode, counting clock is PCLK ● 0b01: Counter mode, counting clock is TRS signal ● 0b10: Trigger start mode, counting clock is PCLK; TRS signal triggers counter start ● 0b11: Gating mode, counting clock is PCLK; ETR pin input signal used as gate	RW
0	CEN	Timer enable control ● 0b0: Timer stopped ● 0b1: Timer enabled **Note** After trigger start mode is triggered, CEN is set to 1 by hardware.	RW

14.4.2.4 Interrupt Enable Control Register (BTIM_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																													TI	UI	
Access																														RW	RW	

Bit	Name	Functional Description	Access
31:2	*Reserved*	-	-
1	TI	Trigger interrupt enable control ● 0b0: Trigger interrupt disabled ● 0b1: Trigger interrupt enabled	RW
0	UI	Overflow interrupt enable control ● 0b0: Overflow interrupt disabled ● 0b1: Overflow interrupt enabled	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

14.4.2.5 Interrupt Flag Register (BTIM_IFR)

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																													TI	UI	
Access																														RO	RO	

Bit	Name	Description	Access
31:2	*Reserved*	-	-
1	TI	Trigger flag ● 0b0: No trigger event occurred ● 0b1: A trigger event occurred	RO
0	UI	Counter overflow flag ● 0b0: Counter has not overflowed ● 0b1: Counter overflowed	RO

14.4.2.6 Interrupt Flag Clear Register (BTIM_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	0x00000003																															
Name	Reserved																													TI	UI	
Access																														WO	WO	

Bit	Name	Functional Description	Access
31:2	*Reserved*	-	-
1	TI	Trigger flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
0	UI	Counter overflow flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

14.4.2.7 Composite Interrupt Flag Register (BTIM_AIFR)

Offset	Bit Position																															
0x48	31	30	29	28	27	26	25	24	23	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																										TI25	UI25	TI14	UI14	TI03	UI03
Access																											RO	RO	RO	RO	RO	RO

Bit	Name	Functional Description	Access
31:6	*Reserved*	-	-
5	TI25	BTIM2/5 trigger flag ● 0b0: No trigger event occurred ● 0b1: A trigger event occurred	RO
4	UI25	BTIM2/5 overflow flag ● 0b0: Counter has not overflowed ● 0b1: Counter overflowed	RO
3	TI14	BTIM1/4 trigger flag ● 0b0: No trigger event occurred ● 0b1: A trigger event occurred	RO
2	UI14	BTIM1/4 overflow flag ● 0b0: Counter has not overflowed ● 0b1: Counter overflowed	RO
1	TI03	BTIM0/3 trigger flag ● 0b0: No trigger event occurred ● 0b1: A trigger event occurred	RO
0	UI03	BTIM0/3 overflow flag ● 0b0: Counter has not overflowed ● 0b1: Counter overflowed	RO

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

14.4.2.8 Composite Interrupt Flag Clear Register (BTIM_AICR)

Offset	Bit Position																													
0x4C	31	30	29	28	27	26	25	24	23	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																										TI25	UI25	TI14	UI14	TI03	UI03
Access																											WO	WO	WO	WO	WO	WO

Bit	Name	Functional Description	Access
31:6	*Reserved*	-	-
5	TI25	BTIM2/5 trigger flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
4	UI25	BTIM2/5 overflow flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
3	TI14	BTIM1/4 trigger flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
2	UI14	BTIM1/4 overflow flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
1	TI03	BTIM0/3 trigger flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
0	UI03	BTIM0/3 overflow flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

14.4.3 GTIM Register

14.4.3.1 Reload Register (GTIM_ARR)

Offset	Bit Position																													
0x300	31	30	29	28	27	26	25	24	23	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	Timer reload value	RW

14.4.3.2 Count Register (GTIM_CNT)

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																CNT															
Access																	RW															

Bit	Name	Functional Description	Access
31:16	*Reserved*	-	-
15:0	CNT	Timer count value	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

14.4.3.3 Control Register 1 (GTIM_CR1)

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																									ETRFLT			Reserved			
Access																										RW						

Bit	Name	Functional Description	Access
31:7	*Reserved*	-	-
6:4	ETRFLT	ETR pin input signal filter configuration The sampling clock is PCLK or PCLK frequency division, and the number of sampling points is N ● 0b000: No filter ● 0b001: F_{sample}=PCLK, N=2 ● 0b010: F_{sample}=PCLK, N=4 ● 0b011: F_{sample}=PCLK, N=6 ● 0b100: F_{sample}=PCLK/4, N=4 ● 0b101: F_{sample}=PCLK/4, N=6 ● 0b110: F_{sample}=PCLK/8, N=4 ● 0b111: F_{sample}=PCLK/8, N=6	RW
3:0	*Reserved*	-	-

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

14.4.3.4 Control Register 0 (GTIM_CR0)

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																				ETP	TRS		OST	PRS				TOGEN	MD	CEN	
Access																					RW	RW		RW	RW				RW	RW	RW	

Bit	Name	Functional Description	Access
31:12	*Reserved*	-	-
11	ETP	ETR signal polarity selection for external pin input ● 0b0: ETR forward (rising edge active in trigger mode, high level active in gating mode) ● 0b1: ETR reverse (falling edge active in trigger mode, low level active in gating mode)	RW
10:9	TRS	Trigger source selection ● 0b00: Signal from ETR pin input ● 0b01: Internal interconnection 1 ● 0b10: Internal interconnection 2 ● 0b11: Internal interconnection 3 **Note** Internal interconnection details can be found in the On-chip Peripheral Interconnection section.	RW
8	OST	Single/continuous counting mode control ● 0b0: Continuous counting mode ● 0b1: Single counting mode	RW
7:4	PRS	Prescaler division ratio configuration ● 0b0000: DIV1 ● 0b0001: DIV2 ● 0b0010: DIV4 ● 0b0011: DIV8 ● 0b0100: DIV16 ● 0b0101: DIV32 ● 0b0110: DIV64 ● 0b0111: DIV128 ● 0b1000: DIV256 ● 0b1001: DIV512 ● 0b1010: DIV1024 ● 0b1011: DIV2048 ● 0b1100: DIV4096 ● 0b1101: DIV8192 ● 0b1110: DIV16384 ● 0b1111: DIV32768	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Bit	Name	Functional Description	Access
7:4	PRS	**Note** In capture mode, the prescaler factor can only be set to values 0b000-0b0011.	RW
3	TOGEN	TOG pin output enable control ● 0b0: TOG, TOGN output levels are both 0 ● 0b1: TOG, TOGN output signals with opposite levels	RW
2:1	MD	Timer working mode configuration ● 0b00: Timer mode, counting clock is PCLK ● 0b01: Counter mode, counting clock is TRS signal ● 0b10: Trigger start mode, counting clock is PCLK; TRS signal triggers counter start ● 0b11: Gating mode, counting clock is PCLK, ETR pin input signal as gate control	RW
0	CEN	Timer enable control ● 0b0: Timer stopped ● 0b1: Timer enabled **Note** After trigger start mode is triggered, CEN is set to 1 by hardware.	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

14.4.3.5 Interrupt Enable Control Register (GTIM_IER)

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																				C0TRAD	OVTRAD	DCC3	DCC2	DCC1	DCC0	CC3	CC2	CC1	CC0	TI	UI
Access																					RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Functional Description	Access
31:12	*Reserved*	-	-
11	C0TRAD	Compare capture 0 triggers ADC enable control ● 0b0: Disable ● 0b1: Enable	RW
10	OVTRAD	Overflow triggers ADC enable control ● 0b0: Disable ● 0b1: Enable	RW
9	DCC3	CH3 triggers DMA enable control ● 0b0: Disable ● 0b1: Enable	RW
8	DCC2	CH2 triggers DMA enable control ● 0b0: Disable ● 0b1: Enable	RW
7	DCC1	CH1 triggers DMA enable control ● 0b0: Disable ● 0b1: Enable	RW
6	DCC0	CH0 triggers DMA enable control ● 0b0: Disable ● 0b1: Enable	RW
5	CC3	CH3 capture compare interrupt enable control ● 0b0: Disable ● 0b1: Enable	RW
4	CC2	CH2 capture compare interrupt enable control ● 0b0: Disable ● 0b1: Enable	RW
3	CC1	CH1 capture compare interrupt enable control ● 0b0: Disable ● 0b1: Enable	RW
2	CC0	CH0 capture compare interrupt enable control ● 0b0: Disable ● 0b1: Enable	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Bit	Name	Functional Description	Access
1	TI	Trigger interrupt enable control ● 0b0: Disable ● 0b1: Enable	RW
0	UI	Overflow interrupt enable control ● 0b0: Disable ● 0b1: Enable	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

14.4.3.6 Interrupt Flag Register (GTIM_IFR)

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																										CC3	CC2	CC1	CC0	TI	UI
Access																											RO	RO	RO	RO	RO	RO

Bit	Name	Functional Description	Access
31:6	*Reserved*	-	-
5	CC3	CH3 compare capture interrupt flag ● 0b0: No compare capture event occurred ● 0b1: A compare capture event occurred	RO
4	CC2	CH2 compare capture interrupt flag ● 0b0: No compare capture event occurred ● 0b1: A compare capture event occurred	RO
3	CC1	CH1 compare capture interrupt flag ● 0b0: No compare capture event occurred ● 0b1: A compare capture event occurred	RO
2	CC0	CH0 compare capture interrupt flag ● 0b0: No compare capture event occurred ● 0b1: A compare capture event occurred	RO
1	TI	Trigger interrupt flag ● 0b0: No trigger event occurred ● 0b1: A trigger event occurred	RO
0	UI	Overflow interrupt flag ● 0b0: Counter not overflowed ● 0b1: Counter overflowed	RO

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

14.4.3.7 Interrupt Flag Clear Register (GTIM_ICR)

Offset	Bit Position																													
0x31C	31	30	29	28	27	26	25	24	23	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																										CC3	CC2	CC1	CC0	TI	UI
Access																											WO	WO	WO	WO	WO	WO

Bit	Name	Functional Description	Access
31:6	*Reserved*	-	-
5	CC3	CH3 compare capture interrupt flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
4	CC2	CH2 compare capture interrupt flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
3	CC1	CH1 compare capture interrupt flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
2	CC0	CH0 compare capture interrupt flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
1	TI	Trigger interrupt flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
0	UI	Overflow interrupt flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

14.4.3.8 Compare Capture Control Register (GTIM_CMMR)

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																		CC3M		Reserved		CC2M		Reserved		CC1M		Reserved		CC0M	
Access																			RW				RW				RW				RW	

Bit	Name	Functional Description	Access
31:15	*Reserved*	-	-
14:12	CC3M	CH3 capture compare mode configuration CC0M for function description	RW
11	*Reserved*	-	-
10:8	CC2M	CH2 capture compare mode configuration CC0M for function description	RW
7	*Reserved*	-	-
6:4	CC1M	CH1 capture compare mode configuration CC0M for function description	RW
3	*Reserved*	-	-
2:0	CC0M	CH0 capture compare mode configuration ● 0b000: No function ● 0b001: Rising edge capture ● 0b010: Falling edge capture ● 0b011: Capture on both rising and falling edges ● 0b100: Force output low ● 0b101: Force output high ● 0b110: PWM non-inverted output (outputs high level when CNT≥CCR0) ● 0b111: PWM inverted output (outputs high level when CNT<CCR0)	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

14.4.3.9 Compare Capture Register (GTIM_CCRy) (y=0-3)

Offset: 0x320+0x4*y

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(GTIM_CCR0/GTIM_CCR2); 0x00000000(GTIM_CCR1/GTIM_CCR3)																													
Name	Reserved																CCR													
Access																	RW													

Bit	Name	Functional Description	Access
31:16	*Reserved*	-	-
15:0	CCR	16-bit compare capture register	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

15 LPTIM

15.1 Introduction

LPTIM is an asynchronous 16-bit timer/counter. After the system clock is turned off, it can still timing/counting through the internal low-speed oscillator RCL or the external low-speed crystal XTL, its interrupt signal can wake up the system in low power mode.

15.2 Features

- 16-bit timer counter, DeepSleep mode can work
- 4 types of counting clocks: PCLK, RCL, XTL, ETR pin clock
- Three main functions: timing, counting, and gate control
- Automatic wake up: 30μs-18.2Hour@32.768kHz

15.3 Functional Description

LPTIM is composed of several main functional modules such as clock selection, frequency divider, counter, interrupt, and TOG, and can realize 16-bit timing and counting functions.

15.3.1 Function Block Diagram

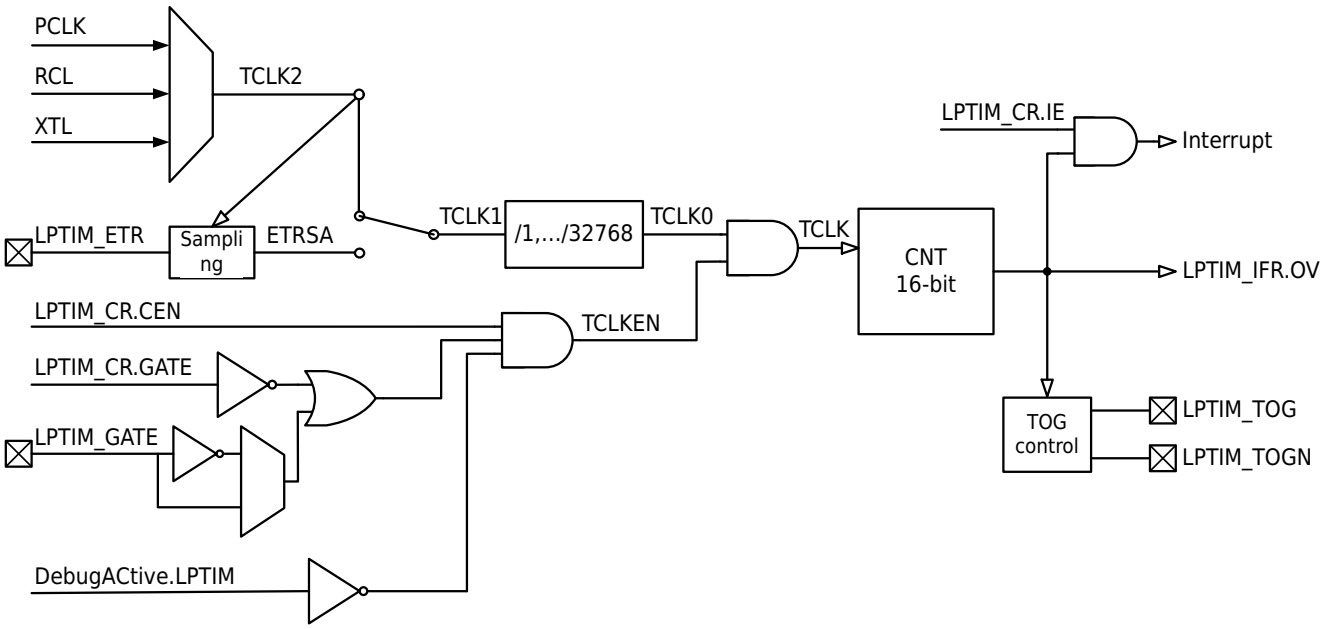


Figure 15-1 Low Power Timer Mode 1

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

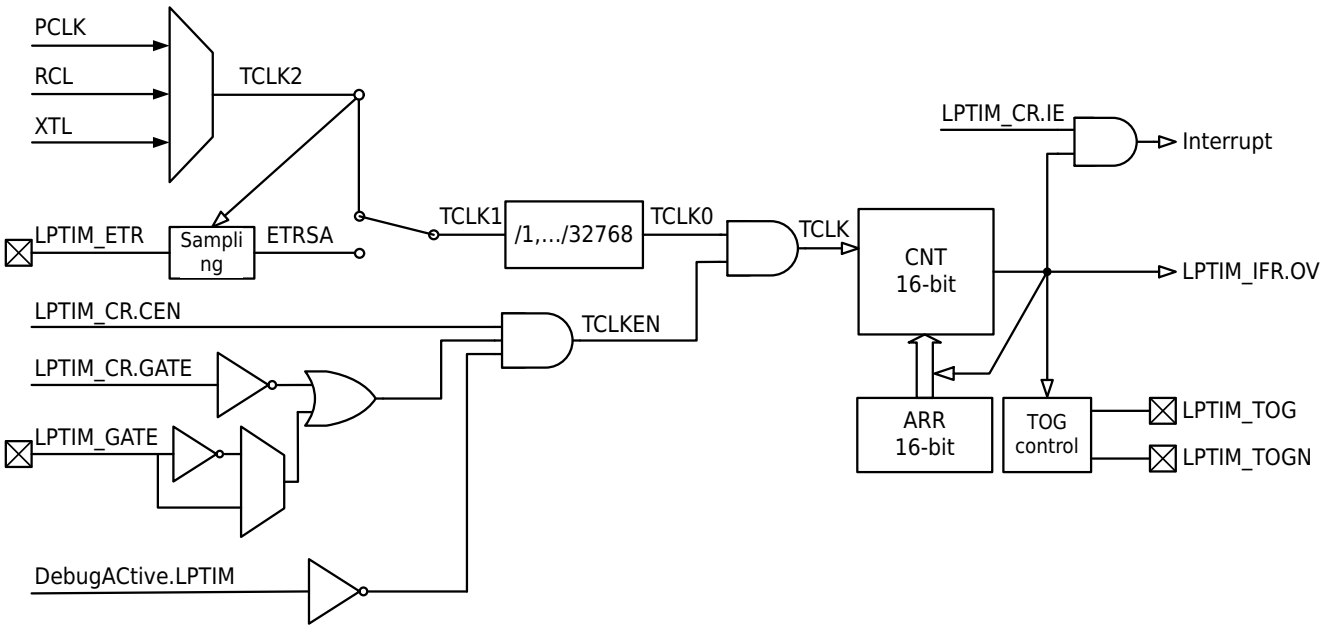


Figure 15-2 Low Power Timer Mode 2

15.3.2 Counting Unit

On each rising edge of TCLK, the internal counter CNT automatically increases by 1, and an overflow signal is generated after counting to 0xFFFF:

- If LPTIM_CR.IE is 1, an overflow interrupt will be generated every time it overflows. The user should clear the interrupt flag in the interrupt service routine.
- If LPTIM_CR.TOGEN is 1, the TOG/TOGN output level will flip once for every overflow.

Users can use the auxiliary function of GPIO to output the TOG/TOGN signal to the pin for driving external circuits.

Counting method:

- When LPTIM works in mode 1, it counts upward from 0x0000 after overflow; its counting period is 65536.
- When LPTIM works in mode 2, it counts upward from ARR after overflow; its counting period is (65536-ARR).

15.3.3 Counting Clock Configuration

Counting clock source selection:

- When setting LPTIM_CR.CT to 0, the counting clock source is determined by LPTIM_CR.TCKSEL (PCLK, RCL, XTL).
- When setting LPTIM_CR.CT to 1, the counting clock source is the clock on the ETR pin.

The selected count clock source TCLK1 is divided by the frequency divider to generate the clock TCLK0, when the TCLKEN signal is 1, TCLK0 is transmitted to the basic counting unit for counting.

If the selected counting clock is RCL, XTL, or ETR, LPTIM can work in DeepSleep mode.

15.3.4 Internal Clock Timing Function

When configuring LPTIM_CR.CT to 0, LPTIM works in timing mode and can generate overflow signals at specified time intervals. The counting unit increments by 1 on each rising edge of TCLK. When the count

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

reaches 0xFFFF, it will overflow and generate an interrupt. When working in mode 2, the timer period of the timer is (65536-ARR).

15.3.5 External Pin Counting Function

When configuring LPTIM_CR.CT to 1, LPTIM works in counting mode and can count the clock input by ETR. The signal input to the ETR pin is sampled by TCLK2 and then enters the frequency divider. The divided clock is used to drive the counting unit to count. Due to the existence of the sampling circuit, the ETR pin input clock frequency should be less than one-half the frequency of the sampling clock TCLK2.

15.3.6 Gating Function

When configuring LPTIM_CR.GATE to 1, LPTIM works in gating mode. The basic counting unit will only count when the level of the GATE pin is as expected. It can be used to measure the width of the high and low levels of the GATE pin or control the running and stopping of the timer by the level of the GATE pin.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

15.4 Register Description

15.4.1 Register List

Base address: 0x40000C00

Table 15-1 Low Power Timer (LPTIM) Register Offset

Offset	Register	Description	Width
0x00	LPTIM_CNT	Counter Count Value Register	32
0x04	LPTIM_ARR	Reload Register	32
0x0C	LPTIM_CR	Control Register	32
0x10	LPTIM_IFR	Interrupt Flag Register	32
0x14	LPTIM_ICLR	Interrupt Flag Clear Register	32

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

15.4.2 Counter Count Value Register (LPTIM_CNT)

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																						CNT									
Access																							RO									

Bit	Name	Functional Description	Access
31:16	*Reserved*	-	-
15:0	CNT	Timer count value	RO

15.4.3 Reload Register (LPTIM_ARR)

Note

This register can only be modified when LPTIM_CR.WTFLAG is 1.

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																						ARR									
Access																							RW									

Bit	Name	Functional Description	Access
31:16	*Reserved*	-	-
15:0	ARR	Timer reload value	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

15.4.4 Control Register (LPTIM_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	0x00000088																															
Name	Reserved																		PRS			IE	GATEP	GATE	WTFLAG	Reserved	TCKSEL	TOGEN	CT	MD	CEN
Access																			RW			RW	RW	RW	RO		RW	RW	RW	RW	RW

Bit	Name	Functional Description	Access
31:15	*Reserved*	-	-
14:11	PRS	Prescaler division ratio configuration ● 0b0000: DIV1 ● 0b0001: DIV2 ● 0b0010: DIV4 ● 0b0011: DIV8 ● 0b0100: DIV16 ● 0b0101: DIV32 ● 0b0110: DIV64 ● 0b0111: DIV128 ● 0b1000: DIV256 ● 0b1001: DIV512 ● 0b1010: DIV1024 ● 0b1011: DIV2048 ● 0b1100: DIV4096 ● 0b1101: DIV8192 ● 0b1110: DIV16384 ● 0b1111: DIV32768	RW
10	IE	Interrupt enable control ● 0b0: Disable ● 0b1: Enable	RW
9	GATEP	Gating polarity control ● 0b0: The counter runs when the GATE pin level is 1 ● 0b1: The counter runs when the GATE pin level is 0	RW
8	GATE	Gating function enable control ● 0b0: Disable gating function ● 0b1: Enable gating function, the effective level is controlled by GATEP	RW
7	WTFLAG	Write register synchronization flag ● 0b0: Synchronizing, the ARR register cannot be changed ● 0b1: Synchronization is completed and the ARR register can be changed	RO
6	*Reserved*	-	-

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Bit	Name	Functional Description	Access
5:4	TCKSEL	Internal count clock selection ● 0b00: PCLK ● 0b01: Setting prohibited ● 0b10: XTL ● 0b11: RCL	RW
3	TOGEN	TOG pin output enable control ● 0b0: TOG, TOGN output levels are both 0 ● 0b1: TOG, TOGN output signals with opposite levels	RW
2	CT	Counter/timer function selection ● 0b0: Timer function, the counting clock is determined by TCKSEL ● 0b1: Counter function, the counting clock comes from the external ETR pin **Note** The frequency of the ETR pin signal should be less than one-half the frequency of the sampling clock TCLK2.	RW
1	MD	Timer working mode selection ● 0b0: Mode 1 no reload mode 16-bit counter/timer ● 0b1: Mode 2 auto-reload 16-bit counter/timer	RW
0	CEN	Timer run control bit ● 0b0: Timer stop ● 0b1: Timer enabled	RW

15.4.5 Interrupt Flag Register (LPTIM_IFR)

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																															OV
Access																																RO

Bit	Name	Functional Description	Access
31:1	*Reserved*	-	-
0	OV	Counter overflow flag ● 0b0: No overflow ● 0b1: Overflow	RO

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

15.4.6 Interrupt Flag Clear Register (LPTIM_ICLR)

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																															OV
Access																																WO

Bit	Name	Functional Description	Access
31:1	*Reserved*	-	-
0	OV	Counter overflow flag clear Write 0 to clear flag, write 1 has no effect; read always returns 0.	WO

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

16 RTC

16.1 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.

16.2 Features

Table 16-1 Basic Features of RTC

Basic Features	Description
Clock source	Off-chip low-speed crystal oscillator XTL (32.768kHz) On-chip low-speed oscillator RCL (32.768kHz, 1% precision) 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
	Intrusion detection (only generates interrupts and set timestamp functions)
Interrupt	Supports periodic interrupts with a minimum period of 0.5 to 32 seconds.
	Alarm interrupt A/B with down-to-the-second precision
	60μs-145Hour wake-up timer interrupt
	Timestamp interrupt

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

16.3 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, and RTC_YEAR 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.

16.3.1 Function Block Diagram

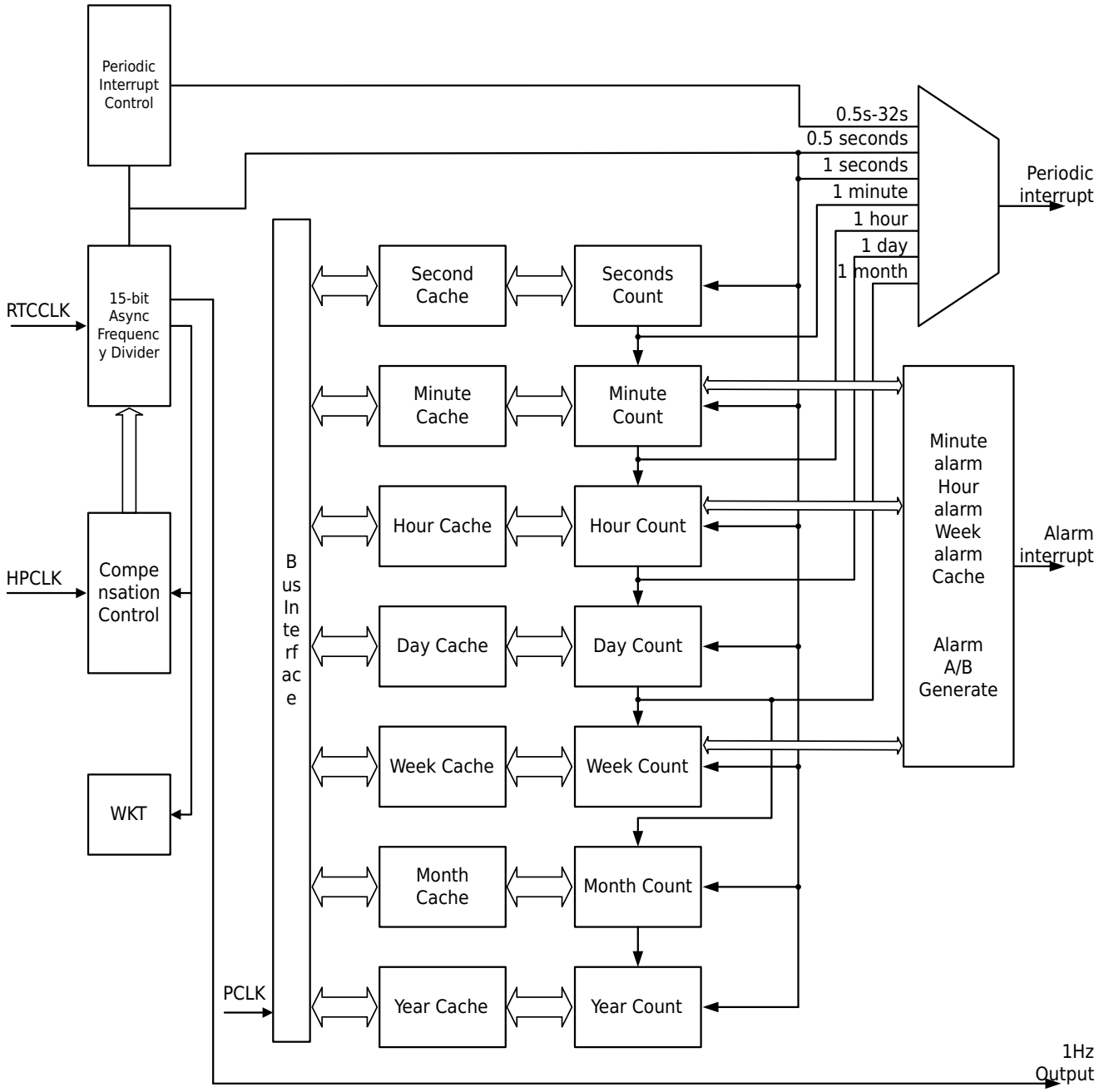


Figure 16-1 RTC Block Diagram

16.3.2 Power-on Setting

The RTC is reset once after power-on. When the system remains powered, no external reset requests can reset the RTC. After power-on, configure the initial calendar values, alarm settings, error compensation, and interrupts, and then start the RTC.

16.3.3 System Low Power Mode Switching

After the RTC starts counting, if the system needs to immediately switch to low power mode, please perform any of the following confirmations before switching modes (controlled via the SYSCTRL_CR1.RTC_LPW register):

- After setting RTC_CR0.START=1, switch modes after more than 2 RTC count clocks.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

- After setting RTC_CR0.START=1, set RTC_CR1.WAIT=1, wait until RTC_CR1.WAITF=1, then set RTC_CR1.WAIT=0, and confirm that RTC_CR1.WAITF=0 before switching modes.

In RTC low power mode, RTC registers cannot be read or written. In low power mode, the RTC consumes less current.

16.3.4 Clock Error Compensation

Due to the inherent error in the external crystal oscillator, compensation is required to achieve high-precision counting results. 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.

Principle and Calculation of Internal Clock Error Compensation

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) ×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:

$$\begin{aligned}\text{CR}[8:0] &= (-20.3 \times 2^{15}/10^6)_{\text{Take 2's complement}} + 0001.00000\text{B} \\ &= (-0.6651904)_{\text{Take 2's complement}} + 0001.00000\text{B} \\ &= 1111.01011\text{B} + 0001.00000\text{B} \\ &= 0000.01011\text{B}\end{aligned}$$

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Principle and Calculation of Error Compensation Based on High-Speed 24MHz Clock:

The calculation method of this method is the same as the error compensation based on its own clock. Due to the introduction of the high-speed clock HPCLK (XTH external high-speed clock), the 1/32768 second error that originally needs to be accumulated within a maximum of 32 seconds can be dispersed to every 1 second, minimum 0.96ppm (23 clock cycles at 24MHz) compensation for every 1 second, achieving an average high-precision 1Hz clock output per second.

16.3.5 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.

16.3.6 Intrusion Detection

Because there are no backup registers, this function only generates interrupts.

16.3.7 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.
- 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.

Note

If the periodic wake-up flag is enabled via RTC_CR2.OSEL, this flag can be connected to the RTC_ALARM output.

16.3.8 RTC Interrupt

RTC supports 4 interrupt types. Alarm A/B interrupt, periodic interrupt, timestamp interrupt, and wake-up timer interrupt.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

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.

16.4 Operation Example

16.4.1 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.

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.

16.4.2 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.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

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.

16.4.3 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.

Step5. Query until RTC_CR1.WAITF=0.

16.4.4 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.

16.4.5 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:

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

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 of 4/6/8/12/16/20/24/32 MHz)**
  - **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.** Configure SYSCTRL_CR1.RTCADJ.
  - **Step6.** Configure RTC_COMPEN.CR[8:0] to set the compensation.
  - **Step7.** Set RTC_COMPEN.EN=1 to enable error compensation.
  - **Step8.** Set RTC_CR0.START=1, counting starts.
  - **Step9.** Wait for more than 2 count cycles.
  - **Step10.** 1Hz output starts.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

16.5 Register Description

16.5.1 Register List

Base address: 0x40001400

Table 16-2 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
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.

16.5.2 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																															
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 ● 0b0: Use the periodic interrupt time interval set by PRDS ● 0b1: Use the periodic interrupt time interval set by PRDX	RW
13:8	PRDX	Set the time interval for generating periodic interrupts The settable range is from 0.5 seconds to 32 seconds, and the step is 0.5 seconds. ● 0b000000: 0.5 seconds ● 0b000001: 1 second ● ..... ● 0b111110: 31.5 seconds ● 0b111111: 32 seconds	RW
7	START	RTC count enable ● 0b0: Stop the RTC count ● 0b1: Enable the RTC count	RW
6	HZ1SEL	1Hz output selection ● 0b0: Normal precision 1Hz output ● 0b1: High precision 1Hz output	RW
5	HZ1OE	1Hz output enable ● 0b0: Disabled 1Hz output ● 0b1: Enable 1Hz output	RW
4	*Reserved*	-	-
3	AMPM	Time format selection ● 0b0: 12-hour time format ● 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 ● 0b000: No periodic interrupt generated ● 0b001: 0.5 seconds ● 0b010: 1 second ● 0b011: 1 minute (At 00 seconds every minute) ● 0b100: 1 hour (At 00:00 every hour) ● 0b101: 1 day (At 00:00:00 every day) ● 0b110/0b111: 1 month (At 00:00:00 on the first day of every month) **Notice** If you need to write and change the time interval of periodic interrupt when START=1, the operation steps are as follows: 1. Turn off the RTC interrupt in the NVIC; 2. Change the time interval of periodic interrupt; 3. Clear the RTC interrupt flag; 4. Enable RTC interrupt.	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

16.5.3 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 ● 0b000/0b001: XTL 32.768k ● 0b010/0b011: RCL 32k ● 0b100: XTH/128 (select this option when the crystal oscillator is 4M) ● 0b101: XTH/256 (select this option when the crystal oscillator is 8M) ● 0b110: XTH/512 (select this option when the crystal oscillator is 16M) ● 0b111: XTH/768 (Select this option when the crystal oscillator is 24M.)	RW
7:2	*Reserved*	-	-
1	WAITF	Read and write wait valid flag ● 0b0: Not write/read status ● 0b1: Write/read status **Notice** 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 ● 0b0: Normal Counting Mode ● 0b1: Write/read mode **Notice** 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.

16.5.4 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 BCD code of second tens digit count value	RW
3:0	SECL	Second count units digit 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.

16.5.5 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 BCD code of minute tens digit count value	RW
3:0	MINL	Minute count units digit 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.

16.5.6 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 BCD code of hour tens digit count value	RW
3:0	HOURL	Hour count units digit 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.

16.5.7 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 BCD code of day tens digit count value	RW
3:0	DAYL	Day count units digit 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.

16.5.8 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 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.

16.5.9 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 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.

16.5.10 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 BCD code of year tens digit count value	RW
3:0	YEARL	Year conut units digit 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.

16.5.11 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																															
Name	Reserved																EN	Reserved												CR		
Access																	RW													RW		

Bit	Name	Functional Description	Access
31:16	*Reserved*	-	-
15	EN	Compensation enable ● 0b0: Disable clock error compensation ● 0b1: Enable clock error compensation	RW
14:9	*Reserved*	-	-
8:0	CR	Compensation value 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.	RW

For more compensation accuracy, see the following table:

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	Uncompensat-ed

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

16.5.12 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																															
Name	Reserved																					ALMBE	ALMAE	Reserved	WUTE	TSE	O2SEL		TSEDG	WUCKSEL		
Access																						RW	RW		RW	RW	RW		RW	RW		

Bit	Name	Functional Description	Access
31:11	*Reserved*	-	-
10	ALMBE	Alarm B enable ● 0b0: Disable ● 0b1: Enable	RW
9	ALMAE	Alarm A enable ● 0b0: Disable ● 0b1: Enable	RW
8	*Reserved*	-	-
7	WUTE	Wake-up timer enable ● 0b0: Disable ● 0b1: Enable	RW
6	TSE	Timestamp enable ● 0b0: Disable ● 0b1: Enable	RW
5:4	O2SEL	RTC_OUT pin output selection ● 0b00: Setting prohibited ● 0b01: Alarm A interrupt flag ● 0b10: Alarm B interrupt flag ● 0b11: Timer interrupt flag	RW
3	TSEDG	Timestamp edge selection ● 0b0: Rising edge ● 0b1: Falling Edge	RW
2:0	WUCKSEL	Wake-up timer clock source selection ● 0b000: RTC/2 ● 0b001: RTC/4 ● 0b010: RTC/8 ● 0b011: RTC/16 ● 0b100: RTC_Sec/1 ● 0b101: RTC_Sec/2 ● 0b110: RTC_Sec/4 ● 0b111: RTC_Sec/8	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

16.5.13 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 Bit24-Bit30 correspond to Sunday-Saturday respectively: When the corresponding bit is set to 1, the alarm is active on that day of the week. For example: ● Setting it to 0b0000001 indicates that the Sunday alarm is enabled ● Setting it to 0b0000011 means the alarm is enabled for Sunday and Monday ● Setting it to 0b1000000 indicates that the Saturday alarm is enabled	RW
23	MSKH	Alarm hour bit mask bit ● 0b0: Alarm A will be generated only when the hour bit match ● 0b1: The hour bit is invalid in alarm A	RW
22	*Reserved*	-	-
21:20	HOURH	Alarm hour tens digit BCD code of the alarm hour tens digit	RW
19:16	HOURL	Alarm hour units digit BCD code of the alarm hour units digit	RW
15	MSKM	Alarm minute bit mask bit ● 0b0: Alarm A will be generated only when the minute bit match ● 0b1: The minute bit is invalid in alarm A	RW
14:12	MINH	Alarm minute tens digit BCD code of the alarm minute tens digit	RW
11:8	MINL	Alarm minute units digit BCD code of the alarm minute units digit	RW
7	MSKS	Alarm second bit mask bit ● 0b0: Alarm A will be generated only when the second bit match ● 0b1: The second bit is invalid in alarm A	RW
6:4	SECH	Alarm seconds tens digit 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 BCD code of the alarm second units digit	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

16.5.14 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 Bit24-Bit30 correspond to Sunday-Saturday, respectively: When the corresponding bit is set to 1, the alarm is active on that day of the week. For example: ● Setting it to 0b0000001 indicates that the Sunday alarm is enabled ● Setting it to 0b0000011 means the alarm is enabled for Sunday and Monday ● Setting it to 0b1000000 indicates that the Saturday alarm is enabled	RW
23	MSKH	Alarm hour mask bit ● 0b0: Alarm B will be generated only when the hour bit matches ● 0b1: The hour bit is invalid in alarm B	RW
22	*Reserved*	-	-
21:20	HOURH	Alarm hour tens digit BCD code of the alarm hour tens digit	RW
19:16	HOURL	Alarm hour units digit BCD code of the alarm hour units digit	RW
15	MSKM	Alarm minute bit shielding bit ● 0b0: Alarm B will be generated only when the minute bit match ● 0b1: The minute bit is invalid in alarm B	RW
14:12	MINH	Alarm minute tens digit BCD code of the alarm minute tens digit	RW
11:8	MINL	Alarm minute units digit BCD code of the alarm minute units digit	RW
7	MSKS	Alarm seconds digit mask bit ● 0b0: Alarm B will be generated only when the second bit match ● 0b1: The second bit is invalid in alarm B	RW
6:4	SECH	Alarm seconds tens digit 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 BCD code of the alarm second units digit	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

16.5.15 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 BCD code of hour tens digit count value **Note** This bit uses 24-hour format and is not affected by RTC_CR0.AMPM.	RO
19:16	HOURL	Hour count units digit BCD code of hour units digit count value **Note** This bit uses 24-hour format and is not affected by RTC_CR0.AMPM.	RO
15	*Reserved*	-	-
14:12	MINH	Minute count tens digit BCD code of minute tens digit count value	RO
11:8	MINL	Minute count units digit BCD code of minute units digit count value	RO
7	*Reserved*	-	-
6:4	SECH	Second count tens digit BCD code of second tens digit count value	RO
3:0	SECL	Second count units digit BCD code of second units digit count value	RO

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

16.5.16 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 Week count value, the format is the same as RTC_WEEK	RO
12:8	MON	Month count value Month count value, the format is the same as RTC_MON	RO
7:6	*Reserved*	-	-
5:4	DAYH	Day count tens digit BCD code of day tens digit count value	RO
3:0	DAYL	Day count units digit BCD code of day units digit count value	RO

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

16.5.17 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

16.5.18 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 ● 0b0: Disable ● 0b1: Enable	RW
5:4	*Reserved*	-	-
3	TS	Timestamp interrupt enable ● 0b0: Disable ● 0b1: Enable	RW
2	WU	Timing interrupt enable ● 0b0: Disable ● 0b1: Enable	RW
1	ALMB	Alarm B interrupt enable ● 0b0: Disable ● 0b1: Enable	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Bit	Name	Functional Description	Access
0	ALMA	Alarm A interrupt enable ● 0b0: Disable ● 0b1: Enable	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

16.5.19 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																															
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 ● 0b0: No interrupt occurred ● 0b1: Interrupt is occurred	RO
5	*Reserved*	-	-
4	TSOV	Timestamp overflow flag ● 0b0: No overflow occurred ● 0b1: Overflow is occurred	RO
3	TS	Timestamp interrupt flag ● 0b0: No interrupt occurred ● 0b1: Interrupt is occurred	RO
2	WU	Timer interrupt flag ● 0b0: No interrupt occurred ● 0b1: Interrupt is occurred	RO
1	ALMB	Alarm B interrupt flag ● 0b0: No interrupt occurred ● 0b1: Interrupt is occurred	RO
0	ALMA	Alarm A interrupt flag ● 0b0: No interrupt occurred ● 0b1: Interrupt is occurred	RO

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

16.5.20 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	0x0000007F																															
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 Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
5	*Reserved*	-	-
4	TSOV	Timestamp overflow flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
3	TS	Timestamp interrupt flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
2	WU	Timing interrupt flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
1	ALMB	Alarm B interrupt flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
0	ALMA	Alarm A interrupt flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

16.5.21 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	0xXXXXXXXX																															
Name	Reserved																								WPR							
Access																									WO							

Bit	Name	Functional Description	Access
31:8	*Reserved*	-	-
7:0	WPR	Protection sequence 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. 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.

17 IWDT

17.1 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 RC32K), and it can still work normally even if the main clock fails.

17.2 Features

- Independent RC oscillator
- Overflow period 125μs-65s
- 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

17.3 Functional Description

17.3.1 Function Block Diagram

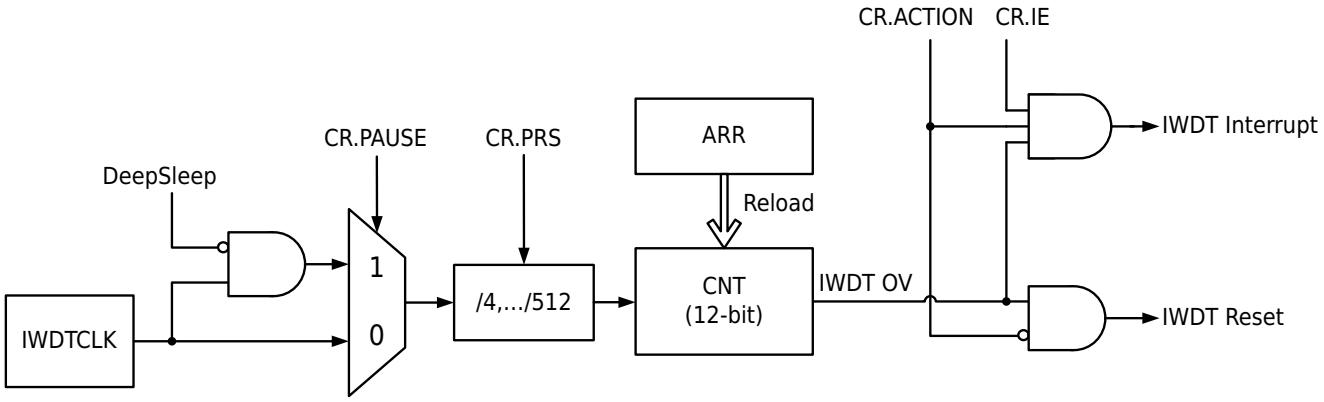


Figure 17-1 IWDT Function Block Diagram

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.

17.3.2 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

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

reload the watchdog and reset the down counter value to IWDT_ARR. The user program should reload 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.

17.3.3 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.

17.3.4 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.

17.3.5 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.

17.3.6 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.

17.3.7 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; writing other values to the IWDT_KR register will enable the write protection of these registers.

17.4 Operation Example

17.4.1 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. Wait for IWDT_SR.PRSF to be 0.

Step5. Configure IWDT_ARR to select the overflow period of the watchdog.

Step6. Wait for IWDT_SR.ARRF to be 0.

Step7. Write the required value to IWDT_WINR to configure the window size.

Step8. Wait for IWDT_SR.WINRF to become 0, and automatically refresh the counter value to IWDT_ARR.

Step9. 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.

17.4.2 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. Wait for IWDT_SR.PRSF to be 0.

Step5. Configure IWDT_ARR to select the overflow period of the watchdog.

Step6. Wait for IWDT_SR.ARRF to be 0

Step7. Refresh the counter value to the value of IWDT_ARR(IWDT_KR=0x0000AAAA).

Step8. Before the count value is decremented to 0, write 0xAAAA to the IWDT_KR register to reload the IWDT.

17.4.3 Watchdog Timeout Time (32kHz Independent Clock)

Table 17-1 IWDT Timeout Time (32kHz 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	0.03125*4=0.125	0.03125*4*4096=512
8 frequency division	0b001	0.03125*8=0.25	0.03125*8*4096=1024
16 frequency division	0b010	0.03125*16=0.5	0.03125*16*4096=2048
32 frequency division	0b011	0.03125*32=1	0.03125*32*4096=4096
64 frequency division	0b100	0.03125*64=2	0.03125*64*4096=8192
128 frequency division	0b101	0.03125*128=4	0.03125*128*4096=16384
256 frequency division	0b110	0.03125*256=8	0.03125*256*4096=32768
512 frequency division	0b111	0.03125*512=16	0.03125*512*4096=65536

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

17.5 Register Description

17.5.1 Register List

Base address: 0x40001080

Table 17-2 Independent Watchdog (IWDT) Register Offset

Offset	Register	Description	Width
0x00	IWDT_KR	Key Value Register	32
0x04	IWDT_CR	Control Register	32
0x08	IWDT_ARR	Reload Register	32
0x0C	IWDT_SR	Status Register	32
0x10	IWDT_WINR	Window Register	32

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

17.5.2 Key Value Register (IWDT_KR)

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																KR															
Access																	WO															

Bit	Name	Functional Description	Access
31:16	*Reserved*	-	-
15:0	KR	IWDT key value ● Write 0xCCCC: Start IWDT counter ● Write 0xAAAA: Reload IWDT count value ● Write 0x1234, 0x5678 in sequence: Stop watchdog ● Write 0x5555: Release the write protection of IWDT_ARR, IWDT_WINR, IWDT_CR ● Write non-0x5555: Enable the write protection of IWDT_ARR, IWDT_WINR, IWDT_CR	WO

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

17.5.3 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.PRSF is 0.

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																									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 ● 0b0: IWDT continues to run in DeepSleep mode ● 0b1: IWDT automatically pauses in DeepSleep mode	RW
5	ACTION	Action configuration after IWDT overflow ● 0b0: Reset is generated after watchdog overflow ● 0b1: Interrupt is generated after watchdog overflow	RW
4	IE	IWDT interrupt enable control ● 0b0: Disable watchdog interrupt ● 0b1: Enable watchdog interrupt	RW
3	*Reserved*	-	-
2:0	PRS	Configure the watchdog count clock and the frequency division ratio of RC32K oscillator ● 0b000: 4-frequency division ● 0b001: 8-frequency division ● 0b010: 16-frequency division ● 0b011: 32-frequency division ● 0b100: 64-frequency division ● 0b101: 128-frequency division ● 0b110: 256-frequency division ● 0b111: 512-frequency division	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

17.5.4 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																													
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	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.

17.5.5 Status Register (IWDT_SR)

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																											RUN	OV	WINRF	ARRF	PRSF
Access																												RO	RW	RO	RO	RO

Bit	Name	Functional Description	Access
31:5	*Reserved*	-	-
4	RUN	WDT running flag ● 0b0: WDT is not running ● 0b1: WDT is running	RO
3	OV	WDT overflow flag Write 0 to clear the overflow flag, write 1 has no effect. Read: ● 0b0: WDT overflow has not occurred ● 0b1: WDT overflow has occurred	RW
2	WINRF	WINR register update flag ● 0b0: WINR register update completed ● 0b1: WINR register is being updated **Notice** The WINR register can only be updated when the WINRF is 0.	RO
1	ARRF	ARR register update flag ● 0b0: ARR register update completed ● 0b1: ARR register is being updated **Notice** The ARR register can only be updated when the ARRF is 0.	RO
0	PRSF	CR register update flag ● 0b0: CR register update completed ● 0b1: CR register is being updated **Notice** The CR register can only be updated when the PRSF is 0.	RO

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

17.5.6 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																													
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	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.

18 WWDT

18.1 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).

18.2 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

18.3 Functional Description

18.3.1 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.

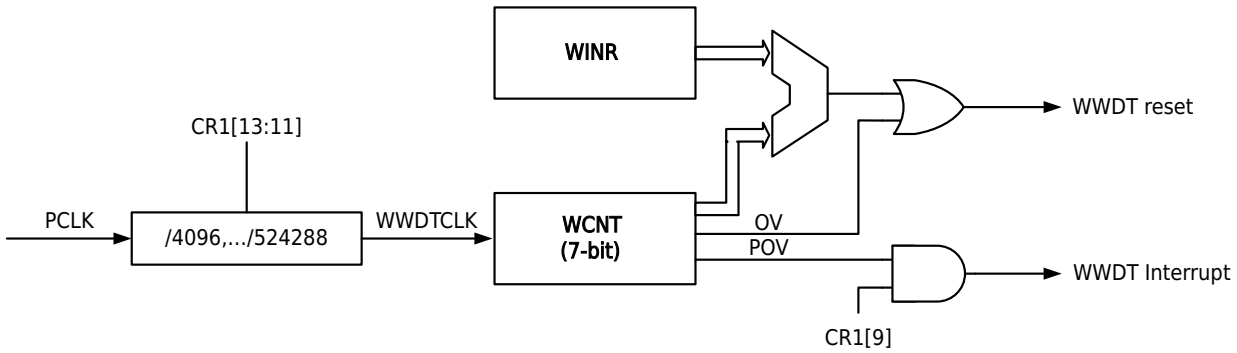


Figure 18-1 WWDT Function Block Diagram

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

18.3.2 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.

18.3.3 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)

18.3.4 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.

18.3.5 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.

18.4 Operation Example

18.4.1 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.

18.5 Register Description

18.5.1 Register List

Base address: 0x40006800

Notice

WWDT affects register value after reset.

Table 18-1 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.

18.5.2 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 Once this bit is set to 1, it cannot be cleared by software ● 0b0: Disable WWDT ● 0b1: Enable WWDT	RW
6:0	WCNT	Read: WWDT counter current count value Write: Update WWDT counter current count value	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

18.5.3 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 ● 0b000: PCLK/4096 ● 0b001: PCLK/8192 ● 0b010: PCLK/16384 ● 0b011: PCLK/32768 ● 0b100: PCLK/65536 ● 0b101: PCLK/131072 ● 0b110: PCLK/262144 ● 0b111: PCLK/524288	RW
10	*Reserved*	-	-
9	IE	WWDT pre-overflow interrupt enable control ● 0b0: Disable pre-overflow interrupt ● 0b1: Enable pre-overflow interrupt	RW
8:7	*Reserved*	-	-
6:0	WINR	Watchdog window value 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.

18.5.4 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 Write 0 to clear the pre-overflow flag, write 1 has no effect. Read: ● 0b0: No pre-overflow has occurred in WWDT ● 0b1: Pre-overflow has occurred in WWDT	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

19 USART

19.1 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.

19.2 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

19.2.1 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

19.2.2 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

19.2.3 Smart Card Interface Features

- Communication protocol: ISO/IEC7816-3
- 8 ETU lengths: 16-512
- Support hardware error detection and retransmission
- Support IO signal edge integer
- Support master interface

19.2.4 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.

19.2.5 LIN Features

- Support synchronous interval transmission and detection
- Support synchronous segment transmission and baud rate automatic detection

19.3 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.

19.3.1 USART General Function

19.3.1.1 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 19-1 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 When external clock source: PCLK/6 (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 19-2 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 19-3 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 19-4 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 19-5 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 19-6 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 19-7 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%					

19.3.1.2 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.

19.3.1.3 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.

19.3.1.4 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.

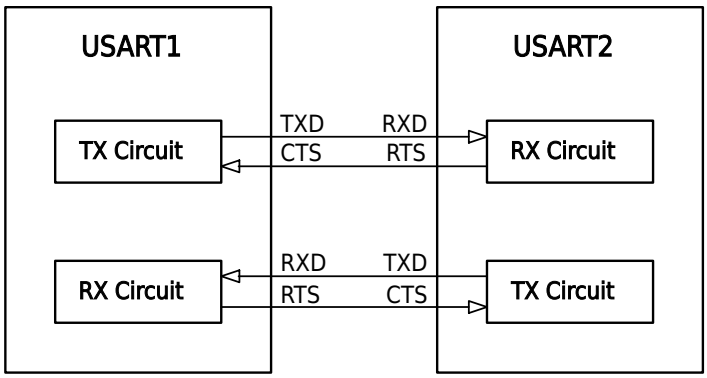


Figure 19-1 Asynchronous Serial Port Mode Hardware Flow Control Example

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.

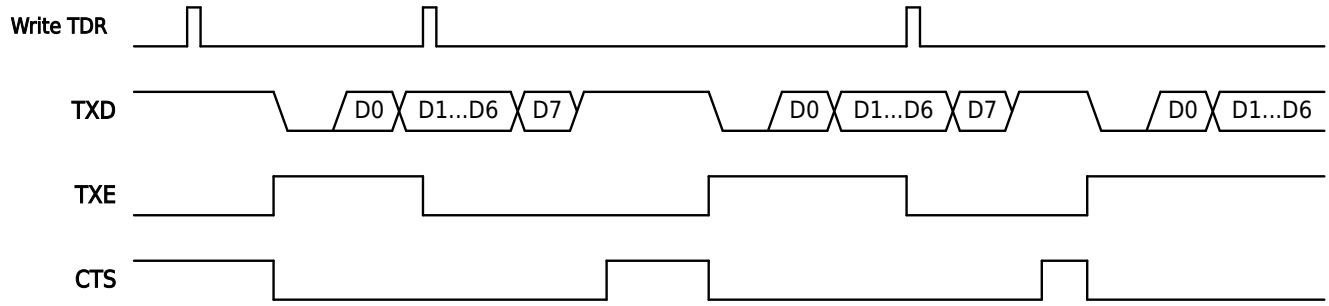


Figure 19-2 CTS Hardware Flow Control Signal

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.

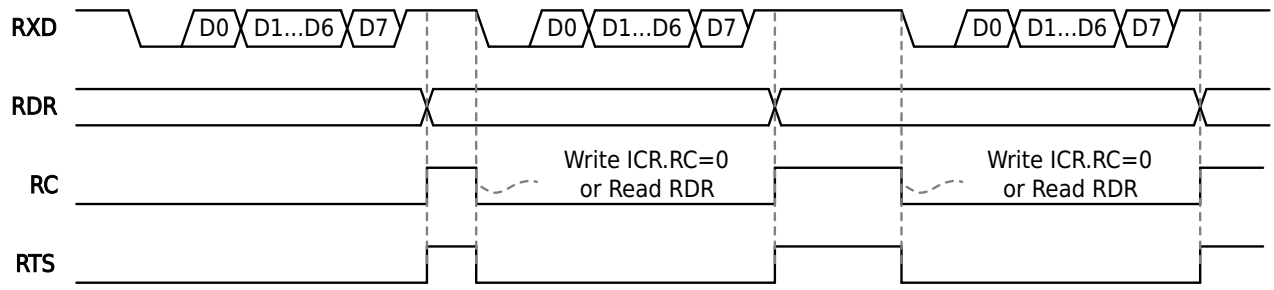


Figure 19-3 RTS Hardware Flow Control Signal

19.3.1.5 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 19-8 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.

19.3.1.6 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 19-9 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	✓	✓	✓	✓	✓
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.

19.3.2 Asynchronous Serial Port Mode

19.3.2.1 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 4 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

19.3.2.2 Function Block Diagram

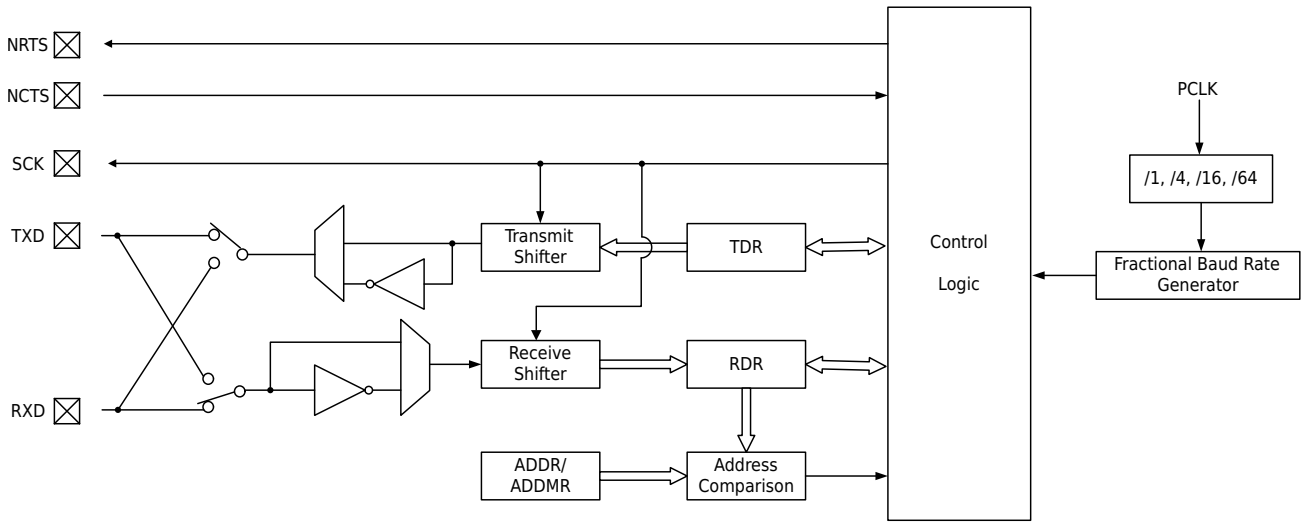


Figure 19-4 Asynchronous Serial Port Function Block Diagram

The function description of each pin is as follows:

- SCK: USART working clock output
- TXD: USART transmit data pins
- RXD: USART receive data pins

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

- NRTS: USART hardware flow control RTS output pin
- NCTS: USART hardware flow control CTS input pin

19.3.2.3 Data Format

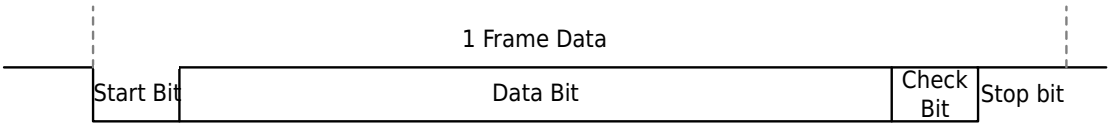


Figure 19-5 Asynchronous Serial Port Mode Data Format

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).

19.3.2.4 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.

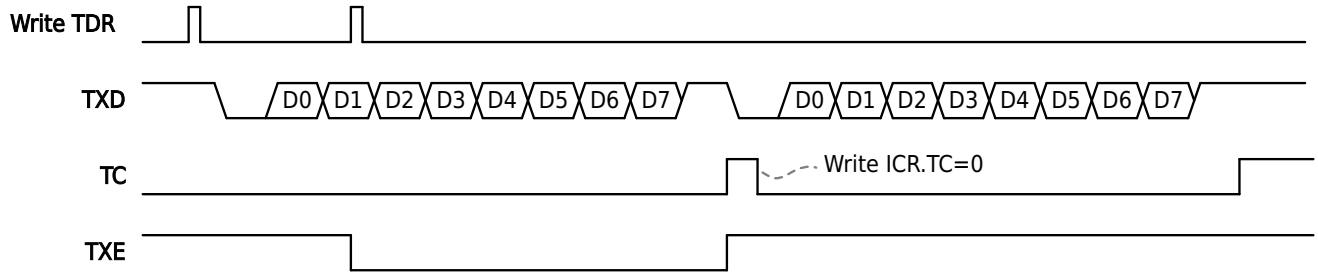


Figure 19-6 Transmit Data Example

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.

19.3.2.5 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.

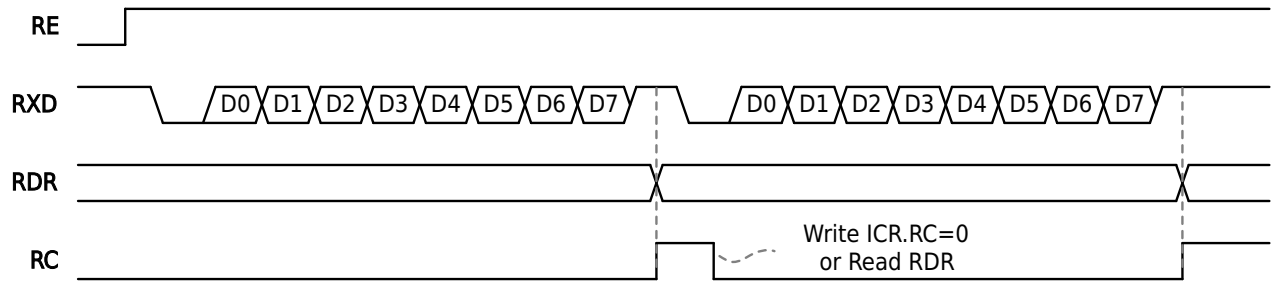


Figure 19-7 Receive Data Example

19.3.2.6 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.

19.3.2.7 Multiprocessor Communication

Note

- Multiprocessor communication is only applicable to asynchronous serial port mode.

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

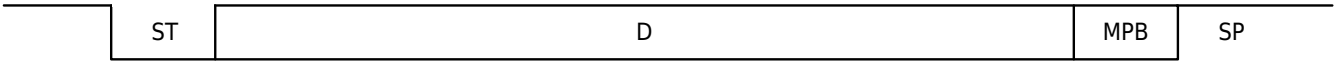


Figure 19-8 Multiprocessor Communication Data Format

- **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..

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

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

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).

19.3.3 Synchronous Serial Port Mode

19.3.3.1 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

19.3.3.2 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.

19.3.3.3 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.

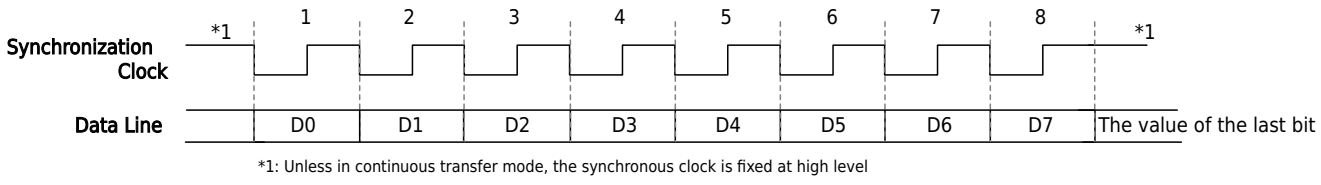


Figure 19-9 Synchronize Serial Port Mode Data Format

19.3.3.4 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 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.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

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

Step1. Set USART_CR1, USART_SR register to reset values;

Step2. Initialize the port. Configure TXD as a digital input with pull-up. Set GPIO_AFRH/AFRL to map TXD to the required pin;

Step3. Select clock source through the USART_CR2.CLKC bit;

Step4. Set USART_CR1 (except TE), USART_CR2, USART_CR3 registers;

Step5. If the internal clock source is selected, it is necessary to set 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.

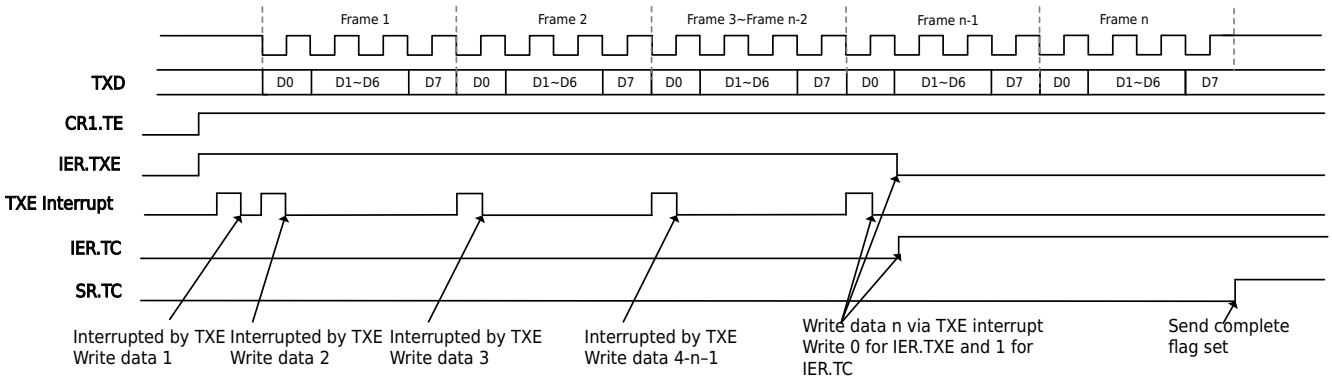


Figure 19-10 Synchronous Serial Port Mode Sending Data Legend 1 (LSB/No CTS Function)

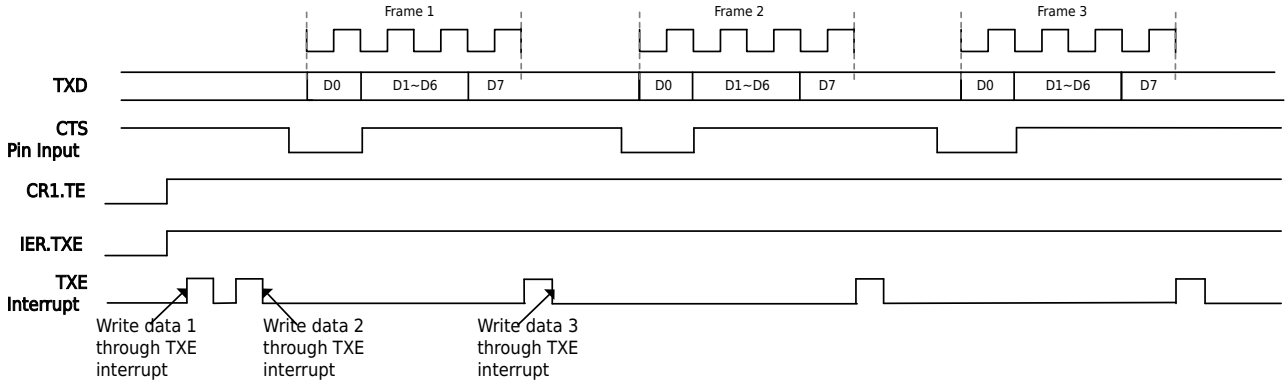


Figure 19-11 Synchronous Serial Port Mode to Send Data Legend 2 (LSB/CTS Function)

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Transmitter Interrupt

The synchronous serial port mode transmitter supports two interrupts, the transmit data register empty interrupt and the transmit complete interrupt.

USART_IER.TXE=1, an interrupt occurs when the value of the USART_TDR register is transferred to the transmit shift register.

USART_IER.TC=1, when the last bit of data is sent, the USART_TDR register is not updated and an interrupt occurs.

19.3.3.5 Receiver

Receive Data Setting Steps

Step1. Set USART_CR1, USART_SR register to reset values;

Step2. Initialize the port. Configure RXD as a digital input with pull-up. Set GPIO_AFRH/AFRL to map RXD 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

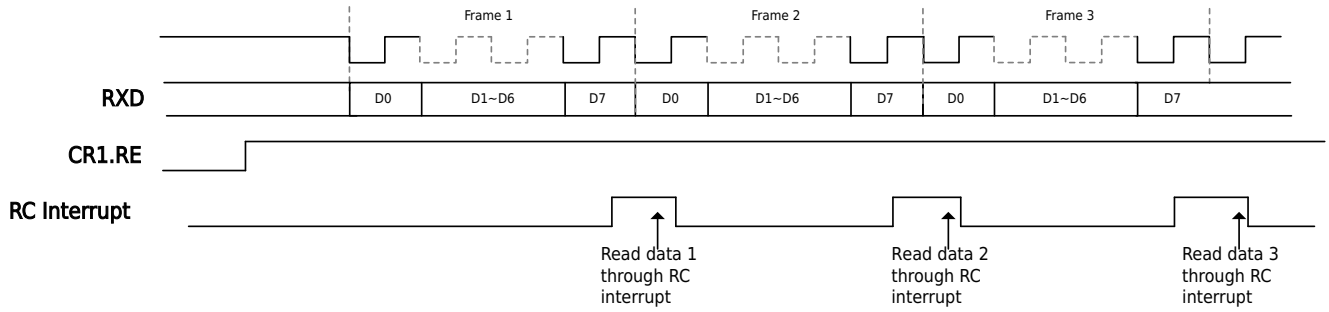


Figure 19-12 Synchronous Serial Port Mode Receiving Data Legend 1 (LSB/No RTS Function)

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

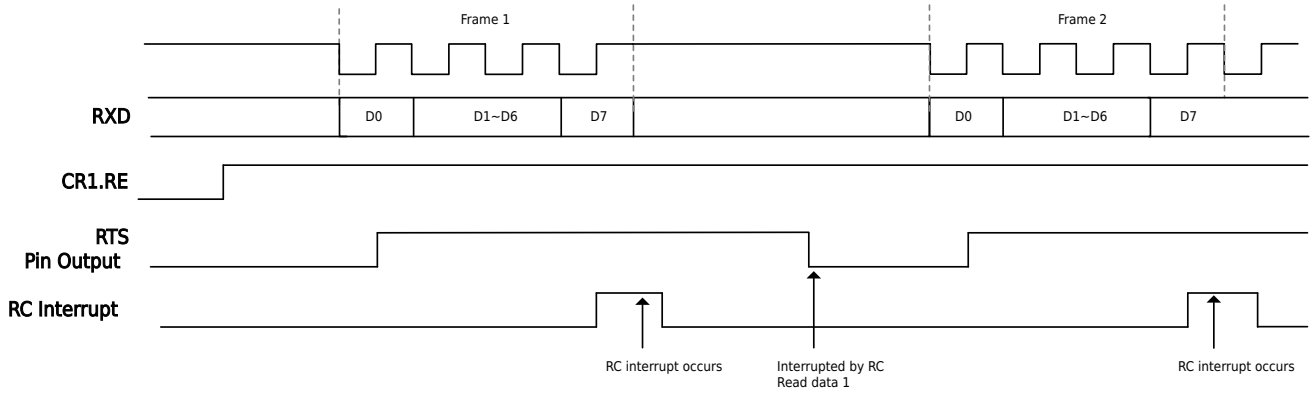


Figure 19-13 Synchronous Serial Port Mode Receiving Data Legend 2 (LSB/RTS Function)

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 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.

19.3.3.6 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.

19.3.3.7 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.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

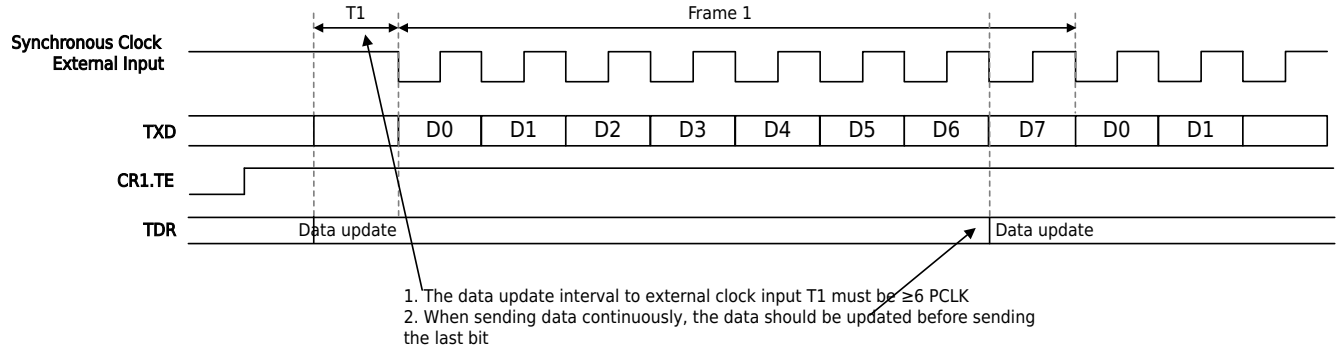


Figure 19-14 Precautions for Synchronous Serial Port Mode

19.3.4 Smart Card (7816) Mode

19.3.4.1 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.

19.3.4.2 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:

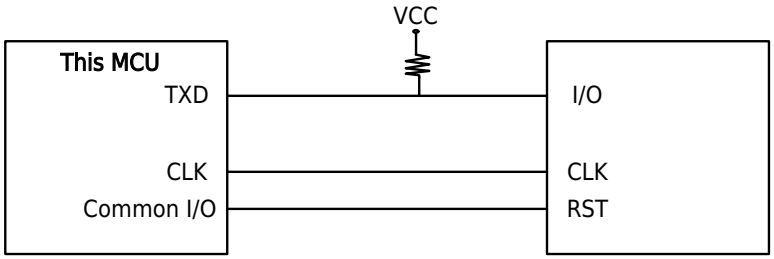


Figure 19-15 Smart Card Connection Schematic Diagram

19.3.4.3 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.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

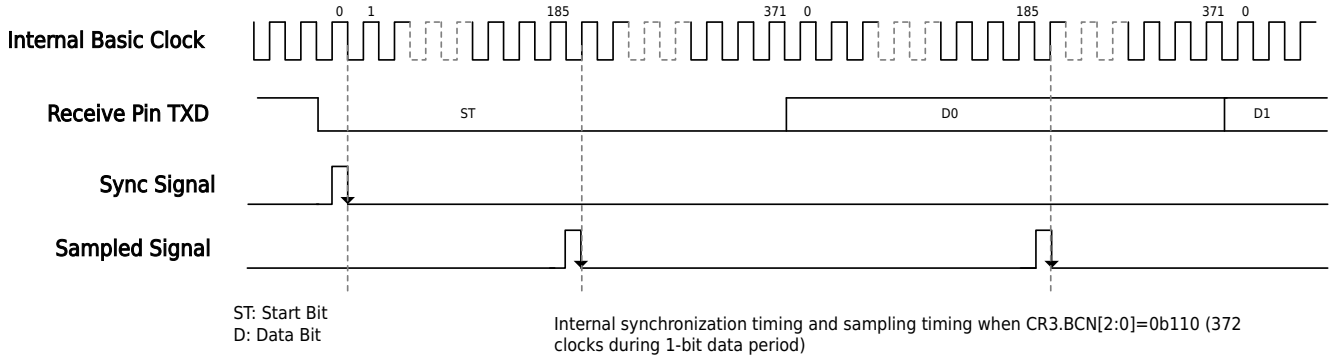


Figure 19-16 Smart Card Mode Internal Synchronization Timing and Sample Timing Diagram

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

19.3.4.4 Data Format

In smart card mode, a frame of data consists of a start bit, data bits, and a check bit.

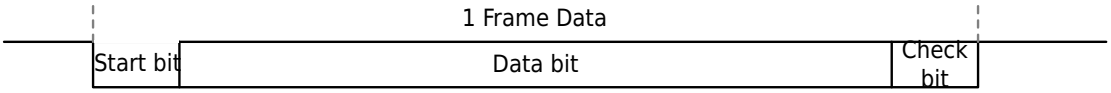


Figure 19-17 Smart Card Mode Data Format

- **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.

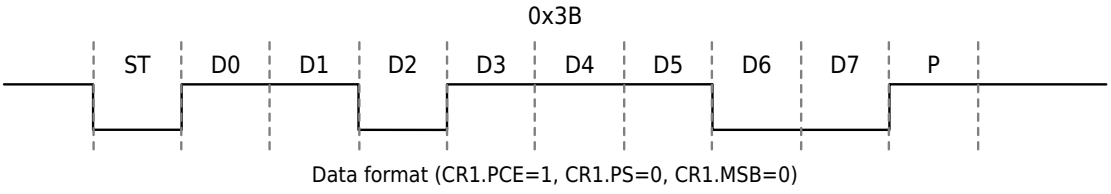


Figure 19-18 Smart Card Mode Data Format Example

19.3.4.5 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;

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

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.

19.3.4.6 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.

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.

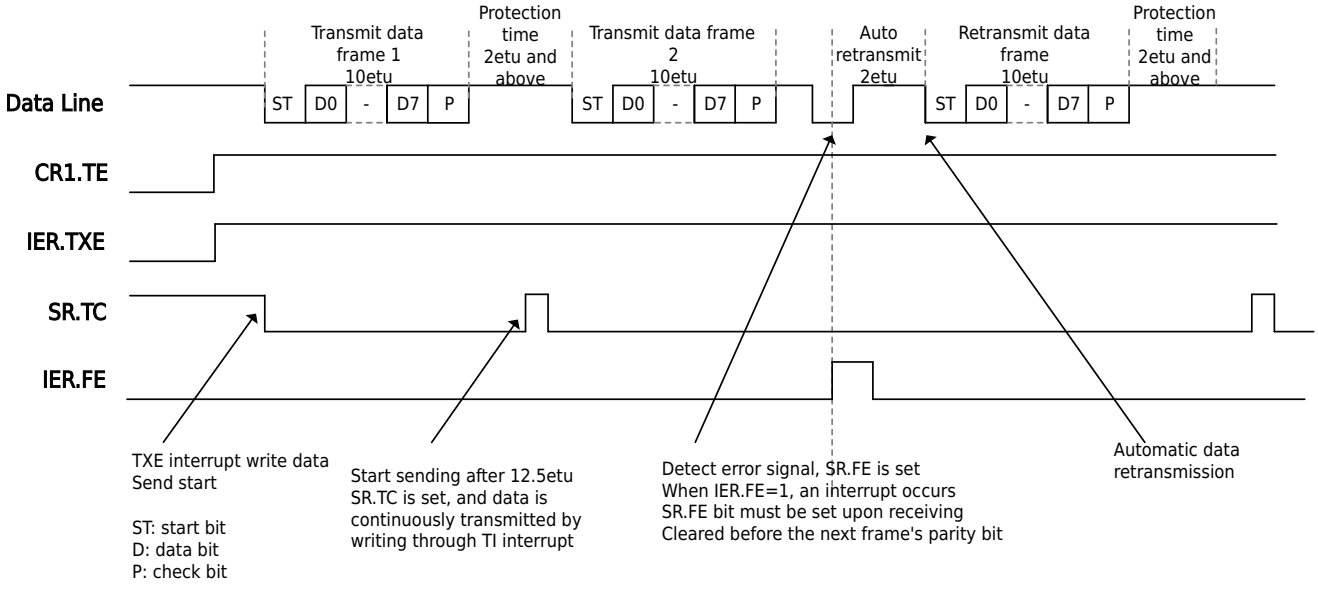


Figure 19-19 Smart Card Mode Transmit Data Legend

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

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.

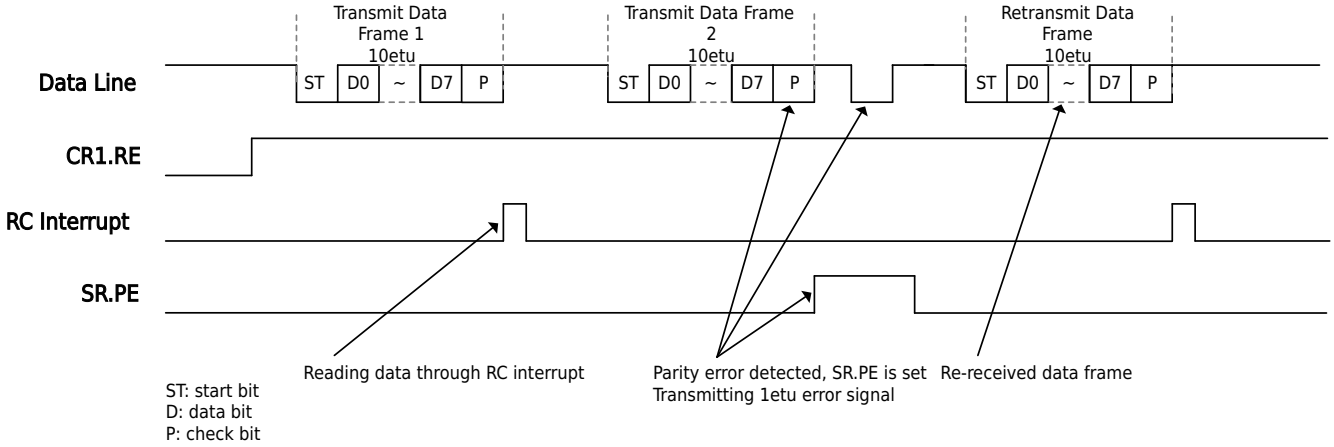


Figure 19-20 Smart Card Mode Receive Data Legend

19.3.5 Infrared Mode

19.3.5.1 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.

19.3.5.2 Infrared Mode 1: Infrared Pulse Width for Baud Rate Period 3/16

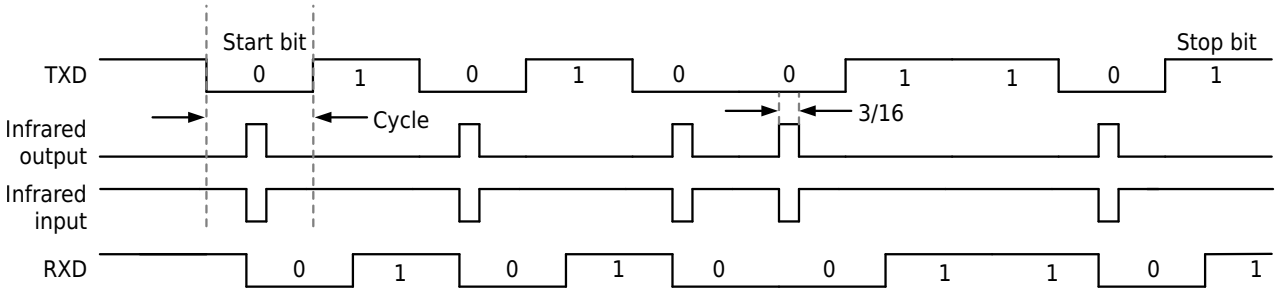


Figure 19-21 Infrared Data Modulation

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

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 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.

19.3.5.3 Infrared Mode 2: 38K Modulation with Adjustable Duty Cycle

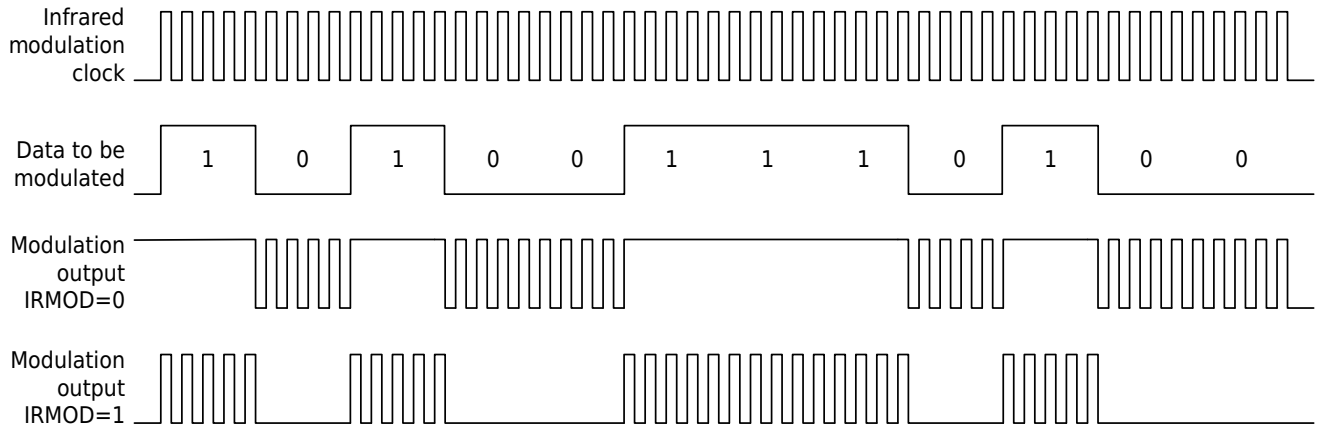


Figure 19-22 38K Infrared Modulation

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

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

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.

19.3.6 LIN Mode

19.3.6.1 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
- 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.

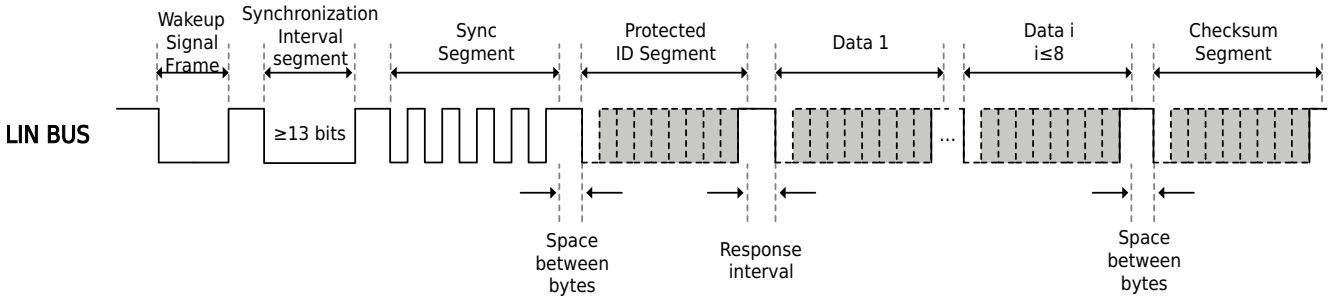


Figure 19-23 LIN Bus Data and Frame Structure

Note

Wake-up signal frame is only required when the device needs to wake up

19.3.6.2 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.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

19.3.6.3 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.

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 set GPIO_AFRH/AFRL to 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.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

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. ......

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.

19.3.6.4 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:

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

- $t_{PCLK:}$ 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".

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 set GPIO_AFRH/AFRL to 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.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

> **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.

19.4 Operation Example

19.4.1 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. Set GPIO_AFRH/AFRL to map TXD to the required pin.

Step2. Set SYSCTRL_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. Set USART_CR1.TE to 1 to enable TXD sending function.

Step10. Write 0x00 to USART_ICR.TC to clear the sending completion flag.

Step11. Write a frame of data to be sent to the USART_TDR register.

Step12. The query waits for USART_SR.TC to become 1, and the data frame written this time is sent.

Step13. If the data to be sent is not completed, then jump to the step10 to continue execution.

19.4.2 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_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 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.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

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.

19.4.3 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.

19.4.4 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.

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.

19.5 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.

19.5.1 Register List

- USART0 base address: 0x40006000
- USART1 base address: 0x40006400

Table 19-10 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.

19.5.2 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 ● 0b0: Normal mode ● 0b1: Silent mode **Note** ● 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. ● 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) ● 0b0: Invalid ● 0b1: Valid **Note** ● The non-asynchronous serial port mode must be set to 0. ● 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 ● 0b0: TXD and RXD are used separately ● 0b1: Asynchronous serial port single-bus mode, TXD and RXD share TXD port **Note** ● The non-asynchronous serial port mode must be set to 0. ● 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 ● 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. ● 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. **Note** ● The non-asynchronous serial port mode must be set to 0. ● 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 ● 0b000: Stop timer counter ● 0b001: General timing function ● 0b010: Receive timing function ● 0b011: Timing function between data frames ● 0b100: In LIN mode, synchronous interval segment counting is enabled ● 0b101: In LIN mode or asynchronous serial port mode, baud rate automatic setection is enabled ● 0b110-0b111: Reserved, the same as 0b000, counting function disabled **Note** ● TOE of all functions are cleared by hardware. ● 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 ● 0b0: 16 times sampling ● 0b1: 8 times sampling **Note** ● Synchronous serial port mode and smart card mode must be set to 0, infrared mode setting is invalid. ● 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 ● 0b0: LSB mode ● 0b1: MSB mode **Note** 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 ● 0b0: 1 stop bit ● 0b1: 2 stop bits **Note** ● Synchronous serial port mode and LIN mode must be set to 0. ● 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 ● 0b0: 8-bit ● 0b1: 9-bit **Note** ● Synchronous serial port mode and LIN mode must be set to 0. ● This bit can only be written when both USART_CR1.TE and USART_CR1.RE are 0.	RW
3	PCE	Parity valid bit ● 0b0: No parity ● 0b1: Verification **Note** ● Synchronous serial port mode and LIN mode must be set to 0. ● This bit can only be written when both USART_CR1.TE and USART_CR1.RE are 0.	RW
2	PS	Parity select bit ● 0b0: Even parity ● 0b1: Odd parity **Note** ● The PS bit is only valid when PCE=1. ● This bit can only be written when both USART_CR1.TE and USART_CR1.RE are 0.	RW
1	TE	Transmit enable bit ● 0b0: Transmit disable ● 0b1: Transmit enable **Note** ● 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. ● 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 ● 0b0: Receive disable ● 0b1: Receive enable **Note** 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.

19.5.3 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 **Asynchronous Serial Port Mode** ● 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 ● 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. ● 0b10/0b11: Setting prohibited **Synchronous Serial Port Mode** ● 0b00/0b01: The clock source is the clock generated by the internal baud rate generator and output to the CK pin. ● 0b10/0b11: Clock source is external input clock, input clock frequency equals baud rate **Smart Card Mode** ● 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 ● 0b01: The clock source is the clock generated by the internal baud rate generator, clock output to the CK pin ● 0b10/0b11: Setting prohibited **Note** ● Infrared mode and LIN mode settings are invalid, neither CK can be output to CK pin. ● 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 ● 0b0: Do not exchange ● 0b1: Exchange **Note** 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 ● 0b0: Asynchronous serial port mode ● 0b1: Synchronous serial port mode **Note** ● Asynchronous serial port mode, smart card mode and LIN mode must write 0. ● Synchronous serial port mode must write 1. ● The infrared mode setting is invalid. ● 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 ● 0b0: Not inverted ● 0b1: TXD output pin signal inversion **Note** 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 ● 0b0: Not inverted ● 0b1: RXD input pin signal inversion **Note** 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 ● 0b0: Disable ● 0b1: Enable **Note** ● Non-asynchronous serial port mode and non-synchronous serial port mode must be set to 0. ● 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 ● 0b0: Disable ● 0b1: Enable **Note** ● Non-asynchronous serial port mode and non-synchronous serial port mode must be set to 0. ● 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 ● 0b0: Turn off hardware handshake signal ● 0b1: Turn on the hardware handshake signal **Note** DMA mode is not supported when working in single-wire half-duplex.	RW
0	RXDMAEN	RX DMAC hardware handshake signal enable bit ● 0b0: Turn off hardware handshake signal ● 0b1: Turn on the hardware handshake signal **Note** DMA mode is not supported when working in single-wire half-duplex.	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

19.5.4 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			SMCPU		BCN		SCEN	BREAK	ABRE	LBDE	LINEN
Access																	RW	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 ● 0b0: Output 38K carrier when data is 0 ● 0b1: Output 38K carrier when data is 1 **Note** Non-infrared mode setting is invalid.	RW
14	IRCW	In infrared mode, infrared carrier selection ● 0b0: 3/16 mode ● 0b1: 38K carrier mode **Note** Non-infrared mode setting is invalid.	RW
13	IREN	In infrared mode, the infrared mode enable bit ● 0b0: Infrared function module off ● 0b1: Infrared function module is turned on **Note** Non-infrared mode must be set to 0.	RW
12:10	*Reserved*	*All bits read as 0, Write as 0*	-
9:8	SMCPU	In smart card mode, output edge shaping ● 0b00: Output without edge shaping when open-drain ● 0b01: When open-drain, strong output CMOS high level for 1 PCLK cycle ● 0b10: When open-drain, strong output CMOS high level for 2 PCLK cycle ● 0b11: When open-drain, strong output CMOS high level for 4 PCLK cycle **Note** The non-smart card mode setting is invalid.	RW

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 ● 0b000: 16 ● 0b001: 32 ● 0b010: 64 ● 0b011: 128 ● 0b100: 186 ● 0b101: 256 ● 0b110: 372 ● 0b111: 512 **Note** The non-smart card mode setting is invalid.	RW
4	SCEN	In smart card mode, the mode enable bit ● 0b0: Smart card mode is disabled ● 0b1: Enable smart card mode **Note** Non-smart card mode must be set to 0.	RW
3	BREAK	In LIN mode, send sync interval enable ● 0b0: The value written to the USART_TDR register represents the data sent normally ● 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. **Note** Non- LIN mode must be set to 0.	RW
2	ABRE	Asynchronous serial port mode/LIN mode, automatic baud rate detection is enabled ● 0b0: Stop detecting baud rate ● 0b1: Start detecting baud rate **Note** 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 ● 0b0: Disable synchronization interval segment detection ● 0b1: Enable synchronization interval segment detection **Note** 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 ● 0b0: Disable LIN Mode ● 0b1: Enable LIN mode **Note** Non- LIN mode must be set to 0.	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

19.5.5 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 ● 0b00: PCLK ● 0b01: PCLK/4 ● 0b10: PCLK/16 ● 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 ● 0b0: Disable ● 0b1: Enable	RW
6:0	DIVF	Fractional frequency divider register 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.

19.5.6 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 Timing counting time = t_{PCLK}* 4^{PSC}* (TOR + 1)	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

19.5.7 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 ● 0b0: No wake-up occurred. ● 0b0: Wake-up occurred. Setting condition: Set to 1 by hardware. Clearing condition: Write 0 to the USART_ICR.DWK bit.	RO
13	LBD	LIN mode, synchronous interval detection flag ● 0b0: No synchronize interval segment detected ● 0b1: Synchronize interval segment detected Setting condition: Set to 1 by hardware. 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 ● 0b0: Detection not completed ● 0b1: Detection completed Setting condition: Set to 1 by hardware. 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 ● 0b0: No CTS signal inversion ● 0b1: CTS signal inversion Setting condition: Set to 1 by hardware. 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 ● 0b0: CTS pin is low level ● 0b1: CTS pin is high level	RO
9	AM	When multiprocessors communicate, the address match flag ● 0b0: Address mismatch ● 0b1: Address match	RO

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Bit	Name	Description	Access
8	MPB	During multiprocessor communication, the multiprocessor flag bit ● 0b0: Received data is data ● 0b1: Received data is ID **Note** 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 ● 0b0: Timer did not match or overflow ● 0b1: Timer has matched or overflowed Setting condition: Set to 1 by hardware. Clearing condition: Write 0 to the USART_ICR.TO bit.	RO
5	OR	Overflow error flag ● 0b0: No overflow ● 0b1: Overflow Setting condition: When the received data register has a value, new data is received. Clearing condition: Write 0 to the USART_ICR.OR bit. **Note** ● RE=0 does not reset the OR bit. ● 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 ● 0b0: No receive frame error ● 0b1: Receive frame error occurred Setting condition: ● Asynchronous serial port mode/infrared mode/LIN mode: the stop bit of the received data frame is 0. ● Smart card mode: sampled low level error signal flag. Clearing condition: Write 0 to the USART_ICR.FE bit. **Note** ● RE=0 does not reset the FE bit. ● 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 ● 0b0: No parity error ● 0b1: Parity error occurred Setting condition: a parity error occurs in the received data. Clearing condition: Write 0 to the USART_ICR.PE bit. **Note** ● RE=0 does not reset the PE bit. ● 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 ● 0b0: Non-receive complete status ● 0b1: Receive completed Setting condition: the stop bit is received, Clearing condition: ● After the data in the USART_RDR register is read by software ● Write 0 to the USART_ICR.PE bit.	RO
1	TC	Send completion flag bit ● 0b0: During data transmission ● 0b1: Data transmission complete Setting condition: ● 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). ● 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). The specific timing for TC set is: 2.5 bit times after the check bit is transmitted. Clearing condition: Write 0 to the USART_ICR.TC bit.	RO
0	TXE	Transmit data register empty flag ● 0b0: Data is not transferred to the shift register ● 0b1: Data is transferred to the shift register Setting condition: After the data is transferred to the shift register, the hardware sets it to 1. 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.

19.5.8 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 ● 0b0: Send data ● 0b1: Transmit ID **Note** The MPID bit is only valid in multiprocessor communication, and must be set to 0 in other modes.	RW
8:0	TDR	Transmit Data **Note** 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.

19.5.9 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 ● 0b0: Disable ● 0b1: Enable **Note** USART_CR2.RXINV=0 corresponds to falling edge, USART_CR2.RXINV=1 corresponds to rising edge.	RW
13	LBD	In LIN mode, synchronous interval segment interrupt enable ● 0b0: Disable ● 0b1: Enable	RW
12	ABR	Interrupt enable for auto baud rate detection in asynchronous serial mode/LIN mode ● 0b0: Disable ● 0b1: Enable	RW
11	CTS	Asynchronous serial port mode/synchronous serial port mode, CTS signal toggle interrupt enable ● 0b0: Disable ● 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 ● 0b0: Disable ● 0b1: Enable	RW
8:7	*Reserved*	*All bits read as 0, Write as 0*	-
6	TO	Timer/counter overflow interrupt enable ● 0b0: Disable ● 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 ● 0b0: Disable ● 0b1: Enable	RW
4	FE	Asynchronous serial port mode/smart card mode/infrared mode/LIN mode, the interrupt enable of receiving frame error ● 0b0: Disable ● 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 ● 0b0: Disable ● 0b1: Enable	RW
2	RC	Interrupt request enable bit after a frame of data is received and written to the receive data register ● 0b0: Disable ● 0b1: Enable	RW
1	TC	Interrupt request enable bit when a frame of data transmission is completed ● 0b0: Disable ● 0b1: Enable	RW
0	TXE	Transmit buffer empty status interrupt enable ● 0b0: Disable ● 0b1: Enable	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

19.5.10 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 Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
13	LBD	LBD flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
12	ABR	ABR flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
11	CTS	CTS flag clear 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 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 Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
5	OR	OR flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
4	FE	FE flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
3	PE	PE flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
2	RC	RC flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
1	TC	TC flag clear 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.

19.5.11 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 **Note** The highest bit RDR[8] is valid only in the asynchronous serial port mode and the data length is set to 9 bits.	RO

19.5.12 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 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.

19.5.13 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 ● 0b0: Corresponding bit does not participate in the matching. ● 0b1: Corresponding bit participates in the matching. **Note** ● Slave Address=USART_ADDR.ADDR & USART_ADDMR.ADDM. ● This bit can only be written when both USART_CR1.TE and USART_CR1.RE are 0.	RW

19.5.14 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 The communication baud rate can be calculated by the following formula: $Baud = \frac{f_{PCLK}}{4^{PSC} \times (BDR+1)}$ **Note** Only applicable to asynchronous serial port mode/LIN mode.	RO

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

19.5.15 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 38K high level hold time t = t_{PCLK}* (DCNTH + 1)	RW
15:11	*Reserved*	-	-
10:0	DCNTL	In infrared mode, low level count value 38K low level hold time t = t_{PCLK}* (DCNTL + 1)	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

20 LPUART

20.1 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.

20.2 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.

20.3 Functional Description

20.3.1 Function Block Diagram

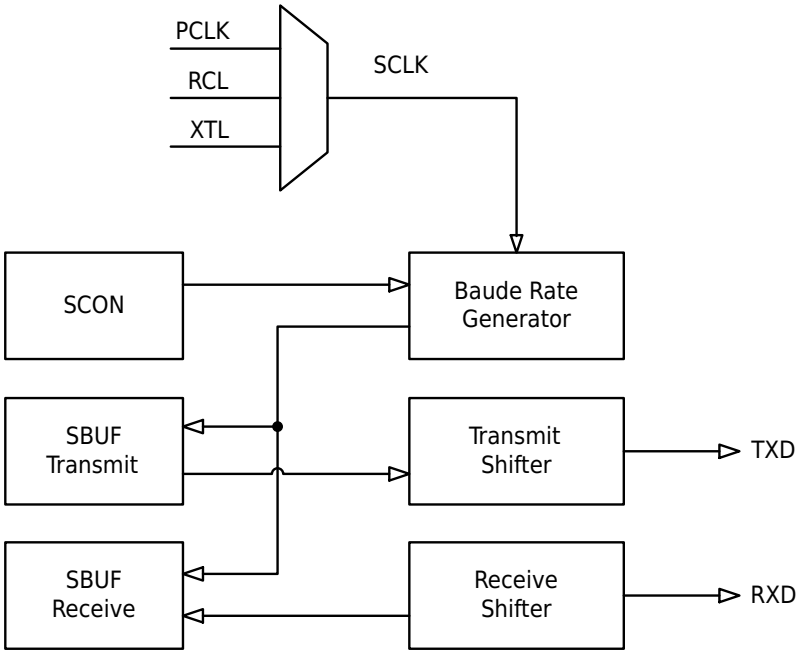


Figure 20-1 LPUART Function Block Diagram

20.3.2 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.

20.3.3 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.MODE with LPUART_SCON.HDSEL, various required working modes can be configured.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

20.3.3.1 Mode0-Mode3 Function Comparison

Configure LPUART_SCON.MODE to select different transmission modes: Mode0-Mode3. The main function comparison of these four working modes is shown in the table below:

Table 20-1 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)	$BaudRate = \frac{f_{SCLK}}{OVER \times SCNT}$

- 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.
- The definition of OVER is detailed in register LPUART_SCON introduction.
- The definition of SCNT is detailed in register LPUART_SCNT introduction.
- DR8 data bit is special and has different meanings in different applications, please refer to the following table:

Table 20-2 DR8 Data Meaning

Application Scenario	LPUART_SCON.ADRDET	LPUART_SCON.DR8CFG[1:0]	DR8 Data Meaning
Parity check	-	0b01/0b10	When receiving, DR8 is the parity check bit of the 8bits of data received. When transmitting, DR8 is the parity check bit of the 8bits of data transmitted.
Multi-machine communication	0b1	-	DR8=1, it means that it is currently an address frame. DR8=0, it means that it is currently a data frame.
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 DR8CFG.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

20.3.3.2 Mode0 Data Transmitting and Reception

- Mode0 Data Transmitting

  When LPUART_SCON.RXEN 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.

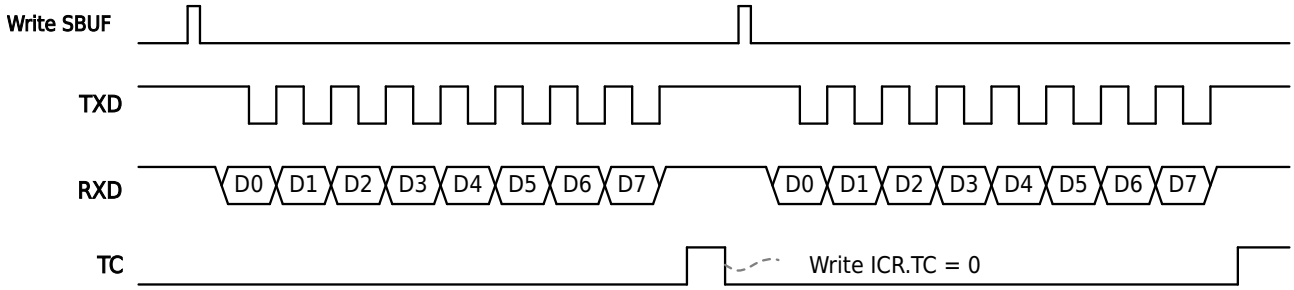


Figure 20-2 Mode0 Data Transmitting

- Mode0 Data Receiving

  When LPUART_SCON.RXEN 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), the LPUART_ISR.RC flag is cleared, the next byte can be received.

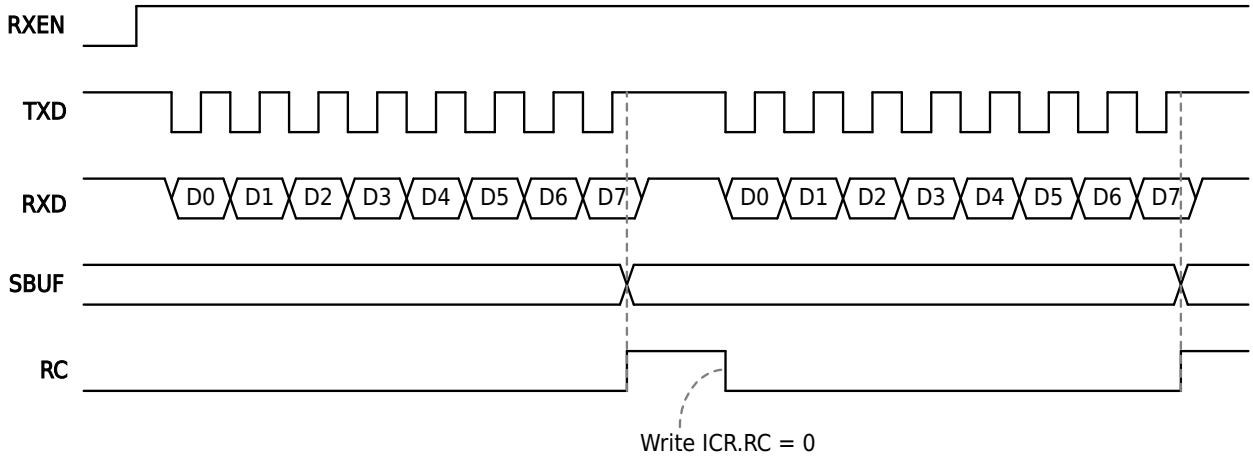


Figure 20-3 Mode0 Data Receiving

20.3.3.3 Mode1 Data Transmitting and Reception

- Mode1 Data Transmitting

  Whether LPUART_SCON.RXEN 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.

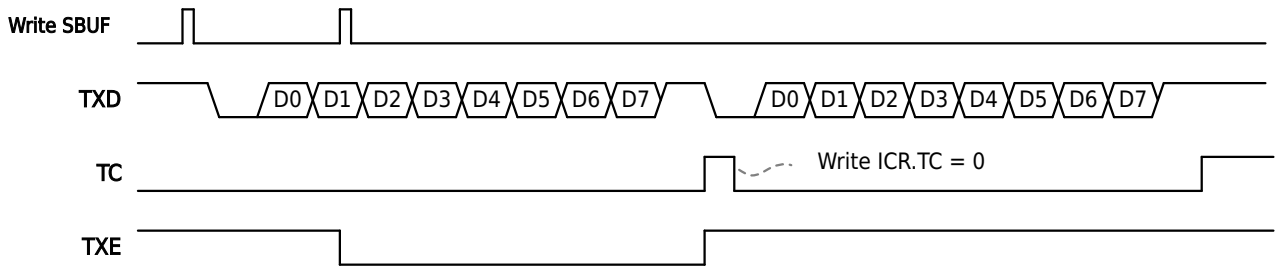


Figure 20-4 Mode1 Data Transmitting

- Mode1 Data Receiving

  When receiving data, set LPUART_SCON.RXEN 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.

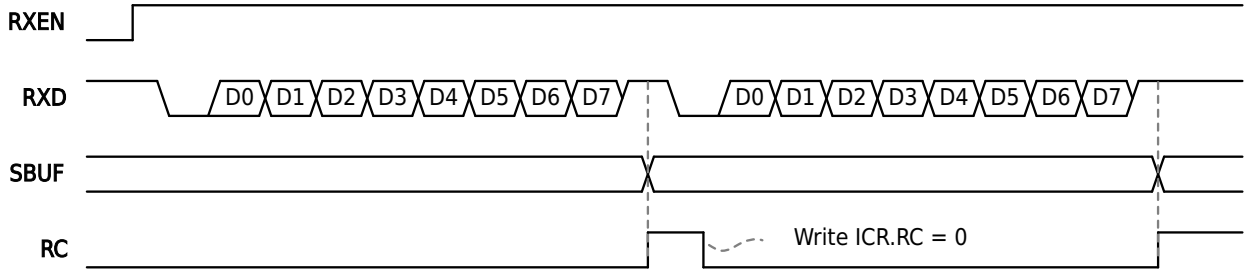


Figure 20-5 Mode1 Data Receiving

20.3.3.4 Mode2/Mode3 Data Transmitting and Reception

- Mode2/3 Data Transmitting

  Whether LPUART_SCON.RXEN 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.

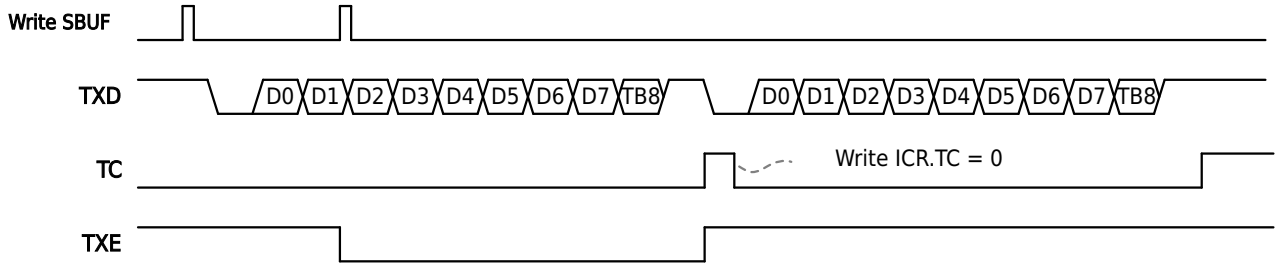


Figure 20-6 Mode2/Mode3 Data Transmitting

- Mode2/3 Data Receiving

  When receiving data, set LPUART_SCON.RXEN to 1.

  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.

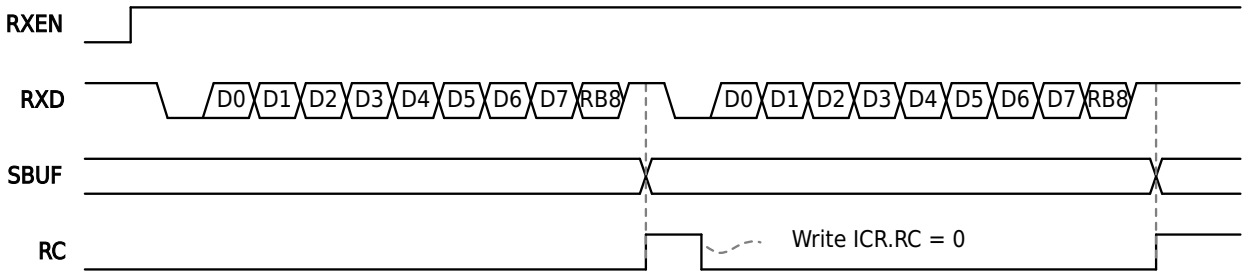


Figure 20-7 Mode2/Mode3 Data Receiving

20.3.3.5 Single-Wire Half-Duplex Data Transmitting and Reception

When the single bus function control bit LPUART_SCON.HDSEL is enabled, LPUART only uses one TXD pin to transmit and receive data. The user needs to configure the TXD pin as an open-drain output and connect an external pull-up resistor.

- When no data is transmitted, 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 LPUART_SBUF register, the TXD pin changes to the output state in order to send the data written by the user to the register in LPUART_SBUF to the bus.
- 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.

To prevent collisions on a single bus, the user should take appropriate application-layer protection measures to ensure that multiple masters do not transmitted data to the bus at the same time.

20.3.4 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.
- The definition of OVER is detailed in register LPUART_SCON introduction.
- The definition of SCNT is detailed in register LPUART_SCNT introduction.

Mode1/Mode3 Baud Rate Setting Example

Table 20-3 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%

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

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%
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	/	/

Table 20-4 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 20-5 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	/	/

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Table 20-6 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%
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 20-7 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 20-8 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%

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%
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%
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	/	/

20.3.5 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.

20.3.6 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.DR8 is 1, the address frame is transmitted, LPUART_SBUF.DR is the address of the target slave.

When LPUART_SBUF.DR8 is 0, the data frame is transmitted, LPUART_SBUF.DR 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

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

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).

20.3.7 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:

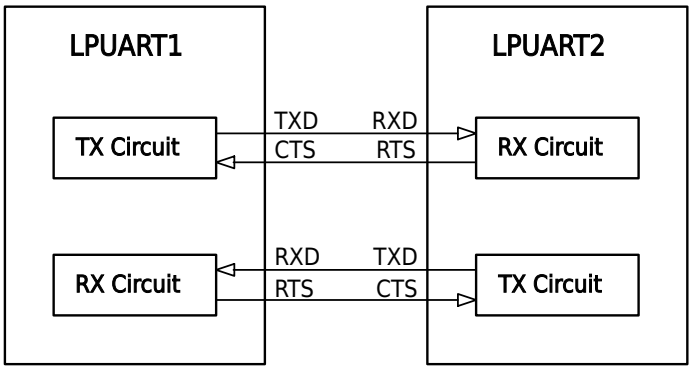


Figure 20-8 LPUART Hardware Flow Control

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.

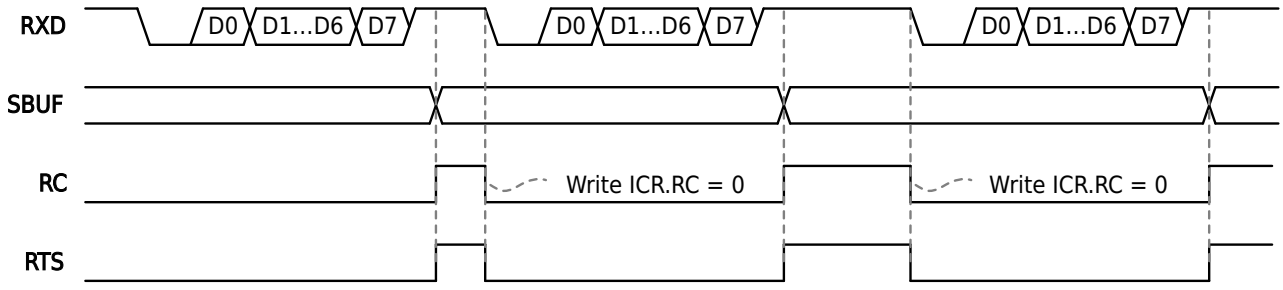


Figure 20-9 RTS Hardware Flow Control Signal

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.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

- 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.

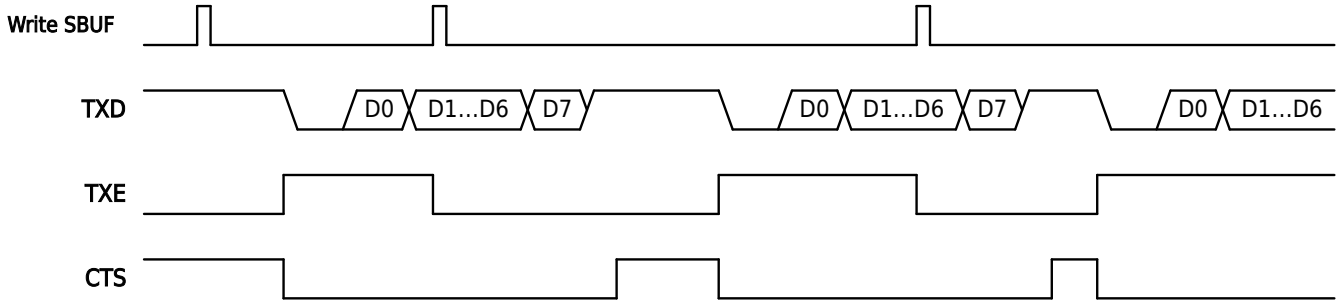


Figure 20-10 CTS Hardware Flow Control Signal

20.3.8 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 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.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

20.4 Operation Example

20.4.1 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_PERI_CLKEN0.LPUARTx to 1 to enable LPUART configuration clock and working clock.

Step3. Configure LPUART_SCON.CLKSRC to select the transmission clock source for LPUART.

Step4. Configure LPUART_SCON.MODE 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.STOP, LPUART_SCON.DR8CFG, 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.TC to clear the transmitting completion flag.

Step11. If the data to be transmitted is not completed, then jump to step8.

20.4.2 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_PERI_CLKEN0.LPUARTx to 1 to enable LPUART configuration clock and working clock.

Step3. Configure LPUART_SCON.CLKSRC to select the transmission clock source for LPUART.

Step4. Configure LPUART_SCON.MODE 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.STOP, LPUART_SCON.DR8CFG, 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.RXEN 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.RC 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.

20.5 Register Description

20.5.1 Register List

- LPUART0 base address: 0x40000000
- LPUART1 base address: 0x40004000

Table 20-9 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

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

20.5.2 Data Register (LPUART_SBUF)

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																							DR8	DR							
Access																								RW	RW							

Bit	Name	Functional Description	Access
31:9	*Reserved*	-	-
8	DR8	DR8 Data Bit ● In Mode 0/1, read this bit is 0, write this bit is invalid ● Under Mode 2/3, there are such situations: ▶ When parity bit is turned on: When receiving data, this bit serves as the parity check bit for the received data; when transmitting data, this bit is invalid. ▶ When the parity bit is turned off, this bit is receive/transmit data Bit8. **Note** 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 DR8CFG.	RW
7:0	DR	Data 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.

20.5.3 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	STOP		PEIE	CLKSRC		OVER	TXEIE	MODE	ADRDET	RXEN	DR8CFG		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 ● 0b0: Disable ● 0b1: Enable	RW
21	FEIE	Frame error interrupt enable ● 0b0: Disable ● 0b1: Enable	RW
20	CTSIE	CTS signal flip interrupt enable ● 0b0: Disable ● 0b1: Enable	RW
19	CTSEN	CTS signal enable control ● 0b0: Disable ● 0b1: Enable	RW
18	RTSEN	RTS signal enable control ● 0b0: Disable ● 0b1: Enable	RW
17	DMATXEN	DMA transmit data enable control ● 0b0: Disable ● 0b1: Enable	RW
16	DMARXEN	DMA receive data enable control ● 0b0: Disable ● 0b1: Enable	RW
15:14	STOP	STOP duration configuration ● 0b00: 1-bit ● 0b01: 1.5-bit ● 0b10: 2-bit ● 0b11: Setting prohibited **Notice** 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 ● 0b0: Disable ● 0b1: Enable	RW
12:11	CLKSRC	Transmission clock SCLK source selection ● 0b00: PCLK ● 0b01: PCLK ● 0b10: XTL ● 0b11: RCL	RW
10:9	OVER	Oversampling control ● Mode0: Invalid ● Mode1/3: ▶ 0b00: 16 sampling frequency division ▶ 0b01: 8 sampling frequency division ▶ 0b10: 4 sampling frequency division ▶ 0b11: Setting prohibited ● Mode2: ▶ 0b00: 32 sampling frequency division ▶ 0b01: 16 sampling frequency division ▶ 0b10: 8 sampling frequency division ▶ 0b11: Setting prohibited	RW
8	TXEIE	TX Buf empty interrupt enable ● 0b0: Disable ● 0b1: Enable	RW
7:6	MODE	Working mode ● 0b00: Mode0, synchronous half-duplex, Data (8bits) ● 0b01: Mode1, asynchronous full-duplex, Start (1bit) + Data (8bits) + Stop (1-2bits) ● 0b10: Mode2, asynchronous full-duplex, Start (1bit) + Data (8bits) + B8 (1bit) + Stop (1-2bits) ● 0b11: Mode3, asynchronous full-duplex, Start (1bit) + Data (8bits) + B8 (1bit) + Stop (1-2bits)	RW
5	ADRDET	Multi-machine address automatic recognition enable ● 0b0: Disabled, during transmission/reception, the function of LPUART_SBUF.DR8 is determined by DR8CFG ● 0b1: Enabled, during reception, address match detection is performed	RW
4	RXEN	Receive enable ● Mode0: ▶ 0b0: Transmit ▶ 0b1: Receivie ● Mode1/2/3: ▶ 0b0: Transmit ▶ 0b1: Receive/Transmit	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Bit	Name	Functional Description	Access
3:2	DR8CFG	DR8 data bit function configuration ● 0b00: Deciding by software read/write LPUART_SBUF.DR8 ● 0b01: Hardware even parity check ● 0b10: Hardware odd parity check ● 0b11: Setting prohibited	RW
1	TCIE	Transmit complete interrupt enable ● 0b0: Disable ● 0b1: Enable	RW
0	RCIE	Receive complete interrupt enable ● 0b0: Disable ● 0b1: Enable	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

20.5.4 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

20.5.5 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 Each bit corresponds to an address bit: ● 0b0: Does not participate in address matching ● 0b1: Participates in address matching	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

20.5.6 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 ● 0b0: CTS signal is low level ● 0b1: CTS signal is high level	RO
5	CTSIF	CTS signal flip flag ● 0b0: CTS signal does not flip ● 0b1: CTS signal fliped	RO
4	PE	Parity check error flag ● 0b0: No parity errors occurred ● 0b1: A parity error occurred	RO
3	TXE	Tx Buffer empty flag ● 0b0: Tx Buffer is not empty ● 0b1: Tx Buffer is empty	RO
2	FE	Frame error flag ● 0b0: No frame error occurred ● 0b1: A frame error occurred	RO
1	TC	Transmit completion flag ● 0b0: Transmission not completed ● 0b1: Transmission completed	RO
0	RC	Receive completion flag ● 0b0: Reception not completed ● 0b1: Reception completed	RO

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

20.5.7 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																										CTSIF	PE	Reserved	FE	TC	RC
Access																											WO	WO		WO	WO	WO

Bit	Name	Functional Description	Access
31:6	*Reserved*	-	-
5	CTSIF	ISR.CTSIF flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
4	PE	ISR.PE flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
3	*Reserved*	-	-
2	FE	ISR.FE flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
1	TC	ISR.TC flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
0	RC	ISR.RC flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO

20.5.8 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.

21 CAN

21.1 Introduction

CAN (Controller Area Network) bus is a bus standard that enables communication between microprocessors or devices without a host.

The CAN bus controller complies with the CAN bus protocol 2.0A and 2.0B and is upward compatible with CAN-FD.

The CAN bus controller can handle data transmission and reception on the bus. In this product, the CAN has 8 filter groups. Filters are used to select messages for the application to receive.

In the CAN bus controller, applications can send data to the bus through 1 high-priority Primary Transmit Buffer (PTB) and 4 Secondary Transmit Buffers (STB), with the transmission sequence determined by the transmit scheduler. Bus data is received through 10 Receive Buffers (RB). The 4 STBs and 10 RBs can be understood as a 4-stage FIFO and an 10-stage FIFO, both fully controlled by hardware.

CAN bus controllers can also support Time-trigger communication.

21.2 Features

- Fully supports CAN2.0A/CAN2.0B protocols
- Upward compatible with CAN-FD
- Supports a maximum communication baud rate of 1Mbit/s
- Supports baud rate prescaler of 1 to 1/256 for flexible baud rate configuration
- 10 receive buffers
  - FIFO mode
  - Messages stored will not be overwritten by erroneous or unreceived data
- 1 high-priority PTB
- 4 STB
  - FIFO Mode
  - Priority arbitration method
- 8 independent filter groups
  - Support 11-bit standard ID and 29-bit extended ID
  - Programmable ID CODE bits and MASK bits
- Both PTB and STB support single send mode
- Supports silent mode
- Supports loopback mode
- Supports capturing transmission error types and locating bit arbitration lost bits
- Programmable error warning value
- Supports ISO11898-4 defined time-trigger CAN and receive timestamps

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

21.3 Functional Description

21.3.1 System Block Diagram

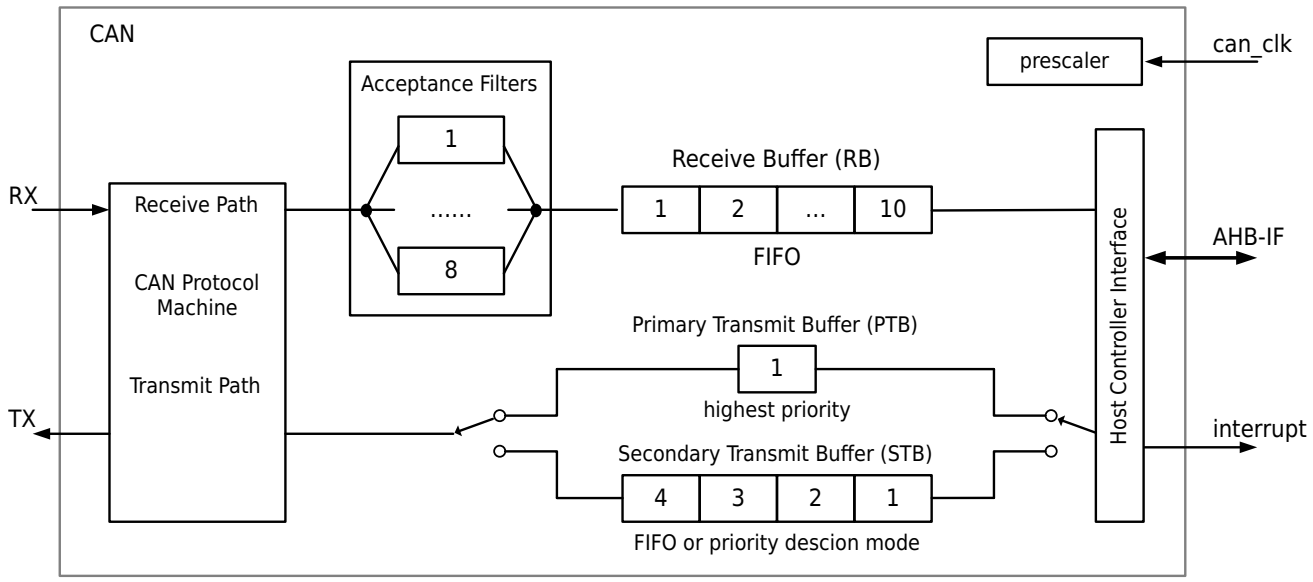


Figure 21-1 CAN System Block Diagram

21.3.2 Pin Description

Table 21-1 CAN Pin Description

Pin Name	Direction	Description
CAN_RxD	Input	CAN receive data signal
CAN_TxD	Output	CAN transmit data signal
CAN_STBY	Output	External transceiver low-power mode control signal

21.3.3 Operating Mode

21.3.3.1 Action Mode

The CAN controller has two operating modes: reset mode (CAN_CFG_STAT.RESET=1) and action mode (CAN_CFG_STAT.RESET=0). When initializing the module, first set the registers that can only be operated in reset mode (see the Register Description section for details) in reset mode, then exit reset mode and operate the remaining registers in action mode.

21.3.3.2 Loopback Mode

CAN_CTRL supports the following two loopback modes:

- Internal loopback
- External loopback

Both loopback modes can receive data frames sent by themselves, mainly used for testing purposes.

Internal loopback mode: Internally connects the receive data line to the transmit data line within the module, and transmit data is not output. In internal loopback mode, the node generates a self-acknowledge signal to avoid ACK errors.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

External loopback mode maintains the connection with the transceiver, so the transmitted data can still appear on the CAN bus. With the help of the transceiver, CAN can receive its own transmitted data. In external loopback mode, the self-acknowledge signal can be generated or not by setting the SACK bit in the RCTRL register. When SACK=0, no self-acknowledge signal is generated; when SACK=1, a self-acknowledge signal is generated.

In external loopback mode with SACK=0, the following two situations may occur:

- Other nodes also receive the data frame sent by this node and send an ACK signal. In this case, the node can successfully send and receive data.
- If no other node returns an ACK signal, an ACK error will occur, causing the data to be retransmitted and the error counter to be incremented. In this case, it is recommended to use single send mode.

When returning from loopback mode to normal mode, in addition to clearing the mode bit, it is also necessary to perform a software reset of CAN_CTRL.

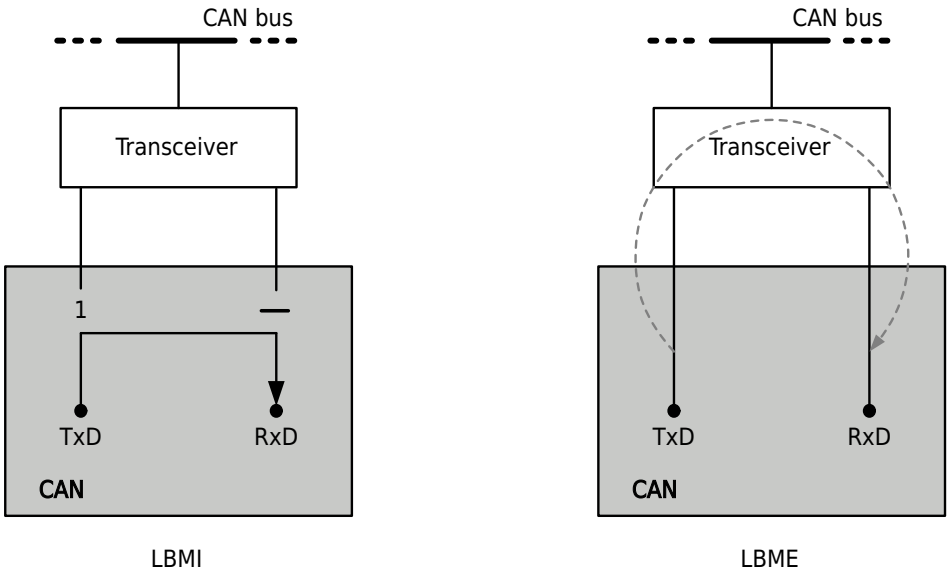


Figure 21-2 CAN LBMI and LBME Schematic Diagram

21.3.3.3 Silent Mode

Silent mode can be used to monitor CAN network data. In silent mode, data can be received from the CAN bus, but no data is sent to the bus. Set TCMD.LOM to 1 to enter silent mode, and clear it to exit silent mode.

External loopback mode can be combined with silent mode to form external loopback silent mode. In this mode, CAN can be considered a silent receiver but can send data when necessary. In external loopback silent mode, frames containing self-acknowledge signals are allowed to be sent, but the node will not generate error flags or overload frames.

21.3.4 Baud Rate Setting

The clock source for CAN communication clock (can_clk) is HCLK. Before using the CAN module, it is necessary to set the CAN communication clock in the SYSCTRL chapter. CAN 2.0B supports bit rates up to 1Mbit/s.

The diagram below shows the CAN bit time definition. The upper part above the dashed line represents the bit time specified by the CAN protocol, and the lower part below the dashed line represents the bit time defined by the CAN-CTRL. Where, segment1 and segment2 can be set through the CAN_BT

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

registers. The CAN_BT registers can only be set when CAN_CFG_STAT.RESET=1, i.e., during CAN software reset.

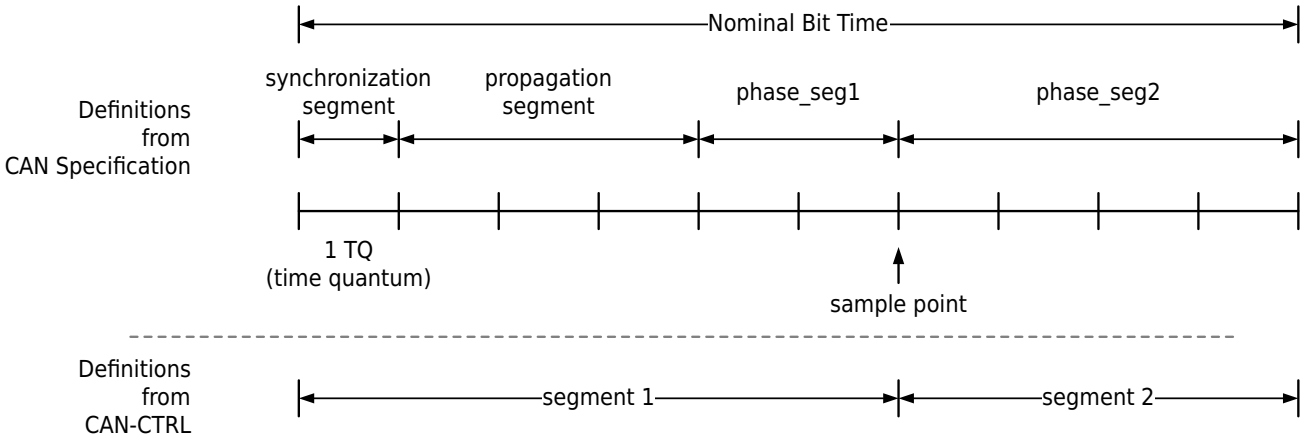


Figure 21-3 CAN Bit Time Definition Diagram

In the diagram, the synchronization segment, propagation segment, and phase_seg1 are combined into segment 1, with its time length represented by t_{SEG_1}; phase_seg2 is segment 2,with its time length represented by t_{SEG_2}.

The CAN bit time (BT) consists of several segments as shown in the diagram, with each segment containing several time quanta TQ, and the total length of BT is n_{TQ}.

Please refer to the following formula for the TQ calculation method, where PRESC is set through the PRESC bit in the BT register. f_{can_clk} is the CAN communication clock frequency.

$$\text{Time Quanta} = \frac{\text{PRSEC}+1}{f_{can\_clk}}$$

Refer to the following formula for the bit time calculation method, where SEG_1 and SEG_2 are set through the SEG_1 bit and SEG_2 bit in the SBT register.

$$BT = t_{SEG\_1} + t_{SEG\_2} = ((SEG\_1 + 2) + (SEG\_2 + 1)) \times TQ$$

Table 21-2 CAN Bit Time Setting Rules

Bit	Predetermined Area	Rule
BT.SEG_1	0-63	SEG_1≥SEG_2 + 1 SEG_2≥SJW
BT.SEG_2	0-7	
BT.SJW	0-7	

21.3.5 Transmission Processing

21.3.5.1 Transmit Buffer

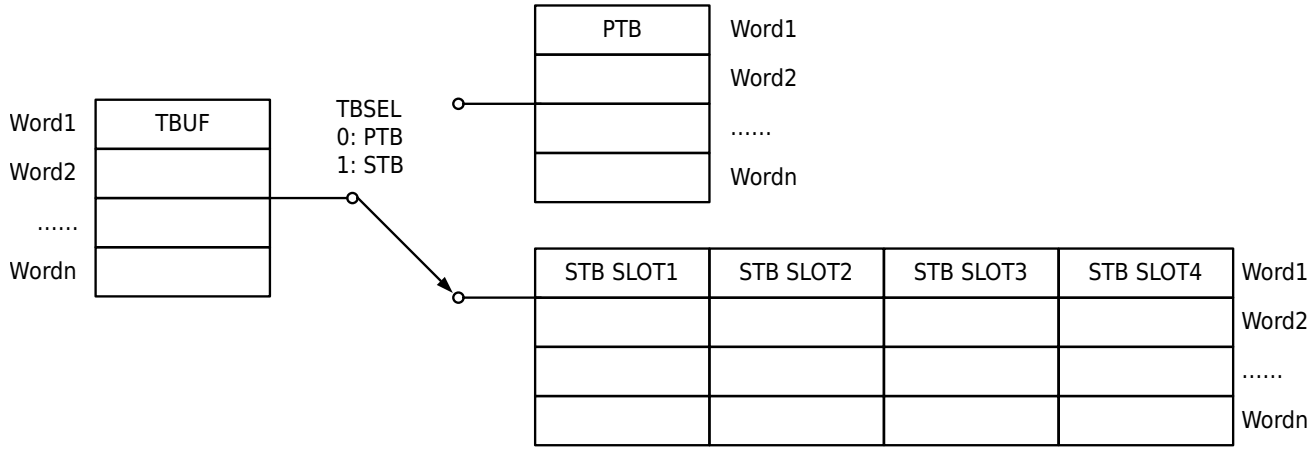
CAN_CTRL provides two transmit buffers for sending data: the Primary Transmit Buffer (PTB) and the Secondary Transmit Buffer (STB). PTB has the highest priority but can only buffer one frame of data. STB has a lower priority than PTB but can buffer 4 frames of data, and the 4 frames in STB can operate in FIFO mode or priority arbitration mode.

The 4 frames in STB can be set to send all at once by setting the TSALL bit in the TCMD register to 1. In FIFO mode, the data written first is sent first. In priority mode, data with a smaller ID is sent first.

Data in PTB has the highest priority, so PTB transmission can delay STB transmission, but STB that has won arbitration and started transmission cannot be delayed by PTB transmission.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

PTB and STB each have one frame of data occupying 4 words, which can be accessed through the TBUF register. Select PTB or STB by setting the TBSEL bit in the TCMD register. TBSEL=0 selects PTB; TBSEL=1 selects STB. Select the next SLOT in STB by setting the TSNEXT bit in the TCTRL register. The correspondence is shown in the diagram below:



Set TSNEXT bit to mark a slot filled and to jump to the next message slot

Figure 21-4 CAN TBUF Register Write Transmit Buffer and Schematic Diagram

21.3.5.2 Data Transmission

Before starting transmission, ensure that at least one frame of data has been loaded in PTB or STB. TPE is locked during PTB transmission, and the filling status of STB can be confirmed through the TSSTAT bit.

The steps for setting up data transmission are as follows:

Step1. Set TBSEL to select the transmit BUF from PTB and STB;

Step2. Write the data to be sent through the TBUF register;

Step3. If STB is selected, set TSNEXT=1 to complete the loading of STB SLOTs;

Step4. Send enable;

- Use TPE for PTB transmission.
- Use TSALL or TSONE for STB transmission.

Step5. Send complete status acknowledgment:

- Use TPIF for PTB transmission completion, and TPIE to enable TPIF;
- Use TSIF for STB transmission completion when using TSONE, and TSIE to enable TSIF;
- Use TSIF for STB transmission completion when using TSALL. In this case, TSIF is set after all STB SLOT data to be sent is sent, and TSIE is used to enable TSIF.

21.3.5.3 Single Data Transmission

When automatic retransmission is not required, single send mode can be set through registers. The CFG_STAT.TPSS bit is used for single send mode setting for PTB, and the CFG_STAT.TSSS bit is used for STB. When data is successfully sent, the actions in single send mode are the same as those in normal send mode. However, when data is not successfully sent, the following results occur:

- TPIF is set (TPIE=1), and the corresponding BUF SLOT data is cleared.
- When there is an error in transmission, KOER is updated, and BEIF is set (BEIE=1).
- In case of arbitration lost, ALIF is set (ALIE=1).

In single send mode, TPIF alone cannot be relied upon to determine transmission completion. It is necessary to use BEIF and ALIF together to determine whether transmission is complete.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

21.3.5.4 Cancel Data Transmission

Data transmission that has been requested but not yet executed can be canceled through TPA or TSA. The following situations may occur when canceling data transmission:

- During arbitration:
  - If the node loses arbitration, data transmission is canceled.
  - If the node wins arbitration, data transmission continues.
- Data sending:
  - Data is successfully sent and ACK is received. The corresponding flags and statuses are set normally. Data transmission is not canceled;
  - Data is successfully sent but no ACK is received. Data transmission is canceled, and the error counter is incremented.
  - For data transmission set by TSALL=1, the STB SLOT data being sent is sent normally, and the STB SLOT that has not started sending is canceled.

The results of canceling data transmission are as follows:

- TPA releases PTB and sets TPE=0.
- TSA releases one STB SLOT or all STB SLOTs depending on whether TSONE or TSALL is enabled for transmission.

21.3.6 Reception Processing

21.3.6.1 Receive Buffer

CAN_CTRL provides 10 SLOT receive buffers for storing received data. These 10 SLOT receive buffers operate in FIFO mode. Each RB SLOT occupies four words, received data can be read through the RBUF register. The earliest received data is always read first, and the read RB SLOT is released by setting RREL=1 in the RCTRL register, pointing to the next RB SLOT.

The diagram below shows the schematic for reading RB SLOT through RBUF.

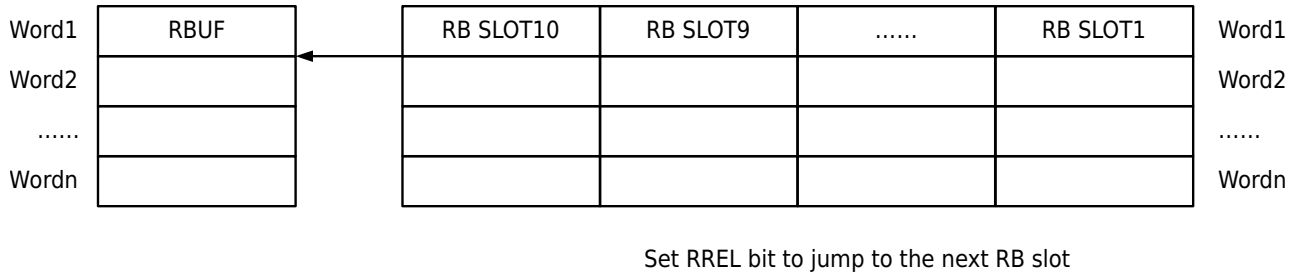


Figure 21-5 CAN RBUF Register Read Receive Buffer Schematic Diagram

21.3.6.2 Receive Filter Register Set

CAN_CTRL provides 8 groups of 32-bit filters for filtering received data to reduce CPU load. The filters can support standard format 11-bit ID or extended format 29-bit ID. Each filter group has a 32-bit ID CODE register and a 32-bit ID MASK register. The ID CODE register is used to compare the received CAN ID, and the ID MASK register is used to select the CAN ID bits for comparison. When the corresponding ID MASK bit is 1, the ID CODE of that bit is not compared.

Received data is received if it passes through any one of the 8 filter groups.The received data is stored in RB; otherwise, it is not received or stored.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Each filter group is enabled or disabled through the ACFEN register. ID CODE and ID MASK are set through the SELMASK bit in the ACFCTRL register. When SELMASK=0, it points to ID CODE; when SELMASK=1, it points to ID MASK. Filters are selected through the ACFADR bit in the ACFCTRL register. ID CODE and ID MASK are accessed through the ACF register and can only be set when CFG_STAT.RESET=1, i.e., during CAN software reset. Rrefer to the diagram below for the method of accessing filter groups through the CAN ACF register.

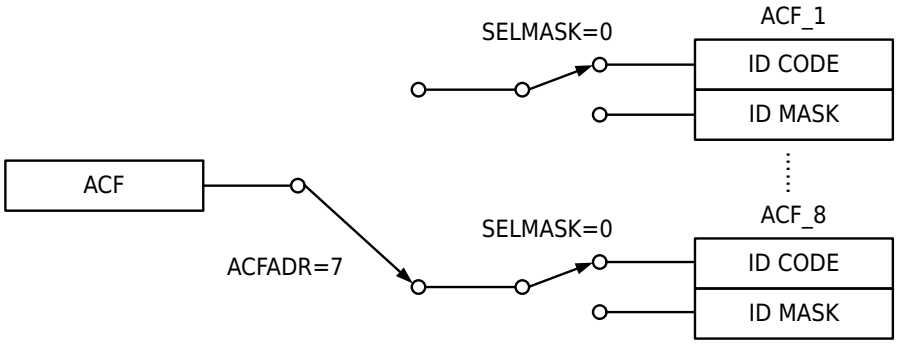


Figure 21-6 CAN ACF Register Access Filter Group Schematic Diagram

21.3.6.3 Data Reception

The receive filter groups can filter out unwanted received data, reducing the occurrence of interrupts and RB reads, thereby reducing CPU load. The steps for setting up received data are as follows:

Step1. Set filter groups.

Step2. Set RFIE, RAFIE, and AFWL.

Step3. Wait for RFIF or RAFIF.

Step4. Read the earliest received data from the RB FIFO through RBUF.

Step5. Set RREL=1 to select the next RB SLOT.

Step6. Repeat steps 4 and 5 until it is confirmed that RB is empty through RSTAT.

21.3.7 Error Handling

CAN_CTRL can not only automatically handle some errors, such as automatically retransmitting data or discarding received frames containing errors, but also report errors to the CPU through interrupts.

CAN nodes have the following 3 error states:

- Error Active: The node automatically sends an active error flag when an error is detected.
- Error Passive: The node automatically sends a passive error flag when an error is detected.
- Bus Off: In this state, the node no longer affects the entire CAN network.

CAN_CTRL provides two counters, TECNT and RECNT, for counting errors. The TECNT and RECNT counters are incremented or decremented according to the rules specified in the CAN2.0B protocol. Additionally, a programmable CAN error warning LIMIT register is provided to generate error interrupts to notify the CPU.

There are the following five types of errors during CAN communication, which can be identified through the KOER bit in the EALCAP register.

- Bit error
- Wrong form
- Stuff Error
- Acknowledgment error

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

- CRC error

21.3.8 BUS OFF

When the number of transmission errors exceeds 255, the CAN node automatically enters Bus Off state and no longer participates in CAN communication until it returns to Error Active state. The Bus Off state of the CAN node can be confirmed by checking the CFG_STAT.BUSOFF bit. An EIF interrupt is generated when BUSOFF is set.

There are the following two methods for CAN to recover from Bus Off state to Error Active state:

- Power-on reset
- Received 128 consecutive 11-bit stealth bit sequences (recovery sequence)

In Bus Off state, the TECNT value remains unchanged, and RECNT is used to count the recovery sequence. After recovery from Bus Off state, TECNT and RECNT are reset to 0.

When the BUSOFF flag is set, the CFG_STAT.RESET bit is also set.

21.3.9 Arbitration lost Position Capture

CAN_CTRL can precisely capture the arbitration lost bit position and reflect it in the ALC register. The ALC register stores the bit position of the most recent arbitration lost bit. If the node wins arbitration, the ALC bit is not updated.

The ALC Value Is Defined As Follows:

After the SOF bit, the ALC for the first ID data bit is 0, the ALC for the second ID data bit is 1, and so on. Since arbitration only occurs in the arbitration field, the maximum value of ALC is 31. For example, in an arbitration between a standard format remote frame and an extended frame, if the extended frame fails at the IDE bit, then ALC=12.

21.3.10 Time-Triggered TTCAN

CAN-CTRL provides partial (level 1) hardware support for the time-triggered communication method specified in ISO11898-4. This chapter introduces the TTCAN function from the following five sections.

21.3.10.1 TBUF Behavior in TTCAN Mode

TTTBM=1

When TTTBM=1, PTB and STB SLOTs together form TB SLOT, and the transmit BUF is specified through the TBPTR register. TBPTR=0 points to PTB, TBPTR=1 points to STB SLOT1, and so on. The host can mark transmit BUF SLOTs through the TBE and TBF registers. At this time, the TBSEL and TSNEXT registers are meaningless and can be ignored.

When TTTBM=1, PTB does not have any special attributes and is the same as STB SLOT. The transmission completion flag also uses TSIF.

In TTCAN mode, the transmit BUF does not have FIFO mode or priority arbitration mode, and only one selected SLOT can transmit data.

In TTCAN mode, transmission start requires a time-triggered method. TPE, TSONE, TSALL, TPSS, and TPA are fixed to 0 and ignored.

TTTBM=0

When TTTBM=0, event-driven communication and receive timestamp functions are used in combination. In this mode, the functions of PTB and STB are the same as when TTEN=0. Therefore, PTB always has the highest priority, and STB can operate in FIFO mode or priority arbitration mode.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

21.3.10.2 TTCAN Function Description

After power-on, the Time Master needs to be initialized according to the ISO 11898-4 protocol. In a CAN network, there can be up to eight potential Time Masters. Each Time Master has its own reference message ID (last three bits of the ID). These potential Time Masters send their respective reference messages based on their priorities.

After TTEN=1, the 16-bit counter starts working. When a reference message is successfully received or the Time Master successfully sends a reference message, the CAN controller copies Sync_Mark to Ref_Mark, and Ref_Mark sets the cycle time to 0. Successfully receiving a reference message sets the RIF flag, and successfully sending a reference message sets the TPIF flag or TSIF flag. At this time, the host needs to prepare the trigger conditions for the next action.

The trigger condition can be a receive trigger. This trigger only triggers an interrupt and can be used to detect when an expected message has not been received.

The trigger condition can also be a transmit trigger. This trigger starts transmitting data in the TBUF SLOT specified by the TTPTR register. If the selected TBUF SLOT is marked as empty, transmission does not start, but the interrupt flag is set.

21.3.10.3 TTCAN Timing

CAN_CTRL supports ISO 11898-4 level 1. It includes a 16-bit counter that operates under the bit time defined by PRESC, SEG_1, SEG_2. If TTEN=1, there is an additional prescaler T_PRESC.

At the SOF of a frame of data, the counter value is Sync_Mark. If the frame is a reference message, Sync_Mark is copied to Ref_Mark. The cycle time is equal to the counter value minus Ref_Mark. This time is used as a timestamp for received messages or a trigger time baseline for transmitted messages.

21.3.10.4 TTCAN Trigger Mode

The TTYPE register defines the TTCAN trigger method, the TTPTR register specifies the transmit SLOT, and TT_TRIG specifies the cycle time of the trigger.

The following 5 trigger methods are included:

- Immediate Trigger
- Time Trigger
- Single Transmit Trigger
- Transmit Start Trigger
- Transmit Stop Trigger

Except for the immediate trigger method, all triggers use the TTIF flag. When TTTBM=1, only the time-triggered method is supported.

Immediate Trigger

The trigger is started by writing the high bit of TT_TRIG (the value written does not matter). In this mode, the data in the TBUF SLOT selected by TTPTR is immediately sent. TTIF is not set.

Time Trigger

The time-triggered method only generates an interrupt by setting the TTIF flag and has no other functions. If a node expects to receive expected data within a specific time window, the time-triggered method can be used. If the TT_TRIG value is less than the actual cycle time, TEIF is set with no other action.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Single Transmit Trigger

The single transmit trigger method is used to send data within an execution time window. At this time, the TSSS bit is ignored.

The maximum of 16 cycle time ticks specified by ISO11898-4 is set through the TEW bit, with a setting range of 1-16. If data is not started to be sent within the specified transmit enable time window, the frame is discarded. The corresponding transmit BUF SLOT is marked as empty, and AIF is set. The data in the corresponding transmit BUF is not overwritten because it can be retransmitted by setting TPF.

If the TT_TRIG value is less than the actual cycle time, TEIF is set with no other action.

Transmit Start Trigger

The transmit start trigger method is used for arbitration within a time window. TSSS is used to determine whether to use auto-retransmit or single send mode. If the message specified by the TTPTR register is not successfully sent, the transmit stop trigger can be used to stop the transmission.

If the TT_TRIG value is less than the actual cycle time, TEIF is set with no other action.

Transmit Stop Trigger

The transmit stop trigger method is used to stop a transmission that has already started through the transmit start trigger method. If the transmission is stopped, the transmit frame is discarded, AIF is set, and the selected TBUF SLOT is marked as empty. However, the data in the TBUF SLOT is not overwritten and can be retransmitted by setting TPF.

If the TT_TRIG value is less than the actual cycle time, TEIF is set and the stop is executed.

21.3.10.5 TTCAN Trigger Gate Time

The TTCAN trigger watchdog time function is similar to the watchdog function and is used when TTTBM=1. It is used to watchdog the time since the last successful reception of a reference message. The reference message can be received within the cycle time or after an event. The application should set an appropriate watchdog time based on specific circumstances.

If the cycle count equals TT_WTRIG, WTIF is set. The watchdog trigger is turned off by writing 0 to WTIE.

If TT_WTRIG is less than the actual cycle time, TEIF is set.

21.3.11 Upward Compatibility with CAN-FD Function

CAN-CTRL automatically ignores CAN FD frames even when received in a network that includes CAN FD, without returning an ACK. It will send or receive the next CAN 2.0B frame once the bus is idle.

21.3.12 Software Reset Function

Software reset function is achieved by setting the CFG_STAT.RESET bit to 1. The reset scope of the software reset function is shown in the table below.

Table 21-3 Software Reset Range Table

Register Bit Name	Software Reset	Remark	Register Bit Name	Software Reset	Remark
ACFADR	No	-	EWL	Yes	-
ACODE	No	Writable only during software reset	KOER	Yes	-
AE_x	No	-	LBME	Yes	-

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Register Bit Name	Software Reset	Remark	Register Bit Name	Software Reset	Remark
AFWL	No	-	LBMI	Yes	-
AIF	Yes	-	RACTIVE	Yes	Immediately stops reception and does not generate ACK
ALC	Yes	-	RAFIE	No	-
ALIE	No	-	RAFIF	Yes	-
ALIF	Yes	-	RBALL	Yes	-
AMASK	No	Writable only during software reset	RBUF	Yes	All RBs are marked as empty, value indeterminate
BEIE	No	-	REF_ID	No	-
BEIF	Yes	-	REF_IDE	No	-
BUSOFF	No	Cleared by writing 1	RFIE	No	-
EIE	No	-	RFIF	Yes	-
EIF	No	-	RIE	No	-
EPASS	No	-	RIF	Yes	-
EPIE	No	-	ROIE	No	-
EPIF	Yes	-	ROIF	Yes	-
EWARN	No	-	ROM	No	-
ROV	Yes	-	TSNEXT	Yes	-
RREL	Yes	-	TSONE	Yes	-
PRESC	No	Writable only during software reset	TPIE	No	-
RSTAT	Yes	-	TPIF	Yes	-
SACK	Yes	-	TPSS	Yes	-
SELMASK	No	-	TSFF	Yes	All STB SLOTs are marked as empty
SEG_1	No	Writable only during software reset	TSIE	No	-
SEG_2	No	Writable only during software reset	TSIF	Yes	-
SJW	No	Writable only during software reset	TSSS	Yes	-
TACTIVE	Yes	Immediately stops transmission	TSSTAT	Yes	All STB SLOTs are marked as empty
TBE	Yes	-	TTEN	Yes	-
TBF	No	-	TTIF	Yes	-

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Register Bit Name	Software Reset	Remark	Register Bit Name	Software Reset	Remark
TBPTR	No	-	TTIE	No	-
TBSEL	Yes	-	TTPTR	No	-
TBUF	Yes	All STBs are marked as empty, pointing to PTB	TTTBM	No	-
TECNT	No	Cleared by BUSOFF=1	TTYPE	No	-
TEIF	Yes	-	TT_TRIG	No	-
TPA	Yes	-	TT_WTRIG	No	-
TPE	Yes	-	T_PRESC	No	-
TSA	Yes	-	WTIE	No	-
TSALL	Yes	-	WTIF	Yes	-
TSMODE	No	-			

21.3.13 Interrupt

Table 21-4 CAN Interrupt Table

Interrupt Flag	Description
RIF	Receive interrupt
ROIF	Receive overflow interrupt
ROIF	Receive BUF full interrupt
RAFIF	Receive BUF almost full interrupt
TPIF	PTB send interrupt
TSIF	STB transmit interrupt
EIF	Error interrupt
AIF	Cancel transmitting interrupt
EPIE	Error passive interrupt
ALIF	Arbitration lost interrupt
BEIF	Bus error interrupt
WTIF	Trigger watchdog interrupt
TEIF	Trigger error interrupt
TTIF	Time triggered interrupt

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

21.4 Register Description

21.4.1 Register List

Base Address: 0x40030000

Table 21-5 Controller Area Network (CAN) Register Offset

Offset	Register	Description	Width
0x00-0x0F	CAN_RBUF	CAN Receive Buffer Register	-
0x50-0x5F	CAN_TBUF	CAN Transmit Buffer Register	-
0xA0	CAN_CFG_STAT	CAN Configuration and Status Register	8
0xA1	CAN_TCMD	CAN Command Register	8
0xA2	CAN_TCTRL	CAN Transmit Control Register	8
0xA3	CAN_RCTRL	CAN Receive Control Register	8
0xA4	CAN_RTIE	CAN Receive and Transmit Interrupt Enable Register	8
0xA5	CAN_RTIF	CAN Receive and Transmit Interrupt Flag Register	8
0xA6	CAN_ERRINT	CAN Error Interrupt Enable and Flag Register	8
0xA7	CAN_LIMIT	CAN Warning Limits Register	8
0xA8	CAN_BT	CAN Bit Timing Register	32
0xB0	CAN_EALCAP	CAN Error and Arbitration Lost Capture Register	8
0xB2	CAN_RECNT	CAN Receive Error Counter Register	8
0xB3	CAN_TECNT	CAN Transmit Error Counter Register	8
0xB4	CAN_ACFCTRL	CAN Acceptance Filter Control Register	8
0xB6	CAN_ACFEN	CAN Acceptance Filter Enable Register	8
0xB8	CAN_ACF	CAN Acceptance Code and Mask Register	32
0xBE	CAN_TBSLOT	TTCAN TB Slot Point Register	8
0xBF	CAN_TTCFG	TTCAN Time Trigger Configuration Register	8
0xC0	CAN_REF_MSG	TTCAN Reference Message Register	32
0xC4	CAN_TRG_CFG	TTCAN Trigger Configuration Register	16
0xC6	CAN_TRG_TRIG	TTCAN Trigger Time Register	16
0xC8	CAN_TRG_WTRIG	TTCAN Watch Trigger Time Register	16

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

21.4.2 CAN Receive Buffer Register (CAN_RBUF)

Offset: 0x00-0x0F

Reset: 0xXXXXXXXX

The RBUF register points to the RB SLOT address of the earliest received CAN mailbox. The RBUF register can be read in any order.

The KOER bit is the EALCAP.KOER register, which is meaningful only when RBALL=1.

The TX bit indicates that a message sent by itself is received in loopback mode.

The CYCLE_TIME bit is valid only in TTCAN mode and indicates the cycle time at the start of SOF.

The data format of the CAN receive mailbox is as follows:

Table 21-6 Standard Format CAN Receive Mailbox Format

Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Function
RBUF	ID[7:0]								ID
RBUF+1	Reserved					ID[10:8]			ID
RBUF+2	Reserved								ID
RBUF+3	Reserved								ID
RBUF+4	IDE=0	RTR	Reserved		DLC[3:0]				Control
RBUF+5	KOER[2:0]			TX	Reserved				Status
RBUF+6	CYCLE_TIME[7:0]								Time-Triggered CAN
RBUF+7	CYCLE_TIME[15:8]								Time-Triggered CAN
RBUF+8	DATA1								Data
RBUF+9	DATA2								Data
RBUF+10	DATA3								Data
...	...								Data
RBUF+14	DATA7								Data
RBUF+15	DATA8								Data

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Table 21-7 Extended Format CAN Receive Mailbox Format

Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Function
RBUF	ID[7:0]								ID
RBUF+1	ID[15:8]								ID
RBUF+2	ID[23:16]								ID
RBUF+3	Reserved			ID[28:24]					ID
RBUF+4	IDE=1	RTR	Reserved		DLC[3:0]				Control
RBUF+5	KOER[2:0]			TX	Reserved				Status
RBUF+6	CYCLE_TIME[7:0]								Time-Triggered CAN
RBUF+7	CYCLE_TIME[15:8]								Time-Triggered CAN
RBUF+8	DATA1								Data
RBUF+9	DATA2								Data
RBUF+10	DATA3								Data
...	...								Data
RBUF+14	DATA7								Data
RBUF+15	DATA8								Data

The meanings of the control bits and status bits are as follows:

Classification	Function	Description
Control Bit	IDE (IDentifier Extension)	● 0b0: standard format ● 0b1: extended Format
	RTR (Remote Transmission Request)	● 0b0: data frame ● 0b1: remote frame
	DLC (Data Length Code)	Data length code, range 0-8, corresponding to data lengths of 0-8 bytes. Table 21-9 DLC Relationship with CAN 2.0 Data length DLC (binary) / Data length 0b0000-0b1000 / 0-8 0b1001-0b1111 / 8
Status Bits	KOER	Same as EALCAP.KOER
	TX	This bit is set to 1 when receiving a message sent by itself in loopback mode

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

21.4.3 CAN Transmit Buffer Register (CAN_TBUF)

Offset: 0x50-0x5F

Reset: 0xXXXXXXXX

The TBUF register points to the next empty CAN transmit BUF SLOT. The TBUF register can be read in any order. The corresponding TBUF SLOT is marked as written by writing 1 to TSNEXT, thereby pointing to the next TBUF SLOT.

CAN_TBUF only supports WORD access.

The data format of the CAN sending mailbox is as follows:

Table 21-10 Standard Format CAN Transmit Mailbox Format

Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Function
TBUF	ID[7:0]								ID
TBUF+1	Reserved					ID[10:8]			ID
TBUF+2	Reserved								ID
TBUF+3	Reserved								ID
TBUF+4	IDE=0	RTR	Reserved		DLC[3:0]				Control
TBUF+5	Reserved								-
TBUF+6	Reserved								-
TBUF+7	Reserved								-
TBUF+8	DATA1								Data
TBUF+9	DATA2								Data
TBUF+10	DATA3								Data
...	...								Data
TBUF+14	DATA7								Data
TBUF+15	DATA8								Data

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Table 21-11 Extended Format CAN Transmit Mailbox Format

Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Function
TBUF	ID[7:0]								ID
TBUF+1	ID[15:8]								ID
TBUF+2	ID[23:16]								ID
TBUF+3	Reserved			ID[28:24]					ID
TBUF+4	IDE=1	RTR	Reserved		DLC[3:0]				Control
TBUF+5	Reserved								-
TBUF+6	Reserved								-
TBUF+7	Reserved								-
TBUF+8	DATA1								Data
TBUF+9	DATA2								Data
TBUF+10	DATA3								Data
...	...								Data
TBUF+14	DATA7								Data
TBUF+15	DATA8								Data

The meaning of the control bits is described in the "CAN Receive Buffer Register (CAN_RBUF)" section.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

21.4.4 CAN Configuration and Status Register (CAN_CFG_STAT)

Offset	Bit Position							
0xA0	7	6	5	4	3	2	1	0
Reset	0x80							
Name	RESET	LBME	LBMI	TPSS	TSSS	RACTIVE	TACTIVE	BUSOFF
Access	RW	RW	RW	RW	RW	RO	RO	RW

Bit	Name	Functional Description	Access
7	RESET	Reset Request ● 0b0: Do not request partial reset ● 0b1: Request partial reset Some registers can only be written when RESET=1. For details, refer to the "Software Reset Function" section. When the node enters BUS OFF state, the hardware automatically sets the RESET bit. **Note** After RESET=0, it takes 11 CAN bit times for the node to participate in communication.	RW
6	LBME	Enable External Loopback Mode ● 0b0: Disable ● 0b1: Enable **Note** This bit cannot be set during communication.	RW
5	LBMI	Internal Loopback Mode Enable ● 0b0: Disable ● 0b1: Enable **Note** This bit cannot be set during communication.	RW
4	TPSS	PTB Single Transmit Mode ● 0b0: Disable PTB single transmit mode ● 0b1: Enable PTB single transmit mode	RW
3	TSSS	STB Single Transmit Mode ● 0b0: Disable STB single transmit mode ● 0b1: Enable STB single transmit mode	RW
2	RACTIVE	Receive Status Signal ● 0b0: Not receiving ● 0b1: Receiving	RO

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Bit	Name	Functional Description	Access
1	TACTIVE	Transmit Status Signal ● 0b0: Not transmitting ● 0b1: Transmitting	RO
0	BUSOFF	Bus Off State ● 0b0: Bus active state ● 0b1: Bus off state **Note** Writing 1 clears the TECNT and RECNT registers, but this is only for debugging purposes.	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

21.4.5 CAN Command Register (CAN_TCMD)

Offset	Bit Position							
0xA1	7	6	5	4	3	2	1	0
Reset	0x00							
Name	TBSEL	LOM	STBY	TPE	TPA	TSONE	TSALL	TSA
Access	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Functional Description	Access
7	TBSEL	Transmit BUF Selection Bit ● 0b0: PTB ● 0b1: STB When TTEN=1 & TTTBM=1, TBSEL is reset to its reset value. **Note** This bit needs to be held constant when writing to the TBUF register or TSNEXT bit.	RW
6	LOM	Silent Mode Enable Bit ● 0b0: Disable silent mode ● 0b1: Enable silent mode When LOM=1 & LBME=0, transmission is disabled. When LOM=1 & LBME=1, acknowledgments to received frames and error frames are disabled, but data can be sent. **Note** This bit cannot be set during communication.	RW
5	STBY	Send Low Power Mode ● 0b0: Disable ● 0b1: Enable This bit is connected to the output CAN_STBY signal to control the low power mode of the external transceiver. If TPE=1, TSONE=1, or TSALL=1, STBY cannot be set to 1. If software sets STBY to 0, a waiting period is required for the transceiver to start up before initiating a new transmission.	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Bit	Name	Functional Description	Access
4	TPE	PTB Transmit Enable Bit ● 0b0: Disable PTB transmission ● 0b1: Enable PTB transmission After this bit is enabled, the mailbox in PTB will be sent at the next available bit position. An ongoing STB transmission will continue, but the next pending STB transmission will be delayed until the PTB transmission is complete. After this bit is written to 1, it will remain 1 until the PTB transmission is completed or the transmission is canceled by TPA. Software cannot clear this bit by writing 0. The TPE is reset to the reset value by hardware in the following cases: ● RESET=1 ● BUSOFF=1 ● LOM=1&LBME=0 ● TTEN=1&TTTBM=1	RW
3	TPA	PTB Transmit Cancel Bit ● 0b0: Do not cancel ● 0b1: Cancel a PTB transmission that has been requested by setting TPE to 1 but has not yet started This bit is written to 1 by software but cleared by hardware. This bit can be cleared by writing 1, so it is written to 1 simultaneously with TPE. This bit is reset to its reset value by hardware in the following situations: ● RESET=1 ● BUSOFF=1 ● TTEN=1&TTTBM=1	RW
2	TSONE	Transmit a Frame STB Data Setting Bit ● 0b0: Do not send ● 0b1: Send one frame of STB data In FIFO mode, the earliest written data is sent. In priority mode, the data with the highest priority is sent. This bit remains set to 1 until the STB transmission is complete or canceled by TSA. Software cannot clear this bit by writing 0. This bit is reset to its reset value by hardware in the following situations: ● RESET=1 ● BUFOFF=1 ● LOM=1&LBME=0 ● TTEN=1&TTTBM=1	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Bit	Name	Functional Description	Access
1	TSALL	Transmit All STB Data Setting Bit ● 0b0: Do not send ● 0b1: Send all data in STB This bit remains set to 1 until the STB transmission is complete or canceled by TSA. Software cannot clear this bit by writing 0. This bit is reset to its reset value by hardware in the following situations: ● RESET=1 ● BUSOFF=1 ● LOM=1&LBME=0 ● TTEN=1&TTTBM=1	RW
0	TSA	STB Transmit Cancel Bit ● 0b0: Do not cancel ● 0b1: Cancel an STB transmission that has been requested by setting TSONE or TSALL to 1 but has not yet started. This bit is written to 1 by software but cleared by hardware. This bit can clear the TSONE or TSALL bit by writing 1. This bit is reset to its reset value by hardware in the following situations: ● RESET=1 ● BUSOFF=1	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

21.4.6 CAN Transmit Control Register (CAN_TCTRL)

Offset	Bit Position							
0xA2	7	6	5	4	3	2	1	0
Reset	0x90							
Name	Reserved	TSNEXT	TSMODE	TTTBM	Reserved		TSSTAT	
Access		RW	RW	RW			RO	

Bit	Name	Functional Description	Access
7	*Reserved*	-	-
6	TSNEXT	Next STB SLOT ● 0b0: No action ● 0b1: The current STB SLOT is filled, pointing to the next SLOT After the application writes data to TBUF, it sets the TSNEXT bit to indicate that the current STB SLOT has been filled, so the hardware points TBUF to the next STB SLOT. The data in the STB SLOT marked by the TSNEXT bit can be sent using the TSONE or TSALL bit. This bit is written to 1 by the application and cleared by hardware. After all STB SLOTs are filled, TSNEXT remains set to 1 until an STB SLOT is released. **Note** This bit is fixed at 0 in TTCAN mode.	RW
5	TSMODE	STB Transmit Mode ● 0b0: FIFO mode ● 0b1: Priority mode In FIFO mode, data frames are sent in the order they are written. In priority mode, the ID is automatically judged, with smaller IDs having higher priority. In either mode, PTB has the highest priority. **Note** The TSMODE bit can only be set when STB is empty.	RW
4	TTTBM	TTCAN BUF Mode ● 0b0: Determined by TSMODE, PTB and STB ● 0b1: Set by TBPTR and TTPTR In TTCAN mode, when only the receive timestamp function is required, this bit can be set to 0, and TSMODE determines whether to use PTB or STB. **Note** The TSMODE bit can only be set when STB is empty. When TTEN=0, TTTBM has no effect.	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Bit	Name	Functional Description	Access
3:2	*Reserved*	-	-
1:0	TSSTAT	STB Status When TTEN=0 or TTEN=1 & TTTBM=0: ● 0b00: STB empty ● 0b01: STB is less than or equal to half full ● 0b10: STB is more than half full ● 0b11: STB is full When TTEN=1 and TTTBM=1: ● 0b00: PTB and STB Empty ● 0b01: Both PTB and STB are not full ● 0b10: Reserved ● 0b11: Both PTB and STB are full	RO

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

21.4.7 CAN Receive Control Register (CAN_RCTRL)

Offset	Bit Position							
0xA3	7	6	5	4	3	2	1	0
Reset	0x00							
Name	SACK	ROM	ROV	RREL	RBALL	Reserved	RSTAT	
Access	RW	RW	RO	RW	RW		RO	

Bit	Name	Functional Description	Access
7	SACK	Self-Acknowledge ● 0b0: No self-acknowledge ● 0b1: When LBME=1, enable the self-acknowledge function	RW
6	ROM	Receive BUF Overflow Mode Setting Bit ● 0b0: The earliest received data is overwritten ● 0b1: Newly received data is not stored	RW
5	ROV	Receive BUF Overflow Flag ● 0b0: No overflow ● 0b1: Overflow, with at least one data loss Cleared by writing 1 to RREL.	RO
4	RREL	Receive Buffer Release ● 0b0: Do not release ● 0b1: Indicates that the receive BUF has been read, and the RBUF register points to the next RB SLOT	RW
3	RBALL	Receive BUF Data Stores All Data Frames ● 0b0: Normal Mode ● 0b1: Store all data, including data with errors	RW
2	*Reserved*	-	-
1:0	RSTAT	Receive BUF Status ● 0b00: RBUF empty ● 0b01: RBUF is not empty but less than the AFWL programmed value ● 0b10: RBUF is greater than or equal to the AFWL programmed value but not full ● 0b11: Full (this value is maintained during overflow)	RO

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

21.4.8 CAN Receive and Transmit Interrupt Enable Register (CAN_RTIE)

Offset	Bit Position							
0xA4	7	6	5	4	3	2	1	0
Reset	0xFE							
Name	RIE	ROIE	RFIE	RAFIE	TPIE	TSIE	EIE	TSFF
Access	RW	RW	RW	RW	RW	RW	RW	RO

Bit	Name	Functional Description	Access
7	RIE	Receive Interrupt Enable ● 0b0: Disable ● 0b1: Enable	RW
6	ROIE	Receive Overflow Interrupt Enable ● 0b0: Disable ● 0b1: Enable	RW
5	RFIE	Receive BUF Full Interrupt Enable ● 0b0: Disable ● 0b1: Enable	RW
4	RAFIE	Receive BUF Almost Full Interrupt Enable ● 0b0: Disable ● 0b1: Enable	RW
3	TPIE	PTB Transmit Interrupt Enable ● 0b0: Disable ● 0b1: Enable	RW
2	TSIE	STB Transmit Interrupt Enable ● 0b0: Disable ● 0b1: Enable	RW
1	EIE	Error Interrupt Enable ● 0b0: Disable ● 0b1: Enable	RW
0	TSFF	Transmit BUF Full Flag When TTEN=0 or TTTBM=0: Transmit Secondary Buffer Full Flag ● 0b0: STB SLOT is not fully filled ● 0b1: STB SLOT is fully filled TTEN=1 and TTTBM=1: Transmit Buffer Full Flag ● 0b0: The transmit BUF selected by TBPTR is not fully filled ● 0b1: The transmit BUF selected by TBPTR is fully filled	RO

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

21.4.9 CAN Receive and Transmit Interrupt Flag Register (CAN_RTIF)

Offset	Bit Position							
0xA5	7	6	5	4	3	2	1	0
Reset	0x00							
Name	RIF	ROIF	RFIF	RAFIF	TPIF	TSIF	EIF	AIF
Access	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Functional Description	Access
7	RIF	Receive Interrupt Flag Write 1 to clear flag, write 0 has no effect Read: ● 0b0: No data frame received ● 0b1: A valid data frame or remote frame is received	RW
6	ROIF	Receive Overflow Interrupt Flag Write 1 to clear flag, write 0 has no effect Read: ● 0b0: No RB is overwritten ● 0b1: At least one RB is overwritten **Note** When overflow occurs, ROIF and RFIF are set simultaneously.	RW
5	RFIF	Receive BUF Full Interrupt Flag Write 1 to clear flag, write 0 has no effect Read: ● 0b0: RB FIFO is not full ● 0b1: RB FIFO is full	RW
4	RAFIF	Receive BUF Almost Full Interrupt Flag Write 1 to clear flag, write 0 has no effect Read: ● 0b0: The number of filled RB SLOTs is less than the AFWL setting ● 0b1: The number of filled RB SLOTs is greater than or equal to the AFWL setting	RW
3	TPIF	PTB Send Interrupt Flag Write 1 to clear flag, write 0 has no effect Read: ● 0b0: No PTB transmission completion ● 0b1: The requested PTB transmission is successfully completed **Note** In TTCAN mode, TPIF is invalid, and only TSIF is used.	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Bit	Name	Functional Description	Access
2	TSIF	STB Transmit Interrupt Flag Write 1 to clear flag, write 0 has no effect Read: ● 0b0: No STB transmission completion ● 0b1: The requested STB transmission is successfully completed **Note** In TTCAN mode, TPIF is invalid, and only TSIF is used.	RW
1	EIF	Error Interrupt Flag Write 1 to clear flag, write 0 has no effect Read: ● 0b0: The BUSOFF bit has not changed, or the relative relationship between the value of the error counter and the ERROR warning limit setting has not changed ● 0b1: The BUSOFF bit has changed, or the relative relationship between the value of the error counter and the ERROR warning limit setting has changed. For example, the value of the error counter changes from less than the set value to greater than the set value, or from greater than the set value to less than the set value	RW
0	AIF	Cancel Transmit Interrupt Flag Write 1 to clear flag, write 0 has no effect Read: ● 0b0: No transmission cancellation ● 0b1: The requested transmission message is successfully canceled through TPA and TSA **Note** It is recommended not to set TPA and TSA simultaneously, as they use the same flag bit (AIF).	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

21.4.10 CAN Error Interrupt Enable and Flag Register (CAN_ERRINT)

Offset	Bit Position							
0xA6	7	6	5	4	3	2	1	0
Reset	0x00							
Name	EWARN	EPASS	EPIE	EPIF	ALIE	ALIF	BEIE	BEIF
Access	RO	RO	RW	RW	RW	RW	RW	RW

Bit	Name	Functional Description	Access
7	EWARN	ERROR WARNING LIMIT Reached ● 0b0: RECNT or TECNT is less than the EWL setting ● 0b1: RECNT or TECNT is greater than or equal to the EWL setting	RO
6	EPASS	Error Passive ● 0b0: The node is in Error Active state ● 0b1: The node is in Error Passive state	RO
5	EPIE	Error Passive Interrupt Enable ● 0b0: Disable ● 0b1: Enable	RW
4	EPIF	Error Passive Interrupt Flag Write 1 to clear flag, write 0 has no effect Read: ● 0b0: No change from Error Active to Error Passive or from Error Passive to Error Active ● 0b1: A change from Error Active to Error Passive or from Error Passive to Error Active occurs	RW
3	ALIE	Arbitration Lost Interrupt Enable ● 0b0: Disable ● 0b1: Enable	RW
2	ALIF	Arbitration Lost Interrupt Flag Write 1 to clear flag, write 0 has no effect Read: ● 0b0: Arbitration success ● 0b1: Arbitration lost	RW
1	BEIE	Bus Error Interrupt Enable ● 0b0: Disable ● 0b1: Enable	RW
0	BEIF	Bus Error Interrupt Flag Write 1 to clear flag, write 0 has no effect Read: ● 0b0: No Bus Error ● 0b1: Bus Error	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

21.4.11 CAN Bit Timing Register (CAN_BT)

Offset	Bit Position																													
0xA8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x01020203																															
Name	PRESC								Reserved	SJW							Reserved	SEG_2							SEG_1							
Access	RW									RW								RW							RW							

Bit	Name	Functional Description	Access
31:24	PRESC	Prescaler Setting The prescaler divides the CAN communication clock to obtain the TQ (Time quanta) clock. The setting range is 0-255, and the prescaler value is PRESC+1	RW
23	*Reserved*	-	-
22:16	SJW	Resynchronization Compensation Width Time Setting Resynchronization Compensation Width Time = (SJW + 1) * TQ	RW
15	*Reserved*	-	-
14:8	SEG_2	Bit Segment 2 Time Setting Segment 2 Time = (SEG_2 + 1) * TQ	RW
7:0	SEG_1	Bit Segment 1 Time Setting Segment 1 Time = (SEG_1 + 2) * TQ	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

21.4.12 CAN Error and Arbitration Lost Capture Register (CAN_EALCAP)

Offset	Bit Position							
0xB0	7	6	5	4	3	2	1	0
Reset	0x00							
Name	KOER			ALC				
Access	RO			RO				

Bit	Name	Functional Description	Access
7:5	KOER	Error Type ● 0b000: No error ● 0b001: Bit error ● 0b010: Format error ● 0b011: Stuff error ● 0b100: Acknowledgment error ● 0b101: CRC error ● 0b110: Other errors ● 0b111: Reserved **Note** The KOER bit is updated when there is an error, and the KOER bit remains unchanged when sending and receiving normally.	RO
4:0	ALC	Arbitration Lost Position Capture When arbitration fails, ALC records the bit position where arbitration fails in a frame of data	RO

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

21.4.13 CAN Warning Limits Register (CAN_LIMIT)

Offset	Bit Position							
0xA7	7	6	5	4	3	2	1	0
Reset	0x1B							
Name	AFWL				EWL			
Access	RW				RW			

Bit	Name	Functional Description	Access
7:4	AFWL	Receive BUF Almost Full Warning Limit The setting range is 1-8 AFWL=0 is meaningless and is treated as AFWL=1	RW
3:0	EWL	Programmable Error Warning Limit Error Waring Limit = (EWL + 1) * 8 The setting value of this register affects the EIF flag	RW

21.4.14 CAN Receive Error Counter Register (CAN_RECNT)

Offset	Bit Position							
0xB2	7	6	5	4	3	2	1	0
Reset	0x00							
Name	RECNT							
Access	RO							

Bit	Name	Functional Description	Access
7:0	RECNT	Receive Error Counter The receive error counter increases or decreases according to the error counting rules specified in the CAN protocol. This counter does not overflow, with 255 as the maximum value.	RO

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

21.4.15 CAN Transmit Error Counter Register (CAN_TECNT)

Offset	Bit Position							
0xB3	7	6	5	4	3	2	1	0
Reset	0x00							
Name	TECNT							
Access	RO							

Bit	Name	Functional Description	Access
7:0	TECNT	Receive Error Counter The transmit error counter increases or decreases according to the error counting rules specified in the CAN protocol. This counter does not overflow, with 255 as the maximum value. **Note** When entering Bus_Off, TECNT may overflow, so it may hold a smaller value.	RO

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

21.4.16 CAN Acceptance Filter Control Register (CAN_ACFCTRL)

Offset	Bit Position							
0xB4	7	6	5	4	3	2	1	0
Reset	0x00							
Name	Reserved		SELMASK	Reserved	ACFADR			
Access			RW		RW			

Bit	Name	Functional Description	Access
7:6	*Reserved*	-	-
5	SELMASK	Filter Selection Mask Register ● 0b0: ACF points to the filter ID register ● 0b1: ACF points to the filter MASK register Select a specific filter register group through ACFADR.	RW
4	*Reserved*	-	-
3:0	ACFADR	Filter Address ACFADR points to a specific filter, and distinguishes between ID and MASK through SELMASK: ● 0b0000: Points to ACF_1 ● 0b0001: Points to ACF_2 ● 0b0010: Points to ACF_3 ● 0b0011: Points to ACF_4 ● 0b0100: Points to ACF_5 ● 0b0101: Points to ACF_6 ● 0b0110: Points to ACF_7 ● 0b0111-0b1111: Points to ACF_8	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

21.4.17 CAN Acceptance Filter Enable Register (CAN_ACFEN)

Offset	Bit Position							
0xB6	7	6	5	4	3	2	1	0
Reset	0x01							
Name	AE_8	AE_7	AE_6	AE_5	AE_4	AE_3	AE_2	AE_1
Access	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Functional Description	Access
7	AE_8	ACF_8 Enable ● 0b0: Disable ● 0b1: Enable	RW
6	AE_7	ACF_7 Enable ● 0b0: Disable ● 0b1: Enable	RW
5	AE_6	ACF_6 Enable ● 0b0: Disable ● 0b1: Enable	RW
4	AE_5	ACF_5 Enable ● 0b0: Disable ● 0b1: Enable	RW
3	AE_4	ACF_4 Enable ● 0b0: Disable ● 0b1: Enable	RW
2	AE_3	ACF_3 Enable ● 0b0: Disable ● 0b1: Enable	RW
1	AE_2	ACF_2 Enable ● 0b0: Disable ● 0b1: Enable	RW
0	AE_1	ACF_1 Enable ● 0b0: Disable ● 0b1: Enable	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

21.4.18 CAN Acceptance Code and Mask Register (CAN_ACF)

Offset	Bit Position																													
0xB8	31	30	29	28	27	26	25	24	23	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	AIDEE	AIDE	ACODE/AMASK																												
Access		RW	RW	RW																												

Bit	Name	Functional Description	Access
31	*Reserved*	-	-
30	AIDEE	IDE Bit Comparison Enable ● 0b0: The filter receives standard format and extended format frames ● 0b1: The filter receives standard format or extended format frames defined by the AIDE bit **Note** Only valid when SELMASK=1.	RW
29	AIDE	IDE Bit MASK ● 0b0: The filter only receives standard format frames ● 0b1: The filter only receives extended format frames	RW
28:0	ACODE/ AMASK	Filter CODE/Filter MASK ACODE: ● Point to a specific filter through ACFADR ● When SELMASK=0, it represents the filter's CODE ● For standard format, use Bit10-Bit0; for extended format, use Bit28-Bit0 AMASK: ● Point to a specific filter through ACFADR ● When SELMASK=1, it represents the filter's MASK ● For standard format, use Bit10-Bit0; for extended format, use Bit28-Bit0	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

21.4.19 TTCAN TB Slot Point Register (CAN_TBSLOT)

Offset	Bit Position							
0xBE	7	6	5	4	3	2	1	0
Reset	0x00							
Name	TBE	TBF	Reserved			TBPTR		
Access	RW	RW				RW		

Bit	Name	Functional Description	Access
7	TBE	Set TB Empty ● 0b0: No operation ● 0b1: The SLOT selected by TBPTR is marked as empty When the SLOT is marked as empty and TSFF=0, TBE is automatically reset to 0. If this bit is set to 1 and there is data being transmitted in the selected SLOT, TBE=1. TBE is reset to 0 after transmission completion, transmission error, or transmission cancellation. TBE has higher priority than TBF.	RW
6	TBF	Set TB as Filled ● 0b0: No operation ● 0b1: The SLOT selected by TBPTR is marked as filled When the SLOT is marked as filled and TSFF=1, TBE is automatically reset to 0.	RW
5:3	*Reserved*	-	-
2:0	TBPTR	TB SLOT Pointer ● 0b000: Points to PTB ● 0b001: Points to STB SLOT1 ● 0b010: Points to STB SLOT2 ● 0b011: Points to STB SLOT3 ● Others: Setting prohibited The pointed TB SLOT can be read/written via TBUF and marked as filled or empty using TBE and TBF. In TTCAN mode, TBSEL and TSNEXT registers are invalid. **Note** This bit can only be written when TSFF=0.	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

21.4.20 TTCAN Time Trigger Configuration Register (CAN_TTCFG)

Offset	Bit Position							
0xBF	7	6	5	4	3	2	1	0
Reset	0x90							
Name	WTIE	WTIF	TEIF	TTIE	TTIF	T_PRESC		TTEN
Access	RW	RW	RW	RW	RW	RW		RW

Bit	Name	Functional Description	Access
7	WTIE	Trigger Watchdog Interrupt Enable ● 0b0: Disable ● 0b1: Enable	RW
6	WTIF	Trigger Watchdog Interrupt Flag Write 1 to clear flag, write 0 has no effect Read: ● 0b0: No flag generated ● 0b1: Generate flag When CYCLE COUNT value = TT_WTRIG setting and WTIE=1, WTIF is set.	RW
5	TEIF	Trigger Error Interrupt Flag Write 1 to clear flag, write 0 has no effect Read: ● 0b0: No flag generated ● 0b1: Generate flag When TT_TTIG setting is less than the actual CYCLE_TIME, TEIF is set.	RW
4	TTIE	Time Trigger Interrupt Enable ● 0b0: Disable ● 0b1: Enable	RW
3	TTIF	Time Trigger Interrupt Flag Write 1 to clear flag, write 0 has no effect Read: ● 0b0: No flag generated ● 0b1: Generate flag When CYCLE COUNT value = TT_TRIG setting and TTIE=1, TTIF is set. If TT_TRIG is not updated, TTIF is set only once, and is not set in the next basic CYCLE.	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Bit	Name	Functional Description	Access
2:1	T_PRESC	TTCAN Counter Prescaler ● 0b00: 1 division of the bit time set by the SBT register ● 0b01: 2 division of the bit time set by the SBT register ● 0b10: 4 division of the bit time set by the SBT register ● 0b11: 8 division of the bit time set by the SBT register **Note** T_PRESC can be written when TTEN=0 or when writing TTEN=1 simultaneously.	RW
0	TTEN	TTCAN Enable ● 0b0: Disable ● 0b1: Enable TTCAN, and the counter starts counting	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

21.4.21 TTCAN Reference Message Register (CAN_REF_MSG)

Offset	Bit Position																													
0xC0	31	30	29	28	27	26	25	24	23	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	REF_IDE	Reserved														REF_ID																
Access	RW															RW																

Bit	Name	Functional Description	Access
31	REF_IDE	IDE Bit for Reference Messages ● 0b0: Standard format ● 0b1: Extended format	RW
30:29	*Reserved*	-	-
28:0	REF_ID	Reference Message ID Bit ● REF_IDE=0: REF_ID[10:0] is valid ● REF_IDE=1: REF_ID[28:0] is valid REF_ID is used to detect reference messages, applicable to both transmission and reception. After detecting a reference message, the Sync_Mark of the current frame becomes Ref_Mark. REF_ID[2:0] is fixed at 0 and its value is not checked, supporting up to 8 potential time masters. After the highest byte write operation of REF_MSG, wait for 6 CAN clock cycles to complete the transfer of REF_MSG to the CAN clock domain.	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

21.4.22 TTCAN Trigger Configuration Register (CAN_TRG_CFG)

Offset	Bit Position															
0xC4	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	TEW				Reserved	TTYPE			Reserved					TTPTR		
Access	RW					RW								RW		

Bit	Name	Functional Description	Access
15:12	TEW	Transmit Enable Window Used for TTCAN Single Shot Transmit Trigger, a window of TEW+1 cycle time can be set, and transmission is only allowed within this window.	RW
11	*Reserved*	-	-
10:8	TTYPE	Trigger Type ● 0b000: Immediate Trigger for immediate transmission ● 0b001: Time Trigger for receive triggers ● 0b010: Single Shot Transmit Trigger for exclusive time windows ● 0b011: Transmit Start Trigger for merged arbitrating time windows ● 0b100: Transmit Stop Trigger for merged arbitrating time windows ● Others: Setting prohibited Trigger time is set through the TRG_TRIG register, and the TB Slot is selected through TTPTR.	RW
7:3	*Reserved*	-	-
2:0	TTPTR	Transmit Trigger TB slot Pointer ● 0b000: Points to PTB ● 0b001: Points to STB SLOT1 ● 0b010: Points to STB SLOT2 ● 0b011: Points to STB SLOT3 ● Others: Setting prohibited If the pointed TB SLOT is marked empty, TEIF is set when the trigger time is reached.	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

21.4.23 TTCAN Trigger Time Register (CAN_TRG_TRIG)

Offset	Bit Position															
0xC6	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	TT_TRIG															
Access	RW															

Bit	Name	Functional Description	Access
15:0	TT_TRIG	Trigger Time Used to specify the cycle time for the trigger. For transmit triggers, the transmit SOF time is approximately TT_TRIG setting + 1. After the highest byte write operation of TT_TRIG, the TT_TRIG value starts transferring to the CAN clock domain. Therefore, if writing by BYTE, write the low byte first and then the high byte.	RW

21.4.24 TTCAN Watch Trigger Time Register (CAN_TRG_WTRIG)

Offset	Bit Position															
0xC8	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xFFFF															
Name	TT_WTRIG															
Access	RW															

Bit	Name	Functional Description	Access
15:0	TT_WTRIG	Trigger Time Used to specify the cycle time for the watchdog trigger. After the highest byte write operation of TT_WTRIG, the TT_WTRIG value starts transferring to the CAN clock domain. Therefore, if writing by BYTE, write the low byte first and then the high byte.	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

21.5 Precautions

21.5.1 CAN Bus Anti-Interference Measures

The CAN bus is widely used in the automotive, industrial control, and other industries. If the electromagnetic environment of the CAN application site is harsh, with factors such as circuit imbalance, spatial electromagnetic fields, and power grid inlet lines, it can cause a large amount of communication noise on the CAN bus due to radiation and conduction interference. This leads to an increase in bus error frames, frequent retransmissions, and a failure for correct data to arrive in a timely manner, seriously affecting data communication quality. Therefore, in practical applications, efforts should be made to eliminate noise interference and ensure stable operation of the CAN bus network.

The following are several commonly used CAN bus anti-interference measures (including but not limited to):

- Add Electrical Isolation for the CAN Bus Interface
- Common transceiver signal ground
- Use shielded twisted pair cables and ground them properly
- Increase the twisting rate of CAN transmission lines
- Add signal protector
- Improve network topology
- Application layer software anti-jamming mechanism

21.5.2 CAN Controller Noise Constraints

In the CAN bus network, ensure that the communication bit time meets the requirements of the standard protocol. If noise interference that does not meet the bit time width is introduced, it may cause abnormal operation of the CAN controller.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

22 CRC

22.1 Introduction

CRC calculation unit takes data stream or block as input and generates output number controlled by generator polynomial. This output number is typically used to verify correctness and integrity of data transmission or storage. This module supports CRC value calculation and verification.

22.2 Features

- One implementation standard: ISO/IEC13239
- Two encoding methods: CRC-16, CRC-32
- Three write bit widths: 8-bit, 16-bit, 32-bit
- Two working modes: CRC encoding mode, CRC check mode
- CRC-16 polynomial: $x^{16}+ x^{12}+ x^5+ 1$
- CRC-32 polynomial: $x^{32}+ x^{26}+ x^{23}+ x^{22}+ x^{16}+ x^{12}+ x^{11}+ x^{10}+ x^8+ x^7+ x^5+ x^4+ x^2+ x + 1$

22.3 Functional Description

22.3.1 Working Mode

This module supports two working modes: CRC encoding mode and CRC check mode.

CRC encoding mode inputs a certain amount of original data to CRC module and obtains output value (CRC_RESULT). CRC check mode inputs a certain amount of encoded data sequence (original data + CRC check value) to verify match between original data and CRC check value (CRC_CSR.FLAG).

22.3.2 Encoding Method

This module supports two encoding methods CRC-16 and CRC-32, with 16-bit and 32-bit results respectively, CRC_CSR.CODE configurable as needed.

CRC-16 Encoding

In this mode, the CRC algorithm model is CRC-16-X25:

- Polynomial: $x^{16}+x^{12}+x^5+1$;
- Initial value: 0xFFFFu;
- Result XOR value: 0xFFFFu;
- Input data inversion: True;
- Output data inversion: True.

CRC-32 Encoding

In this mode, the CRC algorithm model is CRC-32;

- Polynomial: $x^{32}+ x^{26}+ x^{23}+ x^{22}+ x^{16}+ x^{12}+ x^{11}+ x^{10}+ x^8+ x^7+ x^5+ x^4+ x^2+ x + 1$;
- Initial value: 0xFFFFFFFFu;
- Result XOR value: 0xFFFFFFFFu;
- Input data inversion: True;
- Output data inversion: True.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

22.3.3 Write Bit Width

This module supports three write bit widths: 8-bit, 16-bit, 32-bit. Different bit width writes must follow "bit width consistency, low-to-high" principle: "each write must use register matching data width, with lower bits written before higher bits".

Below demonstrates same data sequence written in three bit widths with identical output results.

- 8-bit width write: 0x00, 0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x77
- 16-bit width write: 0x1100, 0x3322, 0x5544, 0x7766
- 32-bit width write: 0x33221100, 0x77665544

22.4 Operation Example

22.4.1 CRC-16 Encoding Mode

Procedure

Step1. Write CRC_CSR.CODE as 0b0 to select CRC-16.

Step2. Write CRC_RESULT[15:0] as 0xFFFF to initialize CRC calculation.

Step3. Write the raw data to be encoded sequentially to the CRC_DATA register, with selectable write bit width of 8-bit, 16-bit, or 32-bit.

Step4. Read CRC_RESULT[15:0] to obtain the CRC-16 calculation result.

22.4.2 CRC-16 Check Mode

Procedure

Step1. Write CRC_CSR.CODE as 0b0 to select CRC-16.

Step2. Write CRC_RESULT [15:0] as 0xFFFF to initialize CRC calculation.

Step3. Write the encoded data sequence (original data + CRC check value) into CRC_DATA register in sequence, with selectable write bit width of 8-bit, 16-bit, or 32-bit.

Step4. Determine whether the encoded data sequence has been tampered based on the value of CRC_CSR.FLAG.

22.4.3 CRC-32 Encoding Mode

Procedure

Step1. Write CRC_CSR.CODE as 0b1 to select CRC-32.

Step2. Write CRC_RESULT[31:0] as 0xFFFFFFFF to initialize CRC calculation.

Step3. Write the raw data to be encoded sequentially to the CRC_DATA register, with selectable write bit width of 8-bit, 16-bit, or 32-bit.

Step4. Read CRC_RESULT[31:0] to obtain the CRC-32 calculation result.

22.4.4 CRC-32 Check Mode

Procedure

Step1. Write CRC_CSR.CODE as 0b1 to select CRC-32.

Step2. Write CRC_RESULT[31:0] as 0xFFFFFFFF to initialize CRC calculation.

Step3. Write the encoded data sequence (original data + CRC check value) into CRC_DATA register in sequence, with selectable write bit width of 8-bit, 16-bit, or 32-bit.

Step4. Determine whether the encoded data sequence has been tampered based on the value of CRC_CSR.FLAG.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

22.5 Register Description

22.5.1 Register List

Base address: 0x40020800

Table 22-1 Cyclic Redundancy Check (CRC) Register Offset

Offset	Register	Description	Width
0x00	CRC_CSR	Control and Status Register	32
0x04	CRC_RESULT	Result Register	32
0x80	CRC_DATA	Data Register	32

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

22.5.2 Control and Status Register (CRC_CSR)

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																													FLAG	CODE	
Access																														RO	RW	

Bit	Name	Functional Description	Access
31:2	*Reserved*	-	-
1	FLAG	CRC Check Result ● 0b0: Current CRC check error ● 0b1: Current CRC check correct	RO
0	CODE	CRC Encoding Method Selection ● 0b0: CRC-16 code ● 0b1: CRC-32 code	RW

22.5.3 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	CRC Calculation Result Read RESULT[15:0] to obtain the CRC-16 calculation result. Read RESULT[31:0] to obtain the CRC-32 calculation result. Write 0xFFFF to RESULT[15:0] to initialize CRC-16 calculation. Write 0xFFFFFFFF to RESULT[31:0] to initialize CRC-32 calculation.	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

22.5.4 Data Register (CRC_DATA)

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	DATA																															
Access	WO																															

Bit	Name	Functional Description	Access
31:0	DATA	Data for Operation Used to write the data to be operated on, supporting 3 write bit widths: ● 8-bit write mode: * ((uint8_t *)0x40020880) = 0xXX ● 16-bit write mode: * ((uint16_t *)0x40020880) = 0xXXXX ● 32-bit write mode: * ((uint32_t *)0x40020880) = 0xXXXXXXXX	WO

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

23 I2CSLV

23.1 Introduction

The I2C slave interface is a 2-wire bidirectional serial bus compatible with the I2C 3.0 bus specification. It achieves a transmission rate of 3.4Mbps in high-speed mode. The software can write the I2C module, and the I2C slave module can automatically control the transmission of serial data. This interface also supports the clock extension function. For example, when sending or processing received data, it may be necessary to temporarily prohibit transmission.

This module only acts as an I2C slave device, and its control function includes SCL clock extension function.

23.2 Features

- Transmission rate: standard (about 100kbps), fast (400kbps), ultra-fast (1Mbps), and high speed (3.4Mbps, the module working clock PCLK must be no less than 24MHz)
- Only supports slave mode
- Supports clock low-level extension function to interface with faster masters
- Supports 7-bit slave address and 10-bit slave address
- Support 4 slave addresses
- Support SCL low level timeout
- Support Device ID reading

23.3 Protocol Description

For protocols other than high-speed communication, please refer to the "I2C Bus (I2C)" chapter. The description of high-speed related protocols is as follows:

The I2C specification supports high-speed transmission, allowing transmission rates up to 3.4Mbps and being backward compatible with lower-speed devices. This allows high-speed mode devices to operate in bus systems with different speeds. For electrical and timing details of high-speed mode, refer to the I2C specification. This module is compatible with high-speed mode and does not require software participation.

The I2C bus only works in fast mode by default, with a maximum rate of 400kbps. The following sequence on the I2C bus switches from fast to high-speed mode:

1. START (S)
2. 8-digit master code (0b00001XXX)
3. NACK bit (N)

The high-speed mode master code is a reserved 8-bit code and is not used for slave addressing or other purposes. A master compatible with high-speed mode can switch to high-speed mode by sending the above sequence when the rate does not exceed 400kbps. After this sequence the master's transfer rate in high speed mode reaches 3.4Mbps. The master returns to fast/standard mode after sending the STOP bit.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

23.4 Functional Description

This module only works in slave mode. The hardware provides timing and shift control for serial transmission; the upper-layer protocol is determined by user software. The hardware interface provides the following applications:

- Serial transfer of data by byte
- SDA data synchronization
- Timeout
- START/STOP detection
- Interrupt generation
- Status information
- High speed mode detection
- The hardware identifies the slave address and automatically sends a response
- After the address is matched, the corresponding matching flag will be set to 1.

23.4.1 Function Block Diagram

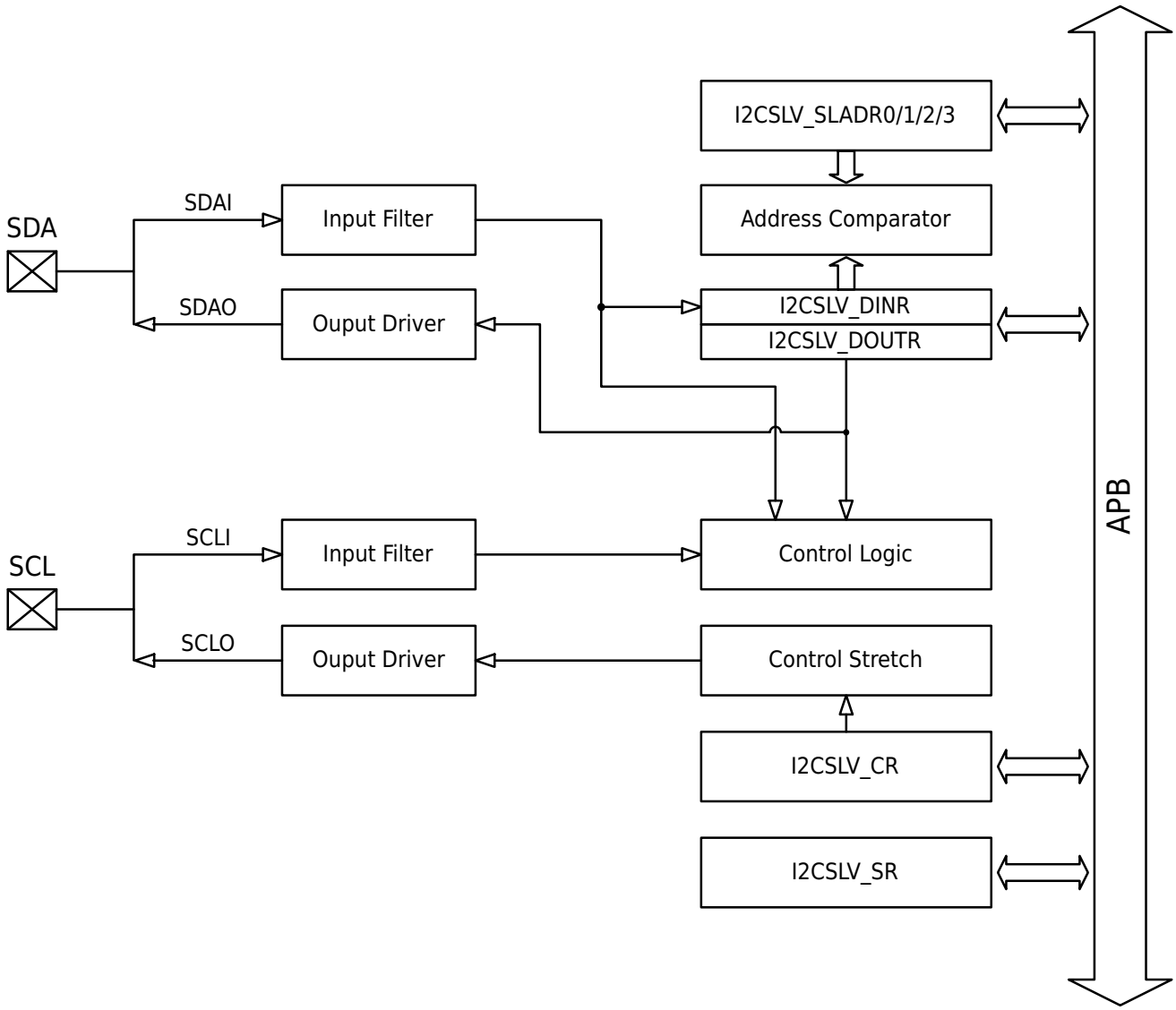


Figure 23-1 I2C SLAVE Module Block Diagram

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

23.4.2 Auto Address Recognition

After this module configures the slave address, the hardware automatically responds with ACK without software involvement. This feature is enabled by software:

1. Set I2CSLV_CR.BUSY=0 so that it automatically responds to ACK.
2. Write the slave address to the I2CSLV_SLADR0/1/2/3 register.
3. Set IO to SCL and SDA pins, that is, open-drain enable.
4. Set I2CSLV_CR.EN=1 to start this module.

The slave address mask (I2CSLV_ADMR0.ADDRMASK) bit is used by the hardware to automatically respond to multiple slave addresses. "0" in I2CSLV_ADMR0.ADDRMASK indicates a don't care bit and does not participate in address matching. If a unique matching address needs to be given, I2CSLV_ADMR0.ADDRMASK must be 0x3FF or 0x7F (10-bit address mode or 7-bit address mode). The relationship between the matching address and these two registers is:

Matching address = I2CSLV_SLADR0.ADDR & I2CSLV_ADMR0.ADDRMASK

For example, if I2CSLV_SLADR0.ADDR = 0b1101001 and I2CSLV_ADMR0.ADDRMASK = 0b1111011, then the matching address would be 0b1101x01, which is 0b1101001 and 0b1101101.

When the addresses match, the matching address will be placed in the slave receive address register I2CSLV_ADIR, allowing software to check which address was used in the transmission. Software can use the I2CSLV_ADIR register to read the address before processing the data. The slave compares the received address with its own four addresses respectively. As long as there is a match, it will send a corresponding response according to the CR.BUSY value.

23.4.3 Operating Mode

This module supports two types of data transmission: transmitting and receiving. The external master initiates both types of transfers and provides an SCL. This section describes the register settings in these two modes in detail. In both modes, this module will determine the receiving or transmitting status after ACK or NACK. If the reception is full or the transmission is empty, the clock will be automatically extended.

Master Transmits Slave Receive Sequence

Steps:

Step1. The input START and "address and W" bytes cause the module to exit idle mode. If the addresses match and CR.BUSY (complete register) = 0, the module automatically replies ACK.

Step2. An interrupt will be generated after the ACK in the address phase. This module automatically determines whether the received data is full. If it is full, it will pull SCL low and wait until the software writes I2CSLV_ICR.I2CINT=0 before releasing it; If the software writes I2CSLV_CR.BUSY=1, SCL will no longer be pulled low after receiving the current data (that is, it will no longer respond to the master). The software executes when Figure 23-3 : I2C Master Transmitting and Slave Receiving Operation Flow Diagram pressed.

Step3. The software reads 1 byte of data in a receive interrupt, and the hardware replies ACK or NACK according to I2CSLV_CR.BUSY.

Step4. The master sends STOP after the entire data transfer is completed.

Figure 23-2 : Example of I2C Master Transmitting and Slave Receiving Sequence example is given, including various states, Figure 23-3 : I2C Master Transmitting and Slave Receiving Operation Flow Diagram describing this sequence and software interrupt operations.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

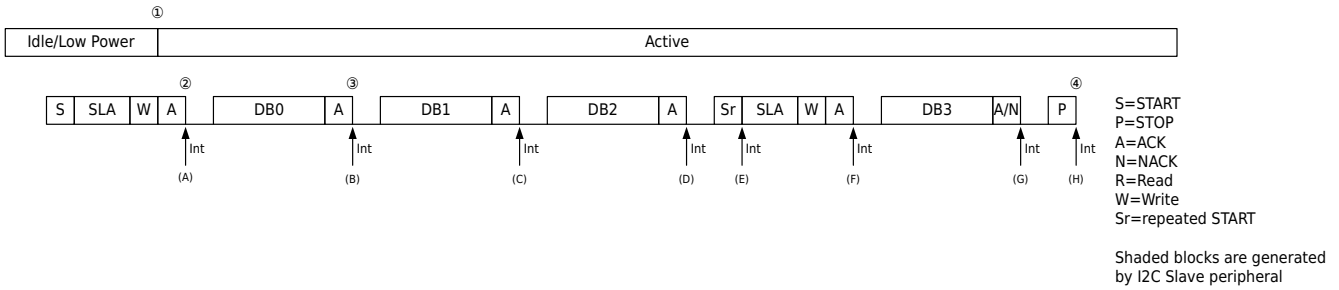


Figure 23-2 Example of I2C Master Transmitting and Slave Receiving Sequence

Note

In step F of the above figure, the software sets I2CSLV_CR.BUSY to 0. If the master receives an ACK in step G, it can still send a STOP after the ACK.

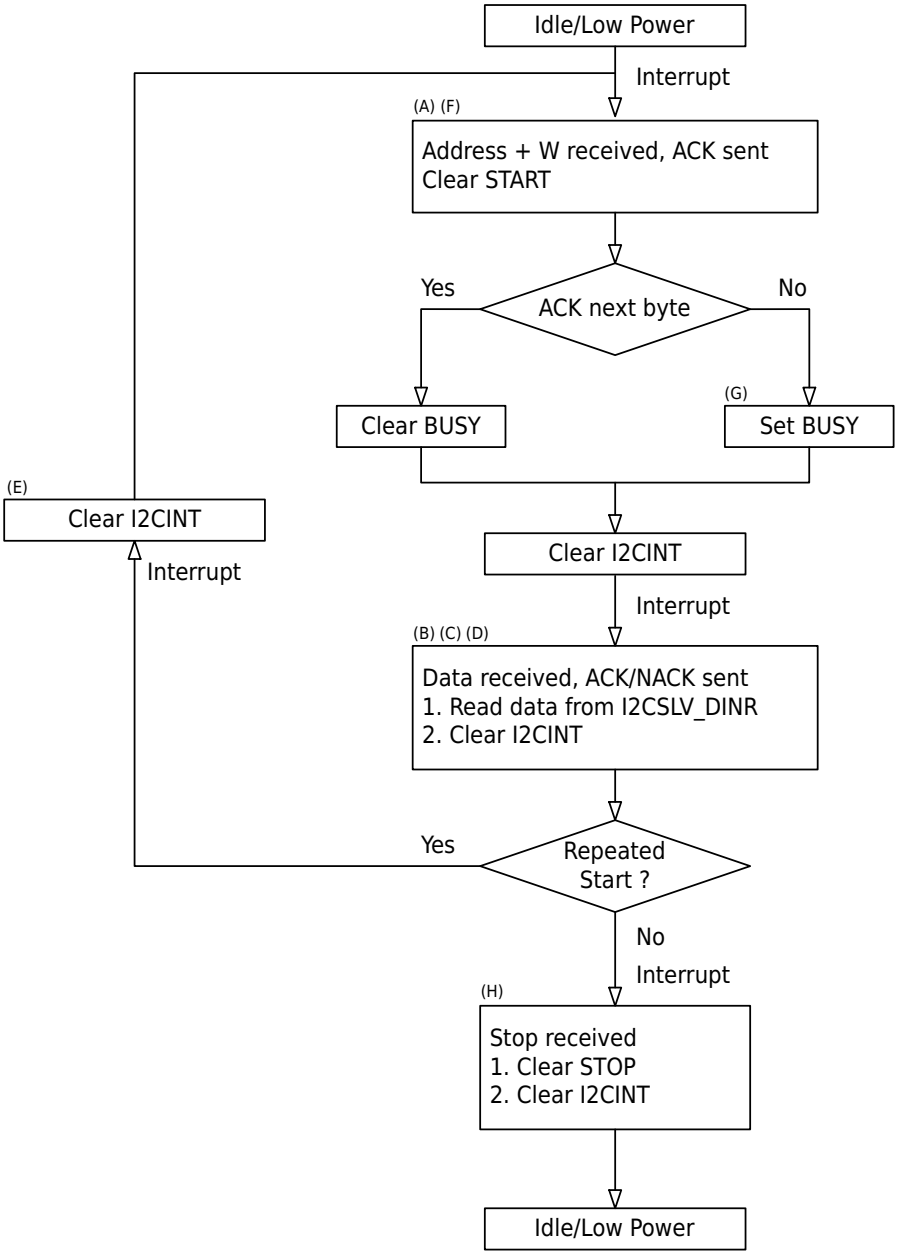


Figure 23-3 I2C Master Transmitting and Slave Receiving Operation Flow Diagram

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

The Master Receives the Sequence Sent by The Slave

Steps:

Step1. The input START and "address and R" bytes cause the module to exit idle mode. If the addresses match and I2CSLV_CR.BUSY=0, this module automatically replies ACK.

Step2. An interrupt will be generated after the ACK in the address phase. This module automatically determines whether the sent data is empty. If it is empty, it will pull SCL low and wait until the software writes I2CSLV_ICR.I2CINT=0 before releasing it; If the hardware receives a NACK signal, it will not pull SCL low. The software executes when Figure 23-5 : I2C Master Receiving and Slave Transmitting Operation Flow Diagram pressed.

Step3. The software writes 1 byte of data in a send interrupt.

Step4. After the current data transmission is completed, the master sends NACK, and then sends START or STOP.

Step5. After all data is transmitted, the master issues STOP.

Figure 23-4 : Example of I2C Master Receiving and Slave Transmitting Sequence Sequence examples are given, including RESTART, Figure 23-5 : I2C Master Receiving and Slave Transmitting Operation Flow Diagram describing this sequence and software interrupt operation.

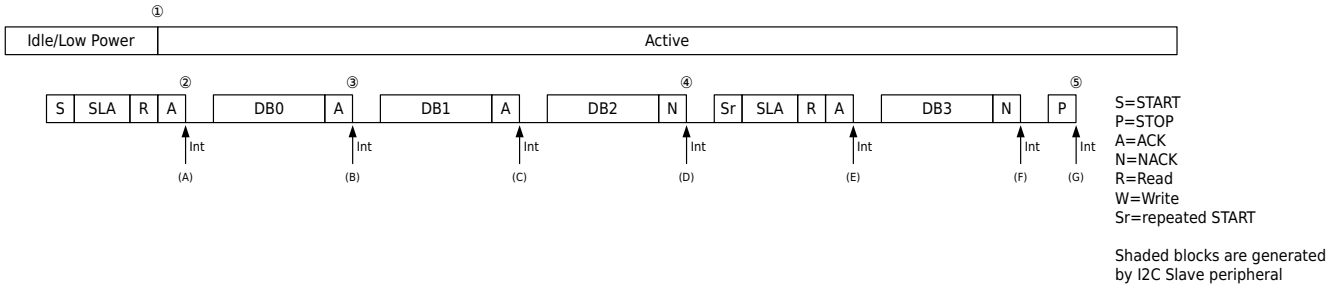


Figure 23-4 Example of I2C Master Receiving and Slave Transmitting Sequence

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

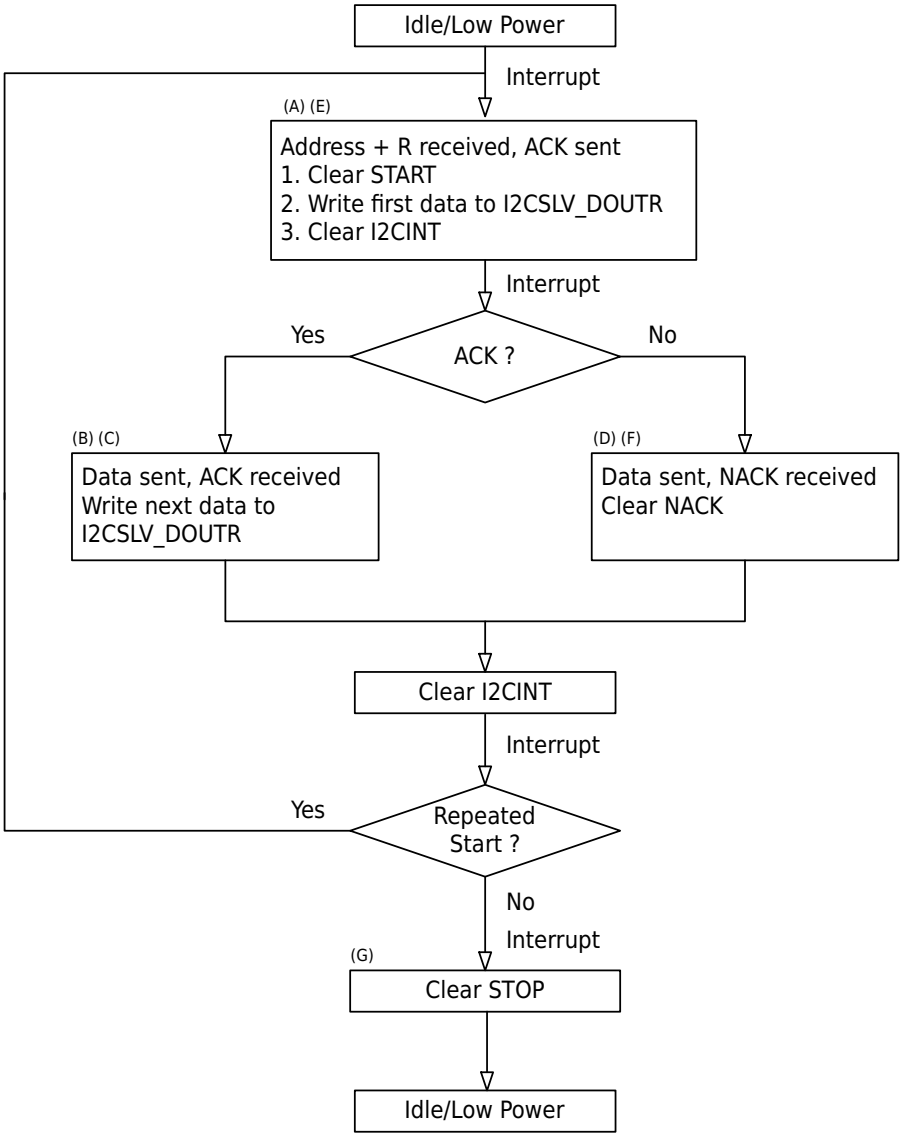


Figure 23-5 I2C Master Receiving and Slave Transmitting Operation Flow Diagram

Note

The master must send NACK before sending RESTART or STOP bit. NACK causes the slave to release the SDA line so that the master can issue a START or STOP signal.

23.4.4 Status Decoding

The current status is easily decoded through the I2CSLV_SR register. The table below describes the typical operations of the software in each state. The state vector in the table refers to the lower 5 bits of the I2CSLV_SR register: NACK, START, STOP, WR and RD. The table is just a typical response, and it is allowed as long as the application process follows the I2C specification.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Note

Multiple interrupt sources (STOP, START, RD, WR, etc.) can accumulate, so the actual state vector may be triggered by other conditions and may not be the same as that listed in the table. In this case, the sequence of operations should be:

1. Handling the STOP bit
2. Handling the START bit
3. Handle START+RD or START+WR bit
4. Handle RD or WR bit

Table 23-1 I2C Status Decoding

Mode	Current Status Vector	I2C Current Status	Desired Operation	Next Expectation Vector
The master transmits and slave receives	0b01010	The slave receives START+address+W, ACK has been transmitted	The slave clears SR.START and SR.I2CINT	0b00010
	0b00010	The slave has received data and ACK has been transmitted	The slave reads the data in I2C_DINR and writes ICR.I2CINT=0 ● If the slave replies NACK on the next byte, write CR.BUSY=1 ● If the slave replies ACK on the next byte, CR.BUSY=0 will be kept.	0b00010 or 0b10010 or 0b00100
	0b10010	The slave has received data and NACK has been transmitted	The slave reads the data in I2C_DINR and writes ICR.I2CINT=0 ● If the slave replies NACK on the next byte, write CR.BUSY=1 ● If the slave replies ACK on the next byte, CR.BUSY=0 will be kept.	0b00010 or 0b10010 or 0b00100
	0b00000	The slave receives RESTART	The slave writes ICR.I2CINT=0	0b01010
	0b00100	The slave receives STOP	The slave writes ICR.STOP=0 and ICR.I2CINT=0	-

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Mode	Current Status Vector	I2C Current Status	Desired Operation	Next Expectation Vector
The master receives and the slave transmits	0b01001	The slave receives START+address+R, ACK has been transmitted	The slave writes ICR.START=0, writes data in I2CDOUT and writes ICR.I2CINT=0	0b00001
	0b00001	The slave has transmitted data and receive ACK from master	The slave writes data in I2CDOUT and writes ICR.I2CINT=0	0b00100
	0b10001	The slave has transmitted data and receive NACK from master	The slave writes ICR.NACK=0 and ICR.I2CINT=0	0b00100
	0b00100	The slave receives STOP	The slave writes ICR.STOP=0 and ICR.I2CINT=0	-

23.4.5 Filtering

Setting the I2CSLV_FLT register can filter the input SCL and SDA. The working clock of the filter is the same as the working clock of this module, and it can filter out noise with an input width of 1/2/3/4/8/12/16 PCLK clocks. By default the filter does not work.

23.4.6 Sampling Location

For the input data on the SDA, the sampling location can be specified to meet the needs of different occasions. By default, SDA is sampled on the falling edge of SCL. Configuring the I2CSLV_RSR register can choose to sample 3 positions before the falling edge of SCL.

23.4.7 Edge Shaping Output

The data line SDA is an open-drain output. In order to shorten the rise time for the low level to high level on the SDA line, the edge shaping register I2CSLV_ODDR can be configured. This causes SDA to have a CMOS high-level output for 1/2/4 working clocks when the low level changes to high level. Once these 1/2/4 clocks end, the SDA is still an open-drain output. By default, SDA has no edge shaping output.

23.4.8 Timeout Function

If the SCL line is pulled low by a slave device on the bus, no further communication is possible. And the master cannot force SCL high to correct this error. To solve this problem, the I2C protocol stipulates that devices participating in transmission must detect the low level of the clock for more than 25ms as a timeout condition. Devices that detect a timeout must restart communication within 10ms of the timeout.

There is a dedicated timer in this module to implement SCL low level timeout counting. Set the TIMEOUT bit in the I2CSLV_CR register to 1 to start the timeout counter. The counter is in reset state when SCL is high and starts counting from zero when it is low. Once the count value exceeds I2CSLV_TOR, an interrupt will be generated (if I2CSLV_IER.TIMEOUT=1), and the interrupt routine should release the SCL line or close/restart the I2C communication.

Timeout length= (TOR + 1)*64*(1/$PCLK_{I2C\_SLV}$)

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Envolves:

- TOR: register I2CSLV_TOR value
- $PCLK_{I2C\_SLV}$: working clock of this module

23.4.9 Device ID Reading

Users can configure the Device ID register, which consists of 3 bytes: 12-digit manufacturer name, 9-digit part identification, and 3-digit revision information. The steps for the master to read the Device ID are as follows:

Step1. The master sends the START command.

Step2. The master sends the Device ID address, which is a reserved address, in which the R/W bit value is 0, that is, "0b11111000".

Step3. The master sends the slave address, the lowest bit of which can be 1 or 0.

Step4. The master sends RESTART command.

Step5. The master sends the Device ID address again, in which the R/W bit value is 1, that is, "0b11111001".

Step6. The slave first sends 12-bit manufacturer name data (the first byte and the high 4 bits of the second byte), then the 9-bit part identification data (the lower 4 bits of the second byte and the upper 5 bits of the third byte), the last is 3 bits of revision information data (lower 3 bits of the third byte).

Step7. The master sends NACK at the last byte to end the Device ID reading operation, and then sends a STOP command..

23.4.10 Interrupt

This module has 17 interrupt sources that can trigger the CPU to enter the interrupt program. Once any bit in the interrupt enable register I2CSLV_IER is set to 1, if the corresponding bit in the status register is also set to 1, an interrupt will be triggered.

23.5 Operation Example

23.5.1 Master Transmitting and Slave Receiving Data in Non-High-Speed Mode

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. Configure the SYSCTRL_PERI_CLKEN0: turn on the I2C SLAVE working clock.

Step3. Configure the I2CSLV register: Except for CR.EN, the registers can be configured in any order depending on the required performance.

Step4. Clear all I2CSLV interrupts: write I2CSLV_ICR=0.

Step5. Start I2CSLV: write I2CSLV_CR.EN=1.

Step6. Wait for I2CSLV_SR.RXNE to become 1, and the slave software reads the data in I2CSLV_DINR; repeat this step as many times as needed.

Step7. After reading the penultimate data, wait for I2CSLV_SR.ACK=1 and then write I2CSLV_CR.BUSY=1.

Step8. After reading the last data, write I2CSLV_CR.EN=0.

Steps 6-8 can also be completed with an I2CSLV_IER.WR interrupt program.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

23.5.2 Master Receiving and Slave Transmitting Data in Non-High-Speed Mode

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. Configure the SYSCTRL_PERI_CLKEN0: turn on the I2C SLAVE working clock.

Step3. Configure the I2CSLV register: Except for I2CSLV_CR.EN, the registers can be configured in any order depending on the required performance.

Step4. Clear all I2CSLV interrupts: write I2CSLV_ICR=0.

Step5. Start I2CSLV: write I2CSLV_CR.EN=1.

Step6. Wait for I2CSLV_SR.RD to become 1.

Step7. Wait for I2CSLV_SR.TXE to be 1 and the slave software writes data to I2CSLV_DOUTR; repeat this step as many times as needed.

Step8. Wait for I2CSLV_SR.NACK to become 1 and then write I2CSLV_CR.EN=0.

Steps 6-8 can also be completed with an interrupt program.

23.5.3 Master Transmitting and Slave Receiving Data in High-Speed Mode

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. Configure the SYSCTRL_PERI_CLKEN0: turn on the I2C SLAVE working clock.

Step3. Configure the I2CSLV register: Except for I2CSLV_CR.EN, the registers can be configured in any order depending on the required performance.

Step4. Clear all I2CSLV interrupts: write I2CSLV_ICR=0.

Step5. Start I2CSLV: write I2CSLV_CR.EN=1.

Step6. Wait for I2CSLV_SR.HSMODE to become 1. If the current module clock is less than 24M, increase the clock frequency.

Step7. Wait for I2CSLV_SR.RXNE to become 1, and the slave software reads the data in I2CSLV_DINR; repeat this step as many times as needed.

Step8. After reading the penultimate data, wait for I2CSLV_SR.ACK=1 and then write I2CSLV_CR.BUSY=1.

Step9. After reading the last data, write I2CSLV_CR.EN=0.

Steps 7-9 can also be completed with an I2CSLV_IER.WR interrupt program.

23.5.4 Master Receiving and Slave Transmitting Data in High-Speed Mode

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. Configure the SYSCTRL_PERI_CLKEN0: turn on the I2C SLAVE working clock.

Step3. Configure the I2CSLV register: Except for I2CSLV_CR.EN, the registers can be configured in any order depending on the required performance.

Step4. Clear all I2CSLV interrupts: write I2CSLV_ICR=0.

Step5. Start I2CSLV: write I2CSLV_CR.EN=1.

Step6. Wait for I2CSLV_SR.HSMODE to become 1. If the current module clock is less than 24M, increase the clock frequency..

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Step7. Wait for I2CSLV_SR.RD to become 1.

Step8. Wait for I2CSLV_SR.TXE to be 1 and the slave software writes data to I2CSLV_DOUTR; repeat this step as many times as needed.

Step9. Wait for I2CSLV_SR.NACK to become 1 and then write I2CSLV_CR.EN=0.

Steps 7-9 can also be completed with an interrupt program.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

23.6 Register Description

23.6.1 Register List

Base address: 0x40004C00

Table 23-2 I2CSLV Slave Module (I2CSLV) Register Offset

Offset	Register	Description	Width
0x00	I2CSLV_SR	Status Register	32
0x04	I2CSLV_DOUTR	Data Transmit Register	32
0x08	I2CSLV_ADIR	Slave Receive Address Register	32
0x0C+0x4*n	I2CSLV_SLADRn	Slave Address n Register (n=0-3)	32
0x1C	I2CSLV_DINR	Data Receive Register	32
0x20	I2CSLV_CR	Control Register	32
0x30	I2CSLV_ADMR0	Slave Address 0 Mask Register	32
0x40	I2CSLV_FLT	Filter Register	32
0x44	I2CSLV_TOR	Timeout Register	32
0x48	I2CSLV_ODDR	Edge Shaping Register	32
0x4C	I2CSLV_RSR	Receive Sampling Location Register	32
0x50	I2CSLV_IER	Interrupt Enable Register	32
0x54	I2CSLV_ICR	Interrupt Flag Clear Register	32
0x58	I2CSLV_DIDR	Device ID Register	32

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

23.6.2 Status Register (I2CSLV_SR)

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	0x00004000																															
Name	Reserved															GC	RXNE	TXE	SLADDR3	SLADDR2	SLADDR1	SLADDR0	ACK	TIMEOUT	HSMODE	ACTIVE	I2CINT	NACK	START	STOP	WR	RD
Access																RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO

Bit	Name	Functional Description	Access
31:17	*Reserved*	-	-
16	GC	Broadcast address flag ● 0b0: No match with the broadcast address. ● 0b1: Match with the broadcast address. It is automatically cleared after receiving STOP; it can also be cleared by software writing I2CSLV_ICR.GC=0.	RO
15	RXNE	Receive register not empty flag ● 0b0: There is no data in the receiving register ● 0b1: The receiving register already has data Software reads the data in I2CSLV_DINR and clears it; software can also write I2CSLV_ICR.RXNE=0 to clear it.	RO
14	TXE	Send register empty flag ● 0b0: There is data in the sending register ● 0b1: The sending register is empty When I2CSLV_CR.EN=1, the software writes data to I2CSLV_DOUTR, and TXE is cleared by hardware; TXE is set to 1 after the hardware moves the data into the transmit shift register. Software can also write I2CSLV_ICR.TXE=0 to set I2CSLV_SR.TXE to 1.	RO
13	SLADDR3	Flag matching slave address 3 ● 0b0: No Address matching slave address 3 detected ● 0b1: Address matching slave address 3 detected Software writes I2CSLV_ICR.SLADDR3=0 to clear.	RO
12	SLADDR2	Flag matching slave address 2 ● 0b0: No Address matching slave address 2 detected ● 0b1: Address matching slave address 2 detected Software writes I2CSLV_ICR.SLADDR2=0 to clear.	RO
11	SLADDR1	Flag matching slave address 1 ● 0b0: No Address matching slave address 1 detected ● 0b1: Address matching slave address 1 detected Software writes I2CSLV_ICR.SLADDR1=0 to clear.	RO

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Bit	Name	Functional Description	Access
10	SLADDR0	Flag matching slave address 0 ● 0b0: No Address matching slave address 0 detected ● 0b1: Address matching slave address 0 detected Software writes I2CSLV_ICR.SLADDR0=0 to clear.	RO
9	ACK	Acknowledge bit It will be set to 1 by hardware when any of the following conditions are met: ● When I2CSLV_SR.ACTIVE=1, the master or slave sends an ACK signal ● The slave sends an ACK signal to the matching address The hardware automatically clears to 0 at the end of the 8th data transmitted next time or when START/RESTART or STOP is received. Software can also write I2CSLV_ICR.ACK=0 to clear it.	RO
8	TIMEOUT	Timeout flag Set to 1 by hardware after timeout; cleared to 0 by software when writing I2CSLV_ICR.TIMEOUT=0.	RO
7	HSMODE	High speed mode Set to 1 by hardware when high-speed master code is received, and cleared to 0 automatically after STOP. Under abnormal circumstances, software can write I2CSLV_ICR.HSMODE=0 to clear it.	RO
6	ACTIVE	Bus active It is set to 1 by hardware when the address matches. After the transmission is completed, whether it is STOP or NACK, it is automatically cleared to 0; After the broadcast address is enabled, it will also be set to 1 by hardware when receiving the broadcast address; it will also be automatically cleared to 0 if RESTART is received. Under abnormal circumstances (such as address mismatch or read and write bit errors when the master reads the Device ID), it will also be automatically cleared to 0. Under abnormal circumstances, software can write I2CSLV_ICR.ACTIVE=0 to clear it.	RO
5	I2CINT	I2C interruption Set to 1 when RD, (WR) or STOP occurs. When I2CSLV_SR.ACTIVE becomes 0, it will also be set to 1, indicating that the transfer is completed. If BUSY=0 when receiving the broadcast address, this bit is set to 1. Set to 1 when HSMODE reserved address (00001) is detected. This bit will generate an interrupt, and the hardware will automatically clear it to 0 after the RD and WR bits are cleared. Or Software writes I2CSLV_ICR.I2CINT=0 to clear.	RO
4	NACK	Non-acknowledge bit It will be set to 1 by hardware when any of the following conditions are met: ● When I2CSLV_SR.ACTIVE=1, the master or slave sends an NACK signal ● The slave sends an NACK signal to the matching address Software can write I2CSLV_ICR.NACK=0 to clear it.	RO

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Bit	Name	Functional Description	Access
3	START	Start bit Set by hardware when receiving START signal. Software must write I2CSLV_ICR.START=0 to clear this bit.	RO
2	STOP	Stop bit When STOP is received when I2CSLV_SR.ACTIVE=1, it will be set to 1 by hardware. Software must write I2CSLV_ICR.STOP=0 to clear this bit.	RO
1	WR	Master sends and slave receives When any of the following conditions is met, the hardware sets it to 1 on the falling edge of the ninth SCL: ● The slave responds with ACK and I2CSLV_DINR is full ● The slave responds with NACK and I2CSLV_DINR is full ● The current transmission is a matching address, bit 8 is "W" (0, master transmits and slave receives) and I2CSLV_ADMR0.FACS is 1. This bit will set I2CSLV_SR.I2CINT to 1 and generate an interrupt. Software must write I2CSLV_ICR.WR=0 to clear this bit.	RO
0	RD	The master receives and the slave sends When any of the following conditions is met, the hardware sets it to 1 on the falling edge of the ninth SCL: ● The master responds with ACK (data or address) and I2CSLV_DOUTR has no data ● The master responds with NACK ● The current transmission is a matching address, bit 8 is "R" (1, the master receives the slave transmission) and I2CSLV_ADMR0.FACS is 1. This bit will set I2CSLV_SR.I2CINT to 1 and generate an interrupt. Software must write I2CSLV_ICR.RD=0 to clear this bit. The host will not set it to 1 when reading the Device ID.	RO

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

23.6.3 Data Transmit Register (I2CSLV_DOUTR)

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																								DOUT							
Access																									RW							

Bit	Name	Functional Description	Access
31:8	*Reserved*	-	-
7:0	DOUT	I2C data to be sent The software writes the data to be sent into this register	RW

23.6.4 Slave Receive Address Register (I2CSLV_ADIR)

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																					ADRLEN	ADIR									
Access																						RO	RO									

Bit	Name	Functional Description	Access
31:11	*Reserved*	-	-
10	ADRLEN	Address length indication received ● 0b0: 7-bit address mode ● 0b1: 10-bit address mode	RO
9:0	ADIR	Received address If it is 7-bit address mode, ADIR[9:7]=0b000 If it is a 10-bit address mode, all ADIR bits are valid	RO

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

23.6.5 Slave Address n Register (I2CSLV_SLADRn) (n=0-3)

Offset: 0x0C+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	Reserved																					AM	ADDR									
Access																						RW	RW									

Bit	Name	Functional Description	Access
31:11	*Reserved*	-	-
10	AM	Slave address length ● 0b0: 7-bit address ● 0b1: 10-bit address	RW
9:0	ADDR	I2C slave address This area is combined with I2CSLV_ADMR0.ADDRMASK to form the slave address of hardware automatic response: ● AM= 0b0: ADDR[6:0] is a valid slave address ● AM= 0b1: ADDR[9:0] is a valid slave address	RW

23.6.6 Data Receive Register (I2CSLV_DINR)

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																								DIN							
Access																									RO							

Bit	Name	Functional Description	Access
31:8	*Reserved*	-	-
7:0	DIN	Data received by I2C	RO

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

23.6.7 Control Register (I2CSLV_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	0x00000002																															
Name	Reserved																						DMARE	DMATE	GC	Reserved			TIMEOUT	FSCLC	BUSY	EN
Access																							RW	RW	RW				RW	RW	RW	RW

Bit	Name	Functional Description	Access
31:10	*Reserved*	-	-
9	DMARE	The master sends the DMA enable that the slave receives ● 0b0: DMA disabled ● 0b1: DMA enabled If DMARE=1, the hardware detects that data is written to I2C_DINR and issues a DMA request.	RW
8	DMATE	The master receives the DMA enable sent from the slave ● 0b0: DMA disabled ● 0b1: DMA enabled If DMATE=1, the hardware detects that the I2C_DOUTR data is read and issues a DMA request.	RW
7	GC	Broadcast address enable bit ● 0b0: Disable ● 0b1: Enable	RW
6:4	*Reserved*	-	-
3	TIMEOUT	SCL low level timeout enable ● 0b0: SCL low level timeout disabled ● 0b1: SCL low level timeout enable The timeout counter starts counting after SCL goes low; the counter is automatically cleared after SCL goes high. The timeout interrupt routine should restart I2C communication.	RW
2	FSCLC	High speed (3.4M) mode control ● 0b0: Standard and fast mode When the master receives the slave transmission operation, the data does not need to be preloaded into I2CSLV_DOUTR, and the data can be written in the interrupt program. If no data has been written (I2CSLV_SR.TXE=1), this module sends an ACK (acknowledge) signal in the ACK stage, and then forcefully pull SCL low to wait for software to write data to I2CSLV_DOUTR. ● 0b1: 3.4M When the master receives the slave sending operation, it must write the data to the I2CSLV_DOUTR register before the 8th clock of the matching address stage. If no data has been written (I2CSLV_SR.TXE=1), the module sends a NACK (non acknowledge) signal in the ACK stage.	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Bit	Name	Functional Description	Access
2	FSCLC	**Note** If I2CSLV_SR.HSMODE=1 but the program does not set FSCLC=1 and I2CSLV_SR.TXE=1, then in high-speed (3.4M) mode, this module will force SCL low to wait for the software to write data to I2CSLV_DOUTR.	RW
1	BUSY	The slave responds to control ● 0b0: The slave will respond to the master ● 0b1: The slave will not respond to the master. All data sent to the slave receives a NACK response	RW
0	EN	I2C enable Slave module is enabled. Before enabling the module, the open-drain enable function for the ports must be completed first: ● 0b0: Disable slave module ● 0b1: Enable the slave module	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

23.6.8 Slave Address 0 Mask Register (I2CSLV_ADMR0)

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	0x000007FF																															
Name	Reserved																					FACS	ADDRMASK									
Access																						RW	RW									

Bit	Name	Functional Description	Access
31:11	*Reserved*	-	-
10	FACS	Address clock forced extension ● 0b0: The I2CSLV_SR.I2CINT bit is not set to 1 by the address response operation alone and requires other conditions to be set to 1 ● 0b1: After the address matches and ACK is returned, the I2CSLV_SR.I2CINT bit will always be set to 1. This will force the clock to stretch until software writes I2CSLV_ICR.I2CINT=0 After the address is matched and ACK is returned, control whether the clock is forcibly extended. If FACS=1, the clock is pulled low and extended after the ACK of the reply address byte, and is not released until I2CSLV_SR.I2CINT=0. If FACS=0, the I2CSLV_SR.I2CINT bit will not be set to 1 after the ACK of the reply address byte, however, I2CSLV_SR.I2CINT may be set to 1 due to other conditions, which are described in the I2CSLV_SR.RD and I2CSLV_SR.WR bits.	RW
9:0	ADDRMASK	Slave address mask Define which bits in the I2CSLV_SLADR0 register need to be compared with the received address and which ones are ignored. Those set to 1 in ADDRMASK need to be compared, and 0 needs to be ignored (the corresponding bit of the received address can be 0 or 1).	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

23.6.9 Filter Register (I2CSLV_FLT)

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																													FW		
Access																														RW		

Bit	Name	Functional Description	Access
31:3	*Reserved*	-	-
2:0	FW	SDA and SCL input filter width ● 0b000: No filter ● 0b001: One PCLK ● 0b010: Two PCLK ● 0b011: Three PCLK ● 0b100: Four PCLK ● 0b101: Eight PCLK ● 0b110: Twelve PCLK ● 0b111: Sixteen PCLK	RW

23.6.10 Timeout Register (I2CSLV_TOR)

Offset	Bit Position																													
0x44	31	30	29	28	27	26	25	24	23	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
31:16	*Reserved*	-	-
15:0	TOR	SCL low level timeout length value Timeout length= (TOR+1)*64*(1/$PCLK_{I2C\_SLV}$) TOR cannot be changed when I2CSLV_CR.EN is 1, and data can be written to TOR only when I2CSLV_CR.EN=0.	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

23.6.11 Edge Shaping Register (I2CSLV_ODDR)

Offset	Bit Position																													
0x48	31	30	29	28	27	26	25	24	23	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																														ODPU	
Access																															RW	

Bit	Name	Functional Description	Access
31:2	*Reserved*	-	-
1:0	ODPU	Bus drive capability configuration when SDA changes from 0 to 1 ● 0b00: Pulled high by an external pull-up resistor, no strong drive ● 0b01: Outputs a strong drive high level of 1 PCLK widths, which is subsequently maintained by a pull-up resistor. ● 0b10: Outputs a strong drive high level of 2 PCLK widths, which is subsequently maintained by a pull-up resistor. ● 0b11: Outputs a strong drive high level of 4 PCLK widths, which is subsequently maintained by a pull-up resistor. **Note** At 0b00, the rising time of SDA is determined by the pull-up resistor; at other values, a rapid rise from low level to high level strong driving can be achieved.	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

23.6.12 Receive Sampling Location Register (I2CSLV_RSR)

Offset	Bit Position																													
0x4C	31	30	29	28	27	26	25	24	23	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																														NS	
Access																															RW	

Bit	Name	Functional Description	Access
31:2	*Reserved*	-	-
1:0	NS	Receive sampling location setup When SCL is high, the built-in counter starts counting, and the NS value determines the position of system sampling. The internal PCLK samples SCL, and the falling edge of the sampled SCL is regarded as the 0 value of the counter, and the counter value corresponding to the PCLK edge before the falling edge is regarded as 1, and so on. ● 0b00: When SCL is high, the SDA sample value when the counter is 0 is saved to I2CSLV_DINR ● 0b01: When SCL is high, the SDA sample value when the counter is 1 is saved to I2CSLV_DINR ● 0b10: When SCL is high, the SDA sample value when the counter is 2 is saved to I2CSLV_DINR ● 0b11: When SCL is high, the SDA sample value when the counter is 3 is saved to I2CSLV_DINR	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

23.6.13 Interrupt Enable Register (I2CSLV_IER)

Offset	Bit Position																													
0x50	31	30	29	28	27	26	25	24	23	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															GC	RXNE	TXE	SLADDR3	SLADDR2	SLADDR1	SLADDR0	ACK	TIMEOUT	HSMODE	ACTIVE	I2CINT	NACK	START	STOP	WR	RD
Access																RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Description	Access
31:17	*Reserved*	-	-
16	GC	Broadcast interrupt enable ● 0b0: Disable ● 0b1: Enable	RW
15	RXNE	Receive non-null interrupt enable ● 0b0: Disable ● 0b1: Enable	RW
14	TXE	Send null interrupt enable ● 0b0: Disable ● 0b1: Enable	RW
13	SLADDR3	Slave address 3 match interrupt enable ● 0b0: Disable ● 0b1: Enable	RW
12	SLADDR2	Slave address 2 match interrupt enable ● 0b0: Disable ● 0b1: Enable	RW
11	SLADDR1	Slave address 1 match interrupt enable ● 0b0: Disable ● 0b1: Enable	RW
10	SLADDR0	Slave address 0 match interrupt enable ● 0b0: Disable ● 0b1: Enable	RW
9	ACK	ACK interrupt enable ● 0b0: Disable ● 0b1: Enable	RW
8	TIMEOUT	TIMEOUT interrupt enable ● 0b0: Disable ● 0b1: Enable	RW
7	HSMODE	HSMODE interrupt enable ● 0b0: Disable ● 0b1: Enable	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Bit	Name	Description	Access
6	ACTIVE	ACTIVE interrupt enable ● 0b0: Disable ● 0b1: Enable	RW
5	I2CINT	I2CINT interrupt enable ● 0b0: Disable ● 0b1: Enable	RW
4	NACK	NACK interrupt enable ● 0b0: Disable ● 0b1: Enable	RW
3	START	START interrupt enable ● 0b0: Disable ● 0b1: Enable	RW
2	STOP	STOP interrupt enable ● 0b0: Disable ● 0b1: Enable	RW
1	WR	WR interrupt enable ● 0b0: Disable ● 0b1: Enable	RW
0	RD	RD interrupt enable ● 0b0: Disable ● 0b1: Enable	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

23.6.14 Interrupt Flag Clear Register (I2CSLV_ICR)

Offset	Bit Position																													
0x54	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0001FFFF																															
Name	Reserved															GC	RXNE	TXE	SLADDR3	SLADDR2	SLADDR1	SLADDR0	ACK	TIMEOUT	HSMODE	ACTIVE	I2CINT	NACK	START	STOP	WR	RD
Access																WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO

Bit	Name	Functional Description	Access
31:17	*Reserved*	-	-
16	GC	Broadcasting interrupt flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
15	RXNE	Receive non-empty interrupt flag cleared Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
14	TXE	Transmit empty interrupt flag cleared Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
13	SLADDR3	Slave address 3 match interrupt flag cleared Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
12	SLADDR2	Slave address 2 match interrupt flag cleared Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
11	SLADDR1	Slave address 1 match interrupt flag cleared Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
10	SLADDR0	Slave address 0 match interrupt flag cleared Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
9	ACK	ACK interrupt flag cleared Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
8	TIMEOUT	TIMEOUT interrupt flag cleared Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
7	HSMODE	HSMODE interrupt flag cleared Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
6	ACTIVE	ACTIVE interrupt flag cleared Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
5	I2CINT	I2CINT interrupt flag cleared Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
4	NACK	NACK interrupt flag cleared Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
3	START	START interrupt flag cleared 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
2	STOP	STOP interrupt flag cleared Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
1	WR	WR interrupt flag cleared Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
0	RD	RD interrupt flag cleared Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

23.6.15 Device ID Register (I2CSLV_DIDR)

Offset	Bit Position																													
0x58	31	30	29	28	27	26	25	24	23	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								MID												PID									RID		
Access									RW												RW									RW		

Bit	Name	Functional Description	Access
31:24	*Reserved*	-	-
23:12	MID	Manufacturer identification	RW
11:3	PID	Part identification	RW
2:0	RID	Revision identification	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

24 EAU

24.1 Introduction

The Extended Arithmetic Unit (EAU) is a coprocessor designed specifically for arithmetic operations. It efficiently performs signed integer division, unsigned integer division, and integer square root operations, providing effective supplement and extension to the arithmetic commands of the Cortex-M0+ core.

24.2 Features

- Supports the square root operation of 32-bit unsigned integers, and can obtain the square root and remainder.
- Supports 32-bit by 32-bit signed and unsigned integer division, and can obtain the quotient and remainder.
- Supports an unsigned division by zero flag.
- Supports a signed division by zero flag and an overflow flag.

24.3 Functional Description

24.3.1 Function Block Diagram

The EAU supports three types of operations: 32-bit unsigned integer square root, 32-bit unsigned integer division by 32-bit unsigned integer, and 32-bit signed integer division by 32-bit signed integer. The square root operation requires 17 HCLK cycles, and the division operation requires 2 to 35 HCLK cycles.

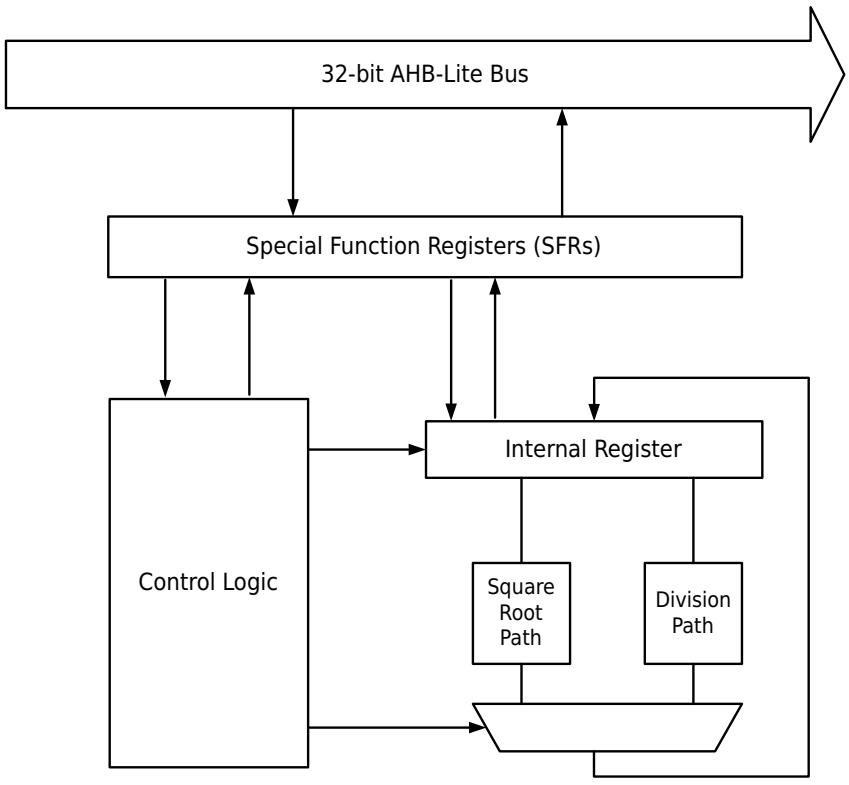


Figure 24-1 EAU Block Diagram

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Note

- The EAU only supports 32-bit operations. Any operand data written is treated as 32-bit data for computation, and the results are stored in the corresponding registers in 32-bit format.
- The EAU only supports 32-bit register access. 8/16-bit register accesses are ignored.

24.3.2 Operation Definition

- **Unsigned Division**

  X = A/B

  Y = A%B

  Given A = B*X + Y

  Where:

  - A: Dividend, 32-bit unsigned integer
  - B: Divisor, 32-bit unsigned integer
  - X: Quotient, 32-bit unsigned integer
  - Y: Remainder, 32-bit unsigned integer

- **Signed Division**

  X = A/B

  Y = A%B

  Given A = B*X + Y

  Where:

  - A: Dividend, 32-bit signed integer
  - B: Divisor, 32-bit signed integer
  - X: Quotient, 32-bit signed integer
  - Y: Remainder, 32-bit signed integer with the same sign as the dividend

- **Square Root**

  $X = \sqrt{A}$

  $Y = A\ X^2$

  Where:

  - A: Operand, 32-bit unsigned integer
  - X: Square root result, 32-bit unsigned integer
  - Y: Remainder, 32-bit unsigned integer

24.3.3 Status Flags

The EAU provides multiple status flag bits to indicate the current working status, including the BUSY flag, ZERO flag, and OVR flag.

Busy Flag

The bus busy flag bit is used to indicate whether the current EAU is in the calculation end or idle state, corresponding to the register EAU_CSR.BUSY.

- When EAU_CSR.BUSY=0, the EAU is in the end of calculation or idle state, and results can be read or a new operation can be started.
- When EAU_CSR.BUSY=1, the EAU is busy, and the current data operation is in progress, unable to provide results.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Zero Divisor Flag

This flag bit indicates whether the just-completed division operation was a meaningless operation with a zero divisor, corresponding to the register EAU_CSR.ZERO.

- When EAU_CSR.ZERO=0, the divisor of the division operation just-completed by EAU is not zero, which is a meaningful operation.
- When EAU_CSR.ZERO=1, the divisor of the division operation just-completed by EAU is zero, which is a meaningless operation.

Note

This flag bit applies only to division operations and not to square root operations.

Signed Division Overflow Flag

This flag bit indicates whether the just-completed signed division operation of EAU is overflowed, corresponding to the register EAU_CSR.OVR.

Due to the limitation of the encoding range, taking 32-bit as an example, when performing signed integer division, if the dividend is the smallest negative number (0x80000000 in 32-bit) and the divisor is -1 (0xFFFFFFFF in 32-bit), the resulting quotient is a very large positive number, one greater than the maximum integer that can be represented by a 32-bit signed integer, exceeding the legal encoding range. This is called signed division overflow.

- When EAU_CSR.OVR=0, the just-completed signed division operation of EAU did not overflow.
- When EAU_CSR.OVR=1, the just-completed signed division operation of EAU overflowed.

Note

- This flag bit applies only to signed division operations and not to square root or unsigned division operations.
- Setting the ZERO and OVR flag bits does not affect the continuation of a new round of operations. When a new round of operations starts, the flag bits are automatically cleared and re-indicated based on the status of the new operation.

24.4 Operation Example

24.4.1 Division Operation

Step1. Enable the EAU clock and release its reset (refer to the "System Controller (SYSCTRL)" and "Reset Control (RESET)" chapter).

Step2. Configure the EAU operation mode as unsigned division (EAU_CSR.MODE=0x00) or signed division (EAU_CSR.MODE=0x01) as needed.

Step3. Write the dividend to the Dividend/Square Root Operand Register EAU_DIVIDEND.DIVIDEND.

Step4. Write the divisor to the Divisor Register EAU_DIVISOR.DIVISOR, and the operation starts automatically (EAU_CSR.BUSY is set to 1).

Step5. Continuously poll the BUSY flag bit EAU_CSR.BUSY until it becomes 0, indicating the end of the operation.

Step6. Depending on the operation mode:

- For unsigned division, check if the ZERO flag bit EAU_CSR.ZERO is 0, indicating no abnormalities in the operation; otherwise, the divisor was zero, making the operation meaningless.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

- For signed division, check both the ZERO flag bit EAU_CSR.ZERO and the OVR flag bit EAU_CSR.OVR; if both are 0, indicating no abnormalities in the operation; otherwise, there was a corresponding abnormality, making the result meaningless.

Step7. If no abnormal state is found, read the results from the respective registers as needed.

Step8. Repeat steps 2-7 for a new round of operations.



Note

In division operation mode, writing to the divisor register triggers the automatic start of the operation. Strictly follow the order of "writing the dividend first, then the divisor".

24.4.2 Square Root Operation

Step1. Enable the EAU clock and release its reset (refer to the "System Controller (SYSCTRL)" and "Reset Control (RESET)" chapter).

Step2. Configure the EAU operation mode as square root mode (EAU_CSR.MODE=0x02).

Step3. Write the number to be squared to the Dividend/Square Root Operand Register (EAU_DIVIDEND.DIVIDEND), and the operation starts automatically.

Step4. Continuously poll the BUSY flag bit EAU_CSR.BUSY until it becomes 0, indicating the end of the operation.

Step5. Read the results from the respective registers as needed.

Step6. Repeat steps 2-5 for a new round of operations.



Note

In square root mode, the value in the divisor register does not participate in the operation, so any value in it will not affect the operation result.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

24.5 Register Description

24.5.1 Register List

Base address: 0x40021800

Table 24-1 Extended Arithmetic Unit (EAU) Register Offset

Offset	Register	Description	Width
0x00	EAU_CSR	Control and Status Registers	32
0x04	EAU_DIVIDEND	Dividend/Square Root Operand Register	32
0x08	EAU_DIVISOR	Divisor Register	32
0x0C	EAU_QUOTIENT	Quotient/Square Root Register	32
0x10	EAU_REMAINDER	Remainder Register	32

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

24.5.2 Control and Status Register (EAU_CSR)

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																											BUSY	ZERO	OVR	MODE	
Access																												RO	RO	RO	RW	

Bit	Name	Functional Description	Access
31:5	*Reserved*	-	-
4	BUSY	Busy flag ● 0b0: Operation ended or idle ● 0b1: Operation in progress	RO
3	ZERO	Zero divisor flag ● 0b0: Divisor is not zero ● 0b1: Divisor is zero	RO
2	OVR	Signed division overflow flag ● 0b0: No overflow ● 0b1: Overflow occurred	RO
1:0	MODE	Operation mode selection ● 0b00: Unsigned division ● 0b01: Signed division ● 0b10: Square root operation ● 0b11: Setting prohibited **Note** Writable only when EAU_CSR.BUSY=0, cannot be written while operation is in progress.	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

24.5.3 Dividend/Square Root Operand Register (EAU_DIVIDEND)

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	DIVIDEND																															
Access	RW																															

Bit	Name	Functional Description	Access
31:0	DIVIDEND	Dividend/square root operand ● For square root operations, write the number to be squared into this register. Writing to this register triggers the operation to start immediately ● For division operations, write the dividend into this register **Note** Writes to this register are only allowed when EAU_CSR.BUSY=0. Writes during an ongoing operation are invalid.	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

24.5.4 Divisor Register (EAU_DIVISOR)

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	DIVISOR																															
Access	RW																															

Bit	Name	Description	Access
31:0	DIVISOR	Divisor ● For square root operations, this register has no effect. ● For division operations, write the divisor into this register. Writing to this register triggers the division operation to start immediately **Note** Writes to this register are only allowed when EAU_CSR.BUSY=0. Writes during an ongoing operation are invalid.	RW

24.5.5 Quotient/Square Root Register (EAU_QUOTIENT)

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	QUOTIENT																															
Access	RO																															

Bit	Name	Functional Description	Access
31:0	QUOTIENT	Quotient/square root register ● For square root operations, this register holds the calculated square root ● For division operations, this register holds the calculated quotient **Note** This register is readable only after the operation is complete. Reading during the operation returns zero.	RO

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

24.5.6 Remainder Register (EAU_REMAINDER)

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	REMAINDER																															
Access	RO																															

Bit	Name	Functional Description	Access
31:0	REMAINDER	Remainder ● For square root operations, this register holds the remainder after the square root calculation. ● For division operations, this register holds the remainder that is not divided. **Note** This register is readable only after the operation is complete. Reading during the operation returns zero.	RO

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

25 ADC

25.1 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.

25.2 Features

- 12-bit conversion accuracy
- 1Msps conversion speed
- Up to 18 input channels are supported, including 16 external pin inputs, 1 internal temperature sensor voltage channel, and 1 channel of 1/3 AVCC voltage
- 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 single sequence conversion, SQR scan continuous conversion, SQR scan discontinuous conversion, JQR scan conversion, continuous conversion accumulation
- 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
- Input channel voltage threshold monitoring

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

25.3 Functional Description

25.3.1 ADC Block Diagram

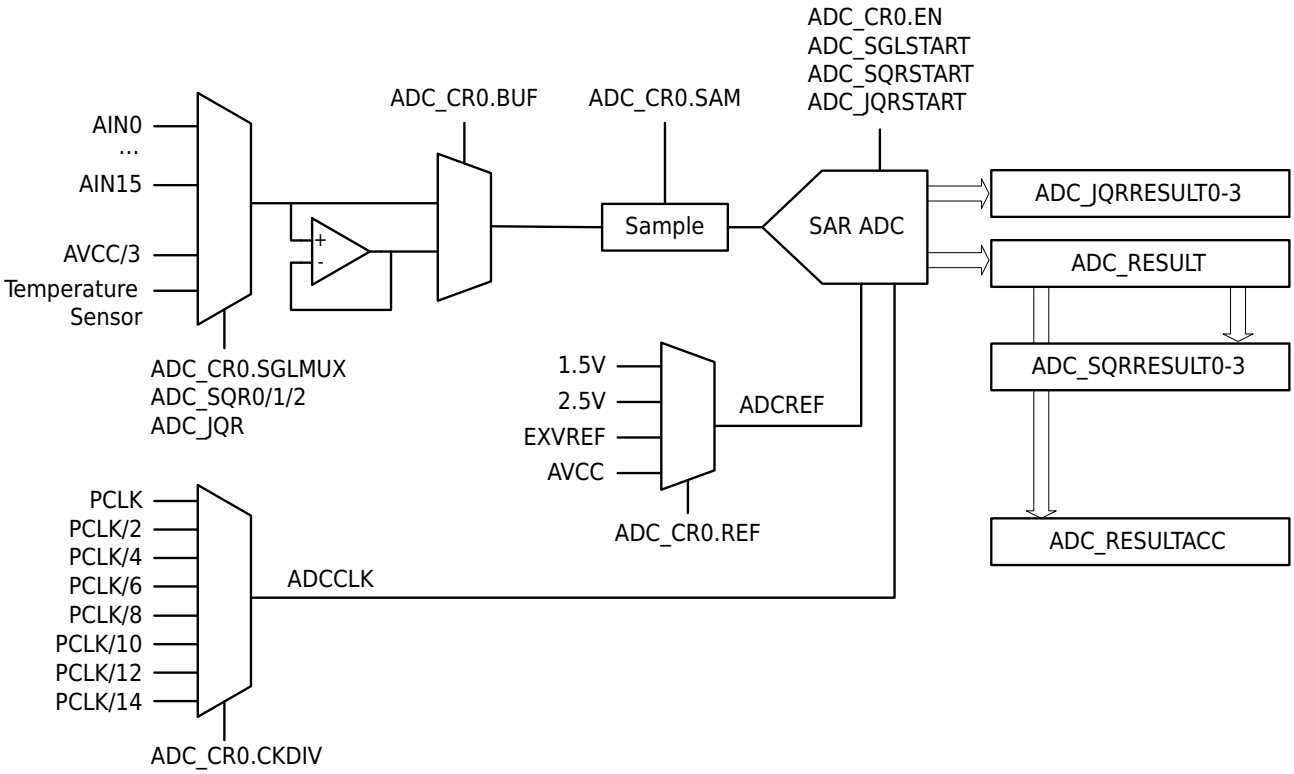


Figure 25-1 ADC Block Diagram

25.3.2 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 4-10 ADCCLK cycles, configured by ADC_CR0.SAM; the successive comparison process requires 12 ADCCLK cycles. Therefore, an ADC conversion requires a total of 16-22 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.

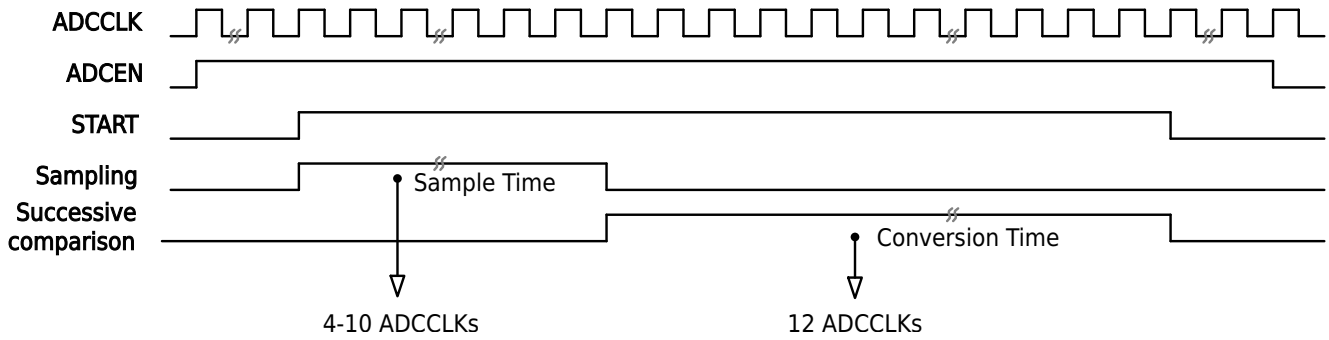


Figure 25-2 ADC Conversion Timing Diagram

The ADC conversion speed is related to the ADC reference voltage and AVCC voltage. The maximum conversion speed is shown in the following table:

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

ADC Reference Voltage	AVCC Voltage	Maximum Conversion Speed	Maximum ADCCLK Frequency
Internal 1.5V	1.8-5.5V	200ksps	4.4MHz
Internal 2.5V	2.8-5.5V	200ksps	4.4MHz
AVCC/EXVREF	1.8-2.4V	200ksps	4.4MHz
AVCC/EXVREF	2.4-2.7V	500ksps	11MHz
AVCC/EXVREF	2.7-5.5V	1Msps	16MHz

25.3.3 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, After the ADC conversion for each channel is completed, ADC_IFR.ESG bit is automatically set to 1, and the conversion result is saved in the ADC_RESULT register.

25.3.4 Scan Conversion Mode

The scan conversion mode is divided into four types: SQR scan single sequence conversion, SQR scan continuous conversion, SQR scan discontinuous conversion and JQR scan conversion. When the four modes work independently, multiple conversions can be performed on multiple channels continuously.

SQR Scan Single Sequence Conversion Mode

SQR Scan conversion mode supports up to 16 consecutive conversions. The total number of conversions is configured by ADC_SQR2.CNT, and all channels can be configured for conversion through ADC_SQRx.CHxMUX. This mode can be started either by setting the ADC_SQRSTART.START bit or by configuring an external trigger via ADC_EXTTRIGGER0. After the conversion is started, the ADC module sequentially converts the channels configured in CH0MUX-CHxMUX until the total number of conversions is completed. ADC_RESULT register after each conversion, and the ADC_IFR.EOC bit is automatically set to 1. The conversion result needs to be read out in time. The conversion results of ADC_SQRx.CH0MUX-ADC_SQRx.CH3MUX will also be saved in the ADC_SQRRESULT0-3 registers. The ADC_IFR.EOS bit is automatically set to 1 after the ADC module completes the total number of conversions. The following diagram illustrates the SQR scan conversion process for performing 8 conversions on AIN0, AIN1, and AIN5. In this configuration, SQR scan conversion channels 7/4/1 are set to AIN0, channels 6/3/0 are set to AIN1, and channels 5/2 are set to AIN5. ADC_SQRSTART.START is set to 1, the ADC module will sequentially perform conversions on SQR scan conversion channels 0 to 7.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Note

The EOC flag will be set after the conversion of each channel is completed, so the interval between EOC setting is related to the sampling rate of SQR scan single sequence or continuous conversion mode. If you obtain the sampling conversion result of each channel by reading the ADC_RESULT register during the EOC interrupt, you need to consider whether the time is sufficient. If the time interval between each EOC setting is too short, the value of the ADC_RESULT register may not be read and be overwritten by a new value. At this time, the IFR.OV flag will be set.

- If the sampling rate of SQR scan single sequence or continuous conversion mode is high, you can consider enabling ADC_IER.DEOC to trigger DMA, and the result data will be transmitted after the conversion of each channel is completed.
- If you only use ADC_SQR0.CH0MUX-ADC_SQR0.CH3MUX, you can also wait until the EOS flag is set, and then obtain the sampling conversion result by reading the ADC_SQRRESULT0-3 registers within the EOS interrupt.

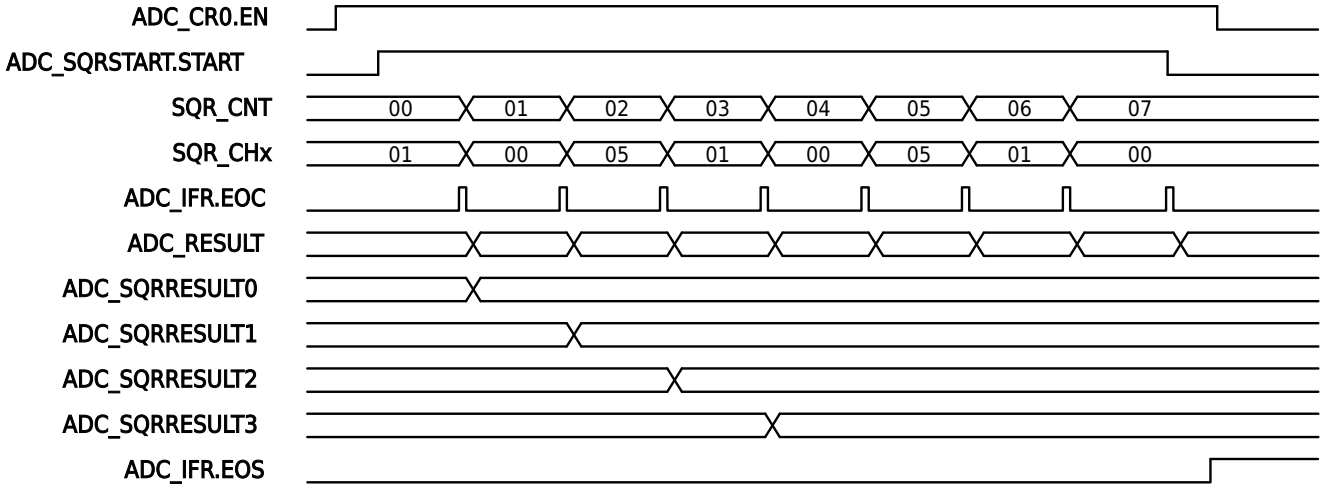


Figure 25-3 Examples of ADC SQR Scan Conversion Process

SQR Scan Continuous Conversion Mode

SQR scan continuous conversion mode can continuously convert the ADC, and the number of single conversions is configured by ADC_SQR2.CNT; it can be configured to select any channel for conversion, and the channel to be converted is configured by ADC_SQRx.CHxMUX. This mode can be started either by setting the ADC_SQRSTART.START bit or by configuring an external trigger via ADC_EXTTRIGGER0. After starting the conversion, the ADC module converts the channel conversions configured in CH0MUX-CHxMUX in turn. After a channel conversion is completed, the ADC_IFR.EOC bit will be automatically set to 1, and the conversion result is saved in the ADC_RESULT register, and then switch to the next channel. After ADC module completes the number of conversions configured in ADC_SQR2.CNT, the ADC_IFR.EOS bit is automatically set to 1. And automatically start the next conversion until the software writes 0 to ADC_SQRSTART.START, and stops after converting the channel currently being converted.

SQR Scan Discontinuous Conversion Mode

In SQR scan discontinuous conversion mode, channels configured in ADC_SQR2.CNT and ADC_SQRx.CHxMUX are converted sequentially. This mode can be started either by setting the ADC_SQRSTART.START bit or by configuring an external trigger via ADC_EXTTRIGGER0. Convert one channel each time, and then wait for the next channel to be converted, the ADC_IFR.EOC bit will be automatically set to 1, and the conversion result will be saved in the ADC_RESULT register.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

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 ADC_JQRSTART.START bit or by configuring an external trigger via ADC_EXTTRIGGER1. After starting the conversion, the ADC module sequentially converts the channels configured from CHxMUX-CH0MUX, until the total number of conversions is completed. After the ADC module completes the total conversion count, the ADC_IFR.EOJ 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.

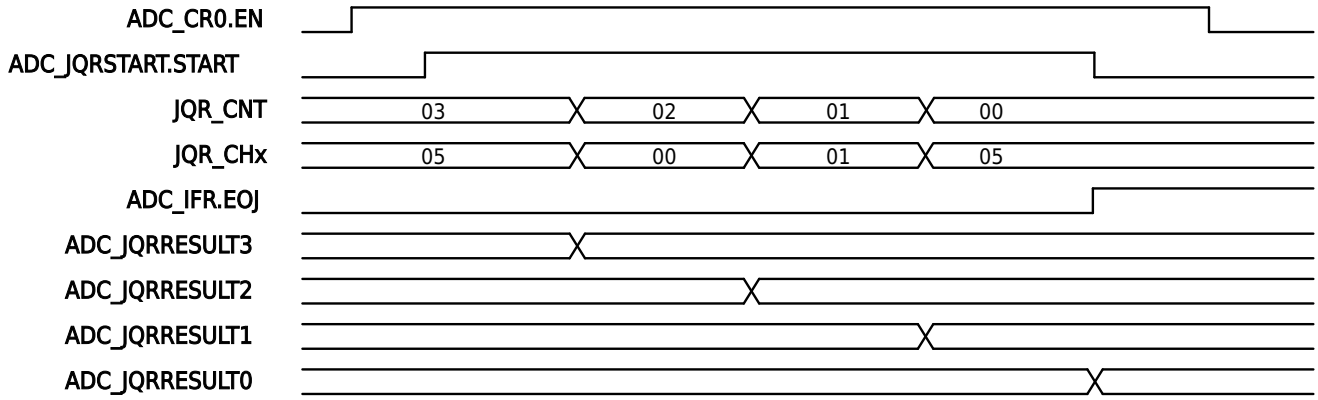


Figure 25-4 Examples of ADC JQR Scan Conversion Process

Note

JQR scan conversion cannot be started while SQR scan conversion is in progress; SQR scan conversion cannot be started while JQR scan conversion is in progress.

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_IER.DEOC to 1, which triggers DMA after each channel conversion; in JQR scan mode, enable the function by setting ADC_IER.DEOJ to 1, which triggers DMA after all JQR scan channels are converted.

25.3.5 Continuous Conversion Accumulation Mode

In the continuous conversion and accumulation mode, starting the ADC once can perform multiple conversions on multiple channels and accumulate the results of each conversion; all channels can be configured to convert. The total number of ADC conversions is configured by ADC_SQR2.CNT; the channel to be converted is configured by ADC_SQRx.CHxMUX. This mode can be initiated either by setting the ADC_SQRSTART.START bit or by setting the external trigger of ADC_EXTTRIGGER0. After the continuous conversion is started, the ADC module sequentially converts the channels configured in CH0MUX-CHxMUX until the total number of conversions is completed. After the ADC module completes the total number of conversions, the ADC_IFR.EOS bit will be automatically set to 1, and the accumulated value of the conversion result will be saved in the ADC_RESULTACC register.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

The figure below demonstrates the process of accumulating 10 consecutive conversions of AIN0, AIN1, and AIN5. SQR scan conversion channels 9/6/3/0 are configured as AIN0; channels 8/5/2 as AIN5; channels 7/4/1 as AIN1. After ADC_SQRSTART.START is set to 1, the ADC module will sequentially perform conversions according to the SQR scan 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.

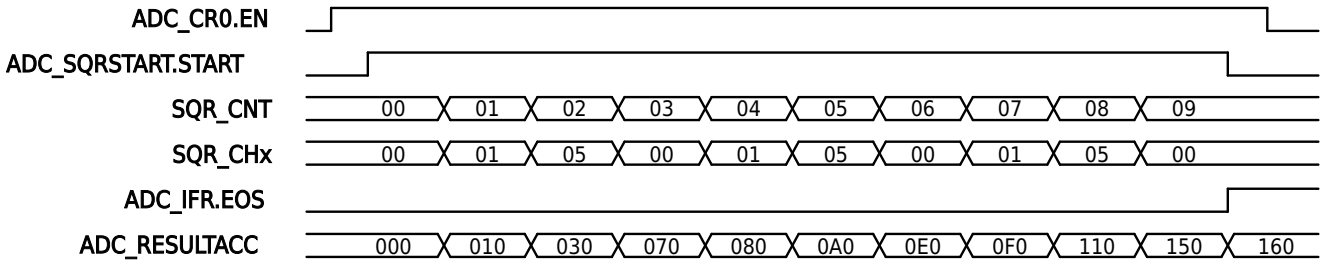


Figure 25-5 Examples of Continuous ADC Conversion Accumulation Process

25.3.6 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.

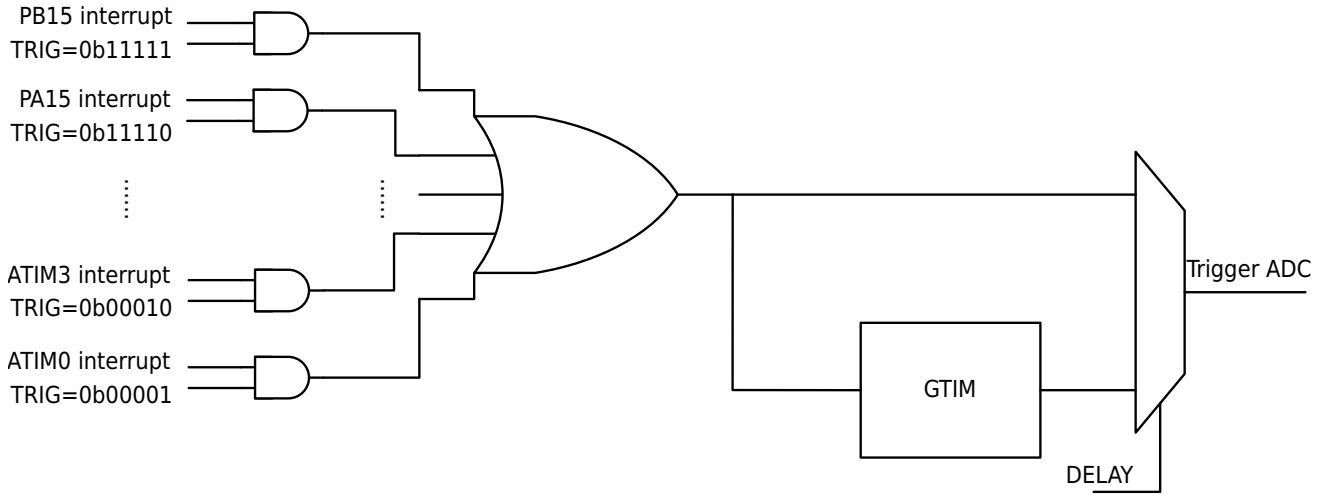


Figure 25-6 Diagram of External Trigger Sources for ADC Single and SQR Scan Conversions

The ADC_EXTTRIGGER1 register can be configured to set the external trigger source for ADC JQR scan conversions.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

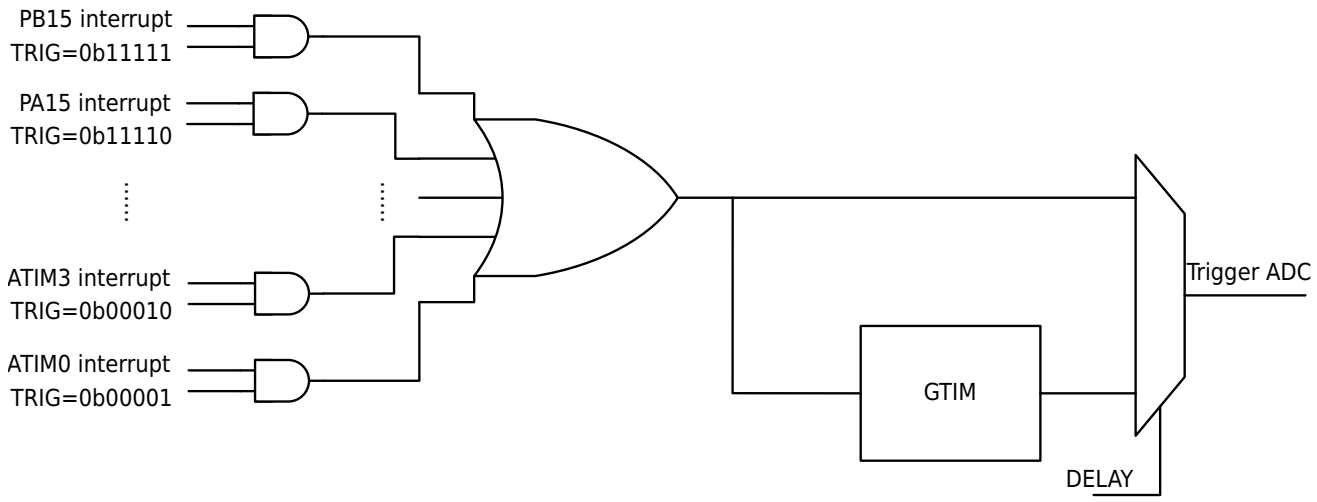


Figure 25-7 Diagram of External Trigger Sources for ADC JQR Scan Conversion

25.3.7 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.HTI bit is set to 1; writing 0 to ADC_ICR.HTI clears ADC_IFR.HTI.
- Low Threshold Comparison: If the ADC conversion result falls within the range [0, ADC_LT), ADC_IFR.LTI is set to 1; writing 0 to ADC_ICR.LTI clears ADC_IFR.LTI.
- Interval Value Comparison: If the ADC conversion result falls within the range [ADC_LT, ADC_HT), ADC_IFR.RGI is set to 1; writing 0 to ADC_ICR.RGI clears ADC_IFR.RGI.

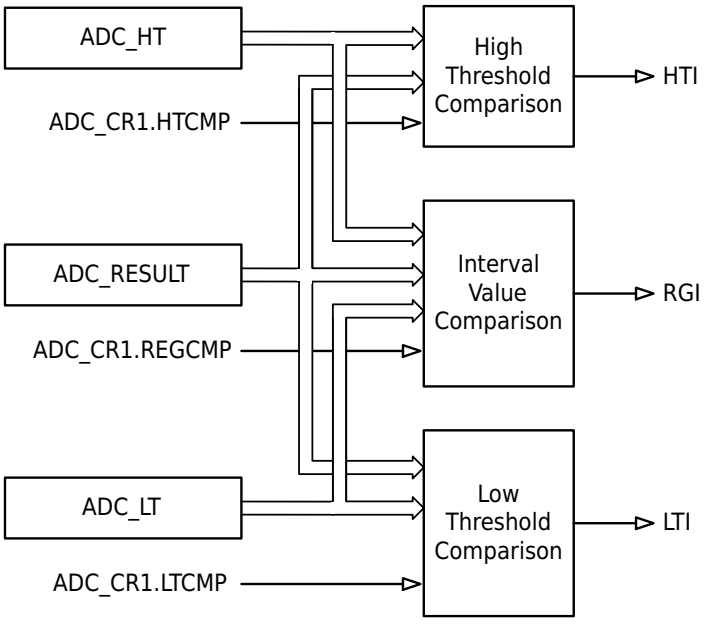


Figure 25-8 ADC Conversion Result Comparison Diagram

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

25.3.8 ADC Interrupt

The ADC interrupt request is shown in the table below:

Table 25-2 ADC Interrupt

Interrupt Source	Interrupt Flag	Interrupt Enable
ADC conversion result overflow	ADC_IFR.OV	ADC_IER.OV
ADC SQR scan single channel conversion complete	ADC_IFR.EOC	ADC_IER.EOC
ADC JQR scan conversion complete	ADC_IFR.EOJ	ADC_IER.EOJ
ADC SQR scan conversion complete	ADC_IFR.EOS	ADC_IER.EOS
ADC conversion result is within the range (greater than or equal to the low threshold and less than the high threshold)	ADC_IFR.RGI	ADC_IER. RGI
ADC conversion result is within the high threshold region (greater than or equal to the high threshold)	ADC_IFR.HTI	ADC_IER.HTI
ADC conversion result is within the low threshold region (less than the low threshold)	ADC_IFR.LTI	ADC_IER.LTI
ADC single conversion completed	ADC_IFR.ESG	ADC_IER.ESG

25.3.9 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:

$$T_A\,(℃) = \frac{1}{16} * [\,T_N + \frac{T_N + T_L}{D_{N15} - D_{L15}} * (D_{ADC} - D_{N15})], \ldots\ldots\ldots (V_{ref} = 1.5\ V)$$

$$T_A\,(℃) = \frac{1}{16} * [\,T_N + \frac{T_N + T_L}{D_{N25} - D_{L25}} * (D_{ADC} - D_{N25})], \ldots\ldots\ldots (V_{ref} = 2.5\ V)$$

Where:

- Vref is the reference voltage of the current ADC module, with typical values of 1.5V or 2.5V.
- D_{ADC} is the result of the ADC module measuring the output voltage of the temperature sensor, and the value is 0-4095.
- T_L, T_N are 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.
- D_{L15},D_{L25},D_{N15},D_{N25},T_L,T_N are calibration values obtained from temperature testing. These values need to be read from Flash memory during calculation. Their storage addresses and bit widths are detailed in the table below.

Table 25-3 Temperature Sensor Calibration Value Information List

Temperature	Calibration Value Name	Calibration Value Storage Address	Width	Calibration Accuracy
Low temperature	D_{L15}	0x00101834-0x00101835	12 bits	TBD
	D_{L25}	0x00101836-0x00101837	12 bits	TBD
	T_L	0x0010183A-0x0010183B	16 bits	TBD

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Temperature	Calibration Value Name	Calibration Value Storage Address	Width	Calibration Accuracy
Room temperature	D_{N15}	0x00101C34-0x00101C35	12 bits	TBD
	D_{N25}	0x00101C36-0x00101C37	12 bits	TBD
	T_N	0x00101C3A-0x00101C3B	16 bits	TBD

An Example Calculation is as Follows:

Condition 1: Vref=2.5, T_L=640, T_N=400, D_{L25}=917, D_{N25}=1227, D_{ADC}=1500:

Temperature 1:

$$T_A(℃) = \frac{1}{16} * [T_N + \frac{T_N + T_L}{D_{N25} - D_{L25}} * (D_{ADC} - D_{N25})] = \frac{1}{16} * [400 + \frac{400 + 640}{1227 - 917} * (1500 - 1227)] = 82.24 ℃$$

Condition 2: Vref=1.5, T_L=640, T_N=400, D_{L15}=1528, D_{N15}=2047, D_{ADC}=2500:

Temperature 2:

$$T_A(℃) = \frac{1}{16} * [T_N + \frac{T_N + T_L}{D_{N15} - D_{L15}} * (D_{ADC} - D_{N15})] = \frac{1}{16} * [400 + \frac{400 + 640}{2047 - 1528} * (2500 - 2047)] = 81.73 ℃$$

25.4 Operation Example

25.4.1 Start the ADC Single Conversion Operation Process through the ADC_SGLSTART.START Bit

Procedure

Step1. Configure the corresponding bits of GPIO_ADS, and configure the ADC channel to be converted as an analog port.

Step2. Set GPIOB_ADS[1] to 1 to configure the ADC external reference voltage pin (PB01) 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 (ADC_IFR.RDY=1) and BGR module to complete startup.

Step6. Set ADC_CR1.MODE to 0 to select single conversion mode.

Step7. Configure ADC_CR0.REF and select the reference voltage of the ADC. If the internal reference voltage is selected, a delay of more than 20μs is required after switching.

Step8. Configure ADC_CR0.SAM and ADC_CR0.CKDIV to set the conversion speed of ADC.

Step9. Configure ADC_CR0.SGLMUX to select the channel to be converted.

Step10. Set ADC_ICR.ESG to 0, and clear ADC_IFR.ESG flag.

Step11. Set ADC_SGLSTART.START to 1 to start ADC single conversion.

Step12. Wait for ADC_IFR.ESG to become 1, read the ADC_RESULT register to get the ADC conversion result, and set ADC_ICR.ESG to 0, clear the ADC_IFR.ESG flag.

Step13. To convert other channels, repeat steps9-12.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Step14. Set ADC_CR0.EN and ADC_BGR.BGR_EN to 0, turn off ADC module and BGR module.

25.4.2 ADC Single Conversion Operation Process through an External Trigger

Procedure

Step1. Configure the corresponding bits of GPIO_ADS, and configure the ADC channel to be converted as an analog port.

Step2. Set GPIOB_ADS[1] to 1 to configure the ADC external reference voltage pin (PB01) 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 (ADC_IFR.RDY=1) and BGR module to complete startup.

Step6. Set ADC_CR1.MODE to 0 to select single conversion mode.

Step7. Set ADC_IER.ESG 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. If the internal reference voltage is selected, a delay of more than 20μs is required after switching.

Step10. Configure ADC_CR0.SAM and ADC_CR0.CKDIV 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 and clear the associated interrupt flag.

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.

25.4.3 Start the ADC SQR Scan Conversion Operation Process through the ADC_SQRSTART.START Bit

Procedure

Step1. Configure the corresponding bits of GPIO_ADS, and configure the ADC channel to be converted as an analog port.

Step2. Set GPIOB_ADS[1] to 1 to configure the ADC external reference voltage pin (PB01) 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.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

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 (ADC_IFR.RDY=1) and BGR module to complete startup.

Step6. Set ADC_CR1.MODE to 1 to choose the SQR scan single-sequence conversion mode.

Step7. Configure ADC_CR0.REF and select the reference voltage of the ADC. If the internal reference voltage is selected, a delay of more than 20μs is required after switching.

Step8. Configure ADC_CR0.SAM and ADC_CR0.CKDIV to set the conversion speed of ADC.

Step9. Configure ADC_SQR0.CHxMUX or ADC_SQR1.CHxMUX or ADC_SQR2.CHxMUX, and select SQR scan conversion channels.

Step10. Configure ADC_SQR2.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.EOS/EOC to 0, clear ADC_IFR.EOC/EOS flag.

Step12. Set ADC_SQRSTART.START to 1 to start ADC SQR scan conversion.

Step13. Wait for ADC_IFR.EOC to become 1, read ADC_RESULT registers to obtain the conversion result of the corresponding channel. Wait for ADC_IFR.EOS to become 1 to end the SQR scan conversion, set ADC_ICR.EOS to 0, and clear the ADC_IFR.EOS flag.

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.

25.4.4 External Trigger Start ADC SQR Scan Conversion Operation Flow

Procedure

Step1. Configure the corresponding bits of GPIO_ADS, and configure the ADC channel to be converted as an analog port.

Step2. Set GPIOB_ADS[1] to 1 to configure the ADC external reference voltage pin (PB01) 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 (ADC_IFR.RDY=1) and BGR module to complete startup.

Step6. Set ADC_CR1.MODE to 1 to select choose the SQR scan single-sequence conversion mode.

Step7. Set ADC_IER.EOS 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. If the internal reference voltage is selected, a delay of more than 20μs is required after switching.

Step10. Configure ADC_CR0.SAM and ADC_CR0.CKDIV to set the conversion speed of ADC.

Step11. Configure ADC_SQR0.CHxMUX or ADC_SQR1.CHxMUX or ADC_SQR2.CHxMUX, and select SQR scan conversion channels.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Step12. Configure ADC_SQR2.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 the external trigger condition triggers the ADC, the ADC SQR scan conversion is started. When a channel conversion is completed, ADC_IFR.EOC becomes 1 and the ADC_RESULT register is read to obtain the conversion result of the corresponding channel. When the EOS interrupt occurs, the current sequence conversion ends and the EOS interrupt flag is cleared.

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.

25.4.5 Start ADC JQR Scan Conversion Operation Flow via ADC_JQRSTART.START Bit

Procedure

Step1. Configure the corresponding bits of GPIO_ADS, and configure the ADC channel to be converted as an analog port.

Step2. Set GPIOB_ADS[1] to 1 to configure the ADC external reference voltage pin (PB01) 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 (ADC_IFR.RDY=1) and BGR module to complete startup.

Step6. Set ADC_CR1.MODE to 2 to choose JQR scan conversion mode.

Step7. Configure ADC_CR0.REF and select the reference voltage of the ADC. If the internal reference voltage is selected, a delay of more than 20μs is required after switching.

Step8. Configure ADC_CR0.SAM and ADC_CR0.CKDIV 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.EOJ to 0, clear ADC_IFR.EOJ flag.

Step12. Set ADC_JQRSTART.START to 1 to start ADC JQR scan conversion.

Step13. Wait for ADC_IFR.EOJ to become 1, read ADC_JQRRESULTx-ADC_JQRRESULT0 registers to obtain the conversion result of the corresponding channel and clear the ADC_IFR.EOJ flag.

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.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

25.4.6 External Trigger Start ADC JQR Scan Conversion Operation Flow

Procedure

Step1. Configure the corresponding bits of GPIO_ADS, and configure the ADC channel to be converted as an analog port.

Step2. Set GPIOB_ADS[1] to 1 to configure the ADC external reference voltage pin (PB01) 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 (ADC_IFR.RDY=1) and BGR module to complete startup.

Step6. Set ADC_CR1.MODE to 2 to choose JQR scan conversion mode.

Step7. Set ADC_IER.EOJ 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. If the internal reference voltage is selected, a delay of more than 20μs is required after switching.

Step10. Configure ADC_CR0.SAM and ADC_CR0.CKDIV 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 and clear the ADC_IFR.EOJ flag.

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.

25.4.7 Start the ADC Continuous Conversion Accumulation Operation Process through the ADC_SQRSTART.START Bit

Procedure

Step1. Configure the corresponding bits of GPIO_ADS, and configure the ADC channel to be converted as an analog port.

Step2. Set GPIOB_ADS[1] to 1 to configure the ADC external reference voltage pin (PB01) as an analog port.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

> **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 (ADC_IFR.RDY=1) and BGR module to complete startup.

Step6. Set ADC_CR1.MODE to 1 to choose the SQR scan single-sequence 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. If the internal reference voltage is selected, a delay of more than 20μs is required after switching.

Step9. Configure ADC_CR0.SAM and ADC_CR0.CKDIV to set the conversion speed of ADC.

Step10. Configure ADC_SQR0.CHxMUX or ADC_SQR1.CHxMUX or ADC_SQR2.CHxMUX, and select SQR scan conversion channels.

Step11. Configure ADC_SQR2.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.EOS to 0, clear ADC_IFR.EOS 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.EOS to become 1, read ADC_RESULTACC registers to obtain accumulated value of the conversion result and clear the ADC_IFR.EOS flag.

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.

25.4.8 Start ADC Continuous Conversion Accumulation Operation Process through External Trigger

Procedure

Step1. Configure the corresponding bits of GPIO_ADS, and configure the ADC channel to be converted as an analog port.

Step2. Set GPIOB_ADS[1] to 1 to configure the ADC external reference voltage pin (PB01) 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 (ADC_IFR.RDY=1) and BGR module to complete startup.

Step6. Set ADC_CR1.MODE to 1 to choose the SQR scan single-sequence conversion mode.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Step7. Set ADC_IER.EOS 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. If the internal reference voltage is selected, a delay of more than 20μs is required after switching.

Step11. Configure ADC_CR0.SAM and ADC_CR0.CKDIV to set the conversion speed of ADC.

Step12. Configure ADC_SQR0.CHxMUX or ADC_SQR1.CHxMUX or ADC_SQR2.CHxMUX, and select SQR scan conversion channels.

Step13. Configure ADC_SQR2.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 and clear the relevant interrupt flag.

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.

25.4.9 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 (ADC_IFR.RDY=1) 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, and delay more than 20μs.

Step6. Configure ADC_CR0.SAM and ADC_CR0.CKDIV to set the conversion speed of ADC.

Step7. Set ADC_CR0.SGLMUX to 0x10001, 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.ESG to 0, and clear ADC_IFR.ESG flag.

Step10. Set ADC_SGLSTART.START to 1 to start ADC single conversion.

Step11. Wait for ADC_IFR.ESG to become 1, read ADC_RESULT registers to obtain accumulated value of the conversion result and clear the ADC_IFR.ESG flag.

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.

25.5 Register Description

25.5.1 Register List

Base address: 0x40002500

Table 25-4 Analog-to-Digital Converter (ADC) Register Offset

Offset	Register	Description	Width
0x000	ADC_BGR	ADC Analog Device Select 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
0x044	ADC_SQR1	ADC SQR Scan Conversion Channel Configuration Register 1	32
0x048	ADC_SQR2	ADC SQR Scan Conversion Channel Configuration Register 2	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
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
0x0A8	ADC_HT	ADC Comparison High Threshold Register	32
0x0AC	ADC_LT	ADC Comparison Low Threshold Register	32

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Offset	Register	Description	Width
0x030	ADC_IER	ADC Interrupt and DMA Trigger Enable 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

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

25.5.2 ADC Analog Device Select 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 ● 0b0: Disable internal temperature sensor ● 0b1: Enable internal temperature sensor, and BGR must be enabled **Note** After the temperature sensor is enabled for 20μs, it can output a stable signal.	RW
0	BGR_EN	BGR enable control ● 0b0: Disable BGR ● 0b1: Enable BGR **Note** ● This register can only be operated when SYSCTRL_PERI_CLKEN0.ADC is 1. ● 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. ● When using ADC/OPA/PLL, BGR must be enabled. ● When using VC, it is necessary to decide whether to enable BGR according to the configuration of the VC register.	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

25.5.3 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	0x000027F0																															
Name	Reserved																		SAM		BUF		REF		SGLMUX				CKDIV			EN
Access																			RW		RW		RW		RW				RW			RW

Bit	Name	Functional Description	Access
31:14	*Reserved*	-	-
13:12	SAM	ADC sampling period selection ● 0b00: 4 ADCCLK cycles ● 0b01: 6 ADCCLK cycles ● 0b10: 9 ADCCLK cycles ● 0b11: 10 ADCCLK cycles **Note** The successive comparison process alone requires 12 ADCCLK cycles.	RW
11	BUF	Built-in follower enable control ● 0b0: Disable the follower, and the external input signal is directly connected to the ADC ● 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 **Note** When this function is enabled, the maximum rate is 200ksps. The following conditions must enable this function: ● The output impedance of the signal to be converted is high ● The signals to be converted are: AVCC/3, internal temperature sensor output voltage.	RW
10:9	REF	ADC reference voltage selection ● 0b00: Internal 1.5V ● 0b01: Internal 2.5V (power supply voltage must exceed 2.8V) ● 0b10: External reference voltage EXVREF (PB01) ● 0b11: AVCC voltage	RW
8:4	SGLMUX	Single conversion mode conversion channel selection ● 0b00000: AIN0 (PA00) ● 0b00001: AIN1 (PA01) ● 0b00010: AIN2 (PA02) ● 0b00011: AIN3 (PA03) ● 0b00100: AIN4 (PA04) ● 0b00101: AIN5 (PA05) ● 0b00110: AIN6 (PA06)	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Bit	Name	Functional Description	Access
8:4	SGLMUX	● 0b00111: AIN7 (PA07) ● 0b01000: AIN8 (PB00) ● 0b01001: AIN9 (PB01) ● 0b01010: AIN10 (PB02) ● 0b01011: AIN11 (PB10) ● 0b01100: AIN12 (PB11) ● 0b01101: AIN13 (PB12) ● 0b01110: AIN14 (PB13) ● 0b01111: AIN15 (PB14) ● 0b10000: AVCC/3 ● 0b10001: Internal temperature sensor output voltage ● Others: Setting prohibited **Note** When set to 0b10000 and 0b10001, ADC_CR0.BUF must be 1.	RW
3:1	CKDIV	ADC clock selection ● 0b000: PCLK clock divided by 2 ● 0b001: PCLK clock divided by 4 ● 0b010: PCLK clock divided by 6 ● 0b011: PCLK clock divided by 8 ● 0b100: PCLK clock divided by 10 ● 0b101: PCLK clock divided by 12 ● 0b110: PCLK clock divided by 14 ● 0b111: PCLK clock	RW
0	EN	ADC enable control ● 0b0: Disable ● 0b1: Enable	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

25.5.4 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	OVMD	Reserved			THCH				ALIGN	MODE	
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 Write 0 to clear the ADC_RESULTACC register, write 1 has no effect; read always returns 1.	WO
14	REGCMP	ADC interval comparison control ● 0b0: Disable ● 0b1: Enable	RW
13	HTCMP	ADC high threshold compare control ● 0b0: Disable ● 0b1: Enable	RW
12	LTCMP	ADC low threshold compare control ● 0b0: Disable ● 0b1: Enable	RW
11	RACCEN	ADC conversion result automatic accumulation control ● 0b0: Disable ● 0b1: Enable	RW
10	OVMD	Storage method of ADC_RESULT when ADC conversion result overflows ● 0b1: Overwrite the last data and keep the new data **Note** This bit must be set to 1 in SQR Scan Conversion Mode. Only supports SQR scan mode.	RW
9:8	*Reserved*	-	-
7:3	THCH	Threshold comparison channel selection ● 0b00000: Select channel 0 for threshold comparison ● 0b00001: Select channel 1 for threshold comparison ● 0b00010: Select channel 2 for threshold comparison ● ..... ● 0b10010: Select channel 18 for threshold comparison	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Bit	Name	Functional Description	Access
2	ALIGN	Conversion result alignment control ● 0b0: Conversion result stored as 16-bit right-aligned ● 0b1: Conversion result stored as 16-bit left-aligned	RW
1:0	MODE	ADC conversion mode selection ● 0b00: single conversion mode ● 0b01: SQR scan single sequence conversion mode ● 0b10: SQR scan continuous-sequence conversion mode or JQR scan conversion mode ● 0b11: SQR scan discontinuous sequences conversion mode	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

25.5.5 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 For more information, see ADC_CR0.SGLMUX	RW
24:20	CH4MUX	SQR scan conversion channel 4 selection For more information, see ADC_CR0.SGLMUX	RW
19:15	CH3MUX	SQR scan conversion channel 3 selection For more information, see ADC_CR0.SGLMUX	RW
14:10	CH2MUX	SQR scan conversion channel 2 selection For more information, see ADC_CR0.SGLMUX	RW
9:5	CH1MUX	SQR scan conversion channel 1 selection For more information, see ADC_CR0.SGLMUX	RW
4:0	CH0MUX	SQR scan conversion channel 0 selection For more information, see ADC_CR0.SGLMUX	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

25.5.6 ADC SQR Scan Conversion Channel Configuration Register 1 (ADC_SQR1)

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	0x00000000																															
Name	Reserved	CH11MUX						CH10MUX					CH9MUX					CH8MUX					CH7MUX					CH6MUX			
Access		RW						RW					RW					RW					RW					RW			

Bit	Name	Functional Description	Access
31:30	*Reserved*	-	-
29:25	CH11MUX	SQR scan conversion channel 11 selection For more information, see ADC_CR0.SGLMUX	RW
24:20	CH10MUX	SQR scan conversion channel 10 selection For more information, see ADC_CR0.SGLMUX	RW
19:15	CH9MUX	SQR scan conversion channel 9 selection For more information, see ADC_CR0.SGLMUX	RW
14:10	CH8MUX	SQR scan conversion channel 8 selection For more information, see ADC_CR0.SGLMUX	RW
9:5	CH7MUX	SQR scan conversion channel 7 selection For more information, see ADC_CR0.SGLMUX	RW
4:0	CH6MUX	SQR scan conversion channel 6 selection For more information, see ADC_CR0.SGLMUX	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

25.5.7 ADC SQR Scan Conversion Channel Configuration Register 2 (ADC_SQR2)

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				CH15MUX					CH14MUX					CH13MUX					CH12MUX			
Access									RW				RW					RW					RW					RW			

Bit	Name	Functional Description	Access
31:24	*Reserved*	-	-
23:20	CNT	Number of SQR scan conversions The actual number of conversions is CNT + 1. Channels selected by CH0MUX-CHxMUX are scanned by SQR in sequence (where x=CNT).	RW
19:15	CH15MUX	SQR scan conversion channel 15 selection For more information, see ADC_CR0.SGLMUX	RW
14:10	CH14MUX	SQR scan conversion channel 14 selection For more information, see ADC_CR0.SGLMUX	RW
9:5	CH13MUX	SQR scan conversion channel 13 selection For more information, see ADC_CR0.SGLMUX	RW
4:0	CH12MUX	SQR Scan conversion channel 12 selection For more information, see ADC_CR0.SGLMUX	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

25.5.8 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 The actual number of conversions is CNT + 1. Channels selected by CHxMUX-CH0MUX (x=CNT) are scanned by JQR in sequence.	RW
19:15	CH3MUX	JQR scan conversion channel 3 selection For more information, see ADC_CR0.SGLMUX	RW
14:10	CH2MUX	JQR scan conversion channel 2 selection For more information, see ADC_CR0.SGLMUX	RW
9:5	CH1MUX	JQR scan conversion channel 1 selection For more information, see ADC_CR0.SGLMUX	RW
4:0	CH0MUX	JQR scan conversion channel 0 selection For more information, see ADC_CR0.SGLMUX	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

25.5.9 ADC SQR Scan Conversion Channel x Result Register (ADC_SQRRESULTx) (x=0-3)

Offset: 0x050 (SQRRESULT0), 0x054 (SQRRESULT1), 0x058 (SQRRESULT2), 0x05C (SQRRESULT3)

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
31:16	*Reserved*	-	-
15:0	RESULT	ADC SQR scan conversion channelx conversion result	RO

25.5.10 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
31:16	*Reserved*	-	-
15:0	RESULT	ADC JQR scan conversion channelx conversion result	RO

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

25.5.11 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
31:16	*Reserved*	-	-
15:0	RESULT	ADC conversion results	RO

25.5.12 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.

25.5.13 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

25.5.14 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.

25.5.15 ADC Interrupt and DMA Trigger Enable Register (ADC_IER)

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																						OV	DEOJ	DEOC	EOC	EOJ	EOS	RGI	HTI	LTI	ESG
Access																							RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Functional Description	Access
31:10	*Reserved*	-	-
9	OV	ADC conversion result overflow interrupt enable ● 0b0: Disable ● 0b1: Enable	RW
8	DEOJ	JQR scan conversion triggers DMA ● 0b0: Disable ● 0b1: Enable	RW
7	DEOC	SQR scan single-channel conversion completion triggers DMA ● 0b0: Disable ● 0b1: Enable	RW
6	EOC	SQR scan single channel conversion complete interrupt enable ● 0b0: Disable ● 0b1: Enable	RW
5	EOJ	JQR scan conversion complete interrupt enable ● 0b0: Disable ● 0b1: Enable	RW
4	EOS	SQR scan conversion complete interrupt enable ● 0b0: Disable ● 0b1: Enable	RW
3	RGI	Interval value comparison interrupt enable ● 0b0: Disable ● 0b1: Enable	RW
2	HTI	High threshold comparison interrupt enable ● 0b0: Disable ● 0b1: Enable	RW
1	LTI	Low threshold comparison interrupt enable ● 0b0: Disable ● 0b1: Enable	RW
0	ESG	One-shot conversion interrupt enable ● 0b0: Disable ● 0b1: Enable	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

25.5.16 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																					RDY	OV	Reserved		EOC	EOJ	EOS	RGI	HTI	LTI	ESG
Access																						RO	RO			RO	RO	RO	RO	RO	RO	RO

Bit	Name	Functional Description	Access
31:11	*Reserved*	-	-
10	RDY	Initial state After enabling the ADC, 32 ADC clock cycles are required before starting the conversion: ● 0b0: ADC is initializing after enabling ADC. Starting ADC is invalid. ● 0b1: ADC enable initialization is completed and ADC conversion can be started.	RO
9	OV	ADC conversion result overflow flag ● 0b0: No overflow in conversion result ● 0b1: Overflow occurred in conversion result **Note** When a conversion result overflow occurs, the data storage method is handled according to the configuration of ADC_CR1.OVMD.	RO
8:7	*Reserved*	-	-
6	EOC	ADC SQR scan conversion single channel complete flag ● 0b0: Conversion not complete ● 0b1: Conversion complete	RO
5	EOJ	ADC JQR scan conversion complete flag ● 0b0: Conversion not complete ● 0b1: Conversion complete	RO
4	EOS	ADC SQR scan conversion complete flag ● 0b0: Conversion not complete ● 0b1: Conversion complete	RO
3	RGI	ADC conversion result comparison interval flag ● 0b0: ADC conversion result is outside the [ADC_LT, ADC_HT) range ● 0b1: ADC conversion result is within the range of [ADC_LT, ADC_HT)	RO
2	HTI	ADC conversion result comparison upper threshold flag ● 0b0: ADC conversion result is outside [ADC_HT, 4095] range ● 0b1: ADC conversion result is within the range [ADC_HT, 4095]	RO

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Bit	Name	Functional Description	Access
1	LTI	ADC conversion result comparison lower threshold flag ● 0b0: ADC conversion result is outside the interval [0, ADC_LT) ● 0b1: ADC conversion result is within the range [0, ADC_LT)	RO
0	ESG	ADC single conversion complete flag ● 0b0: Conversion not complete ● 0b1: Conversion complete	RO

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

25.5.17 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	0x000003FF																															
Name	Reserved																						OV	Reserved		EOC	EOJ	EOS	RGI	HTI	LTI	ESG
Access																							WO			WO	WO	WO	WO	WO	WO	WO

Bit	Name	Functional Description	Access
31:10	*Reserved*	-	-
9	OV	ADC conversion result overflow flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
8:7	*Reserved*	-	-
6	EOC	ADC SQR scan conversion single channel complete flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
5	EOJ	ADC JQR scan conversion complete flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
4	EOS	ADC SQR scan conversion complete flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
3	RGI	ADC conversion result comparison interval flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
2	HTI	ADC conversion result comparison upper threshold flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
1	LTI	ADC conversion result comparison lower threshold flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
0	ESG	ADC single conversion complete flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

25.5.18 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																										DELAY	TRIG				
Access																											RW	RW				

Bit	Name	Functional Description	Access
31:6	*Reserved*	-	-
5	DELAY	ADC trigger delay control ● 0b0: Directly use the TRIG selection signal to trigger ADC ● 0b1: TRIG-selected trigger signal activates GTIM, which then triggers the ADC	RW
4:0	TRIG	ADC trigger signal selection ● 0b00000: No peripheral trigger ● 0b00001: ATIM0 signal triggers ADC conversion ● 0b00010: ATIM3 signal triggers ADC conversion ● 0b00011: GTIM0 signal triggers ADC conversion ● 0b00100: GTIM1 signal triggers ADC conversion ● 0b00101: ATIM2 signal triggers ADC conversion ● 0b00110: USART0 interrupt triggers ADC conversion ● 0b00111: USART1 interrupt triggers ADC conversion ● 0b01000: LPUART0 interrupt triggers ADC conversion ● 0b01001: LPUART1 interrupt triggers ADC conversion ● 0b01010: SPI0 interrupt triggers ADC conversion ● 0b01011: SPI1 interrupt triggers ADC conversion ● 0b01100: DMA interrupt triggers ADC conversion ● 0b01101: VC0 interrupt triggers ADC conversion ● 0b01110: VC1 interrupt triggers ADC conversion. ● 0b01111: LVD interrupt triggers ADC conversion ● 0b10000: PA01 interrupt triggers ADC conversion ● 0b10001: PB01 interrupt triggers ADC conversion ● 0b10010: PA03 interrupt triggers ADC conversion ● 0b10011: PB03 interrupt triggers ADC conversion ● 0b10100: PA05 interrupt triggers ADC conversion ● 0b10101: PB05 interrupt triggers ADC conversion ● 0b10110: PA07 interrupt triggers ADC conversion ● 0b10111: PB07 interrupt triggers ADC conversion ● 0b11000: PA09 interrupt triggers ADC conversion ● 0b11001: PB09 interrupt triggers ADC conversion ● 0b11010: PA11 interrupt triggers ADC conversion ● 0b11011: PB11 interrupt triggers ADC conversion	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Bit	Name	Functional Description	Access
4:0	TRIG	● 0b11100: PA13 interrupt triggers ADC conversion ● 0b11101: PB13 interrupt triggers ADC conversion ● 0b11110: PA15 interrupt triggers ADC conversion ● 0b11111: PB15 interrupt triggers ADC conversion **Note** PA13 is configured as SWD by default and needs to be multiplexed as GPIO.	RW

Note

- The timer trigger ADC has a dedicated control signal, which is controlled inside the timer.
- The ADC is triggered using the rising edge of each interrupt flag bit. If repeated triggering is required, the interrupt flag needs to be cleared. If you do not need to enter the interrupt service routine, please do not enable the interrupt enable of the NVIC.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

25.5.19 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																										DELAY	TRIG				
Access																											RW	RW				

Bit	Name	Functional Description	Access
31:6	*Reserved*	-	-
5	DELAY	ADC trigger delay control ● 0b0: Directly use the TRIG selection signal to trigger ADC ● 0b1: The trigger signal selected by TRIG triggers GTIM0, and the GTIM0 signal triggers the ADC	RW
4:0	TRIG	ADC trigger signal selection ● 0b00000: No peripheral trigger ● 0b00001: ATIM0 signal triggers ADC conversion. ● 0b00010: ATIM3 signal triggers ADC conversion. ● 0b00011: GTIM0 signal triggers ADC conversion. ● 0b00100: GTIM1 signal triggers ADC conversion. ● 0b00101: ATIM2 signal triggers ADC conversion. ● 0b00110: USART0 interrupt triggers ADC conversion. ● 0b00111: USART1 interrupt triggers ADC conversion. ● 0b01000: LPUART0 interrupt triggers ADC conversion. ● 0b01001: LPUART1 interrupt triggers ADC conversion. ● 0b01010: SPI0 interrupt triggers ADC conversion. ● 0b01011: SPI1 interrupt triggers ADC conversion. ● 0b01100: DMA interrupt triggers ADC conversion. ● 0b01101: VC0 interrupt triggers ADC conversion. ● 0b01110: VC1 interrupt triggers ADC conversion. ● 0b01111: LVD interrupt triggers ADC conversion. ● 0b10000: PA01 interrupt triggers ADC conversion. ● 0b10001: PB01 interrupt triggers ADC conversion. ● 0b10010: PA03 interrupt triggers ADC conversion. ● 0b10011: PB03 interrupt triggers ADC conversion. ● 0b10100: PA05 interrupt triggers ADC conversion. ● 0b10101: PB05 interrupt triggers ADC conversion. ● 0b10110: PA07 interrupt triggers ADC conversion. ● 0b10111: PB07 interrupt triggers ADC conversion. ● 0b11000: PA09 interrupt triggers ADC conversion. ● 0b11001: PB09 interrupt triggers ADC conversion. ● 0b11010: PA11 interrupt triggers ADC conversion. ● 0b11011: PB11 interrupt triggers ADC conversion.	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Bit	Name	Functional Description	Access
4:0	TRIG	● 0b11100: PA13 interrupt triggers ADC conversion. ● 0b11101: PB13 interrupt triggers ADC conversion. ● 0b11110: PA15 interrupt triggers ADC conversion. ● 0b11111: PB15 interrupt triggers ADC conversion. **Note** PA13 is configured as SWD by default and needs to be multiplexed as GPIO.	RW

Note

- The timer trigger ADC has a dedicated control signal, which is controlled inside the timer.
- The ADC is triggered using the rising edge of each interrupt flag bit. If repeated triggering is required, the interrupt flag needs to be cleared. If you do not need to enter the interrupt service routine, please do not enable the interrupt enable of the NVIC.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

25.5.20 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 Can be read back correctly when CR0.EN=1 ● 0b0: Stop ● 0b1: Start	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

25.5.21 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 Can be read back correctly when CR0.EN=1 ● 0b0: Stop ● 0b1: Start	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

25.5.22 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 Can be read back correctly when CR0.EN=1 ● 0b0: Stop ● 0b1: Start	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

26 VC

26.1 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.

26.2 Features

- Support 64-stage voltage divider reference
- Support 11 external inputs and 4 on-chip 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

26.3 Functional Description

26.3.1 Function Block Diagram

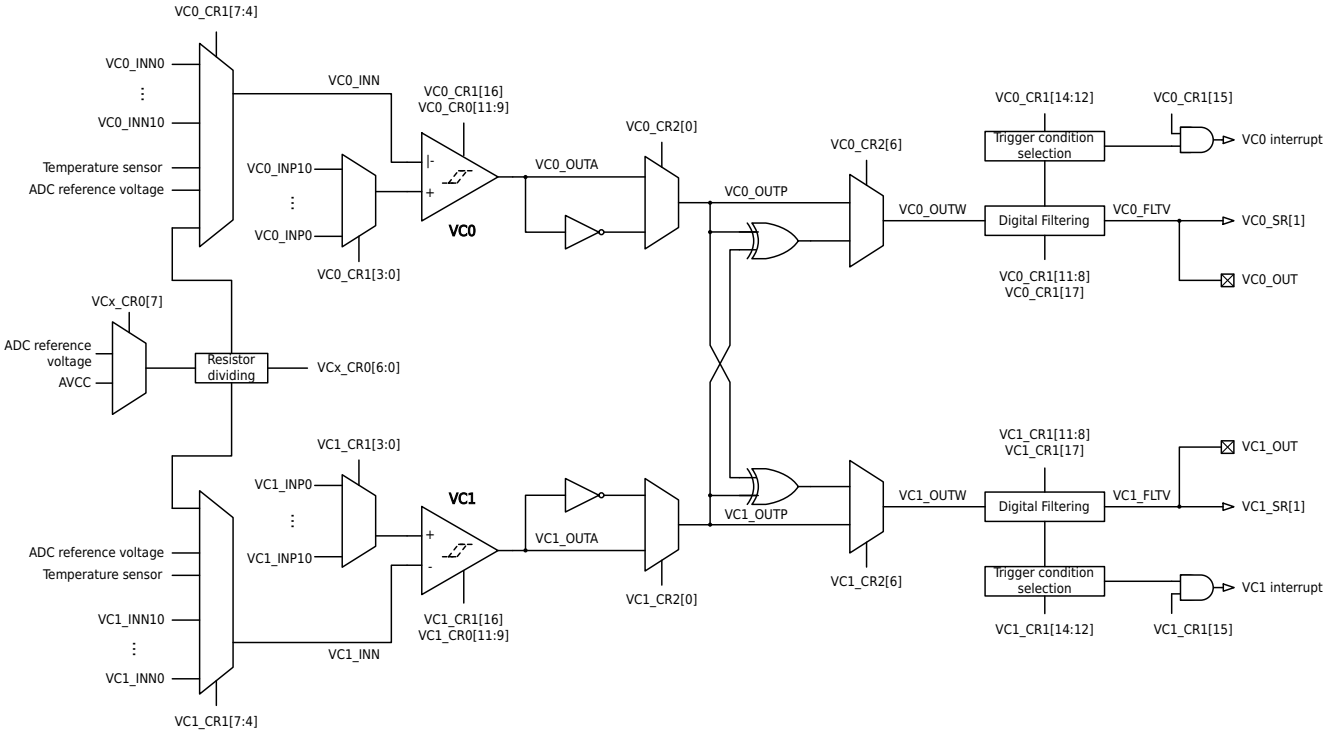


Figure 26-1 VC Function Block Diagram

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

26.3.2 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 the ADC reference voltage (1.5V or 2.5V) is used as the comparison reference voltage, or when VCx_CR0.BIAS=0b10/0b11, the BGR needs to be started (For the stabilization time T_{warmup} 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.

26.3.3 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.

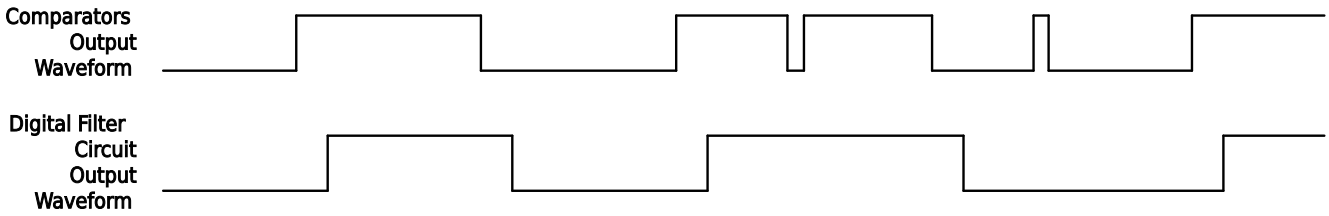


Figure 26-2 VC Filter Response Time

26.3.4 Hysteresis Function

The hysteresis function can be selected for the voltage comparator, and the diagram after the hysteresis function is enabled is as follows:

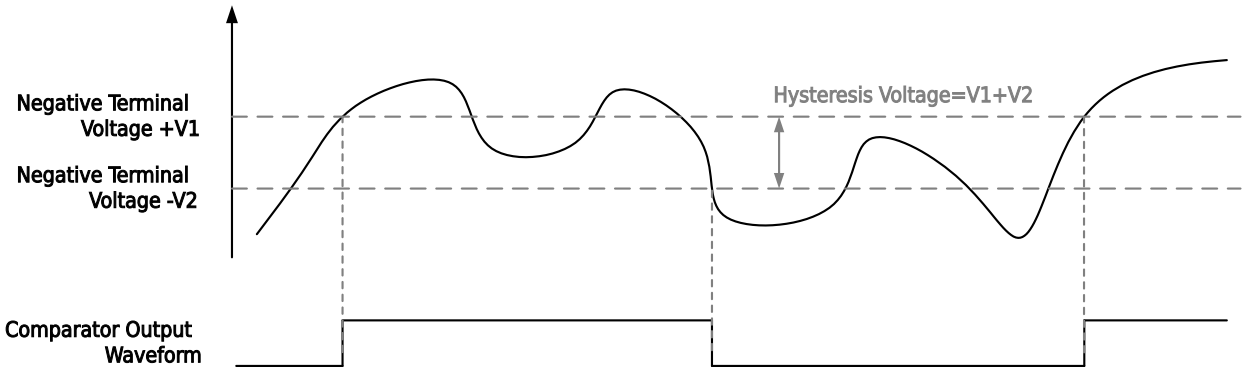


Figure 26-3 VC Hysteresis Function

26.3.5 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.

26.3.6 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.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

26.4 Register Description

26.4.1 Register List

Base address: 0x40002400

Table 26-1 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

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

26.4.2 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 ● 0b00: No hysteresis ● 0b01: Hysteresis voltage approximately 10mV ● 0b10: Hysteresis voltage approximately 20mV ● 0b11: Hysteresis voltage approximately 30mV	RW
9:8	BIAS	Power selection ● 0b00: Low power consumption level ● 0b01: Medium and low power consumption level ● 0b10: Medium power consumption level ● 0b11: High power consumption level **Note** 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". 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 ● 0b0: AVCC ● 0b1: The reference voltage of the ADC. The ADC and BGR need to be enabled **Note** VC0/VC1 share this bit setting.	RW
6	DIVEN	Resistor voltage divider circuit enable control ● 0b0: Disable ● 0b1: Enable **Note** 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 ● 0b000000: 1/64 Vref ● 0b000001: 2/64 Vref ● ... ● 0b111110: 63/64 Vref ● 0b111111: 64/64 Vref **Note** 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.

26.4.3 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														FLTCLK	EN	IE	LEVEL	RISING	FALLING	FLTTIME			FLTEN	NSEL			PSEL			
Access															RW	RW	RW	RW	RW	RW	RW			RW	RW			RW			

Bit	Name	Functional Description	Access
31:18	*Reserved*	-	-
17	FLTCLK	Filter clock selection ● 0b0: Built-in RC oscillator, frequency about 150kHz ● 0b1: PCLK **Note** When the filter clock is PCLK, VC cannot wake up the MCU in Deep-Sleep mode.	RW
16	EN	VC enable control ● 0b0: Disable ● 0b1: Enable	RW
15	IE	VC interrupt enable control ● 0b0: Disable ● 0b1: Enable	RW
14	LEVEL	Filter signal high level trigger interrupt enable control ● 0b0: Disable ● 0b1: Enable	RW
13	RISING	Filter signal rising edge trigger interrupt enable control ● 0b0: Disable ● 0b1: Enable	RW
12	FALLING	Filter signal falling edge trigger interrupt enable control ● 0b0: Disable ● 0b1: Enable	RW
11:9	FLTTIME	Filter time configuration ● 0b000: Filter out signals with a width less than 1 filter clock ● 0b001: Filter out signals with a width less than 3 filter clocks ● 0b010: Filter out signals with a width less than 7 filter clocks ● 0b011: Filter out signals with a width less than 15 filter clocks ● 0b100: Filter out signals with a width less than 63 filter clocks ● 0b101: Filter out signals with a width less than 255 filter clocks ● 0b110: Filter out signals with a width less than 1023 filter clocks	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Bit	Name	Functional Description	Access
11:9	FLTTIME	● 0b111: Filter out signals with a width less than 4095 filter clocks	RW
8	FLTEN	Filter function enable configuration ● 0b0: Disable ● 0b1: Enable	RW
7:4	NSEL	Voltage comparator negative input signal selection ● 0b0000: VCx_INN0 (PA00) ● 0b0001: VCx_INN1 (PA01) ● 0b0010: VCx_INN2 (PA02) ● 0b0011: VCx_INN3 (PA03) ● 0b0100: VCx_INN4 (PA04) ● 0b0101: VCx_INN5 (PA05) ● 0b0110: VCx_INN6 (PB00) ● 0b0111: VCx_INN7 (PB01) ● 0b1000: VCx_INN8 (PB12) ● 0b1001: VCx_INN9 (PB04) ● 0b1010: VCx_INN10 (PB05) ● 0b1011: Built-in resistor voltage divider output voltage ● 0b1100: Internal temperature sensor output voltage ● 0b1101: Setting prohibited ● 0b1110: The reference voltage configured for the ADC module (the ADC needs to be enabled) ● 0b1111: Core voltage	RW
3:0	PSEL	Voltage comparator positive input signal selection ● 0b0000: VCx_INP0 (PA00) ● 0b0001: VCx_INP1 (PA01) ● 0b0010: VCx_INP2 (PA02) ● 0b0011: VCx_INP3 (PA03) ● 0b0100: VCx_INP4 (PA04) ● 0b0101: VCx_INP5 (PA05) ● 0b0110: VCx_INP6 (PB00) ● 0b0111: VCx_INP7 (PB01) ● 0b1000: VCx_INP8 (PB12) ● 0b1001: VCx_INP9 (PB04) ● 0b1010: VCx_INP10 (PB05) ● 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.

26.4.4 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																									WINDOW	ATIMBK	ATIM3CLR	ATIM2CLR	ATIM1CLR	ATIM0CLR	POL
Access																										RW	RW	RW	RW	RW	RW	RW

Bit	Name	Functional Description	Access
31:7	*Reserved*	-	-
6	WINDOW	Window comparison function configuration ● 0b0: Disable ● 0b1: Enable	RW
5	ATIMBK	VCx filtered signal output to the ATIM0/1/2/3 brake enable control ● 0b0: Disable ● 0b1: Enable	RW
4	ATIM3CLR	VCx filtered signal output to the ATIM3 REFCLR enable control ● 0b0: Disable ● 0b1: Enable	RW
3	ATIM2CLR	VCx filtered signal output to the ATIM2 REFCLR enable control ● 0b0: Disable ● 0b1: Enable	RW
2	ATIM1CLR	VCx filtered signal output to the ATIM1 REFCLR enable control ● 0b0: Disable ● 0b1: Enable	RW
1	ATIM0CLR	VCx filtered signal output to the ATIM0 REFCLR enable control ● 0b0: Disable ● 0b1: Enable	RW
0	POL	VCx_OUTP signal polarity configuration ● 0b0: VCx_OUTP is high when the positive terminal is greater than the negative terminal ● 0b1: VCx_OUTP is low when the positive terminal is greater than the negative terminal	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

26.4.5 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 ● 0b0: Low level ● 0b1: High level	RO
0	INTF	Interrupt flag Write 0 to clear the flag, write 1 has no effect Read: ● 0b0: No interrupt occurred ● 0b1: Interrupt has occurred	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

27 LVD

27.1 Introduction

LVD can be used to monitor the voltage of VCC 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.

27.2 Features

- 4 monitoring sources: AVCC, PC13, PB08, PB07
- 16-stage threshold voltage, optional
- 8 trigger conditions, combinations of high level, rising edge and falling edge
- 2 trigger results: reset and interrupt
- 8-stage filter configuration to prevent false triggering
- With hysteresis function, strong anti-interference

27.3 Functional Description

27.3.1 LVD Block Diagram

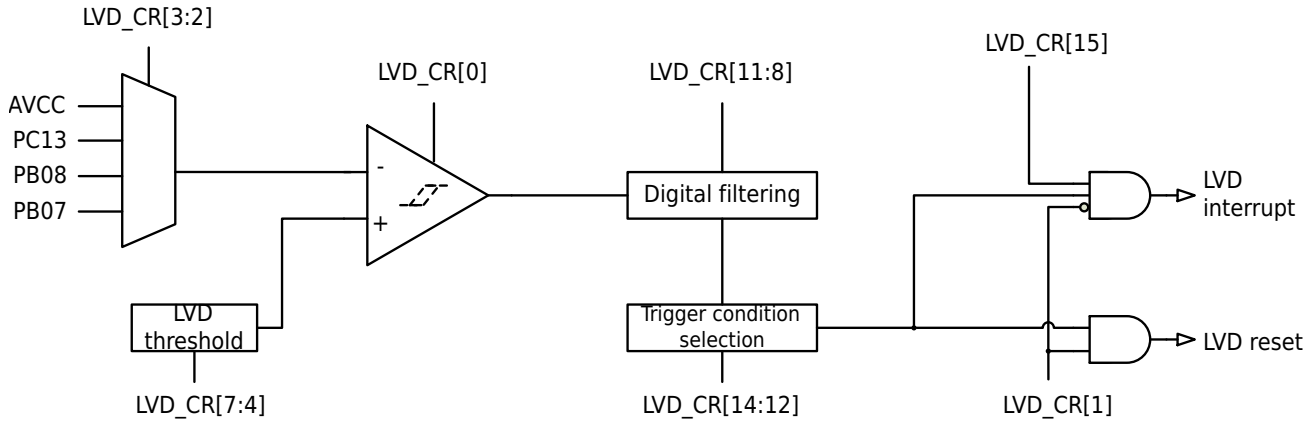


Figure 27-1 LVD Block Diagram

27.3.2 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.DEBOUNCETIME 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 150kHz.

Enable the digital filter module, the filter illustration is as follows:

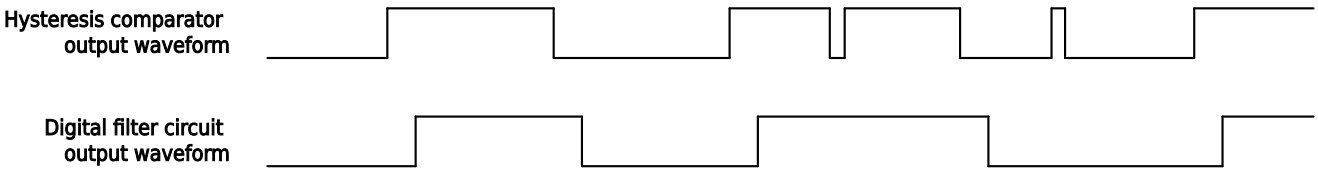


Figure 27-2 LVD Filter Output

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

27.3.3 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:

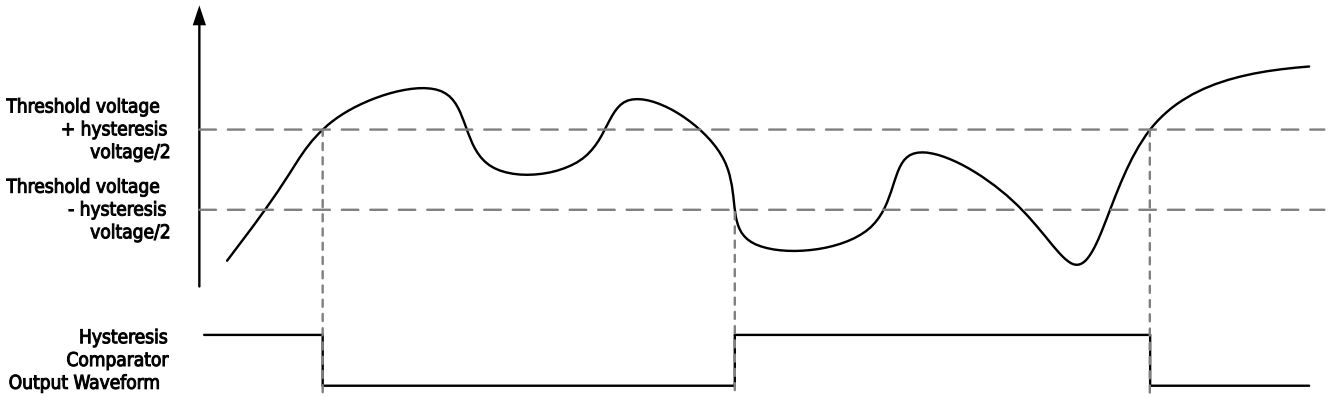


Figure 27-3 LVD Hysteresis Response

27.4 Operation Example

27.4.1 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 to select the voltage source to be monitored.

Step2. Configure LVD_CR.VTDS to select the threshold voltage of LVD.

Step3. Configure LVD_CR.DEBOUNCETIME to select the LVD filter time.

Step4. Set LVD_CR.FLTEN to enable LVD filtering.

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.

27.4.2 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 to select the voltage source to be monitored.

Step2. Configure LVD_CR.VTDS to select the threshold voltage of LVD.

Step3. Configure LVD_CR.DEBOUNCETIME to select the LVD filter time.

Step4. Set LVD_CR.FLTEN to enable LVD filtering.

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.IE to 1 to enable LVD interrupt.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Step8. Enable the LVD interrupt in the NVIC interrupt vector table.

Step9. Set LVD_CR.LVDEN to 1 to enable LVD.

Step10. Execute the operations required by the user in the LVD interrupt service routine; write 0x00 to LVD_SR to clear the interrupt flag before exiting the interrupt service routine.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

27.5 Register Description

27.5.1 Register List

Base address: 0x40002480

Table 27-1 Low Voltage Detector (LVD) Register Offset

Offset	Register	Description	Width
0x000	LVD_CR	LVD Configuration Register	32
0x004	LVD_SR	LVD Status Register	32

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

27.5.2 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																IE	HTEN	RTEN	FTEN	DEBOUNCETIME			FLTEN	VTDS			SOURCE		ACT	LVDEN
Access																	RW	RW	RW	RW	RW			RW	RW			RW		RW	RW

Bit	Name	Functional Description	Access
31:16	*Reserved*	-	-
15	IE	LVD interrupt enable ● 0b0: Disable ● 0b1: Enable	RW
14	HTEN	High level trigger enable The monitored voltage is lower than the threshold voltage ● 0b0: Disable ● 0b1: Enable	RW
13	RTEN	Rising edge trigger enable The monitored voltage changes from above the threshold voltage to below the threshold voltage ● 0b0: Disable ● 0b1: Enable	RW
12	FTEN	Falling edge trigger enable The monitored voltage changes from below the threshold voltage to above the threshold voltage ● 0b0: Disable ● 0b1: Enable	RW
11:9	DEBOUNCE-TIME	Digital filter time configuration ● 0b000: The filter width is less than 1 clock cycles ● 0b001: The filter width is less than 3 clock cycles ● 0b010: The filter width is less than 7 clock cycles ● 0b011: The filter width is less than 15 clock cycles ● 0b100: The filter width is less than 63 clock cycles ● 0b101: The filter width is less than 255 clock cycles ● 0b110: The filter width is less than 1023 clock cycles ● 0b111: The filter width is less than 4095 clock cycles **Note** The filter time configuration is only valid when FLTEN = 0b1.	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Bit	Name	Functional Description	Access
8	FLTEN	Digital filtering enable ● 0b0: Disable ● 0b1: Enable **Note** When the digital filter is enabled, the system will automatically enable the RCL clock.	RW
7:4	VTDS	LVD threshold voltage selection For the threshold voltage data, please refer to the "Electrical Specifications-Embedded Reset and LVD Module Characteristics" section of the "Datasheet".	RW
3:2	SOURCE	LVD monitoring source selection ● 0b00: AVCC power voltage ● 0b01: PC13 port input voltage ● 0b10: PB08 port input voltage ● 0b11: PB07 port input voltage	RW
1	ACT	LVD trigger action selection ● 0b0: NVIC interrupt ● 0b1: System reset	RW
0	LVDEN	LVD enable control ● 0b0: Disable ● 0b1: Enable	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

27.5.3 LVD Status Register (LVD_SR)

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 ● 0b0: Digital filter output low level ● 0b1: Digital filter output high level	RO
0	INTF	LVD interrupt flag Write 0 to clear the interrupt flag, write 1 has no effect. Read: ● 0b0: No interrupt occurred ● 0b1: Interrupt occurred	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

28 OPA

28.1 Introduction

This module is suitable for applications such as operational amplifiers and voltage followers. The input range is 0.2V to AVCC-0.2V, and the output range is 0.2V to AVCC-0.2V. According to actual needs, through the correct connection of external components, structures such as non-inverting amplifiers and unity gain amplifiers can be produced.

28.2 Features

- 2 independently configured op amps
- The input range of the OPA is 0.2V to AVCC-0.2V, and the output range is 0.2V to AVCC-0.2V
- Can be configured in the following modes:

  General purpose OPA
- Input offset supports user calibration

28.3 Functional Description

28.3.1 Function Block Diagram

OPAs can amplify small-signal analog input signals by using external components to form an amplifier, and output the amplified signal.

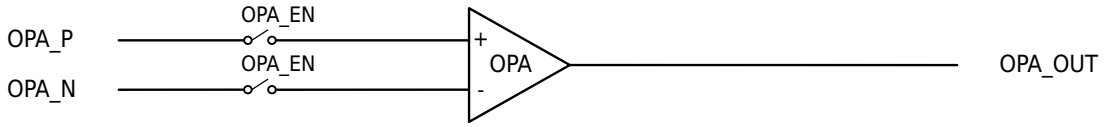


Figure 28-1 OPA Function Block Diagram

Note

OPA load resistance > 5 kΩ.

28.3.2 OPA Zero Calibration

OPA supports different modes of automatic zero calibration functions, which can be controlled by software or hardware.

For high-precision OPA applications, OPA zero calibration must be performed before each use of OPA. The zero calibration time is recommended to be more than 10μs. Please sample immediately after zero calibration.

Software Control

Step1. Set the automatic zero calibration bit OPAx_CR.AZEN to 1.

Step2. Set the automatic zero calibration selection bit OPA_AZ.CLK_SW_SET to 1.

Step3. Set the software calibration bit OPA_AZ.PULSE to 1.

Step4. Set the zero calibration time, wait for the zero calibration time to end, and clear OPA_AZ.PULSE.

Software Trigger Control

Step1. Set the automatic zero calibration bit OPAx_CR.AZEN to 1.

Step2. Set the time parameter bit OPA_AZ.CLK_SEL.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Step3. Set the software trigger calibration bit OPA_AZ.TRIGGER to 1.

ADC Trigger Control

Step1. Set the automatic zero calibration bit OPAx_CR.AZEN to 1.

Step2. Set time parameter OPA_AZ.CLK_SEL.

Step3. Set the ADC trigger bit OPA_AZ.ADCTR_EN to 1.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

28.4 Register Description

28.4.1 Register List

Base address: 0x40002600

Table 28-1 Operational Amplifier (OPA) Register Offset

Offset	Register	Description	Width
0x00	OPA0_CR	OPA0 Control Register	32
0x04	OPA1_CR	OPA1 Control Register	32
0x08	OPA_AZ	OPA Zero Calibration Control Register	32

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

28.4.2 OPAx Control Register (OPAx_CR) (x=0-1)

Offset: 0x00 (OPA0_CR), 0x04 (OPA1_CR)

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	0x00000180																															
Name	Reserved																						BIASSEL			Reserved					AZEN	EN
Access																							RW								RW	RW

Bit	Name	Functional Description	Access
31:10	*Reserved*	-	-
9:7	BIASSEL	OPAx bias current selection Must retain the default value	RW
6:2	*Reserved*	-	-
1	AZEN	Automatic zero calibration enabled ● 0b0: Disable ● 0b1: Enable	RW
0	EN	OPAx enabled ● 0b0: Disable ● 0b1: Enable	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

28.4.3 OPA Zero Calibration Control Register (OPA_AZ)

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																								CLK_SEL				CLK_SW_SET	PULSE	TRIGGER	ADCTR_EN
Access																									RW				RW	RW	WO	RW

Bit	Name	Functional Description	Access
31:8	*Reserved*	-	-
7:4	CLK_SEL	Automatic calibration of pulse width settings ● 0b0000: 1 PCLK cycle ● 0b0001: 2 PCLK cycles ● 0b0010: 4 PCLK cycles ● 0b0011: 8 PCLK cycles ● 0b0100: 16 PCLK cycles ● 0b0101: 32 PCLK cycles ● 0b0110: 64 PCLK cycles ● 0b0111: 128 PCLK cycles ● 0b1000: 256 PCLK cycles ● 0b1001: 512 PCLK cycles ● 0b1010: 1024 PCLK cycles ● 0b1011: 2048 PCLK cycles ● 0b11xx: 4096 PCLK cycles **Note** The recommended automatic calibration time is 10μs.	RW
3	CLK_SW_SET	Auto zeroing selection ● 0b0: software calibration disabled, software trigger calibration enabled ● 0b1: Software calibration enabled	RW
2	PULSE	Software calibration ● 0b0: Invalid ● 0b1: Enable	RW
1	TRIGGER	Software triggered calibration ● 0b0: Invalid ● 0b1: Enable **Note** This bit auto-clears.	WO

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Bit	Name	Functional Description	Access
0	ADCTR_EN	ADC startup triggers OPA auto-calibration ● 0b0: Disable ● 0b1: Enable **Note** For high-precision OPA applications, OPA zero calibration is required before each use of OPA. The calibration time should be longer than 10μs. After calibration, sampling should be performed immediately.	RW

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.

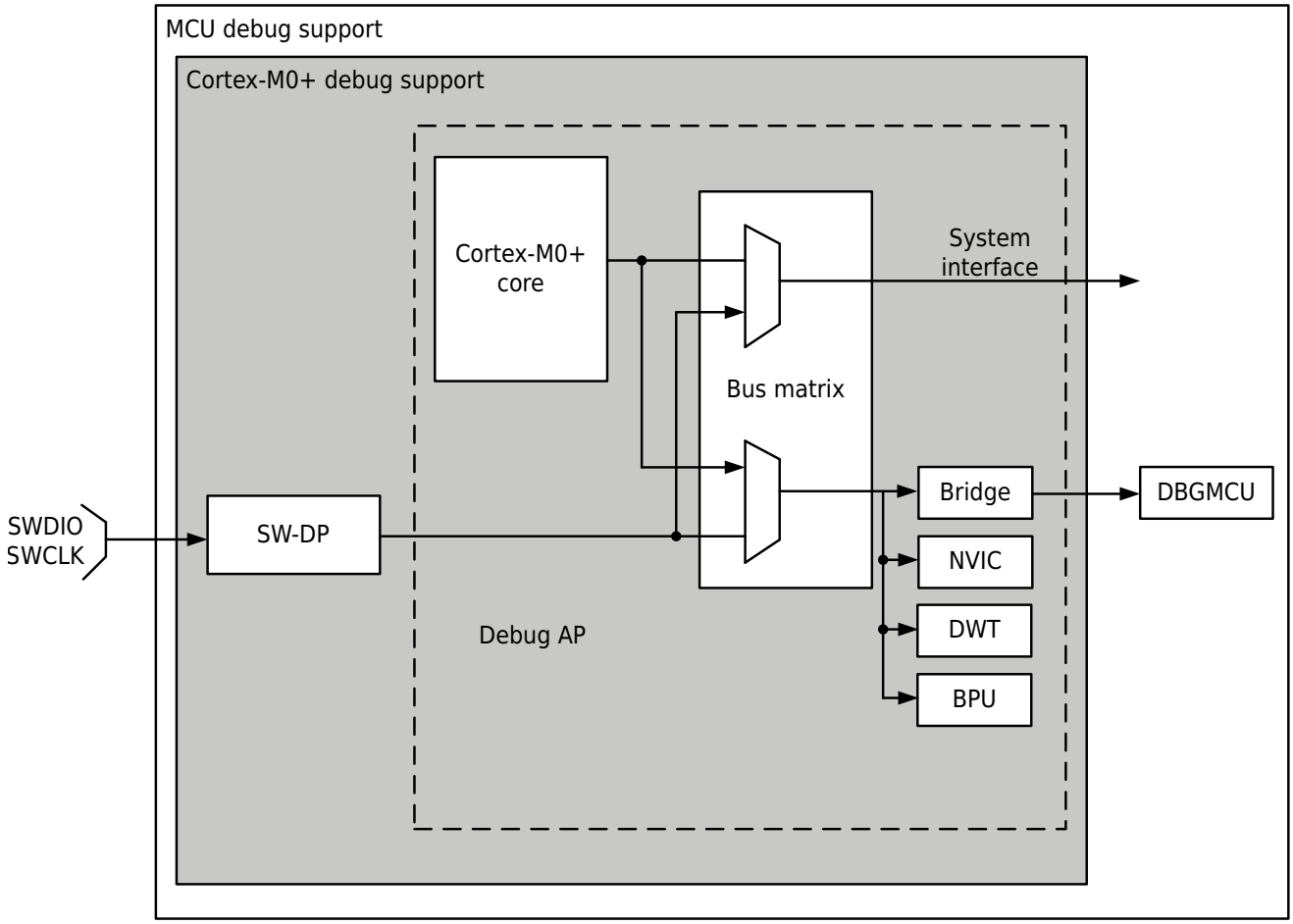


Figure 29-1 Debug Support Block Diagram

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 SYSCTRL_CR1.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 ● 0b1: AP access
2	RnW	● 0b0: Write request ● 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 ● 0b010: WAIT ● 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: ● Request system or debug power-on ● Configure transport operations for AP access ● Control compare and verify operations ● 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: ● Request system or debug power-on ● Configure transport operations for AP access ● Control compare and verify operations ● 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. ● Bit[31:24]: APSEL, select the current AP ● Bit[23:8]: Reserved ● Bit[7:4]: APBANKSEL, select the active 4-word register window within the current AP ● 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 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 This register selects the processor register to be read or written.
DCRDR	32-bit debug core register data registers 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 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: 0x00101F74

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

0x00101B40-0x00101B5F store the ASCII code representing the product model. If the product model is less than 32 bytes, fill it with 0x00.

Example: The product model represented by 48433332463035324A4154422D4C513438000000000000000000000000000000 is HC32F052JATB-LQ48.

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 In unit of Byte 0x00010000 means the Flash capacity is 64K Byte 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 In unit of Byte 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 0x0020 means the number of product pins is 32 0x0030 means the number of product pins is 48	RO
15:0	FLASHSEC-TORSIZE	Flash sector size In unit of Byte 0x0200 represents 512Byte.	RO

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

31 SysTick Timer

31.1 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.

31.2 Operation Example

31.2.1 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.

31.3 Register Description

31.3.1 Register List

Base address: 0xE000E010

Table 31-2 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.

31.3.2 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 Reading this register can clear COUNTFLAG flag ● 0b0: SysTick timer has not overflowed ● 0b1: SysTick timer has underflowed	RO
15:3	*Reserved*	-	-
2	CLKSOURCE	SysTick clock source selection ● 0b0: Setting prohibited ● 0b1: Use core clock (HCLK)	RW
1	TICKINT	SysTick interrupt enable ● 0b0: Disable ● 0b1: Enable	RW
0	ENABLE	SysTick timer enable ● 0b0: Disable ● 0b1: Enable	RW

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

31.3.3 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.

31.3.4 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 Read the register to obtain the current count value of the SysTick timer. Writing any value to the register will clear both the register itself and the COUNTFLAG.	RW

31.3.5 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 ● 0b0: Support reference clock (XTL) ● 0b1: Does not support reference clock (XTL) **Note** When this bit is set to 1, CTRL.CLKSOURCE must be set to 1.	RO
30	SKEW	TENMS accuracy indication ● 0b0: TENMS value represents an exact 10ms ● 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.

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
Rev1.50 Aug. 28, 2026	1. Operating Mode: In the "Functional/Module Working Status" chapter, RTC function is added. 2. SYSCTRL: In the "Clock Status Register (SYSCTRL_STATE)" chapter, the PLLLOCK bit is deleted. 3. 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. 4. LPUART: In the "Multi-Machine Communication" chapter, the description of Slave Address Configuration is optimized. 5. ATIM: 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. 6. ADC: In the "ADC Basic Configuration Register 1 (ADC_CR1)" chapter, the Bit10 function description removes the term "single conversion". At the same time, in the "ADC Interrupt Flag Register (ADC_IFR)" chapter, the Bit9 description "Only supports SQR scan mode" is moved to this bit. 7. SWD Debug Interface: The internal ID of SW-DP is modified to "0x0BC11477".
Rev1.41 Feb. 12, 2026	1. SYSCTRL: In the "Reset Flag Register (SYSCTRL_RSTFLAG)" chapter, add 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", simultaneously optimize the functional descriptions of each bit. 2. ATIM: In the "Mode2/3 Compare Capture Mode - Capture Input" chapter, add notice "When GATE is configured for compare output, its output mode only supports being set to force the output to 0 or 1 and having the PWM mode active (that is, the GATE output is valid when OCMB0 is configured as 0/1/6/7)". 3. LPUART: In the "Single-Wire Half-Duplex Data Transmitting and Reception" chapter, the description "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" is added.
Rev1.40 Sep. 12, 2025	1. SYSCTRL: In the "XTH Control Register (SYSCTRL_XTHCR)" chapter, the Bit[5:4] functional description is modified to specify the correct cycle count. 2. ATIM: In the "Mode 0 Counting Timer Function" chapter, the counting function description is optimized; in the "Mode 2/3 Compare-Capture Mode - Single Pulse Output" chapter, pulse counting timing diagram details are optimized with additional explanations; in the "Master-Slave Mode Control Register (ATIM_MSCR)" chapter, the Bit12 functional description removes the note "Automatically selects GATE port as CH0B input during PWM center-aligned complementary output"; in the "Mode 2/3 Compare-Capture Mode Registers - Master-Slave Mode Control Register (TIM_MSCR)" chapter, the Bit[2:0] functional description adds support for "Software update TG, write M23CR.TG" when MMS=0 for ATIM3.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Revision/Date	Changes
Rev1.40 Sep. 12, 2025	3. RTC: In the "Interrupt Enable Register (RTC_IER)" chapter, TSOV is deleted. 4. IWDT: In the "Control Register (IWDT_CR)" chapter, the PRS bit explanation is moved to the section overview with description changed to "This register can only be modified when IWDT_SR.PRSF=0"; in the "Status Register (IWDT_SR)" chapter, Bit0 name is changed from "IWDT_CR.PRS update flag" to "CR register update flag" with corresponding modifications to the functional description values, and the explanation is updated to "CR register can only be updated when PRSF=0"; 5. USART: In the "Features" chapter, the description "Supports wake-up from Sleep mode" is deleted; 6. LPUART: In the "Baud Rate Generation" chapter, the baud rate setting example description format is modified, referencing the original description from Rev1.10 version, with the omitted description "Mode1/Mode3 Baud Rate Setting Example" supplemented; 7. ADC: in this whole chapter, "sequential scan" description is modified to "SQR scan", "jumping scan" description is modified to "JQR scan"; in the "Scan Conversion Mode" chapter, the JQR scan function and examples during the ADC SQR scan process are deleted and the note of SQR scan and JQR scan cannot be performed simultaneously is added; In the "ADC Interrupt" chapter, descriptions related to ADC conversion result overflow interrupt (OV) are added; in the "Operation Example" chapter, the ADC startup completion condition "ADC_IFR.RDY=1" is added; the naming format for ADC_IER.OV, ADC_IFR.OV, and ADC_ICR.OV bits is unified as "ADC Conversion Result Overflow XXX"; in the "ADC Basic Configuration Register 1 (ADC_CR1)" chapter, the Bit10 description removes 0b0, a new note "This bit must be set to 1 in Single Conversion Mode and SQR Scan Conversion Mode" is added; in the "ADC Interrupt Flag Register (ADC_IFR)" chapter, Bit9 adds the note "Only SQR scan mode is supported".
Rev1.30 Apr. 3, 2025	1. Register bit/bit field function description: The value "Reserved" is modified to "Setting prohibited". 2. SPI: The description regarding maximum communication speed for master/slave modes is deleted from the "Features" section. 3. CTRIM: In the "Operation Example - RCH Clock Auto-Calibration Example" chapter, the configuration data is corrected based on the RCH target calibration value of 12MHz. 4. FLASH: Added 64KB FLASH introduction to maintain consistency with the datasheet. 5. RAM: Descriptions of unsupported RAM specifications are deleted to maintain consistency with the datasheet. 6. DMAC: The description of DMA-supported hardware request signals in the "Hardware Block Transfer Mode" and "Hardware Burst Transfer Mode" chapters is optimized. 7. ATIM: In the "Cycle-by-Cycle Current Limiting and Brake Control" chapter, the current limiting description adds "CR2.POL determines whether the REFCLR signal is active high (CR2.POL=0) or active low (CR2.POL=1)"; the brake control

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Revision/Date	Changes
Rev1.30 Apr. 3, 2025	description adds "VC0/1's CR2.POL determines whether the VC output brake level signal is active high (CR2.POL=0) or active low (CR2.POL=1)". 8. CTIM: Throughout the entire chapter, all "xxx_TOGP" descriptions are modified to "xxx_TOG". 9. LPTIM: Throughout the entire chapter, all "xxx_TOGP" descriptions are modified to "xxx_TOG"; in the "Interrupt Flag Clear Register (LPTIM_ICLR)" chapter, the bit description is changed from "Reads always return 1" to "Reads always return 0". 10. RTC: In the "Features" chapter, timestamp interrupt is added; the clock error compensation accuracy is modified from "0.95ppm" to "0.96ppm"; the description in the "RTC Interrupt" chapter is optimized with register simplification; in the "Timestamp Date Register (RTC_TSDR)" chapter, Bits[12:8] are merged into "MON". 11. ADC: In the "Conversion Timing and Speed" chapter, the language description is optimized with "conversion process" changed to "sampling process"; in the "ADC Basic Configuration Register 0 (ADC_CR0)" chapter, the Bit[13:12] functional description is modified from "conversion cycle" to "ADCCLK". 12. SysTick: In the "SysTick Calibration Value Register (CALIB)" chapter, the Bit31 name description and functional explanation language are optimized.
Rev1.20 Dec. 25, 2024	1. The manual style is refreshed according to the structured development process. 2. SYSCTRL: In the "System Clock Introduction" chapter, the description is changed from "External high-speed crystal oscillator clock XTL" to "External low-speed crystal oscillator clock XTL"; In the "PLL and RCHCLKD Switching Example with RCH as Reference Clock" chapter, BGR startup and shutdown steps are added. 3. RESET: In the "External Reset Pin Reset" chapter, the description is modified from "The glitch filter circuit will filter out glitch signals shorter than 20μS (typical value), therefore, the low-level signal applied to the reset pin must be longer than 20μS to ensure reliable chip reset" to "The glitch filter circuit will filter out glitch signals shorter than a certain time width, therefore, the low-level signal applied to the reset pin must exceed this time width to ensure reliable chip reset. For specific filtering time parameters, please refer to the corresponding 'Datasheet' ". 4. I2C: In the "Operating Modes" chapter, the slave receive mode description is modified from "If the master loses arbitration, it will directly enter slave receive mode" to "If the master loses arbitration, it will directly enter unaddressed slave receive mode"; in the "Operation Example" chapter, the description "I2C_CR.EN" is changed to "I2C_CR.ENS"; register naming is updated from "I2C_BRREN" to "I2C_TMRUN", "I2C_BRR" to "I2C_TM", and "I2C_DR" to "I2C_DATA"; all register bit name markings are refreshed to maintain consistency with the driver; the "Slave Address Match Register (I2C_MATCH)" chapter adds new clear operation instructions. 5. SPI: In the "SPI Status Register (SPI_SR)" chapter, the reset value is changed from "0x00000001" to "0x00000201". 6. FLASH: In the "Continuous Write Operation (Cont-Program)" chapter, an additional register modification step is added after completing all data writes;In the

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Revision/Date	Changes
Rev1.20 Dec. 25, 2024	"FLASH Security Protection" chapter, the description of register write protection is modified from "No write operations can be inserted between the two steps of writing 0x5A5A and writing 0xA5A5; otherwise, the target register value cannot be modified. If the modification fails, these three steps need to be performed again" to "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." in the "Chip Erase" chapter, step 5 is split into individual register operations, following the format of modifying enable registers. 7. RAM: In the "Parity Error Address Register (RAM_ERRADDR)" chapter, the Bit0 description is changed from "Parity error byte address" to "This address is cleared simultaneously when the interrupt flag is cleared". 8. DMAC: The description in the "DMA Transfer Pause" chapter is optimized. 9. ATIM: In the "Mode 0 Counting Timer Function" chapter, the register name CR is changed to M0CR, with added description "Gate control can enable/disable counting"; in the "Mode 0 Free Counting Block Diagram" and "Mode 0 Reload Counting Block Diagram", the description is modified from "PRS[2:0]" to "M0CR.PRS", and outputs "CHA/CHB" are changed to "TOG/TOGN"; in the "Mode 2/3 Compare-Capture Mode - Capture Input" chapter, the CH0B capture input selection diagram adds GATE signal with corresponding textual description; in the "Quadrature Encoder Counting Function" chapter, additional description is added for the table data: 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"; the "Timer Triggering ADC" chapter description is optimized; the "Cycle-by-Cycle Current Limiting and Brake Control" chapter description is optimized; in the "Mode 1 Pulse Width Measurement PWC Configuration Example" chapter, step 2 adds description "Then configure MSCR.IA0S/IB0S according to MSCR.TS settings (IB0S cannot be configured as 1) to select the channel to be measured"; in the "Master-Slave Mode Control Register (ATIM_MSCR)" chapter, Bit12 functional description adds "Automatically selects GATE port as CH0B input when PWM center-aligned complementary output is enabled" and Bit11 description is optimized; in the "Control Register (ATIM_M23CR)" chapter, Bits[22:21] are renamed from "Channel A Compare Interrupt Mode" to "Channel A Compare Match Mode" with optimized value description, and Bit16 description is changed from "VC selection is set in VCx_OUTCFG register" to "VC selection is set in VCx_CR2 register"; in the "Mode 2/3 Compare-Capture Mode Registers - Interrupt Flag Register (ATIM_IFR)" and "Mode 2/3 Compare-Capture Mode Registers - Interrupt Flag Clear Register (ATIM_ICLR)" chapters, Bits 19/18/13/12/10/9/7/6/4/3 add note "Only exists in ATIM3"; in the "Mode 2/3 Compare-Capture Mode Registers - ADC Trigger Control Register (ATIM_ADTR)" chapter, Bits 6/5/3/2 add note "Only exists in ATIM3"; in the "Channel 0 Control Register (ATIM_CRCH0)", "Channel 1 Control Register

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Revision/Date	Changes
Rev1.20 Dec. 25, 2024	(ATIM_CRCH1)" and "Channel 2 Control Register (ATIM_CRCH2)" chapters, Bits 15/14/3/2/1/0 descriptions are optimized; in the "Mode 2/3 Compare-Capture Mode Registers - Dead Time Register (ATIM_DTR)" chapter, Bits 12-7 markings are modified to be consistent with driver code and Bit8 description is optimized. 10. LPTIM: In the "Control Register (LPTIM_CR)" chapter, the Bit3 description is optimized. 11. RTC: In the "Control Register 0 (RTC_CR0)" chapter, the Bit15 functional description is deleted. 12. LPUART: In the "Transmit Data Example" chapter, step 8 description is changed from "Write data to be sent" to "When ISR.TXE is detected as 1, write one frame of data to be sent into SBUF register"; the original step 10 is modified to step 11 with description changed from "Jump to step 7" to "Jump to step 8"; a new step 10 description is added; in the "Receive Data Example", step 10 description is optimized, and step 13 description is changed from "Jump to step 8" to "Jump to step 9"; 13. CAN: In the "Receive Filter Register Group" description, "When SELMASK=0, points to ID MASK" is modified to "When SELMASK=1, points to ID MASK"; in the "Data Reception" chapter, "RAIE" and "RAIF" are changed to "RAFIE" and "RAFIF"; in the "CAN Transmission Control Register (CAN_TCTRL)" chapter, the Bit[1:0] functional description is updated from "When TTEN=0 and TTTBM=1" to "When TTEN=1 and TTTBM=1", with added description for 0b11 value; in the "CAN Reception Control Register (CAN_RCTRL)" chapter, the Bit[1:0] name is changed from "RSSTAT" to "RSTAT"; in the "CAN Filter Group Code and Mask Registers (CAN_ACF)" chapter, the Bit[28:0] functional description is optimized; in the "TTCAN Time Trigger Configuration Register (CAN_TTCFG)" chapter, the Bit5 functional description adds "Write 1 to clear, write 0 has no effect"; in the "TTCAN Reference Message Register (CAN_REF_MSG)" chapter, the Bit[28:0] functional description is changed from "When REF_ID=0: REF_ID[28:0] is valid" to "When REF_ID=0: REF_ID[10:0] is valid"; in the "TTCAN Trigger Configuration Register (CAN_TRG_CFG)" chapter, the Bit[10:8] functional description is modified from "Trigger time is set by TT_TRIG register" to "Trigger time is set by TRG_TRIG register". 14. CRC: The "Encoding Methods" chapter is supplemented with algorithm model information for both CRC-16 and CRC-32 encoding schemes. 15. I2CSLV: In the "Control Register (I2CSLV_CR)" chapter, Bit1 name is changed from "Busy" to "Slave Response Control"; in the "Receive Sample Position Register (I2CSLV_RSR)" chapter, the bit name description is modified to "Receive Sample Position Setting", with original content moved to functional description. 16. ADC: In the "Introduction" chapter, the description is changed from "2 12-bit" to "one 12-bit"; in the "Scan Conversion Mode - Interrupt Scan Conversion Mode" chapter, the example diagram of interrupt scan during sequential scan adds ADC_SQRRESULT1-3; in the "Continuous Conversion Accumulation Mode" chapter, the description is modified from "After starting continuous conversion, the ADC module sequentially converts channels configured in CHxMUX-CH0MUX until total conversion count is completed" to "After starting continuous conversion,

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.

Revision/Date	Changes
Rev1.20 Dec. 25, 2024	the ADC module sequentially converts channels configured in CH0MUX-CHxMUX until total conversion count is completed"; in the "Operation Flow for Starting ADC Sequential Scan Conversion via External Trigger" chapter, step 7 description is changed from "Set ADC_CR0.IE to 1" to "Set ADC_IER.EOS to 1"; in the "Operation Flow for Starting ADC Interrupt Scan Conversion via External Trigger" chapter, step 6 description is modified from "Set ADC_CR1.MODE to 1" to "Set ADC_CR1.MODE to 2"; in the "ADC Conversion External Trigger Source" chapter, the diagram description is changed from "TRIG=0X" to "TRIG=0b"; in the "ADC Interrupt" chapter, the description is modified from "ADC interrupt scan single channel conversion complete" to "ADC sequential scan single channel conversion complete"; in the "ADC Sequential Scan Conversion Channel Configuration Register 2 (ADC_SQR2)" chapter and "ADC Interrupt Scan Conversion Channel Configuration Register (ADC_JQR)" chapter, functional descriptions are added for the CNT bit. 17. Device Electronic Signature: In the "Pin Count/Flash Sector Size Configuration Register" chapter, a FLASHSECTORSIZE bit description is added, and the register offset address is modified accordingly.
Rev1.10 May 16, 2024	1. SYSCTRL: Modified partial content descriptions in sections 3.1.4 and 3.1.5; revised functional description notes for EXXTL and EXXTH bits in register SYSCTRL_CR2 (section 3.5.3). 2. NVIC: Modified the interrupt source names in Table 5-3. 3. FLASH: Modified the Flash erase/write programming description in section 10.4, adding operational instructions for FLASH_CR.OP; revised the functional description of the OP bit in the FLASH_CR register (section 10.6.2). 4. DMAC: Added ATIM trigger source description for DMAC_CONFA0 register in section 12.5.3. 5. ATIM: Modified the description of capture input CH0A signal list in Table 13-5; revised the DIR bit description in ATIMx_M23CR register (section 13.4.3.3). 6. ADC: In section 25.1 "Module Introduction", changed "19 input channels" to "18 input channels" and removed descriptions related to 1.2V; deleted 1.2V related descriptions in Figure 25-1; in section 25.11.2 "ADC_CR0 Register", removed 1.2V related descriptions and added a note to the REF bit that selecting 2.5V reference voltage requires operating voltage exceeding 2.8V. 7. VC: Deleted 1.2V-related descriptions in Figure 26-1 and VC_CR1 register; in section 26.2 "Features", changed "5 on-chip input signals" to "4 on-chip input signals".
Rev1.00 Nov. 15, 2023	First official release.

All Rights Reserved © XIAOHUA SEMICONDUCTOR Co., Ltd.