

HC32F002 Series

32-bit ARM[®] Cortex[®]-M0+ Microcontroller

Reference Manual

Rev1.30
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 HC32F002 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.

Contents

Preface	iii
1 Product Introduction	18
2 Storage and Architectures	19
2.1 Overview	19
2.2 System Address Division.....	20
2.3 Memory and Module Address Assignment.....	21
3 Operating Mode	22
3.1 Overview.....	22
3.2 Working Mode Switching.....	22
3.3 Active Mode.....	24
3.4 Sleep Mode.....	24
3.5 DeepSleep Mode.....	24
3.6 Functional/Module Working Status.....	25
3.7 Register Description.....	26
3.7.1 System Control Register (SCB_SCR) (Cortex-M0+ Core System Control Register).....	26
4 SYSCTRL	27
4.1 System Clock Introduction.....	27
4.1.1 Clock Architecture Diagram.....	27
4.1.2 Clock Types.....	27
4.2 System Clock Switching.....	28
4.2.1 System Clock Switching Introduction.....	28
4.2.2 Switching Process.....	28
4.2.3 Examples of Switching.....	29
4.2.4 Clock Switching Diagram.....	30
4.3 On-Chip Peripheral Clock Control.....	30
4.4 Interrupt Wakeup Control.....	30
4.4.1 Executing Interrupt Service after Waking from DeepSleep mode.....	31
4.4.2 Interrupt Service Not Executed after Waking from DeepSleep mode.....	31
4.4.3 Using Sleep-on-exit Feature.....	32
4.5 Reset Control.....	32
4.5.1 Reset Controller Overview.....	32
4.5.2 Reset Function.....	33
4.6 Register Description.....	34
4.6.1 Register List.....	34
4.6.2 System Control Register 0 (CR0).....	35
4.6.3 System Control Register 1 (CR1).....	36
4.6.4 System Control Register 2 (CR2).....	37
4.6.5 System Control Register 3 (CR3).....	38
4.6.6 RCH Control Register (RCH).....	39
4.6.7 RCL Control Register (RCL).....	40
4.6.8 On-Chip Peripheral Clock Control Register 0 (PERICKEN0).....	41
4.6.9 On-Chip Peripheral Clock Control Register 1 (PERICKEN1).....	43

4.6.10 On-Chip Peripheral Reset Control Register 0 (PERIRESET0).....	44
4.6.11 On-Chip Peripheral Reset Control Register 1 (PERIRESET1).....	45
4.6.12 Reset Flag Register (RESET_FLAG).....	46
4.6.13 Peripheral Debug Function Enable Register (DEBUGACTIVE).....	48
5 NVIC.....	49
5.1 Introduction.....	49
5.2 Functional Description.....	49
5.2.1 Interrupt Priority.....	49
5.2.2 Interrupt Digraph.....	50
5.2.3 Interrupt Input and Pending Behavior.....	51
5.2.4 Interrupt Waiting.....	53
5.2.5 Interrupt Source.....	53
5.2.6 Interrupt Structure Diagram.....	55
5.3 Operation Example.....	57
5.3.1 External Interrupt Enable.....	57
5.3.2 NVIC Interrupt Enable and Clear Enable.....	57
5.3.3 NVIC Interrupt Pending and Clear Pending.....	57
5.3.4 NVIC Interrupt Priority.....	57
5.3.5 NVIC Interrupt Mask.....	57
5.4 Register Description.....	59
5.4.1 Register List.....	59
5.4.2 Interrupt Enable Set Register (NVIC_ISER).....	60
5.4.3 Interrupt Enable Clear Register (NVIC_ICER).....	60
5.4.4 Interrupt Pending State Set Register (NVIC_ISPR).....	61
5.4.5 Interrupt Pending State Clear Register (NVIC_ICPR).....	61
5.4.6 Interrupt Priority Register (NVIC_IPR0).....	62
5.4.7 Interrupt Priority Register (NVIC_IPR1).....	62
5.4.8 Interrupt Priority Register (NVIC_IPR2).....	63
5.4.9 Interrupt Priority Register (NVIC_IPR3).....	63
5.4.10 Interrupt Priority Register (NVIC_IPR4).....	64
5.4.11 Interrupt Priority Register (NVIC_IPR5).....	64
5.4.12 Interrupt Priority Register (NVIC_IPR6).....	65
5.4.13 Interrupt Priority Register (NVIC_IPR7).....	65
5.4.14 Interrupt Mask Special Register (PRIMASK).....	66
6 GPIO.....	67
6.1 Introduction.....	67
6.2 Features.....	67
6.3 Functional Description.....	68
6.3.1 Port Circuit Block Diagram.....	68
6.3.2 Port Mode Configuration.....	68
6.3.3 Port Reset Status.....	69
6.3.4 Port Analog Function.....	69
6.3.5 Port General Input/Output Function.....	70
6.3.6 Port Multiplexing Function.....	72
6.3.7 Port Clock Output.....	73

6.3.8 Port External Interrupt.....	74
6.4 Register Description.....	75
6.4.1 Register List.....	75
6.4.2 Port Digital-to-Analog Configuration Register (GPIO_ADS).....	77
6.4.3 Port Input/Output Direction Configuration Register (GPIO_DIR).....	78
6.4.4 Port Output Type Register (GPIO_OpenDrain).....	79
6.4.5 Port Pull-Up Register (GPIO_PU).....	80
6.4.6 Port Data Input Register (GPIO_IN).....	80
6.4.7 Port Data Output Register (GPIO_OUT).....	81
6.4.8 Port Reset Register (GPIO_BRR).....	82
6.4.9 Port Set Reset Register (GPIO_BSRR).....	83
6.4.10 Port Multiplexing Function Low Register (GPIO_AFRL).....	84
6.4.11 Port High Level Interrupt Enable Configuration Register (GPIO_HIGHIE).....	85
6.4.12 Port Low Level Interrupt Enable Configuration Register (GPIO_LOWIE).....	86
6.4.13 Port Rising Edge Interrupt Enable Configuration Register (GPIO_RISEIE).....	87
6.4.14 Port Falling Edge Interrupt Enable Configuration Register (GPIO_FALLIE).....	88
6.4.15 Port Interrupt Status Register (GPIO_IFR).....	89
6.4.16 Port Interrupt Clear Register (GPIO_ICR).....	89
6.4.17 Port Auxiliary Function Configuration Register 1 (GPIO_CR1).....	90
6.4.18 Port Auxiliary Function Configuration Register 4 (GPIO_CR4).....	91
7 FLASH.....	92
7.1 Introduction.....	92
7.2 Functional Description.....	92
7.2.1 Capacity Division.....	92
7.2.2 Read Wait Period.....	92
7.2.3 Flash Security Protection.....	92
7.3 Operation Example.....	94
7.3.1 Sector Erase.....	94
7.3.2 Chip Erase.....	94
7.3.3 Program.....	95
7.3.4 Read.....	97
7.4 Register Description.....	98
7.4.1 Register List.....	98
7.4.2 Control Register (FLASH_CR).....	99
7.4.3 Interrupt Flag Register (FLASH_IFR).....	100
7.4.4 Interrupt Flag Clear Register (FLASH_ICLR).....	100
7.4.5 Sequential Register (FLASH_BYPASS).....	101
7.4.6 Erase Write Protection Register (FLASH_SLOCK).....	102
7.4.7 Read Waiting Period Register (FLASH_WAIT).....	103
7.4.8 Read Protection Status Register (FLASH_LOCKSTATE).....	103
8 RAM.....	104
8.1 Introduction.....	104
8.2 Functional Description.....	104
8.2.1 RAM Address Range.....	104
8.2.2 Read and Write Bit Width.....	104

9 BTIM	105
9.1 Introduction	105
9.2 Features	105
9.3 Functional Description	106
9.3.1 Function Block Diagram	106
9.3.2 Filter Unit	106
9.3.3 Counting Unit	106
9.3.4 Timer Mode	107
9.3.5 Counter Mode	108
9.3.6 Trigger Start Mode	108
9.3.7 Gating Mode	109
9.3.8 Timer Cascade	110
9.4 Register Description	111
9.4.1 Register List	111
9.4.2 Reload Register (BTIM_ARR)	112
9.4.3 Count Register (BTIM_CNT)	112
9.4.4 Control Register (BTIM_CR)	113
9.4.5 Interrupt Enable Control Register (BTIM_IER)	114
9.4.6 Interrupt Flag Register (BTIM_IFR)	115
9.4.7 Interrupt Flag Clear Register (BTIM_ICR)	115
9.4.8 Composite Interrupt Flag Register (BTIM345_AIFR)	116
9.4.9 Composite Interrupt Flag Clear Register (BTIM345_AICR)	117
10 GTIM	118
10.1 Introduction	118
10.2 Features	118
10.3 Functional Description	119
10.3.1 Function Block Diagram	119
10.3.2 Filter Unit	119
10.3.3 Counting Unit	119
10.3.4 Timer Mode	120
10.3.5 Counter Mode	121
10.3.6 Trigger Start Mode	121
10.3.7 Gating Mode	122
10.3.8 Compare Capture Function	123
10.3.9 Timer Cascade	125
10.3.10 On-Chip Peripheral Interconnection	125
10.4 Register Description	126
10.4.1 Register List	126
10.4.2 Reload Register (GTIM_ARR)	127
10.4.3 Count Register (GTIM_CNT)	127
10.4.4 Control Register 1 (GTIM_CR1)	128
10.4.5 Control Register 0 (GTIM_CR0)	129
10.4.6 Interrupt Enable Control Register (GTIM_IER)	131
10.4.7 Interrupt Flag Register (GTIM_IFR)	132
10.4.8 Interrupt Flag Clear Register (GTIM_ICR)	133

10.4.9 Compare Capture Control Register (GTIM_CMMR).....	134
10.4.10 Compare Capture Register (GTIM_CCRy) (y=0-3).....	135
11 ATIM.....	136
11.1 Introduction.....	136
11.2 Features.....	136
11.3 Functional Description.....	137
11.3.1 Timer Clock.....	137
11.3.2 Timer Counter.....	137
11.3.3 Timer Prescaler.....	137
11.3.4 Mode 0 Count Timer Function.....	138
11.3.5 Mode 1 Pulse Width Measurement PWC.....	140
11.3.6 Mode2/3 Compare Capture Mode.....	143
11.3.7 Mode2/3 Slave Mode.....	161
11.3.8 Quadrature Encoder Counting Function.....	162
11.3.9 Timer Triggers ADC.....	164
11.3.10 Brake Control.....	165
11.3.11 Timer Interconnection.....	165
11.3.12 Input Interconnection.....	165
11.4 Operation Example.....	166
11.4.1 Mode0 Counting Timer Function Configuration Example.....	166
11.4.2 Mode1 Pulse Width Measurement PWC Configuration Example.....	166
11.4.3 Mode2/3 Compare Capture Mode Configuration Example.....	167
11.4.4 Complementary PWM Output B Compare as ADC Trigger Function.....	168
11.5 Register Description.....	169
11.5.1 Register List.....	169
11.5.2 Mode0 Count Timer Register.....	171
11.5.3 Mode1 Pulse Width Measurement PWC Register.....	177
11.5.4 Mode2/3 Compare Capture Mode Register.....	184
12 CTRIM.....	205
12.1 Introduction.....	205
12.2 Features.....	205
12.3 Functional Description.....	205
12.3.1 RCH Clock Calibration Mode.....	205
12.3.2 RCL Clock Calibration Mode.....	206
12.3.3 Low Power Timer Mode.....	206
12.4 Operation Example.....	207
12.4.1 RCH Clock Automatic Calibration Example.....	207
12.4.2 RCL Clock Automatic Calibration Example.....	207
12.5 Register Description.....	209
12.5.1 Register List.....	209
12.5.2 Autoload Register (CTRIM_ARR).....	210
12.5.3 Error Counter Register (CTRIM_CNT).....	210
12.5.4 Control Register 0 (CTRIM_CR0).....	211
12.5.5 Control Register 1 (CTRIM_CR1).....	213
12.5.6 Interrupt Enable Register (CTRIM_IER).....	215

12.5.7 Interrupt and Status Register (CTRIM_ISR).....	216
12.5.8 Interrupt Status Flag Register (CTRIM_ICR).....	217
12.5.9 Error Result Register (CTRIM_FCAP).....	218
12.5.10 TRIM Result Register (CTRIM_TVAL).....	219
12.5.11 Error Upper Limit Register (CTRIM_FLIM).....	219
13 IWDT.....	220
13.1 Introduction.....	220
13.2 Features.....	220
13.3 Functional Description.....	220
13.3.1 Function Block Diagram.....	220
13.3.2 Window Watchdog Mode.....	220
13.3.3 Independent Watchdog Mode.....	221
13.3.4 Watchdog Overflow Handling.....	221
13.3.5 Watchdog Pause.....	221
13.3.6 Watchdog Stop and Run Again.....	221
13.3.7 Register Write Protection.....	221
13.4 Operation Example.....	221
13.4.1 Configure IWDT as Window Watchdog.....	221
13.4.2 Configure IWDT as Independent Watchdog.....	221
13.5 Register Description.....	223
13.5.1 Register List.....	223
13.5.2 Key Value Register (IWDT_KR).....	224
13.5.3 Control Register (IWDT_CR).....	225
13.5.4 Reload Register (IWDT_ARR).....	226
13.5.5 Status Register (IWDT_SR).....	227
13.5.6 Window Register (IWDT_WINR).....	228
14 WWDT.....	229
14.1 Introduction.....	229
14.2 Features.....	229
14.3 Functional Description.....	229
14.3.1 Function Block Diagram.....	229
14.3.2 Enable Watchdog.....	230
14.3.3 Configure Watchdog Timeout.....	230
14.3.4 Update Down Counter.....	230
14.3.5 Pre-overflow Interrupt.....	230
14.4 Operation Example.....	230
14.4.1 Example of Configuring Watchdog Window Overflow Period.....	230
14.5 Register Description.....	231
14.5.1 Register List.....	231
14.5.2 Control Register 0 (WWDT_CR0).....	232
14.5.3 Control Register 1 (WWDT_CR1).....	233
14.5.4 Status Register (WWDT_SR).....	234
15 LPUART.....	235
15.1 Introduction.....	235
15.2 Features.....	235

15.3 Functional Description.....	236
15.3.1 Function Block Diagram.....	236
15.3.2 Clock Description.....	236
15.3.3 Working Mode.....	236
15.3.4 Baud Rate Generation.....	240
15.3.5 Frame Error Detection.....	243
15.3.6 Multi-Machine Communication.....	243
15.3.7 Hardware Flow Control.....	244
15.4 Operation Example.....	246
15.4.1 Transmit Data Example.....	246
15.4.2 Receive Data Example.....	246
15.5 Register Description.....	247
15.5.1 Register List.....	247
15.5.2 Data Register (LPUART_DR).....	248
15.5.3 Control Register (LPUART_CR).....	249
15.5.4 Address Register (LPUART_ADDR).....	252
15.5.5 Address Mask Register (LPUART_ADDRMASK).....	252
15.5.6 Interrupt Flag Bit Register (LPUART_ISR).....	253
15.5.7 Interrupt Flag Bit Clear Register (LPUART_ICR).....	254
15.5.8 Baud Rate Register (LPUART_BRR).....	254
16 SPI.....	255
16.1 Introduction.....	255
16.2 Features.....	255
16.3 Functional Description.....	255
16.3.1 Function Block Diagram.....	255
16.3.2 Communication Format and Timing.....	255
16.3.3 Slave Select Pin Configuration.....	256
16.3.4 Full-duplex Communication.....	257
16.3.5 Single-wire Half-duplex Communication.....	258
16.3.6 Simplex Communication.....	258
16.3.7 Multi-machine Communication.....	259
16.3.8 Status Flags and Error Flags.....	260
16.3.9 Interrupt Generation and Clearing.....	263
16.4 Operation Example.....	264
16.4.1 Example of Sending and Receiving Data in Full-duplex by the Master.....	264
16.4.2 Example of Sending and Receiving Data in Full-duplex by the Slave.....	264
16.4.3 Single-wire Half-duplex Sending and Receiving Data Example.....	265
16.5 Register Description.....	266
16.5.1 Register List.....	266
16.5.2 SPI Control Register 0 (SPI_CR0).....	267
16.5.3 SPI Control Register 1 (SPI_CR1).....	269
16.5.4 SPI Single-wire Half-duplex Output Enable Register (SPI_HDOE).....	270
16.5.5 SPI Internal Slave Select Register (SPI_SSI).....	270
16.5.6 SPI Status Register (SPI_SR).....	271
16.5.7 SPI Interrupt Clear Register (SPI_ICR).....	273

16.5.8 SPI Data Register (SPI_DR).....	274
17 I2C.....	275
17.1 Introduction.....	275
17.2 Features.....	275
17.3 Protocol Description.....	275
17.3.1 Data Transmission on the I2C Bus.....	275
17.3.2 Acknowledgment Signal on the I2C Bus.....	276
17.3.3 Arbitration on the I2C Bus.....	277
17.4 Functional Description.....	278
17.4.1 Function Block Diagram.....	279
17.4.2 Serial Clock Generator.....	279
17.4.3 Input Filter.....	280
17.4.4 Address Comparator.....	280
17.4.5 Acknowledgement Flag Bit.....	280
17.4.6 Interrupt Generator.....	280
17.4.7 Operating Mode.....	280
17.4.8 Status Code Expression.....	285
17.5 Operation Example.....	287
17.5.1 Master Sending Example.....	287
17.5.2 Master Receiving Example.....	287
17.5.3 Slave Sending Example.....	288
17.5.4 Slave Receiving Example.....	289
17.6 Register Description.....	290
17.6.1 Register List.....	290
17.6.2 Baud Rate Generator Enable Register (I2C_BRREN).....	291
17.6.3 Baud Rate Counter Configuration Register (I2C_BRR).....	291
17.6.4 Configuration Register (I2C_CR).....	292
17.6.5 Data Register (I2C_DATA).....	294
17.6.6 Status Register (I2C_STAT).....	294
17.6.7 Slave Address Register 0 (I2C_ADDR0).....	295
17.6.8 Slave Address Register 1 (I2C_ADDR1).....	295
17.6.9 Slave Address Register 2 (I2C_ADDR2).....	296
17.6.10 Slave Address Match Register (I2C_MATCH).....	297
18 ADC.....	298
18.1 Introduction.....	298
18.2 Features.....	298
18.3 Functional Description.....	298
18.3.1 ADC Block Diagram.....	298
18.3.2 Conversion Timing and Conversion Speed.....	299
18.3.3 Single Conversion Mode.....	299
18.3.4 ADC Conversion External Trigger Source.....	299
18.3.5 Continuous Conversion Mode.....	300
18.3.6 Measure AVCC Voltage.....	300
18.3.7 ADC Interrupt.....	300
18.4 Operation Example.....	301

18.4.1 ADC Single Conversion Operation Flow.....	301
18.4.2 ADC Continuous Conversion Operation Flow.....	301
18.5 Register Description.....	302
18.5.1 Register List.....	302
18.5.2 ADC Basic Configuration Register 0 (ADC_CR0).....	303
18.5.3 ADC Basic Configuration Register 1 (ADC_CR1).....	304
18.5.4 ADC Conversion Result Register (ADC_RESULT).....	305
18.5.5 ADC Interrupt Flag Register (ADC_IFR)	305
18.5.6 ADC Interrupt Enable Register (ADC_IER).....	306
18.5.7 ADC Flag Clear Register (ADC_ICR).....	306
18.5.8 ADC External Interrupt Trigger Source Configuration Register (ADC_EXTTRIGGER).....	307
18.5.9 ADC Conversion Start Control Register (ADC_START).....	308
18.5.10 ADC Continuous Conversion Start Control Register (ADC_ALLSTART).....	309
19 LVD.....	310
19.1 Introduction.....	310
19.2 Features.....	310
19.3 Functional Description.....	310
19.3.1 LVD Block Diagram.....	310
19.3.2 Digital Filtering.....	310
19.3.3 Hysteresis Function.....	311
19.4 Operation Example.....	311
19.4.1 LVD Configured as Low Voltage Reset.....	311
19.4.2 LVD Configured as Voltage Change Interrupt.....	312
19.5 Register Description.....	313
19.5.1 Register List.....	313
19.5.2 LVD Configuration Register (LVD_CR).....	314
19.5.3 LVD Status Register (LVD_SR).....	316
20 Device Electronic Signature.....	317
20.1 Introduction.....	317
20.2 Functional Description.....	317
20.2.1 Product Unique Identifier (UID).....	317
20.2.2 One-Time Programming Area (OTP Area).....	317
20.3 Register Description.....	318
20.3.1 Product Unique Identifier (UID) Register (80 bits).....	318
20.3.2 Product Model Register.....	318
20.3.3 FLASH Capacity Register.....	319
20.3.4 RAM Capacity Register.....	319
20.3.5 Pin Count/Flash Sector Size Configuration Register.....	320
21 SWD Debug Interface.....	321
21.1 Introduction.....	321
21.2 Functional Description.....	321
21.2.1 Debug System Diagram.....	321
21.2.2 ARM® Reference Documentation.....	322
21.2.3 Debug Port Pins.....	322

21.2.4 SWD Port.....	323
21.2.5 Core Debugging.....	327
21.2.6 BPU (Breakpoint Unit).....	327
21.2.7 DWT (Data Watchpoint).....	327
21.2.8 MCU Debug Component (DBG).....	328
22 SysTick Timer.....	329
22.1 Introduction.....	329
22.2 Operation Example.....	329
22.2.1 Set SysTick.....	329
22.3 Register Description.....	330
22.3.1 Register List.....	330
22.3.2 SysTick Control and Status Register (CTRL).....	331
22.3.3 SysTick Reload Register (LOAD).....	332
22.3.4 SysTick Current Value Register (VAL).....	333
22.3.5 SysTick Calibration Value Register (CALIB).....	333
A Glossary.....	334
Revision History.....	335

List of Figures

Figure 2-1 System Architecture Schematic Diagram.....	19
Figure 3-1 Working Mode State Transition Diagram.....	22
Figure 4-1 Clock Architecture Diagram.....	27
Figure 4-2 Clock Switching Diagram.....	30
Figure 4-3 Reset Source Diagram.....	33
Figure 5-1 Interrupt Vector Table.....	51
Figure 5-2 Interrupt Active and Pending States.....	52
Figure 5-3 Interrupt Pending State is Cleared and Then Reasserted.....	52
Figure 5-4 When the Interrupt Exits, If the Interrupt Request Remains High, It Will Cause the Interrupt Processing to Be Executed Again.....	52
Figure 5-5 Interrupt Pending Interrupts Generated During Interrupt Processing Can Also Be Acknowledged.....	53
Figure 5-6 Interrupt Structure Diagram.....	56
Figure 6-1 General Port Circuit Diagram.....	68
Figure 6-2 HiZ-Analog Configuration Diagram.....	70
Figure 6-3 Input Floating/Pull-up/Pull-down Configuration Diagram.....	71
Figure 6-4 Read Port Pin Data Synchronization Diagram.....	71
Figure 6-5 Output Configuration Diagram.....	72
Figure 6-6 Multiplexed Function Configuration Diagram.....	73
Figure 6-7 Port External Interrupt Diagram.....	74
Figure 9-1 BTIM Block Diagram.....	106
Figure 9-2 BTIM Filter Unit Diagram.....	106
Figure 9-3 BTIM Counter Waveform.....	107
Figure 9-4 BTIM Timing Mode Block Diagram.....	108
Figure 9-5 BTIM Counter Mode Block Diagram.....	108
Figure 9-6 BTIM Trigger Start Mode Block Diagram.....	109
Figure 9-7 BTIM Trigger Start Mode Counting Diagram.....	109
Figure 9-8 BTIM Gating Mode Block Diagram.....	110
Figure 9-9 BTIM Gating Mode Counting Diagram.....	110
Figure 9-10 BTIM Timer Cascade Diagram.....	110
Figure 10-1 GTIM Block Diagram.....	119
Figure 10-2 GTIM Filter Unit Diagram.....	119
Figure 10-3 GTIM Counter Waveform.....	120
Figure 10-4 GTIM Timing Mode Block Diagram.....	121
Figure 10-5 GTIM Counter Mode Block Diagram.....	121
Figure 10-6 GTIM Trigger Start Mode Block Diagram.....	122
Figure 10-7 GTIM Trigger Start Mode Counting Diagram.....	122
Figure 10-8 GTIM Gating Mode Block Diagram.....	123
Figure 10-9 GTIM Gating Mode Counting Diagram.....	123
Figure 10-10 GTIM Capture Function Diagram.....	124
Figure 10-11 GTIM Compare Function Diagram.....	125
Figure 11-1 ATIM3 Block Diagram.....	137
Figure 11-2 Mode0 Count Timer Free Counting Block Diagram.....	138
Figure 11-3 Mode0 Count Timer Reload Counting Block Diagram.....	138
Figure 11-4 Mode0 Count Timer 16-Bit Reload Counting Waveform.....	139
Figure 11-5 Mode0 Count Timer 32-Bit Free Counting Waveform.....	139
Figure 11-6 Mode0 Count Timer Free Counting Timing Diagram.....	140
Figure 11-7 Mode0 Count Timer Reload Counting Timing Diagram (Prescaler=2).....	140
Figure 11-8 Mode1 Pulse Width Measurement PWC Function Block Diagram.....	141
Figure 11-9 Mode1: Pulse Width Measurement PWC High-Level Pulse Width Measurement.....	142
Figure 11-10 Mode1: Pulse Width Measurement PWC Falling Edge to Falling Edge Period Measurement.....	142

Figure 11-11	Mode1: Pulse Width Measurement PWC Rising Edge to Rising Edge Period Measurement.....	142
Figure 11-12	Mode1: Pulse Width Measurement PWC Rising Edge to Rising Edge Period Measurement Single Mode.....	143
Figure 11-13	Mode 2/3 Compare Capture Mode Counter Block Diagram.....	143
Figure 11-14	Mode 2 Compare Capture Mode Up-Counting Without Prescaler.....	144
Figure 11-15	Mode 2 Compare Capture Mode Up-Counting With Prescaler.....	144
Figure 11-16	Mode 2 Compare Capture Mode Down-Counting Without Prescaler.....	145
Figure 11-17	Mode 2 Compare Capture Mode Down-Counting With Prescaler.....	145
Figure 11-18	Mode2 Edge-Aligned Timer Waveform (M23CR.DIR=0).....	145
Figure 11-19	Mode2 Edge-Aligned Timer Waveform (M23CR.DIR=1).....	145
Figure 11-20	Mode3 Compare Capture Mode Up/Down-Counting with Prescaler.....	146
Figure 11-21	Mode3 Center-Aligned Counter Waveform.....	146
Figure 11-22	Repetition Counter Generates Update Timing (RCR.OV=0, RCR.UD=0).....	147
Figure 11-23	Update Timing Generated by Repetition Counter when Overflow Is Masked (RCR.OV=1, RCR.UD=0).....	147
Figure 11-24	Update Timing Generated by Repetition Counter when Underflow Is Masked (RCR.OV=0, RCR.UD=1).....	147
Figure 11-25	Mode 3 Triangular Wave Mode with Cache Enable.....	148
Figure 11-26	Mode 3 Triangular Wave Mode with Cache Invalid.....	148
Figure 11-27	Mode 2 Sawtooth Wave Mode Up-Counting with Cache Enable.....	149
Figure 11-28	Mode 2 Sawtooth Wave Mode Up-Counting with Cache Invalid.....	149
Figure 11-29	Mode 2 Sawtooth Wave Mode Down-Counting with Cache Enable.....	149
Figure 11-30	Mode 2 Sawtooth Wave Mode Down-Counting with Cache Invalid.....	150
Figure 11-31	Mode2 Sawtooth Wave Down-Counting Mode with Compare Cache Enable.....	150
Figure 11-32	OCREF Output Block Diagram.....	151
Figure 11-33	Sawtooth Counting Single Point Comparison OCREF Output Waveform (OCMyx=0b111).....	152
Figure 11-34	Triangular Counting Single Point Comparison OCREF Output Waveform (OCMyx=0b111).....	152
Figure 11-35	Sawtooth Counting Dual Point Comparison OCREF Output Waveform (OCMyx=0b111).....	152
Figure 11-36	Triangular Counting Dual Point Comparison OCREF Output Waveform (OCMyx=0b111).....	153
Figure 11-37	Independent PWM Output Block Diagram.....	154
Figure 11-38	PWM Output Waveform When CCPy=0.....	154
Figure 11-39	PWM Output Waveform When CCPy=1.....	154
Figure 11-40	Complementary PWM Output Block Diagram.....	155
Figure 11-41	Complementary PWM Output Waveform.....	155
Figure 11-42	Complementary PWM Output Waveform (with Dead Time).....	156
Figure 11-43	Dead Time.....	156
Figure 11-44	Triangular Wave Mode Single Pulse Counting.....	157
Figure 11-45	Sawtooth Up-Counting Single Pulse Mode.....	157
Figure 11-46	Sawtooth Down-Counting Single Pulse Mode.....	158
Figure 11-47	Compare Interrupt Diagram.....	159
Figure 11-48	Capture Function Block Diagram.....	159
Figure 11-49	Capture Timing Diagram.....	159
Figure 11-50	ATIM3 CH0A Capture Input Port Selection.....	160
Figure 11-51	CH0B Capture Input Port Selection.....	160
Figure 11-52	Mode2/3 Slave Mode Diagram.....	161
Figure 11-53	Mode2/3 Slave Mode Gating Function.....	162
Figure 11-54	Mode2/3 Slave Mode Trigger and Reset Function.....	162
Figure 11-56	Quadrature Encoder Counting Mode3 Counting Diagram.....	164
Figure 11-57	Timer Triggers ADC.....	165
Figure 12-1	RCH Clock Calibration Operational Block Diagram.....	205
Figure 12-2	RCL Clock Calibration Operational Block Diagram.....	206
Figure 12-3	General Low Power Timer Scenario Operational Block Diagram.....	207
Figure 13-1	IWDG Function Block Diagram.....	220
Figure 14-1	WWDT Function Block Diagram.....	229
Figure 15-1	LPUART Function Block Diagram.....	236

Figure 15-2	Mode0 Data Transmitting.....	238
Figure 15-3	Mode0 Data Receiving.....	238
Figure 15-4	Mode1 Data Transmitting.....	239
Figure 15-5	Mode1 Data Receiving.....	239
Figure 15-6	Mode2/Mode3 Data Transmitting.....	239
Figure 15-7	Mode2/Mode3 Data Receiving.....	240
Figure 15-8	LPUART Hardware Flow Control.....	244
Figure 15-9	RTS Hardware Flow Control Signal.....	244
Figure 15-10	CTS Hardware Flow Control Signal.....	245
Figure 16-1	SPI Block Diagram.....	255
Figure 16-2	SPI Communication Format Timing Diagram.....	256
Figure 16-3	Slave Select NSS Pin Configuration.....	257
Figure 16-4	SPI Full-duplex Communication.....	258
Figure 16-5	SPI Single-wire Half-duplex Communication.....	258
Figure 16-6	SPI Simplex Transmit Communication Diagram.....	259
Figure 16-7	SPI Simplex Receive Communication Diagram.....	259
Figure 16-8	SPI Multi-machine System.....	260
Figure 16-9	Explanation Diagram of General Flags in SPI Master Mode.....	262
Figure 16-10	SPI Slave Mode General Flag Bit Description Diagram.....	263
Figure 17-1	I2C Transmission Protocol.....	275
Figure 17-2	START, Repeated START and STOP.....	276
Figure 17-3	Bit Transfer on I2C Bus.....	276
Figure 17-4	Acknowledgment Signal on I2C Bus.....	277
Figure 17-5	Arbitration on the I2C Bus.....	278
Figure 17-6	I2C Function Block Diagram.....	279
Figure 17-7	Data Synchronization Diagram of Master Sending Mode.....	281
Figure 17-8	I2C Master Sending State Diagram.....	281
Figure 17-9	Data Synchronization Diagram of Master Receiving Mode.....	282
Figure 17-10	I2C Master Receiving State Diagram.....	282
Figure 17-11	Slave Receiving Mode Data Synchronization Diagram.....	283
Figure 17-12	Slave Receiving State Diagram.....	283
Figure 17-13	Slave Sending Mode Data Synchronization Diagram.....	284
Figure 17-14	I2C Slave Sending State Diagram.....	284
Figure 17-15	I2C Broadcast Call State Diagram.....	285
Figure 18-1	ADC Block Diagram.....	298
Figure 18-2	ADC Conversion Timing Diagram.....	299
Figure 18-3	ADC External Trigger Source Diagram.....	300
Figure 19-1	LVD Block Diagram.....	310
Figure 19-2	LVD Filter Output.....	310
Figure 19-3	LVD Hysteresis Response.....	311
Figure 21-1	Debug Support Block Diagram.....	322

List of Tables

Table 2-1	Address Division Table.....	21
Table 3-2	Reset Type Response Capabilities in Different Operational Mode.....	23
Table 3-3	Peripheral Working Status Table in Each Mode.....	25
Table 4-2	System Controller (SYSCTRL) Register Offset.....	34
Table 5-1	Cortex-M0+ Processor Interrupt List.....	49
Table 5-4	Interrupt Controller (NVIC) Register Offset Address.....	59
Table 6-1	Port Function Configuration Table.....	68
Table 6-2	Port Digital Multiplexed Function.....	73
Table 6-3	General-Purpose I/O (GPIO) Register Offset.....	75
Table 7-1	Flash Capacity Division Table.....	92
Table 7-3	Flash Controller (FLASH) Register Offset.....	98
Table 8-1	RAM Address Mapping.....	104
Table 9-1	BTIM Register Offset.....	111
Table 10-2	GTIM Register Offset.....	126
Table 11-7	CH0A Capture Input Signal.....	160
Table 11-14	Mode0 Timer Register Offset.....	169
Table 11-15	Mode1 Pulse Width Measurement (PWM) Register Offset.....	169
Table 11-16	Mode2/3 Compare-Capture Mode Register Offset.....	169
Table 12-1	Clock Trim Module (CTRM) Register Offset.....	209
Table 13-1	Independent Watchdog (IWDT) Register Offset.....	223
Table 14-1	Window Watchdog (WWDT) Register Offset.....	231
Table 15-1	LPUART Mode0/1/2/3 Data Structure.....	237
Table 15-2	DR8 Data Meaning.....	237
Table 15-3	SCLK = 4MHz Baud Rate Calculation Table.....	240
Table 15-4	SCLK = 8MHz Baud Rate Calculation Table.....	241
Table 15-5	SCLK = 16MHz Baud Rate Calculation Table.....	241
Table 15-6	SCLK = 24MHz Baud Rate Calculation Table.....	242
Table 15-7	SCLK = 32MHz Baud Rate Calculation Table.....	242
Table 15-8	SCLK = 48MHz Baud Rate Calculation Table.....	242
Table 15-9	Low Power Universal Asynchronous Receiver/Transmitter (LPUART) Register Offset.....	247
Table 16-1	SPI Interrupt.....	263
Table 16-2	Serial Peripheral Interface (SPI) Register Offset.....	266
Table 17-1	I2C Clock Signal Baud Rate.....	279
Table 17-2	I2C Status Code Expression.....	285
Table 17-3	I2C Bus (I2C) Register Offset.....	290
Table 18-2	ADC Interrupt.....	300
Table 18-3	Analog-to-Digital Converter (ADC) Register Offset.....	302
Table 19-1	Low Voltage Detector (LVD) Register Offset.....	313
Table 21-3	Host Transmitted Data Packet Request.....	324
Table 21-4	Target Transmitted Acknowledgment Response.....	324
Table 21-5	Data Transfers Transmitted by Host or Target.....	324
Table 22-2	SysTick Timer Register Offset.....	330

1 Product Introduction

HC32F002 Series is a general-purpose 32-bit MCU with a wide operating voltage range. The chip integrates a 10-bit 1Msps high-speed SAR ADC, high-performance PWM timer, low-power UART, SPI, I2C, and other peripherals, featuring high integration, reliability, and low power consumption. The HC32F002 Series adopts Cortex-M0+ core with a maximum frequency of 48MHz and is combined with the mature Keil & IAR debugging and development software, supporting the development with C language and assembly language.

Typical Application

- Motor control, battery management
- Smart home, medical devices
- Security alarms, intelligent transportation
- Sensor modules, wireless modules, electronic shelf labels

About This Manual

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

2 Storage and Architectures

2.1 Overview

System consists of the following parts:

- 1 AHB bus Master:
 - ▶ Cortex-M0+
- 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.

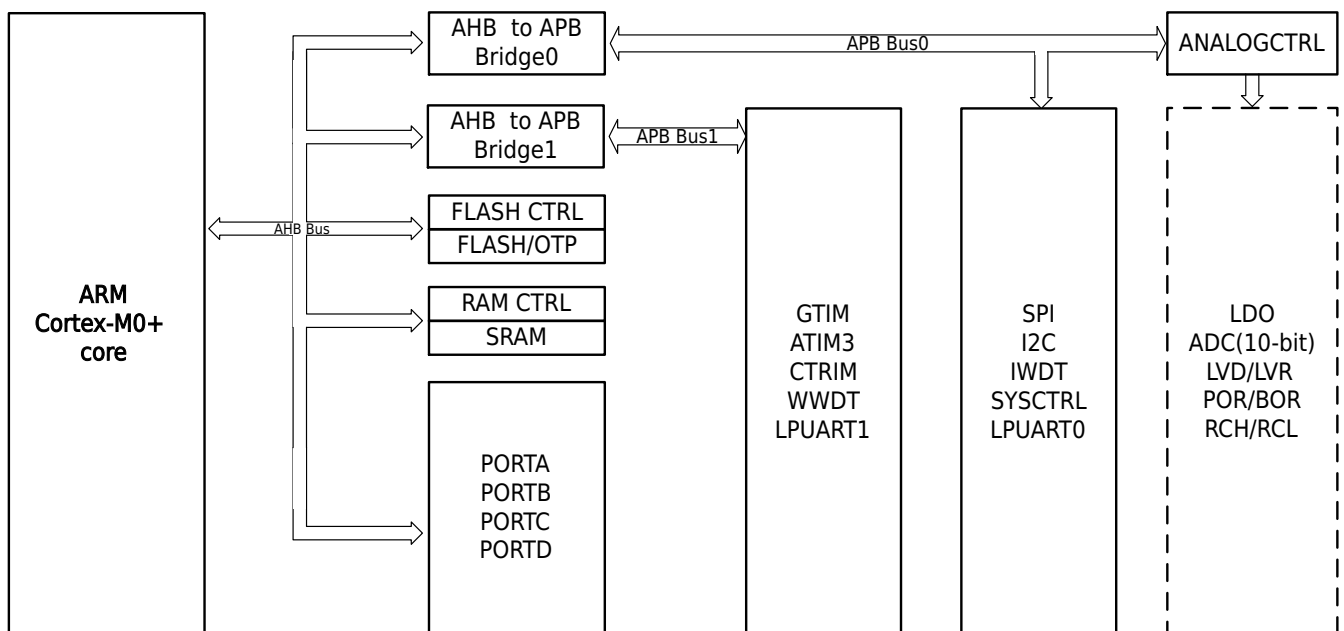
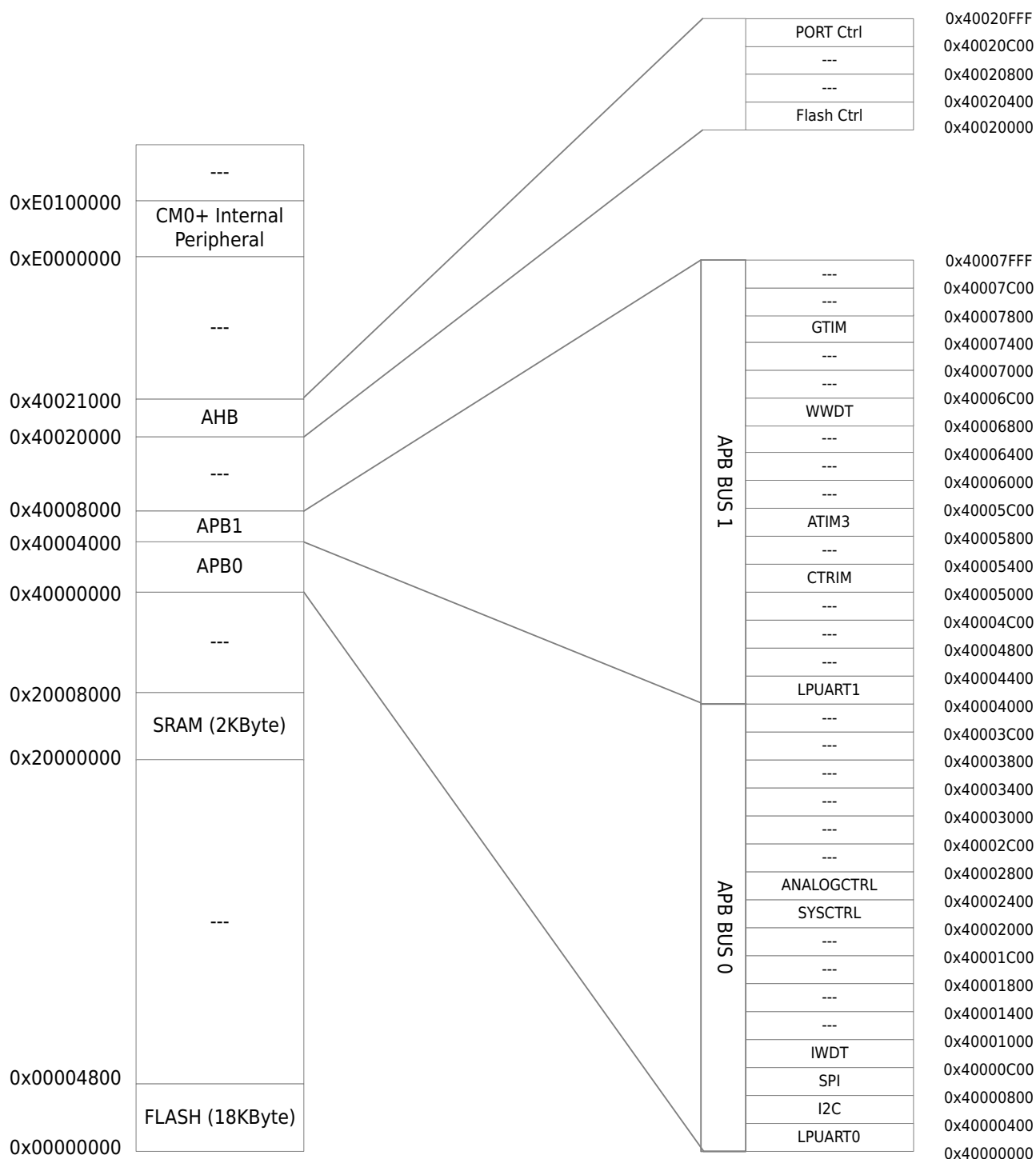


Figure 2-1 System Architecture Schematic Diagram

2.2 System Address Division

System of address region division:



2.3 Memory and Module Address Assignment

Table 2-1 Address Division Table

Categories	Boundary Address	Size	Memory Area
Memory	0x00000000-0x000047FF	18KByte	FLASH
	0x00100F00-0x00100F7F	128Byte	OTP
	0x20000000-0x200007FF	2KByte	SRAM
APB0 peripherals	0x40000000-0x400003FF	1KByte	LPUART0
	0x40000400-0x400007FF	1KByte	I2C
	0x40000800-0x40000BFF	1KByte	SPI
	0x40000F80-0x40000FFF	128Byte	IWDT
	0x40002000-0x400023FF	1KByte	SYSCTRL
	0x40002400-0x400027FF	1KByte	ANALOGCTRL
APB1 peripherals	0x40004000-0x400043FF	1KByte	LPUART1
	0x40005000-0x400053FF	1KByte	CTIM
	0x40005800-0x40005BFF	1KByte	ATIM3
	0x40006800-0x40006BFF	1KByte	WWDT
	0x40007400-0x400077FF	1KByte	GTIM
AHB peripherals	0x40020000-0x400203FF	1KByte	FLASH CTRL
	0x40020C00-0x40020FFF	1KByte	PORT CTRL

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.

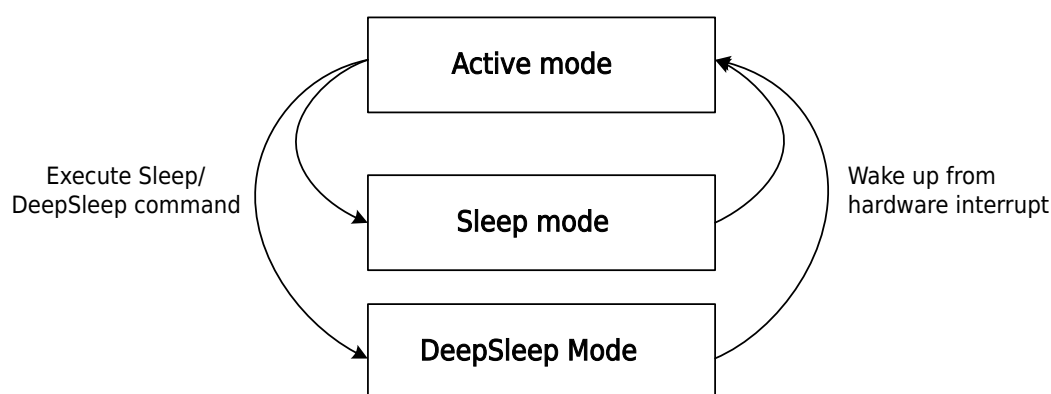


Figure 3-1 Working Mode State Transition Diagram

In various working modes, the interrupts that the CPU can respond type are as follows:

NVIC Interrupt Input	External Interrupt Source	Active Mode	Sleep mode	DeepSleep mode
Interrupt #0	GPIOA	✓	✓	✓
Interrupt #1	GPIOB	✓	✓	✓
Interrupt #2	GPIOC	✓	✓	✓
Interrupt #3	GPIOD	✓	✓	✓
Interrupt #4	-	-	-	-
Interrupt #5	ATIM3	✓	✓	×
Interrupt #6	-	-	-	-
Interrupt #7	-	-	-	-
Interrupt #8	LPUART0	✓	✓	✓
Interrupt #9	LPUART1	✓	✓	✓
Interrupt #10	SPI	✓	✓	×

NVIC Interrupt Input	External Interrupt Source	Active Mode	Sleep mode	DeepSleep mode
Interrupt #11	-	-	-	-
Interrupt #12	I2C	✓	✓	×
Interrupt #13	-	-	-	-
Interrupt #14	-	-	-	-
Interrupt #15	GTIM/BTIM3-5	✓	✓	×
Interrupt #16	-	-	-	-
Interrupt #17	-	-	-	-
Interrupt #18	-	-	-	-
Interrupt #19	-	-	-	-
Interrupt #20	-	-	-	-
Interrupt #21	-	-	-	-
Interrupt #22	IWDT	✓	✓	✓
	WWDT	✓	✓	×
Interrupt #23	-	-	-	-
Interrupt #24	ADC	✓	✓	×
Interrupt #25	-	-	-	-
Interrupt #26	-	-	-	-
Interrupt #27	-	-	-	-
Interrupt #28	LVD	✓	✓	✓
Interrupt #29	-	-	-	-
Interrupt #30	FLASH/RAM	✓	✓	×
Interrupt #31	CTRIM	✓	✓	✓

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

Table 3-2 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	✓	✓	✓

Reset Vector Index Reference	Reset Source	Active Mode	Sleep Mode	DeepSleep Mode
[4]	Hardware Reset Cortex-M0+ LOCKUP	✓	×	×
[5]	Software Reset Cortex-M0+ SYS-RESETREQ	✓	×	×
[6]	Hardware Reset WWDT	✓	✓	×

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 module register PERICKENx.

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_SCR.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 module register PERICKENx.
- 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 (RCL), all on-chip peripherals work normally. The power consumption in this mode is much lower than that in Sleep mode.

If SYSCTRL.CR1.WAKEUPCLK 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 DeepSleep 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 SYSTRL_CR0.CLKSRC.
- Turn off the clocks of unused on-chip low-power peripherals and modify SYSTRL module register PERICKENx.
- 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 3-3 Peripheral Working Status Table in Each Mode

Functional/Module	Active Mode	Sleep Mode	DeepSleep Mode
CPU	✓	×	×
WWDT	✓	✓	×
GTIM/BTIM3-5	✓	✓	×
ATIM3	✓	✓	×
SPI	✓	✓	×
I2C	✓	✓	×
ADC	✓	✓	×
GPIOA	✓	✓	✓
GPIOB	✓	✓	✓
GPIOC	✓	✓	✓
GPIOD	✓	✓	✓
IWDT	✓	✓	✓
LPUART0/LPUART1	✓	✓	✓
CTRM	✓	✓	✓
LVD	✓	✓	✓

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																																						

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	-	-

4 SYSTRL

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:44-48MHz)
- RCL - Internal low-speed RC clock (configurable to 38.4 kHz or 32.768 kHz)



Note

1. When switching the system clock source, please follow the procedure strictly. See the [System Clock Switching](#) section for details.

The HCLK clock signal can be output through the HCLK_OUT pin; the RCH/RCL/HCLK/PCLK clock signals can be output through the TCLK_OUT pin.

4.1.1 Clock Architecture Diagram

The clock architecture of this product is shown as follows:

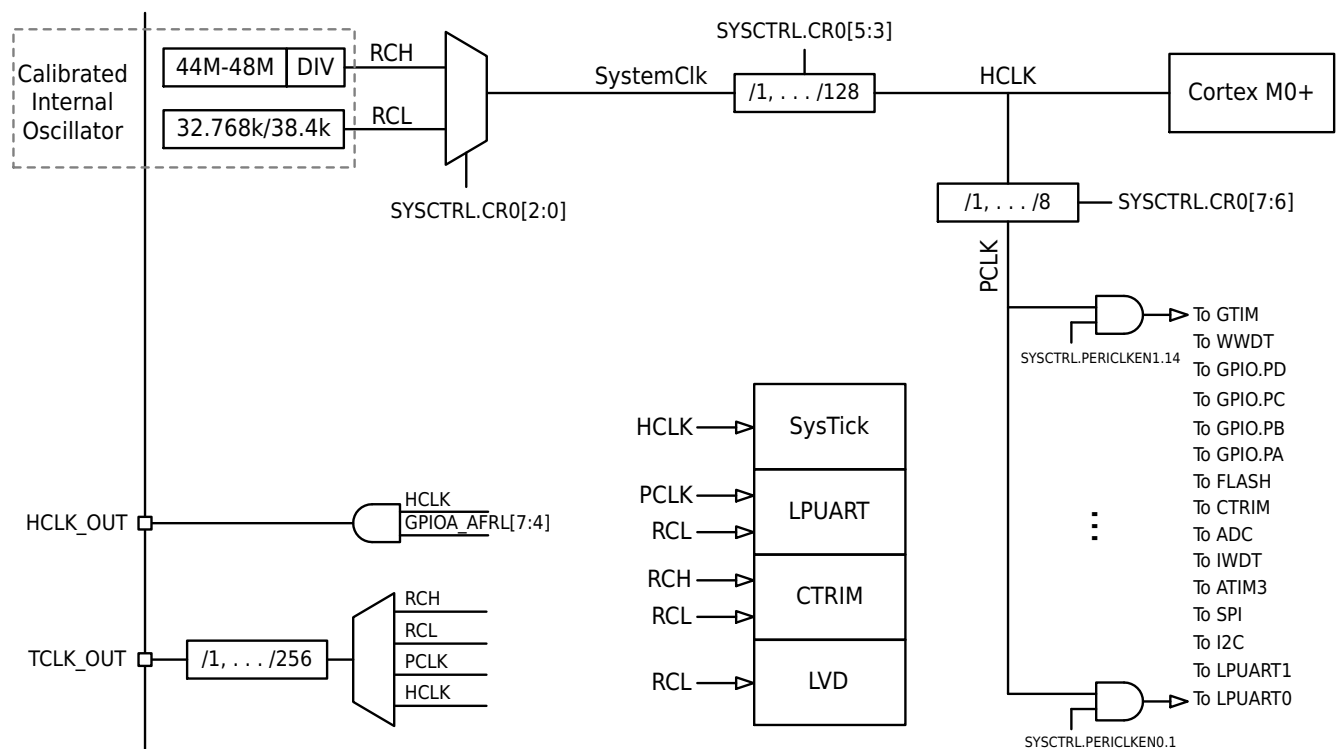


Figure 4-1 Clock Architecture Diagram

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.RCH.TRIM. Every time the register value increases by 1, the oscillation frequency of RCH will increase by about 0.2%. The safe operating range of the RCH oscillator frequency is 44-48MHz, if the RCH oscillator frequency is set above 48MHz, the chip may malfunction. The chip has two pre-adjusted frequency points 44.24MHz and 48MHz before leaving 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 0.5%, 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.RCH.DIV register, with an adjustable range from 1 to 16.

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.RCL.TRIM 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.

4.2 System Clock Switching

4.2.1 System Clock Switching Introduction

The system clock has 2 optional clock sources: RCHCLKD and RCL, which are controlled by the register SYSCTRL.CR0.CLKSRC. The two clock sources RCHCLKD and RCL can be switched between each other; during the switching process, ensure that HCLK does not exceed 48MHz. 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 $HCLK \leq 24MHz$, FLASH_WAIT should be set to 0;
- When $HCLK > 24MHz$, FLASH_WAIT should be set to 1.



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. Configure the oscillation parameters of the new clock source.

- Step2.** Enable the oscillator of the new clock source.
- Step3.** According to the higher frequency of the current clock source and the new clock source, configure the FLASH_WAIT register according to the procedures in the chapter FLASH Controller.
- Step4.** Wait for the new clock source to output a stable frequency.
- Step5.** Configure SYCTRL.CR0.CLKSRC to select the source of the system clock as the new clock source.
- Step6.** According to the frequency of the new clock source, configure the FLASH_WAIT register according to the procedures in the chapter FLASH Controller.
- Step7.** Turn off clock sources that are no longer in use.

4.2.2.2 RCH Switching Process between Different Oscillation Frequencies

Context

Configure SYCTRL.RCH.DIV to make RCH output different clock frequencies.

Procedure

- Step1.** According to the higher frequency of the current RCHCLKD and the target RCHCLKD, configure the FLASH_WAIT register according to the procedures in the chapter FLASH Controller.
- Step2.** Configure SYCTRL.RCH.DIV to select the frequency division ratio between RCH oscillation frequency and RCHCLKD frequency.
- Step3.** According to the current frequency of RCHCLKD, configure the FLASH_WAIT register according to the procedures in the chapter FLASH Controller.

4.2.3 Examples of Switching

4.2.3.1 Example of Switching from Other Clocks to RCL

Procedure

- Step1.** Configure SYCTRL.RCL.TRIM and SYCTRL.RCL.WAITCYCLE.
- Step2.** Write 0x5A5A and 0xA5A5 to SYCTRL.CR3 register in sequence to enable register rewriting.
- Step3.** Set SYCTRL.CR2.RCLEN to 1 to enable the RCL oscillation circuit.
- Step4.** Query and wait for SYCTRL.RCL.STABLE flag to become 1, and the RCL outputs a stable clock.
- Step5.** Write 0x5A5A and 0xA5A5 to SYCTRL.CR3 register in sequence to enable register rewriting.
- Step6.** Set SYCTRL.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 SYCTRL.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 SYCTRL.RCH.TRIM and SYCTRL.RCH.DIV.
- Step2.** Write 0x5A5A and 0xA5A5 to SYCTRL.CR3 register in sequence to enable register rewriting.
- Step3.** Set SYCTRL.CR2.RCHEN to 1 to enable the RCH oscillation circuit.
- Step4.** Query and wait for SYCTRL.RCH.STABLE flag to become 1, and the RCH outputs a stable clock.
- Step5.** According to the higher frequency of the current clock and the target RCHCLKD, configure the FLASH_WAIT register according to the procedures in the chapter FLASH Controller.
- Step6.** Write 0x5A5A and 0xA5A5 to SYCTRL.CR3 register in sequence to enable register rewriting.
- Step7.** Set SYCTRL.CR0.CLKSRC to 0, switch the system clock to RCHCLKD.

- Step8.** According to the frequency of RCHCLKD, set the FLASH_WAIT according to the procedures in the chapter FLASH Controller.
- Step9.** Write 0x5A5A and 0x5A55 to SYSCTRL.CR3 register in sequence to enable register rewriting.
- Step10.** Turn off the clock source that is no longer used.

4.2.4 Clock Switching Diagram

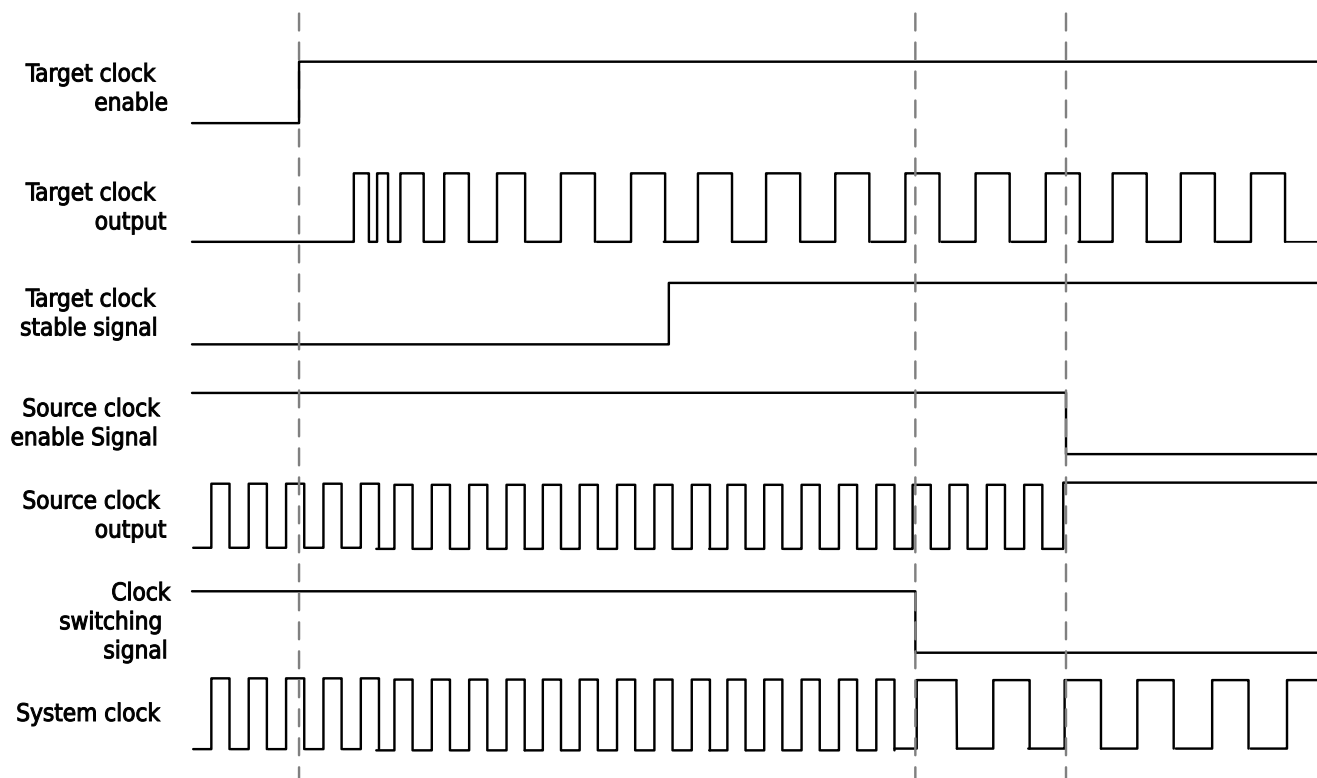


Figure 4-2 Clock Switching Diagram

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 module register PERICKENx 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
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
}

```

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 RESET_FLAG register to obtain the last reset source. Whenever the RESET_FLAG register is queried, the register should be cleared so that the register can normally indicate its reset source during the next reset.

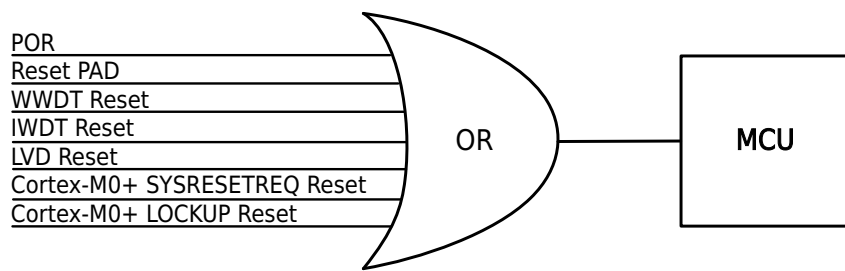


Figure 4-3 Reset Source Diagram

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 (typical value is 1.65V), POR5V signal is generated; when the VCC area is powered off, when the VCC voltage is lower than the BOR threshold voltage (typical value is 1.5V), POR5V signal is generated.
- When the VCORE area is powered on, when the VCORE voltage is lower than the POR threshold voltage, POR15V signal is generated; when the VCORE area is powered off, when the VCORE voltage is lower than the BOR threshold voltage, POR15V signal is generated.

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

4.5.2.2 External Reset Pin Reset

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

4.5.2.3 Watchdog Reset

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

4.5.2.4 LVD Low Voltage Reset

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

4.5.2.5 Cortex-M0+ SYSRESETREQ Reset

Cortex-M0+ software reset.

4.5.2.6 Cortex-M0+ LOCKUP Reset

When Cortex-M0+ encounters a serious exception, it will stop its own PC pointer at the current address and lock it, and reset the entire CORE area after a delay of several clock cycles.

4.6 Register Description

4.6.1 Register List

Base address: 0x40002000

Table 4-2 System Controller (SYCTRL) Register Offset

Offset	Register	Description	Width
0x000	CR0	System Control Register 0	32
0x004	CR1	System Control Register 1	32
0x008	CR2	System Control Register 2	32
0x00C	CR3	System Control Register 3	32
0x010	RCH	RCH Control Register	32
0x018	RCL	RCL Control Register	32
0x030	PERICKEN0	On-Chip Peripheral Clock Control Register 0	32
0x034	PERICKEN1	On-Chip Peripheral Clock Control Register 1	32
0x038	PERIRESET0	On-Chip Peripheral Reset Control Register 0	32
0x03C	PERIRESET1	On-Chip Peripheral Reset Control Register 1	32
0x040	RESET_FLAG	Reset Flag Register	32
0x044	DEBUGACTIVE	Peripheral Debug Function Enable Register	32

4.6.2 System Control Register 0 (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 0b0xx: Internal high-speed clock RCHCLKD 0b1xx: Internal low-speed clock RCL	RW

**Note**

Every time you rewrite the value of CR0/CR1/CR2, you need to write 0x5A5A and 0xA5A5 to CR3 in sequence. Such steps can effectively prevent misuse of CR0/CR1/CR2 registers.

4.6.3 System Control Register 1 (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	0x00000200																																			
Name	Reserved																							GTIMCFG		Reserved			RESIO		WAKEUPCLK		LOCKUPEN		SWDIO	
Access																								RW					RW		RW		RW		RW	

Bit	Name	Functional Description	Access
31:9	Reserved	-	-
8	GTIMCFG	GTIM/BTIM3-5 timer function configuration 0b0: GTIM enabled, BTIM3/4/5 disabled 0b1: BTIM3/4/5 enabled, GTIM disabled	RW
7:4	Reserved	-	-
3	RESIO	RESET pin function configuration 0b0: RESET pin configured as reset function only; IO input function disabled 0b1: RESET pin configured as IO input; reset function disabled	RW
2	WAKEUPCLK	System clock source configuration during DeepSleep Wake-up 0b0: Maintain original system clock source 0b1: Switch the system clock source to RCHCLKD, and keep the original system clock source enabled	RW
1	LOCKUPEN	Cortex-M0+ LockUp function configuration 0b0: Disable 0b1: Enable  Note Enabling this function causes the MCU to reset when the CPU reads an invalid instruction.	RW
0	SWDIO	SWD port function configuration 0b0: PC07 and PD01 are used as SWD ports, and the GPIO function is not available. 0b1: PC07 and PD01 are used as GPIO ports, and the SWD function is not available.	RW

 Note

Every time you rewrite the value of CR0/CR1/CR2, you need to write 0x5A5A and 0xA5A5 to CR3 in sequence. Such steps can effectively prevent misuse of CR0/CR1/CR2 registers.

4.6.4 System Control Register 2 (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																												RCLEN		Reserved	
Access																													RW			

Bit	Name	Functional Description	Access
31:6	Reserved	-	-
5	Reserved	Keep at 0 to disable modification.	-
4:3	Reserved	-	-
2	RCLEN	Internal low-speed clock RCL enable control ● 0b0: Disable ● 0b1: Enable  Note When the system enters DeepSleep, this low-speed clock retains its previous state.	RW
1	Reserved	-	-
0	RCHEN	Internal high-speed clock RCH enable control ● 0b0: Disable ● 0b1: Enable  Note When the system enters DeepSleep, this high-speed clock will be automatically turned off.	RW

**Note**

Every time you rewrite the value of CR0/CR1/CR2, you need to write 0x5A5A and 0xA5A5 to CR3 in sequence. Such steps can effectively prevent misuse of CR0/CR1/CR2 registers.

4.6.5 System Control Register 3 (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																	WO															

Bit	Name	Functional Description	Access
31:16	Reserved	-	-
15:0	CR3	Write protection register for CR0/1/2 Writing 0x5A5A followed by 0xA5A5 to CR3 disables the write protection for CR0/CR1/CR2; once a write operation is performed on CR0/CR1/CR2, the write protection automatically resets. Before each write operation to CR0/CR1/CR2, the sequence 0x5A5A followed by 0xA5A5 must be written to CR3.	WO

4.6.6 RCH Control Register (RCH)

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	0x00000B47																																
Name	Reserved																RO	STABLE	Reserved	DIV				TRIM									
Access																	RO			RW				RW									

Bit	Name	Functional Description	Access
31:16	Reserved	-	-
15	STABLE	RCH clock stable flag 0b0: Indicates RCH is not stable and cannot be used by internal circuits. 0b1: Indicates RCH is stable and can be used by internal circuits.	RO
14:13	Reserved	-	-
12:9	DIV	RCHCLKD output frequency and RCH oscillation frequency division ratio configuration 0b0000: RCHCLKD = RCH/2 0b0001: RCHCLKD = RCH/4 0b0010: RCHCLKD = RCH/6 0b0011: RCHCLKD = RCH/8 0b0100: RCHCLKD = RCH/10 0b0101: RCHCLKD = RCH/12 0b0110: RCHCLKD = RCH/14 0b0111: RCHCLKD = RCH/16 0b1000: RCHCLKD = RCH/1 - Others: Setting prohibited	RW
8:0	TRIM	Clock frequency adjustment Modify the value of this register to adjust the oscillation frequency of RCH. The register value increases by 1, then the oscillation frequency of RCH will increase by about 0.2%, and the total adjustment range is 44-48MHz. The calibration value of 44.24MHz and the 48MHz has been saved in Flash. Read the calibration value in Flash and write it into SYSCTRL.RCH.TRIM to obtain the accurate frequency. 44.24M calibration value address: 0x00100DF0-0x00100DF1 48.00M calibration value address: 0x00100DF2-0x00100DF3	RW

4.6.7 RCL Control Register (RCL)

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	0x0000033F																																
Name	Reserved																STABLE	Reserved				WAITCYCLE	TRIM										
Access																	RO					RW	RW										

Bit	Name	Functional Description	Access
31:16	Reserved	-	-
15	STABLE	RCL clock stable flag 0b0: Indicates RCL is not stable and cannot be used by internal circuits. 0b1: Indicates RCL is stable and can be used by internal circuits.	RO
14:12	Reserved	-	-
11:10	WAITCYCLE	Internal low-speed clock RCL stable time selection 0b00: 4 cycles 0b01: 16 cycles 0b10: 64 cycles 0b11: 256 cycles	RW
9:0	TRIM	Internal low-speed clock frequency adjustment The Flash contains two sets of frequency calibration values. Reading the calibration values from Flash and writing them into SYSCTRL.RCL.TRIM can obtain an accurate frequency. 32.768k calibration value address: 0x00100DF4-0x00100DF5 38.4k calibration value address: 0x00100DF6-0x00100DF7	RW

4.6.8 On-Chip Peripheral Clock Control Register 0 (PERICKEN0)

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	Reserved										CTRIM	Reserved				ADC	IWDT	Reserved				ATIM3	Reserved				SPI	Reserved	I2C	LPUART1	LPUART0	Reserved		
Access	RW											RW					RW	RW					RW					RW		RW	RW	RW	LPUART1	LPUART0	

Bit	Name	Functional Description	Access
31	FLASH	FLASH module configuration clock enable control 0b0: FLASH registers cannot be read or written 0b1: FLASH registers can be read and written	RW
30:22	Reserved	-	-
21	CTRIM	CTRIM module configuration clock enable control 0b0: Disable 0b1: Enable	RW
20:17	Reserved	-	-
16	ADC	ADC module configuration clock and operating clock enable control 0b0: Disable 0b1: Enable	RW
15	IWDT	IWDT module configuration clock enable control 0b0: Disable 0b1: Enable	RW
14:12	Reserved	-	-
11	ATIM3	ATIM3 module configuration clock and operating clock enable control 0b0: Disable 0b1: Enable	RW
10:7	Reserved	-	-
6	SPI	SPI module configuration clock and operating clock enable 0b0: Disable 0b1: Enable	RW
5	Reserved	-	-
4	I2C	I2C module configuration clock and operating clock enable control 0b0: Disable 0b1: Enable	RW
3	LPUART1	LPUART1 module configuration clock enable control 0b0: Disable 0b1: Enable	RW

Bit	Name	Functional Description	Access
2	LPUART0	LPUART0 module configuration clock enable control ● 0b0: Disable ● 0b1: Enable	RW
1:0	Reserved	-	-

4.6.9 On-Chip Peripheral Clock Control Register 1 (PERICKEN1)

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																	GTIM	Reserved			WWDT	Reserved							PD	PC	PB	PA
Access																		RW				RW								RW	RW	RW	RW

Bit	Name	Functional Description	Access
31:15	Reserved	-	-
14	GTIM	GTIM/BTIM3-5 module configuration clock and operating clock enable control 0b0: Disable 0b1: Enable	RW
13:11	Reserved	-	-
10	WWDT	WWDT module configuration clock and operating clock enable control 0b0: Disable 0b1: Enable	RW
9:4	Reserved	-	-
3	PD	PD port configuration clock and clock enable control 0b0: Disable 0b1: Enable	RW
2	PC	PC port configuration clock and clock enable control 0b0: Disable 0b1: Enable	RW
1	PB	PB port configuration clock and clock enable control 0b0: Disable 0b1: Enable	RW
0	PA	PA port configuration clock and clock enable control 0b0: Disable 0b1: Enable	RW

4.6.10 On-Chip Peripheral Reset Control Register 0 (PERIRESET0)

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	0x00210ADC																															
Name	Reserved										CTRIM	Reserved				ADC	Reserved				ATIM3	Reserved				SPI	Reserved	I2C	LPUART1	LPUART0	Reserved	
Access											RW					RW					RW					RW		RW	RW	RW	RW	

Bit	Name	Functional Description	Access
31:22	Reserved	-	-
21	CTRIM	CTRIM module reset control 0b0: The module is in reset state 0b1: The module works normally	RW
20:17	Reserved	-	-
16	ADC	ADC module reset control 0b0: The module is in reset state 0b1: The module works normally	RW
15:12	Reserved	-	-
11	ATIM3	ATIM3 module reset control 0b0: The module is in reset state 0b1: The module works normally	RW
10:7	Reserved	-	-
6	SPI	SPI module reset control 0b0: The module is in reset state 0b1: The module works normally	RW
5	Reserved	-	-
4	I2C	I2C module reset control 0b0: The module is in reset state 0b1: The module works normally	RW
3	LPUART1	LPUART1 module reset control 0b0: The module is in reset state 0b1: The module works normally	RW
2	LPUART0	LPUART0 module reset control 0b0: The module is in reset state 0b1: The module works normally	RW
1:0	Reserved	-	-

4.6.11 On-Chip Peripheral Reset Control Register 1 (PERIRESET1)

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	0x0000440F																																
Name	Reserved																	GTIM	Reserved			WWDT	Reserved							PD	PC	PB	PA
Access																		RW				RW								RW	RW	RW	RW

Bit	Name	Functional Description	Access
31:15	Reserved	-	-
14	GTIM	GTIM/BTIM3-5 module reset control 0b0: The module is in reset state 0b1: The module works normally	RW
13:11	Reserved	-	-
10	WWDT	WWDT module reset control 0b0: The module is in reset state 0b1: The module works normally	RW
9:4	Reserved	-	-
3	PD	PD port reset control 0b0: The module is in reset state 0b1: The module works normally	RW
2	PC	PC port reset control 0b0: The module is in reset state 0b1: The module works normally	RW
1	PB	PB port reset control 0b0: The module is in reset state 0b1: The module works normally	RW
0	PA	PA port reset control 0b0: The module is in reset state 0b1: The module works normally	RW

4.6.12 Reset Flag Register (RESET_FLAG)

**Note**

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

Offset	Bit Position																																	
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		VCORE		VCC				
Access																RW		RW		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	Cortex-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	Cortex-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

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	VCORE	Vcore domain reset flag Write 0 to clear flag, write 1 has no effect Read: ● 0b0: No reset occurred ● 0b1: Reset occurred	RW
0	VCC	VCC power domain reset flag Write 0 to clear flag, write 1 has no effect Read: ● 0b0: No reset occurred ● 0b1: Reset occurred	RW

4.6.13 Peripheral Debug Function Enable Register (DEBUGACTIVE)

**Note**

This register setting is effective only in SWD debug mode.

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	0x00014980																															
Name	Reserved																WWDT	Reserved	GTIM	Reserved	ATIM3	Reserved	IWDT	CTRIM	Reserved							
Access																	RW		RW		RW		RW	RW								

Bit	Name	Functional Description	Access
31:17	Reserved	-	-
16	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
15	Reserved	-	-
14	GTIM	During debugging, GTIM/BTIM3-5 counter function configuration 0b0: In the SWD debugging interface, the GTIM/BTIM3-BTIM5 counts normally 0b1: In the SWD debugging interface, suspend the GTIM/BTIM3-BTIM5 counting function	RW
13:12	Reserved	-	-
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
10:9	Reserved	-	-
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	CTIM	CTIM counting function configuration during debug 0b0: In the SWD debugging interface, the CTIM counts normally 0b1: In the SWD debugging interface, suspend the CTIM counting function	RW
6:0	Reserved	-	-

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:

- 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).

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

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:

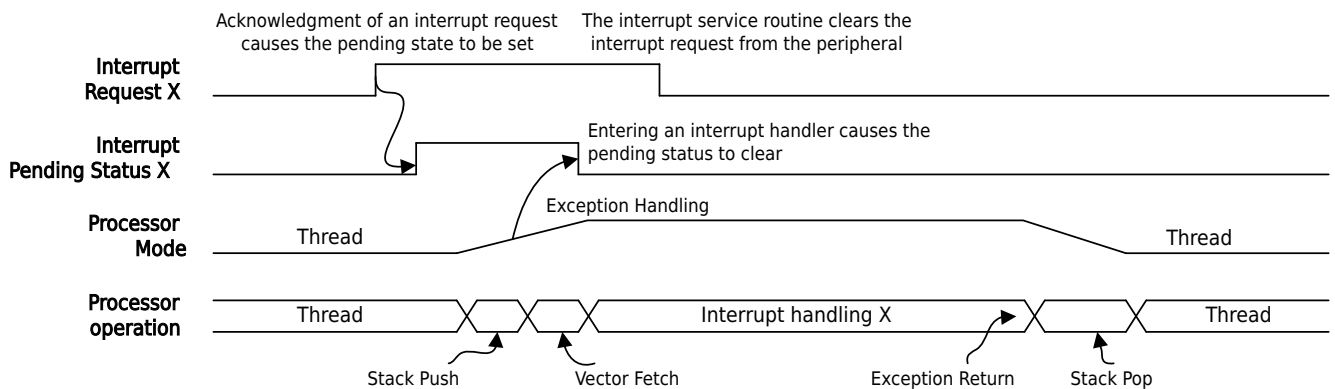


Figure 5-2 Interrupt Active and Pending States

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:

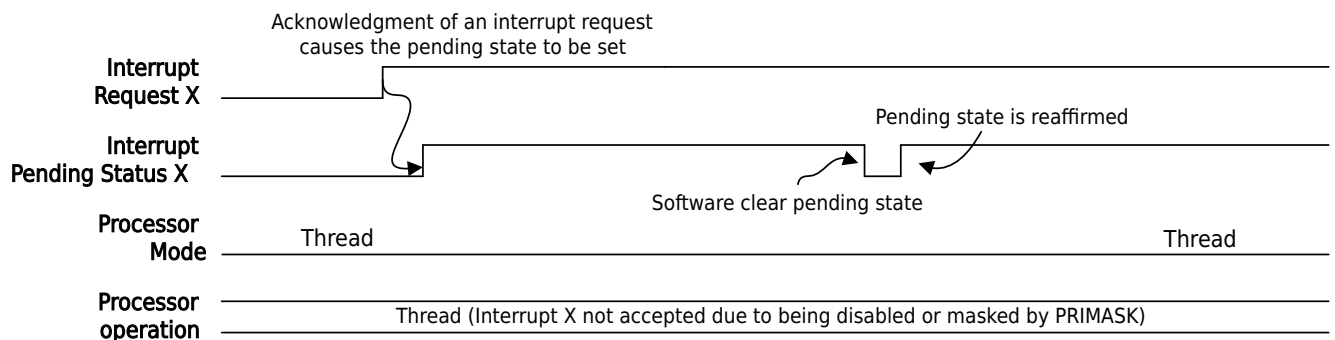


Figure 5-3 Interrupt Pending State is Cleared and Then Reasserted

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:

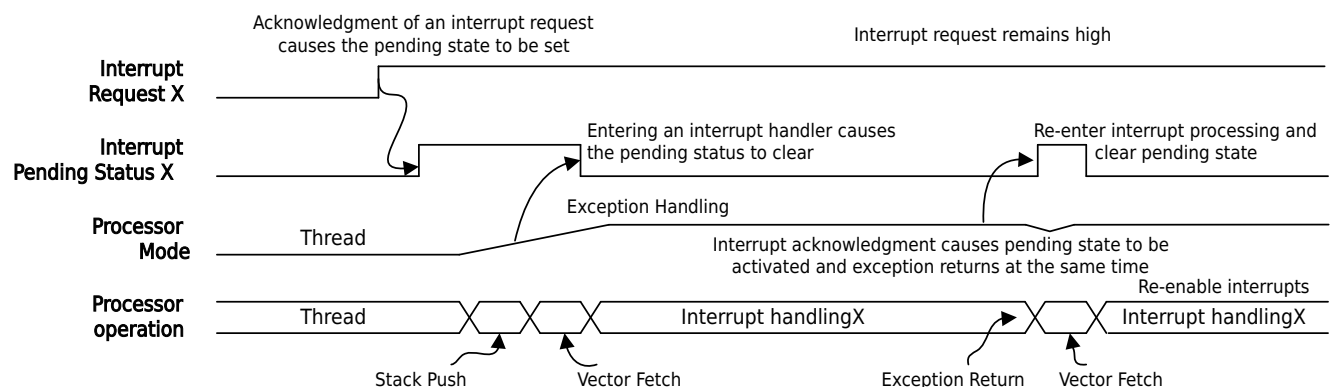


Figure 5-4 When the Interrupt Exits, If the Interrupt Request Remains High, It Will Cause the Interrupt Processing to Be Executed Again

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:

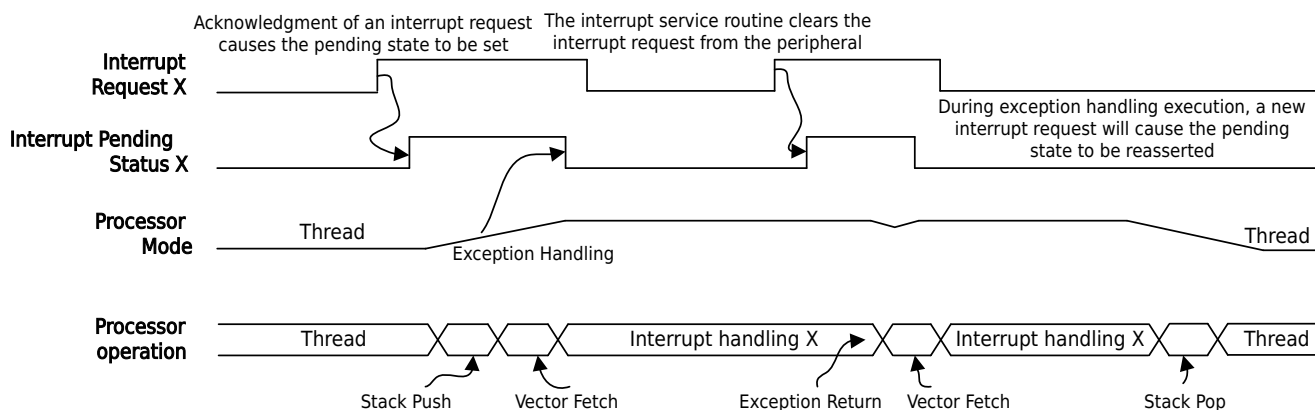


Figure 5-5 Interrupt Pending Interrupts Generated During Interrupt Processing Can Also Be Acknowledged

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 15 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	✓	✓	✓

NVIC Interrupt Input	External Interrupt Source	Active Mode	Sleep mode	DeepSleep mode
Interrupt #3	GPIOD	✓	✓	✓
Interrupt #4	-	-	-	-
Interrupt #5	ATIM3	✓	✓	×
Interrupt #6	-	-	-	-
Interrupt #7	-	-	-	-
Interrupt #8	LPUART0	✓	✓	✓
Interrupt #9	LPUART1	✓	✓	✓
Interrupt #10	SPI	✓	✓	×
Interrupt #11	-	-	-	-
Interrupt #12	I2C	✓	✓	×
Interrupt #13	-	-	-	-
Interrupt #14	-	-	-	-
Interrupt #15	GTIM/BTIM3-5	✓	✓	×
Interrupt #16	-	-	-	-
Interrupt #17	-	-	-	-
Interrupt #18	-	-	-	-
Interrupt #19	-	-	-	-
Interrupt #20	-	-	-	-
Interrupt #21	-	-	-	-
Interrupt #22	IWDT	✓	✓	✓
	WWDT	✓	✓	×
Interrupt #23	-	-	-	-
Interrupt #24	ADC	✓	✓	×
Interrupt #25	-	-	-	-
Interrupt #26	-	-	-	-
Interrupt #27	-	-	-	-
Interrupt #28	LVD	✓	✓	✓
Interrupt #29	-	-	-	-
Interrupt #30	FLASH/RAM	✓	✓	×
Interrupt #31	CTRM	✓	✓	✓

**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.

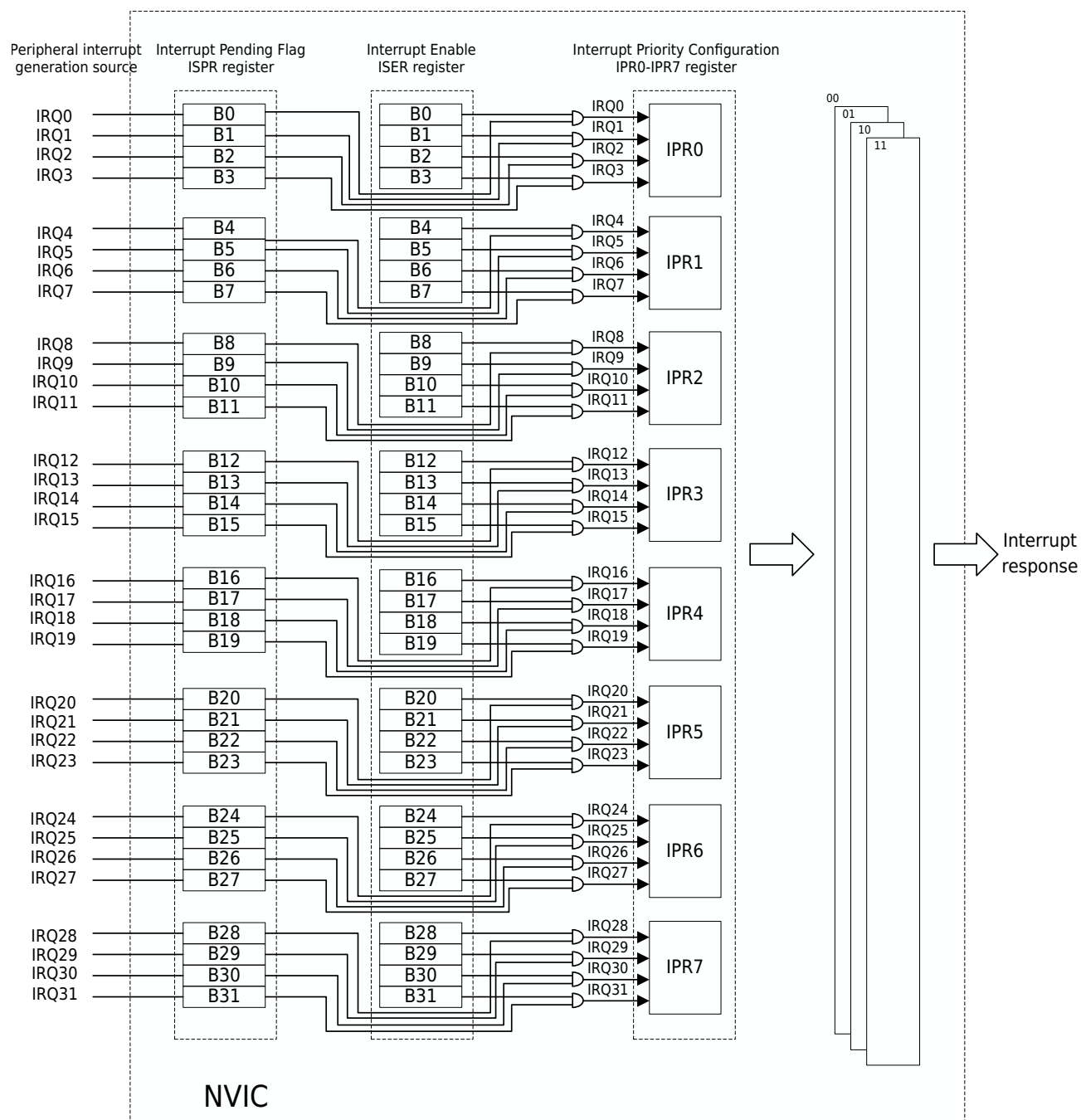


Figure 5-6 Interrupt Structure Diagram

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 register (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).

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
```

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

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

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

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	IPRO																															
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

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

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

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

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

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 is 5 bits, PB is 6 bits, PC is 5bits, and PD is 6 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, PA00 except
- Input mode: analog input, high-impedance input, pull-up 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
- Double drive: the PB04-05 and PC03-04 of TSSOP20 and QFN20 packages have double driving capabilities.

6.3 Functional Description

6.3.1 Port Circuit Block Diagram

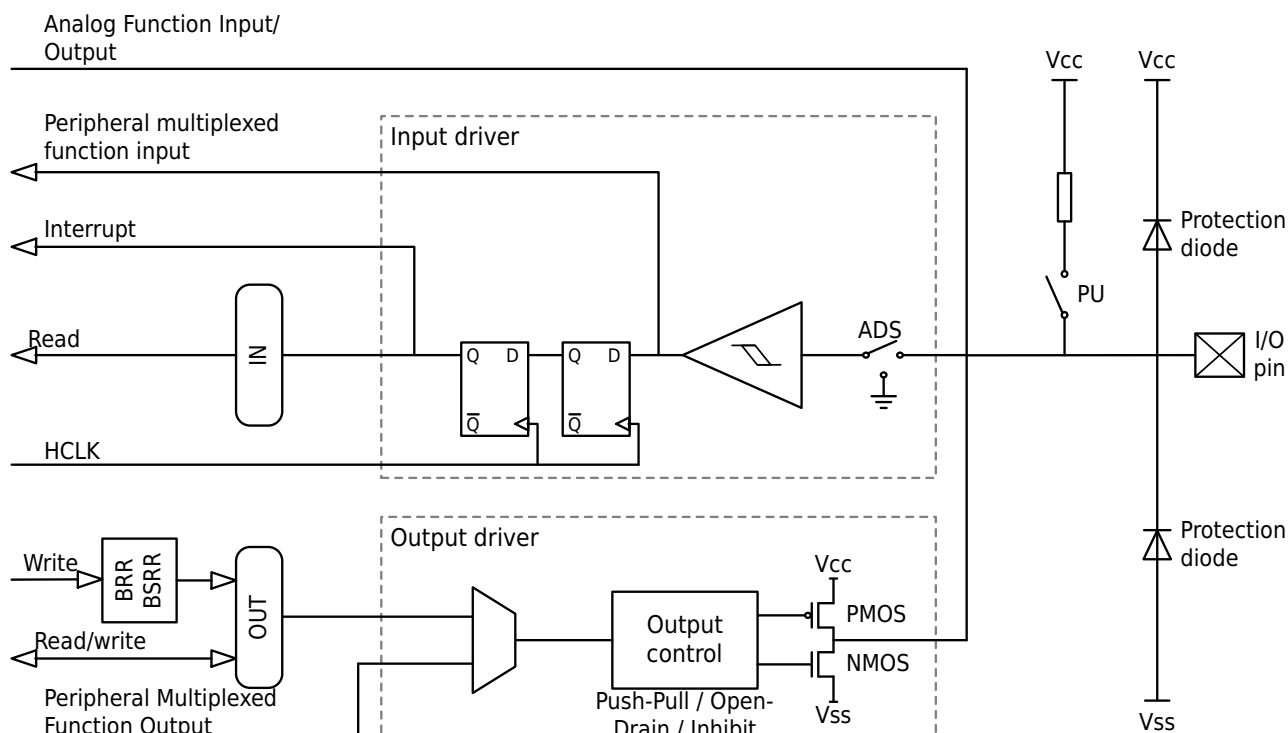


Figure 6-1 General Port Circuit Diagram

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	DIR	Open-Drain	OUT	BRR.B R	BSRR. BR	BSRR. BS	PU
Analog	0	1	-	-	-	-	-	-	-	-
High-impedance input	X	0	0	1	-	-	-	-	-	0
Pull-up input	1	0	0	1	-	-	-	-	-	1
High impedance output	X	0	0	0	1	1	0	0	0	0
High impedance output	X	0	0	0	1	-	0	0	1	0
High impedance output	X	0	0	0	1	-	0	1	1	0
Pull-up output	1	0	0	0	1	1	0	0	0	1
Pull-up output	1	0	0	0	1	-	0	0	1	1
Pull-up output	1	0	0	0	1	-	0	1	1	1

Pin Mode	IN	ADS	AFRL	DIR	Open-Drain	OUT	BRR.BR	BSRR.BR	BSRR.BS	PU
Output 0	0	0	0	0	1	0	0	0	0	-
Output 0	0	0	0	0	1	-	1	0	0	-
Output 0	0	0	0	0	1	-	0	1	0	-
Output 0	0	0	0	0	0	0	0	0	0	-
Output 0	0	0	0	0	0	-	1	0	0	-
Output 0	0	0	0	0	0	-	0	1	0	-
Output 1	0	0	0	0	0	1	0	0	0	-
Output 1	0	0	0	0	0	-	0	0	1	-
Output 1	0	0	0	0	0	-	0	1	1	-
Peripheral output	X	0	>0	0	0	-	-	-	-	-
Peripheral pull-up input	X	0	>0	1	0	-	-	-	-	1

**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 (PD01): Digital input pull-up
- SWDIO (PC07): Digital input pull-up
- PC06: Digital input has no pull-up

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

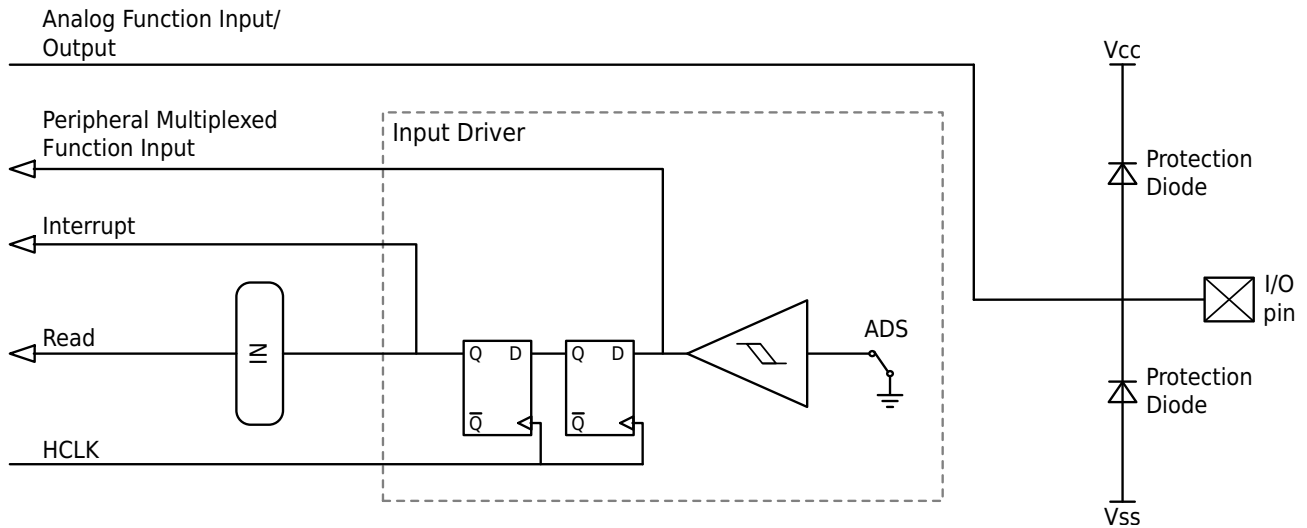


Figure 6-2 HiZ-Analog Configuration Diagram

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 register to 0. Under this configuration, the port has the following features:

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

Each GPIO has 4 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_PU register can enable pull-up of the port
- GPIO_OpenDrain register can configure the port output type as push-pull or open-drain

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, it is not recommended to use this method to modify value of GPIO_OUT register.

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 value of GPIO_OUT register.

6.3.5.2 Input Mode

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

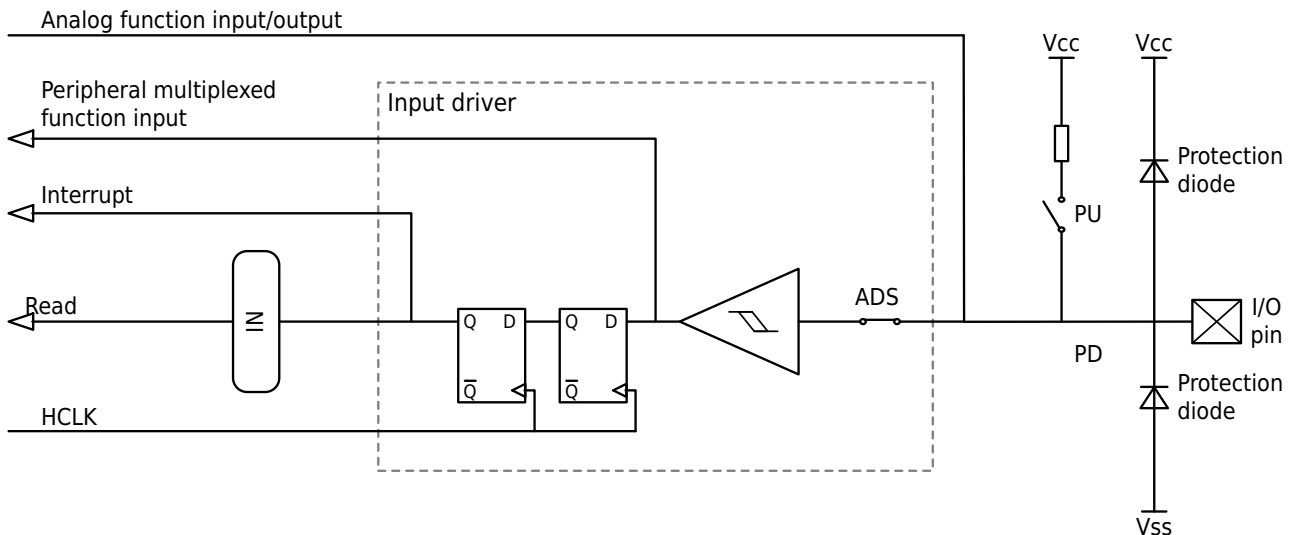


Figure 6-3 Input Floating/Pull-up/Pull-down Configuration Diagram

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:

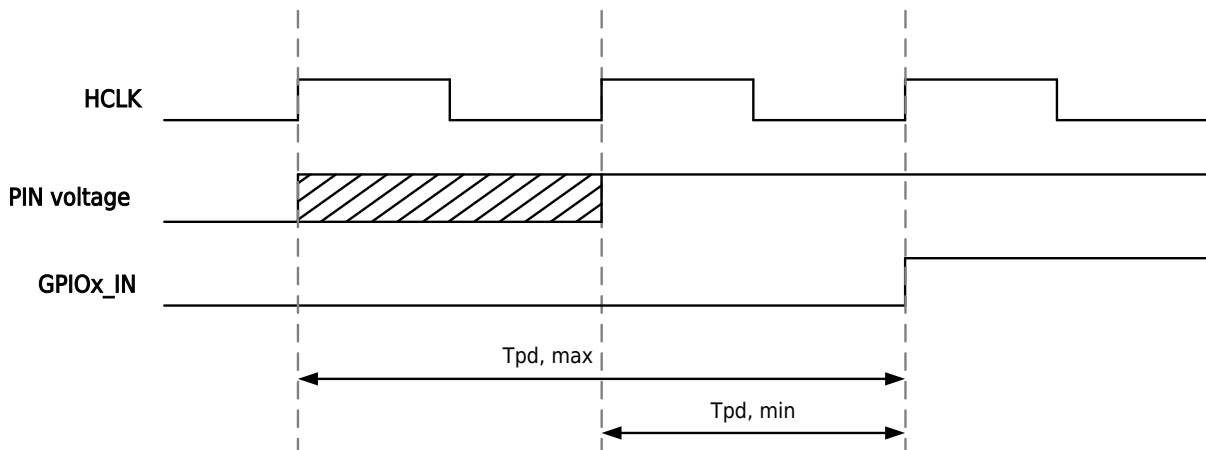


Figure 6-4 Read Port Pin Data Synchronization Diagram

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 T_{pd} is 1-2 system clocks.

6.3.5.3 Output Mode

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

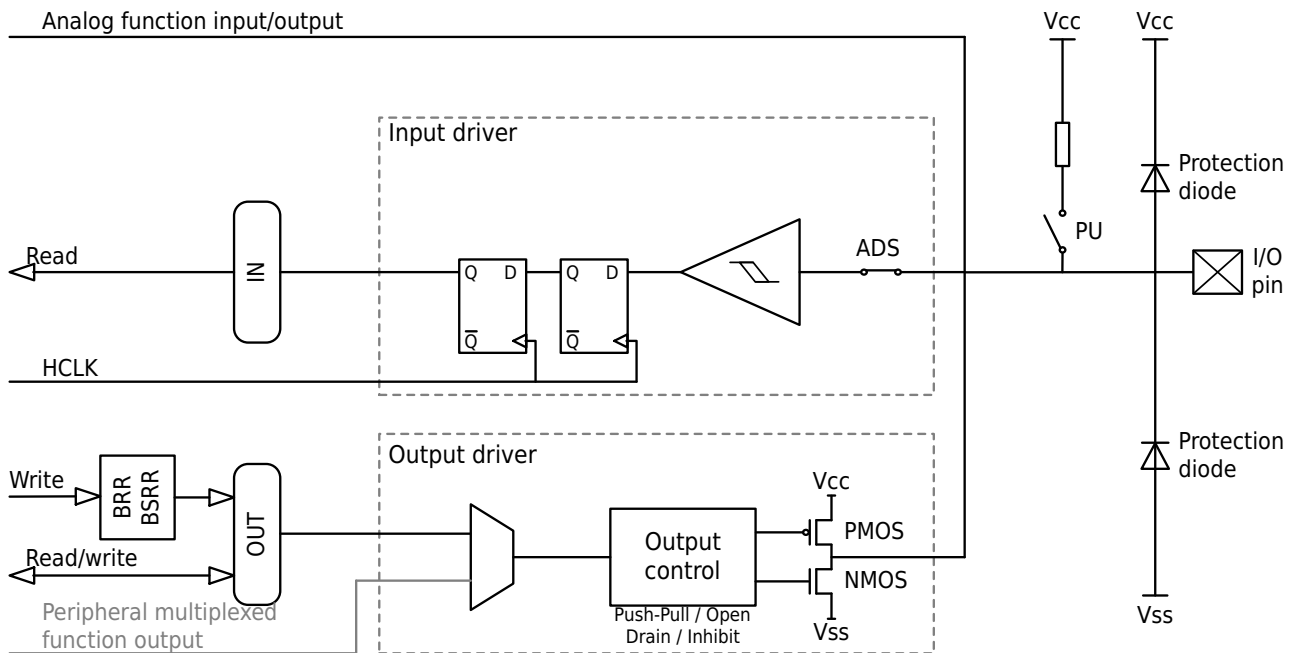


Figure 6-5 Output Configuration Diagram

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 1-6 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, LPUART, 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 to connect a specific on-chip peripheral to the port.

Under this configuration, the port has the following features:

- Input mode: pull-up 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 or push-pull can be selected; pull-up can be enabled; on-chip peripherals output signals to ports; port interrupts cannot be generated

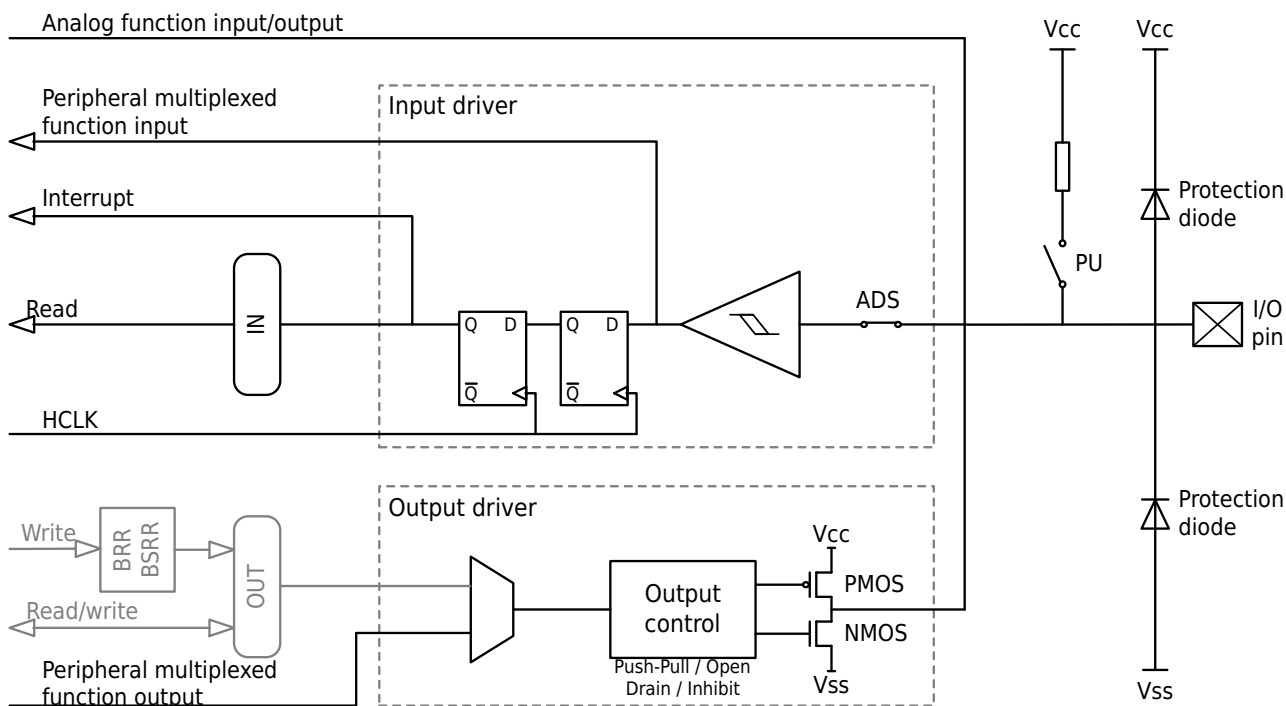


Figure 6-6 Multiplexed Function Configuration Diagram

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

Table 6-2 Port Digital Multiplexed Function

Pin Name	Digital Function				
	AF0	AF1	AF2	AF3	AF4
PA01	GPIO	LPUART0_RXD	I2C_SDA	GTIM_TOGP	HCLK_OUT
PA02	GPIO	LPUART0_TXD	I2C_SCL	GTIM_TOGN	LVD_OUT
PA03	GPIO	GTIM_CH2	SPI_NSS	CTRM_ETR CTRM_TOG	-
PA04	GPIO	GTIM_TOGP	LVD_OUT	SPI_NSS	-
PB00	GPIO	ATIM3_CH0B	GTIM_CH3	LPUART0_RTS	-
PB01	GPIO	ATIM3_CH1B	GTIM_CH2	LPUART0_CTS	-
PB02	GPIO	ATIM3_CH2B	GTIM_CH1	LPUART1_RTS	-
PB03	GPIO	ATIM3_ETR	GTIM_CH0	LPUART1_CTS	-
PB04	GPIO	I2C_SCL	ATIM3_CH2B	LPUART0_CTS	-
PB05	GPIO	I2C_SDA	ATIM3_BK	CTRM_ETR CTRM_TOG	-
PC03	GPIO	ATIM3_CH2A	ATIM3_CH0B	LPUART0_RTS	-
PC04	GPIO	ATIM3_CH0B	ATIM3_CH1B	TCLK_OUT	-
PC05	GPIO	SPI_SCK	GTIM_CH3	ATIM3_ETR	-
PC06	GPIO	SPI_MOSI	ATIM3_CH0A	LPUART1_RTS	-
PC07	GPIO	SPI_MISO	ATIM3_CH1A	LPUART0_RXD	-
PD01	GPIO	GTIM_ETR	ATIM3_BK	LPUART0_TXD	-
PD02	GPIO	GTIM_CH2	ATIM3_CH2A	LPUART1_CTS	-
PD03	GPIO	GTIM_CH1	ATIM3_CH1A	LPUART1_RXD	-
PD04	GPIO	GTIM_CH0	LPUART1_TXD	SPI_SCK	-
PD05	GPIO	LPUART1_TXD	ATIM3_CH0A	SPI_MISO	-
PD06	GPIO	LPUART1_RXD	GTIM_ETR	SPI_MOSI	-

6.3.7 Port Clock Output

6.3.7.1 Output HCLK

Configure the GPIO register to output HCLK. The following example outputs HCLK from PA01:

- Step1.** Set GPIOA_ADS.PIN1 to 0, and configure PA01 as digital port.
- Step2.** Set GPIOA_DIR.PIN1 to 0, and configure PA01 as output mode.

Step3. Set GPIOA_AFRL.AFSEL1 to 4, and configure the multiplexing function of PA01 as HCLK.

6.3.7.2 Output TCLK

Configure the GPIO register to output RCH/RCL/PCLK/HCLK, the example below outputs RCH/8 from PC04:

Step1. Set GPIOA_ADS.PIN4 to 0, and configure PC04 as digital port.

Step2. Set GPIOA_DIR.PIN4 to 0, and configure PC04 as output mode.

Step3. Set GPIOA_AFRL.AFSEL4 to 3, and configure the multiplexing function of PC04 as TCLK.

Step4. Set GPIOC_CR1.TCLKSRC to 0, select TCLK clock source as RCH.

Step5. Set GPIOC_CR1.TCLKDIV to 1, output the 8 frequency division of TCLK

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_RISEIE, GPIO_FALLIE, GPIO_HIGHIE, GPIO_LOWIE. When an interrupt is triggered, it can be 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:

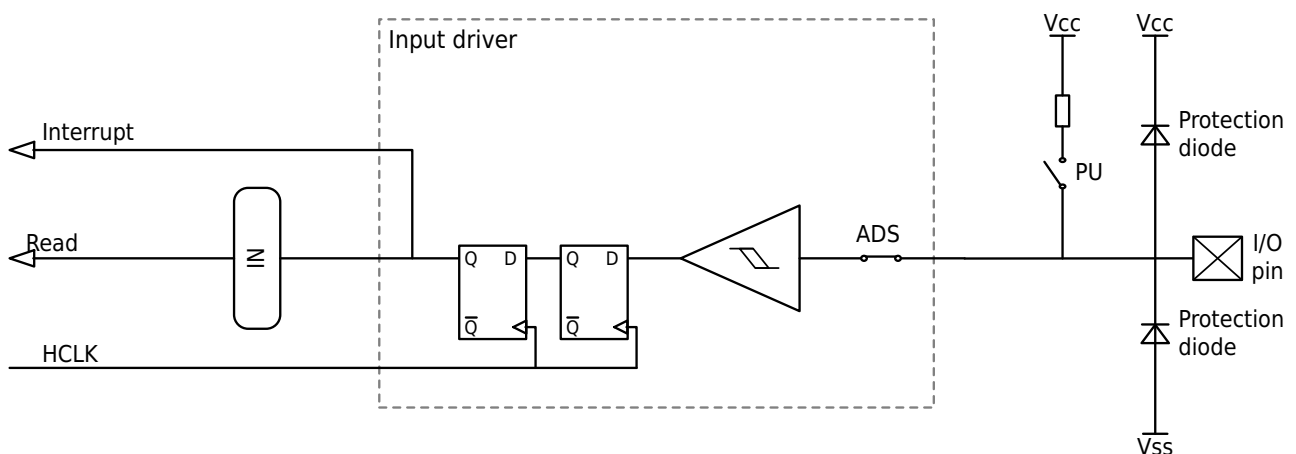


Figure 6-7 Port External Interrupt Diagram

6.4 Register Description

6.4.1 Register List

- GPIOA base address: 0x40020C00
- GPIOB base address: 0x40020D00
- GPIOC base address: 0x40020E00
- GPIOD base address: 0x40020F00



Note

GPIOA features 5 valid bits (PA00-04); GPIOB has 6 valid bits (PB00-05); GPIOC supports 5 valid bits (PC03-07); and GPIOD provides 6 valid bits (PD01-06). Write operations to non-existent bits have no effect, and reads always return 0.

Among them, two auxiliary function registers (CR1, CR4) are shared registers. GPIOA/GPIOB/GPIOC/GPIOD can all be used and are valid for all four ports. That is, 0x40020CA4, 0x40020DA4, 0x40020EA4, and 0x40020FA4 are the same register, and 0x40020CB0, 0x40020DB0, 0x40020EB0, and 0x40020FB0 are the same register.

Table 6-3 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_OpenDrain	Port Output Type Register	32
0x18	GPIO_PU	Port Pull-up 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
0x38	GPIO_HIGHIE	Port High Level Interrupt Enable Configuration Register	32
0x3C	GPIO_LOWIE	Port Low Level Interrupt Enable Configuration Register	32
0x40	GPIO_RISEIE	Port Rising Edge Interrupt Enable Configuration Register	32
0x44	GPIO_FALLIE	Port Falling Edge Interrupt Enable Configuration Register	32
0x48	GPIO_IFR	Port Interrupt Status Register	32
0x4C	GPIO_ICR	Port Interrupt Clear Register	32
0xA4	GPIO_CR1	Port Auxiliary Function Configuration Register 1	32

Offset	Register	Description	Width
0xB0	GPIO_CR4	Port Auxiliary Function Configuration Register 4	32

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	0x0000001E(PA); 0x0000003F(PB); 0x00000038(PC); 0x0000007C(PD)																															
Name	Reserved																								PIN							
Access																																

Bit	Name	Functional Description	Access
31:8	Reserved	-	-
7:0	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, the valid bit of port PA is [4:1], with PA[0] exclusively reserved as a digital port; the valid bit of port PB is [5:0]; the valid bit of port PC is [7:3]; the valid bit of port PD is [6:1].	RW

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	0x0000001F(PA); 0x0000003F(PB); 0x000000F8(PC); 0x0000007E(PD)																															
Name	Reserved																								PIN							
Access																																

Bit	Name	Functional Description	Access
31:8	Reserved	-	-
7:0	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, the valid bit for port PA is [4:1], with PA[0] only input; the valid bit for port PB is [5:0]; the valid bit for port PC is [7:3]; and the valid bit for port PD is [6:1].	RW

6.4.4 Port Output Type Register (GPIO_OpenDrain)

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																																

Bit	Name	Functional Description	Access
31:8	Reserved	-	-
7:0	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, the valid bit for port PA is [4:1]; the valid bit for port PB is [5:0]; the valid bit for port PC is [7:3]; the valid bit for port PD is [6:1].	RW

6.4.5 Port Pull-Up Register (GPIO_PU)

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	0x00000001(PA); 0x00000000(PB); 0x00000000(PC); 0x00000000(PD)																															
Name	Reserved																								PU							
Access																									RW							

Bit	Name	Functional Description	Access
31:8	Reserved	-	-
7:0	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, the valid bit for port PA is [4:0], with PA[0] automatically pulled up when configured as the RESET PAD; the valid bit for port PB is [5:0]; the valid bit for port PC is [7:3]; the valid bit for port PD is [6:1].	RW

6.4.6 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																																

Bit	Name	Functional Description	Access
31:8	Reserved	-	-
7:0	PIN	Port input data  Note Each bit corresponds to one port, please refrain from configuring non-existent ports, the valid bit for port PA is [4:0]; the valid bit for port PB is [5:0]; the valid bit for port PC is [7:3]; the valid bit for port PD is [6:1].	RO

6.4.7 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	0x000000XX																															
Name	Reserved																								PIN							
Access																																

Bit	Name	Functional Description	Access
31:8	Reserved	-	-
7:0	PIN	Port output data  Note Each bit corresponds to one port, please refrain from configuring non-existent ports, the valid bit for port PA is [4:1]; the valid bit for port PB is [5:0]; the valid bit for port PC is [7:3]; the valid bit for port PD is [6:1].	RW

6.4.8 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
31:8	Reserved	-	-
7:0	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, the valid bit for port PA is [4:1]; the valid bit for port PB is [5:0]; the valid bit for port PC is [7:3]; the valid bit for port PD is [6:1].	WO

6.4.9 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	Reserved								BR								Reserved								BS							
Access									WO																WO							

Bit	Name	Functional Description	Access
31:24	Reserved	-	-
23:16	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, the valid bit for port PA is [20:17]; the valid bit for port PB is [21:16]; the valid bit for port PC is [23:19]; the valid bit for port PD is [22:17].	WO
15:8	Reserved	-	-
7:0	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, the valid bit for port PA is [4:1]; the valid bit for port PB is [5:0]; the valid bit for port PC is [7:3]; the valid bit for port PD is [6:1].	WO

**Note**

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

6.4.10 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 ● 0b0101-0b1111: Setting prohibited  Note The specific functions of AF1-AF4 are detailed in "Port Multiplexing Functionality". The valid bit for port PA is [19:4], the valid bit for port PB is [23:0], the valid bit for port PC is [31:12], the valid bit for port PD is [27:4].	RW

6.4.11 Port High Level Interrupt Enable Configuration Register (GPIO_HIGHIE)

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																																

Bit	Name	Functional Description	Access
31:8	Reserved	-	-
7:0	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, the valid bit for port PA is [4:0]; the valid bit for port PB is [5:0]; the valid bit for port PC is [7:3]; the valid bit for port PD is [6:1].	RW

6.4.12 Port Low Level Interrupt Enable Configuration Register (GPIO_LOWIE)

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																																

Bit	Name	Functional Description	Access
31:8	Reserved	-	-
7:0	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, the valid bit for port PA is [4:0]; the valid bit for port PB is [5:0]; the valid bit for port PC is [7:3]; the valid bit for port PD is [6:1].	RW

6.4.13 Port Rising Edge Interrupt Enable Configuration Register (GPIO_RISEIE)

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																																

Bit	Name	Functional Description	Access
31:8	Reserved	-	-
7:0	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, the valid bit for port PA is [4:0]; the valid bit for port PB is [5:0]; the valid bit for port PC is [7:3]; the valid bit for port PD is [6:1].	RW

6.4.14 Port Falling Edge Interrupt Enable Configuration Register (GPIO_FALLIE)

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																																

Bit	Name	Functional Description	Access
31:8	Reserved	-	-
7:0	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, the valid bit for port PA is [4:0]; the valid bit for port PB is [5:0]; the valid bit for port PC is [7:3]; the valid bit for port PD is [6:1].	RW

6.4.15 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																								RO PIN							
Access																																

Bit	Name	Functional Description	Access
31:8	Reserved	-	-
7:0	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 , the valid bit for port PA is [4:0]; the valid bit for port PB is [5:0]; the valid bit for port PC is [7:3]; the valid bit for port PD is [6:1].	RO

6.4.16 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	0x0000001F(PA); 0x0000003F(PB); 0x000000F8(PC); 0x0000007E(PD)																															
Name	Reserved																								PIN							
Access																									WO							

Bit	Name	Functional Description	Access
31:8	Reserved	-	-
7:0	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, the valid bit for port PA is [4:0]; the valid bit for port PB is [5:0]; the valid bit for port PC is [7:3]; the valid bit for port PD is [6:1].	WO

6.4.17 Port Auxiliary Function Configuration Register 1 (GPIO_CR1)

Offset	Bit Position																																
0xA4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reset	0x00000000																																
Name	Reserved																												TCLKDIV		Reserved	TCLKSRC	
Access																													RW			RW	

Bit	Name	Functional Description	Access
31:5	Reserved	-	-
4:3	TCLKDIV	TCLK_OUT output frequency dividing control ● 0b00: TCLK ● 0b01: TCLK/8 ● 0b10: TCLK/64 ● 0b11: TCLK/256	RW
2	Reserved	-	-
1:0	TCLKSRC	TCLK_OUT output signal selection ● 0b00: RCH ● 0b01: RCL ● 0b10: PCLK ● 0b11: HCLK	RW

6.4.18 Port Auxiliary Function Configuration Register 4 (GPIO_CR4)

Offset	Bit Position																															
0xB0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																				GTIMCH0		Reserved		ATIM3CH0B		Reserved					
Access																					RW				RW							

Bit	Name	Functional Description	Access
31:10	Reserved	-	-
9:8	GTIMCH0	Gtimer CH0 input source configuration Parameter selection details refer to GPIO_CR4[5:4].	RW
7:6	Reserved	-	-
5:4	ATIM3CH0B	Atimer3 CH0B input source configuration 0b00: Decided by GPIO_AFRL 0b01: LPUART0_RXD 0b10: LPUART1_RXD 0b11: LVD_OUT	RW
3:0	Reserved	-	-

7 FLASH

7.1 Introduction

This system includes a Flash memory with a capacity of 18KB, which is divided into 36 sectors, and the capacity of each sector is 512 bytes.

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

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

7.2 Functional Description

7.2.1 Capacity Division

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

Table 7-1 Flash Capacity Division Table

Address	Serial Number	...	Address	Serial Number
0x0E00-0x0FFF	Sector7	...	0x4600-0x47FF	Sector35
0x0C00-0x0DFF	Sector6	...	0x4400-0x45FF	Sector34
0x0A00-0x0BFF	Sector5	...	0x4200-0x43FF	Sector33
0x0800-0x09FF	Sector4	...	0x4000-0x41FF	Sector32
0x0600-0x07FF	Sector3	...	0x3E00-0x3FFF	Sector31
0x0400-0x05FF	Sector2	...	0x3C00-0x3DFF	Sector30
0x0200-0x03FF	Sector1	...	0x3A00-0x3BFF	Sector29
0x0000-0x01FF	Sector0	...	0x3800-0x39FF	Sector28

7.2.2 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 1, every two HCLKs can complete a Flash fetch operation.

7.2.3 Flash Security Protection

Sector Erase/Write Protection

The entire 18KB Flash memory is divided into 36 sectors, and every 4 sectors share an erase/write protection bit. When the page is protected, the erasing and writing operations on the sector are invalid and an alarm flag and interrupt signal are generated. When any sector in the Flash memory is protected, the whole-chip erasing of the Flash is invalid, and an alarm flag and an interrupt signal are generated.

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_SLOCK, 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_LOCKSTATE 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	ISP Interface			SWD Interface		
	Flash			Flash		
	Erase	Write	Reading	Erase	Write	Reading
Level0	Y	Y	Y	Y	Y	Y
Level1	Y	Y	N	Y	N	N
Level2	Y	Y	N	N	N	N
Level3	N	N	N	N	N	N

**Note**

1. Level1: Data can be downgraded to Level0 through ISP and SWD. After downgrading, the Flash content is all FF.
2. Level2: Data can be downgraded to Level0 through ISP. After downgrading, the Flash content is all FF.
3. Level3: The ISP and SWD interfaces are prohibited, that is, the chip can only be programmed once.

7.3 Operation Example

7.3.1 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_CR.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_CR.BUSY becomes 0).

Procedure

- Step1.** Write 0x5A5A and 0xA5A5 to the FLASH_BYPASS register in sequence to enable register rewriting.
- Step2.** Configure FLASH_CR.OP to 2, 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_SLOCK to 1 to remove the sector 's erasure protection.
- Step6.** Check whether the corresponding bit of FLASH_SLOCK 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 char *)0x00000200) = 0x00
```

- Step8.** Wait for FLASH_CR.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.

7.3.2 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_CR.BUSY becomes 0).

Procedure

- Step1.** Write 0x5A5A and 0xA5A5 to the FLASH_BYPASS register in sequence to enable register rewriting.
- Step2.** Configure FLASH_CR.OP to 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_SLOCK to 0xFFFFFFFF to remove the erase protection of all Sectors.
- Step6.** Check whether FLASH_SLOCK is 0xFFFFFFFF, if not 0xFFFFFFFF, jump to Step4.
- Step7.** Write any address in the Chip to be erased to trigger Chip erasure.

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

- Step8.** Wait for FLASH_CR.BUSY to become 0, and the chip erase operation is completed.
- Step9.** Write 0x5A5A and 0xA5A5 to the FLASH_BYPASS register in sequence to enable register rewriting.
- Step10.** Configure FLASH_CR.OP to 0, set the Flash operation mode to read.
- Step11.** Check if FLASH_CR.OP is 0, if it is not 0, jump to Step9, if it is 0, end the process.

7.3.3 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_CR.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_CR.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_SLOCK to 1 to remove the sector 's erasure protection.
- Step6.** Check whether the corresponding bit of FLASH_SLOCK is 1, if it is not 1, jump to Step4.
- Step7.** Perform Byte program on the target address to be written to trigger the program.

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

- Step8.** Wait for FLASH_CR.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.
- 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_SLOCK to 1 to remove the sector 's erasure protection.

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

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

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

Step8. Wait for FLASH_CR.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_SLOCK to 1 to remove the sector 's erasure protection.

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

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

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

Step8. Wait for FLASH_CR.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.

7.3.4 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)
```

7.4 Register Description

7.4.1 Register List

Base address: 0x40020000

Table 7-3 Flash Controller (FLASH) Register Offset

Offset	Register	Description	Width
0x20	FLASH_CR	Control Register	32
0x24	FLASH_IFR	Interrupt Flag Register	32
0x28	FLASH_ICLR	Interrupt Flag Clear Register	32
0x2C	FLASH_BYPASS	Sequential Register	32
0x30	FLASH_SLOCK	Erase Write Protection Register	32
0x50	FLASH_WAIT	Read Waiting Period Register	32
0x54	FLASH_LOCKSTATE	Read Protection Status Register	32

7.4.2 Control Register (FLASH_CR)

Offset	Bit Position																															
0x20	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000200																															
Name	Reserved																						LPMODE	Reserved	FLASHIE	PCIE	BUSY	Reserved	OP			
Access																							RW		RW	RW	RO		RW			

Bit	Name	Functional Description	Access
31:10	Reserved	-	-
9	LPMODE	FLASH Low power control 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 Please keep this bit is 1.	RW
8:7	Reserved	-	-
6	FLASHIE	Interrupt Enable Control for Protected FLASH Address Erase/Write Attempts 0b0: Disable 0b1: Enable	RW
5	PCIE	Interrupt Enable Control for Erase/Write Attempts to PC-located FLASH Page 0b0: Disable 0b1: Enable	RW
4	BUSY	FLASH erase/write Status Flag 0b0: erase/write has been completed 0b1: erase/write operation has not been completed	RO
3:2	Reserved	-	-
1:0	OP	FLASH Operate Mode Configuration 0b00: Read 0b01: Program 0b10: Sector erase 0b11: Chip erase  Note After erase/write operations, the OP field must be restored to 0.	RW

7.4.3 Interrupt Flag Register (FLASH_IFR)

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

Bit	Name	Functional Description	Access
31:2	Reserved	-	-
1	FLASH	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

7.4.4 Interrupt Flag Clear Register (FLASH_ICLR)

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

Bit	Name	Functional Description	Access
31:2	Reserved	-	-
1	FLASH	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

7.4.5 Sequential Register (FLASH_BYPASS)

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

Bit	Name	Functional Description	Access
31:16	Reserved	-	-
15:0	BYPASS	Bypass sequence bit Write 0x5A5A first to FLASH_BYPASS, and then write 0xA5A5 to start the program to FLASH module Register (specific registers described in "Flash Security Protection"), as long as the program is done to register, this protection bit automatically recovery to the protection state and needs to be rewritten to open the protection. Subsequent program to the FLASH module registers require repeating the previous procedures.	RW

7.4.6 Erase Write Protection Register (FLASH_SLOCK)

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																									SLOCK							
Access																																

Bit	Name	Functional Description	Access														
31:9	Reserved	-	-														
8:0	SLOCK	Sector Erase/Write Protection Bit	RW														
		● 0b0: Disable erase/write															
		● 0b1: Enable erase/write															
		<table><tr><th>Bit</th><th>Serial Number</th></tr><tr><td>0</td><td>Sector0-3</td></tr><tr><td>1</td><td>Sector4-7</td></tr><tr><td>2</td><td>Sector8-11</td></tr><tr><td>3</td><td>Sector12-15</td></tr><tr><td>...</td><td>...</td></tr><tr><td>8</td><td>Sector32-35</td></tr></table>		Bit	Serial Number	0	Sector0-3	1	Sector4-7	2	Sector8-11	3	Sector12-15	...	...	8	Sector32-35
		Bit		Serial Number													
		0		Sector0-3													
		1		Sector4-7													
		2		Sector8-11													
		3		Sector12-15													
...	...																
8	Sector32-35																

7.4.7 Read Waiting Period Register (FLASH_WAIT)

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																															WAIT
Access																																RW

Bit	Name	Functional Description	Access
31:2	Reserved	-	-
1:0	WAIT	FLASH instruction fetch wait state configuration 0b00: 1 wait state, applicable when HCLK ≤ 24 MHz 0b01: 2 wait states, applicable when 24 MHz < HCLK ≤ 48 MHz - Others: Setting prohibited	RW

7.4.8 Read Protection Status Register (FLASH_LOCKSTATE)

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	0x00000000																															
Name	Reserved																															STATE
Access																																RO

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

8 RAM

8.1 Introduction

This system includes an SRAM block with a capacity of 2KB, 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.

8.2 Functional Description

8.2.1 RAM Address Range

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

Table 8-1 RAM Address Mapping

Address Range	Size	Memory Type
0x20000000-0x200007FF	2KByte	SRAM

8.2.2 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.

9 BTIM

9.1 Introduction

BTIM is a group of timing counters with basic timing functions, each group contains three 16-bit timers with reload function. This product contains a set of basic timers: BTIM3/4/5. BTIM3/4/5 and GTIM functions cannot be used simultaneously. They are controlled by SYSCTRL.CR1.GTIMCFG; when the bit is set to 1, BTIM3/4/5 are enabled.

9.2 Features

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

9.3 Functional Description

9.3.1 Function Block Diagram

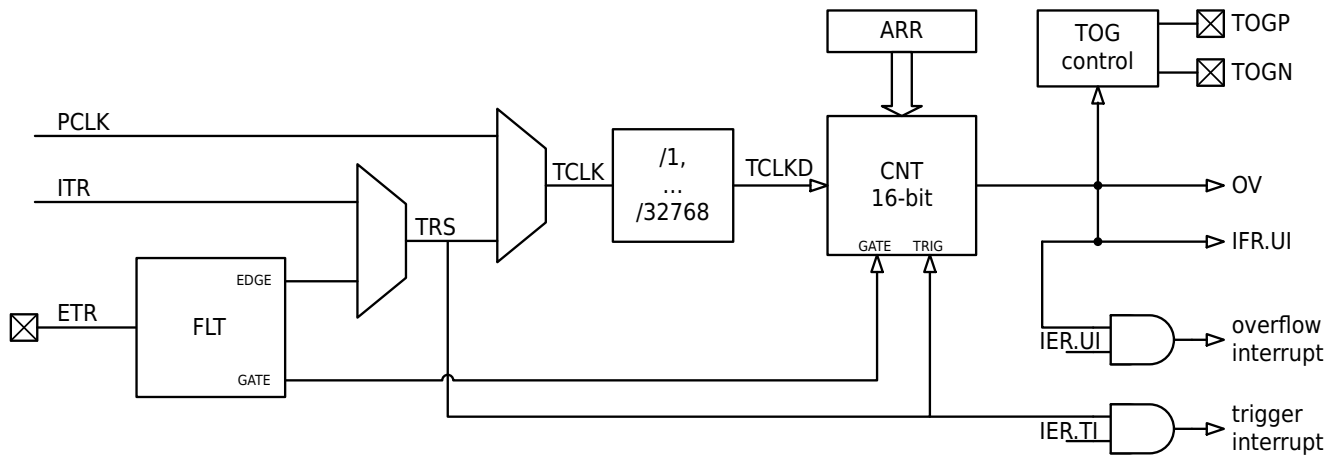


Figure 9-1 BTIM Block Diagram



Note

- BTIM3/4/5 share one ETR.
- For more information on the ITR signal sources, see the section "[Timer Cascade](#)".

9.3.2 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.

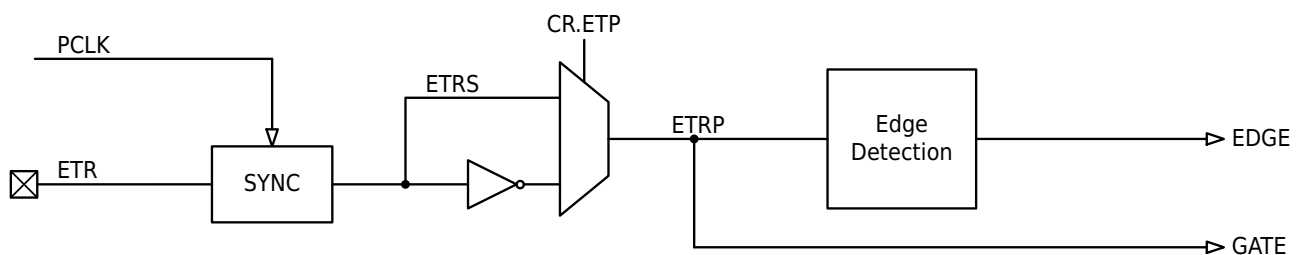


Figure 9-2 BTIM Filter Unit Diagram

9.3.3 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.

- 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 TOGP/TOGN output toggles once on each overflow. The TOGP/TOGN signal can be output to a pin via GPIO multiplexing function to drive external circuits.

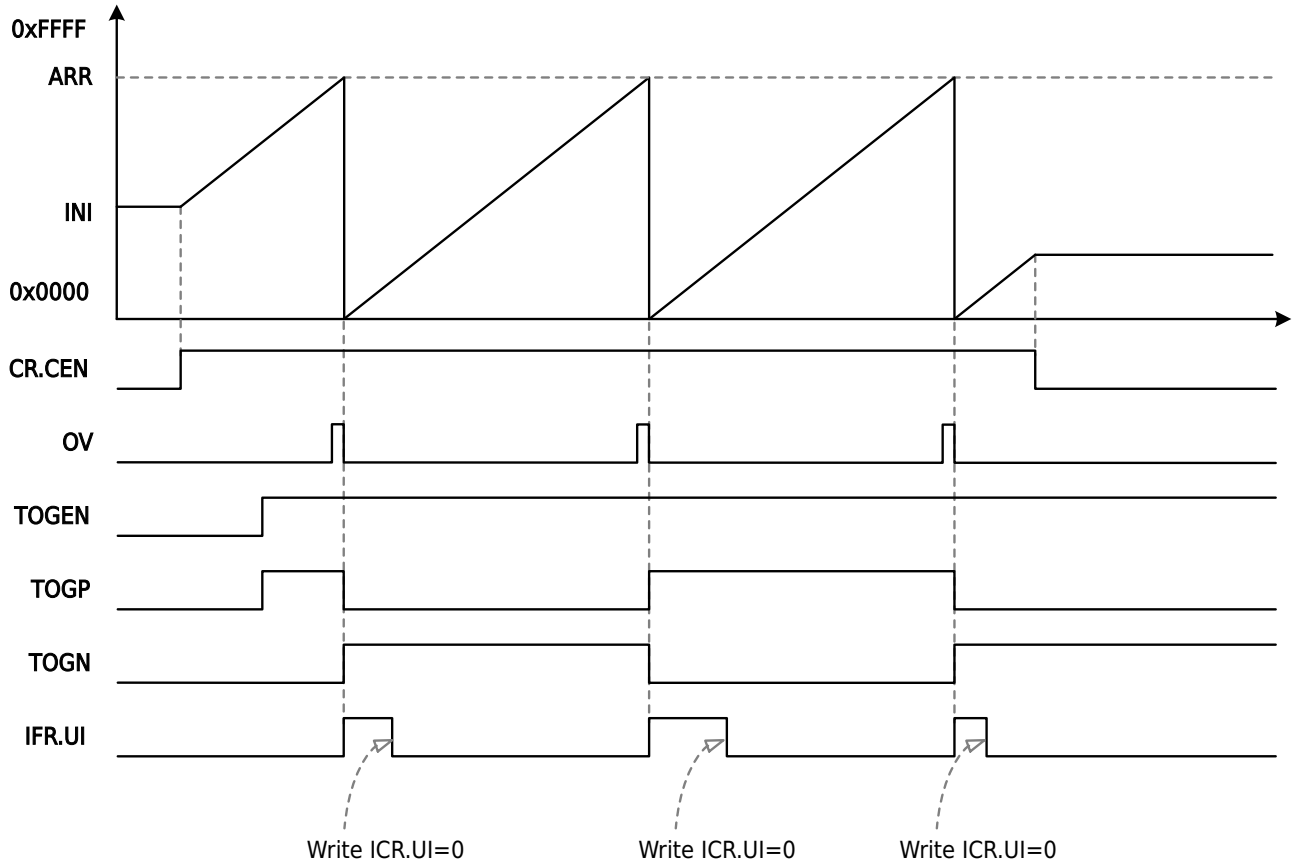


Figure 9-3 BTIM Counter Waveform

9.3.4 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.

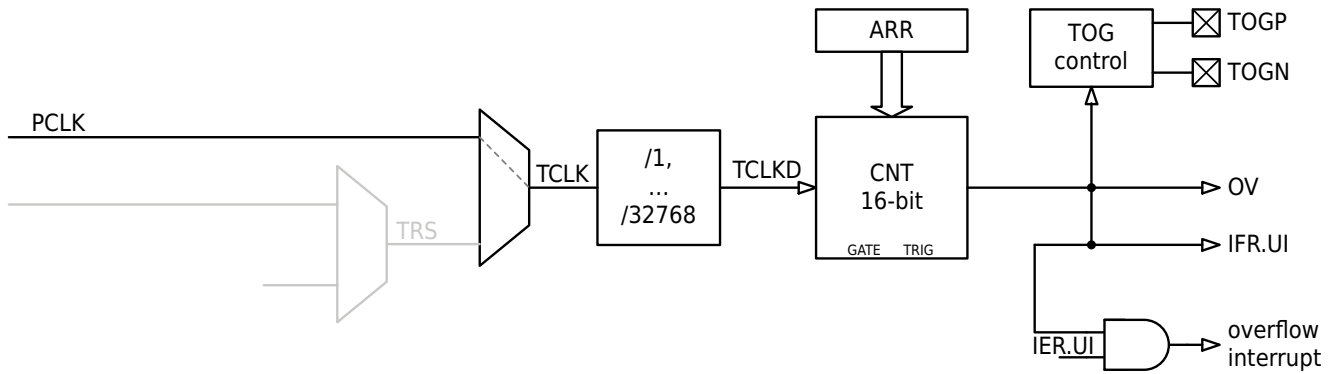


Figure 9-4 BTIM Timing Mode Block Diagram

9.3.5 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.

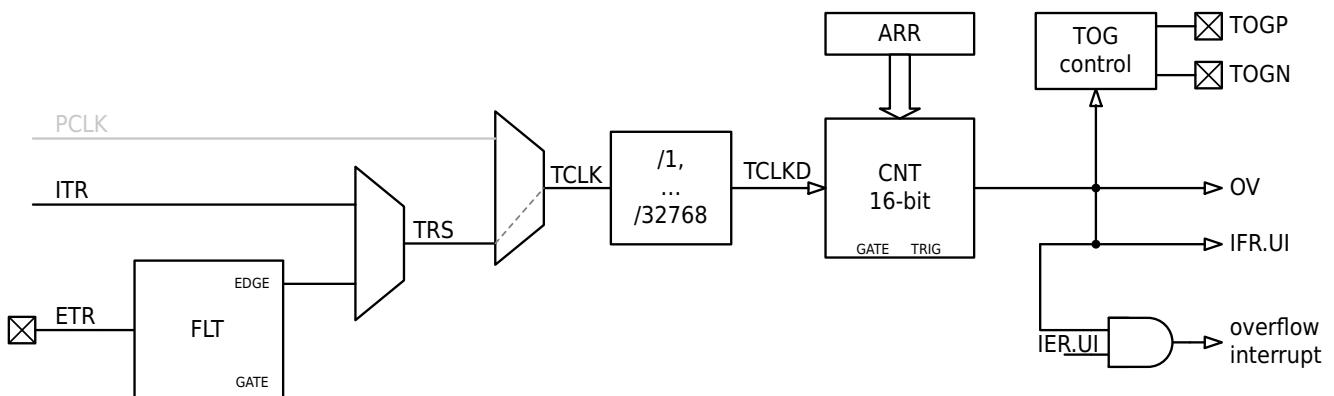


Figure 9-5 BTIM Counter Mode Block Diagram

9.3.6 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 **TRIS** 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 **TRIS** 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.

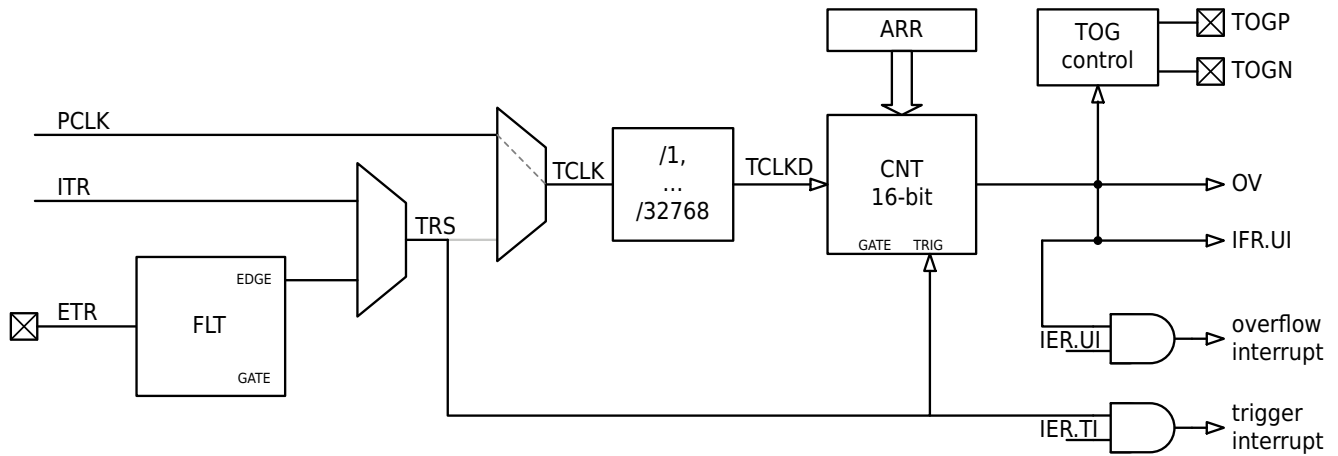


Figure 9-6 BTIM Trigger Start Mode Block Diagram

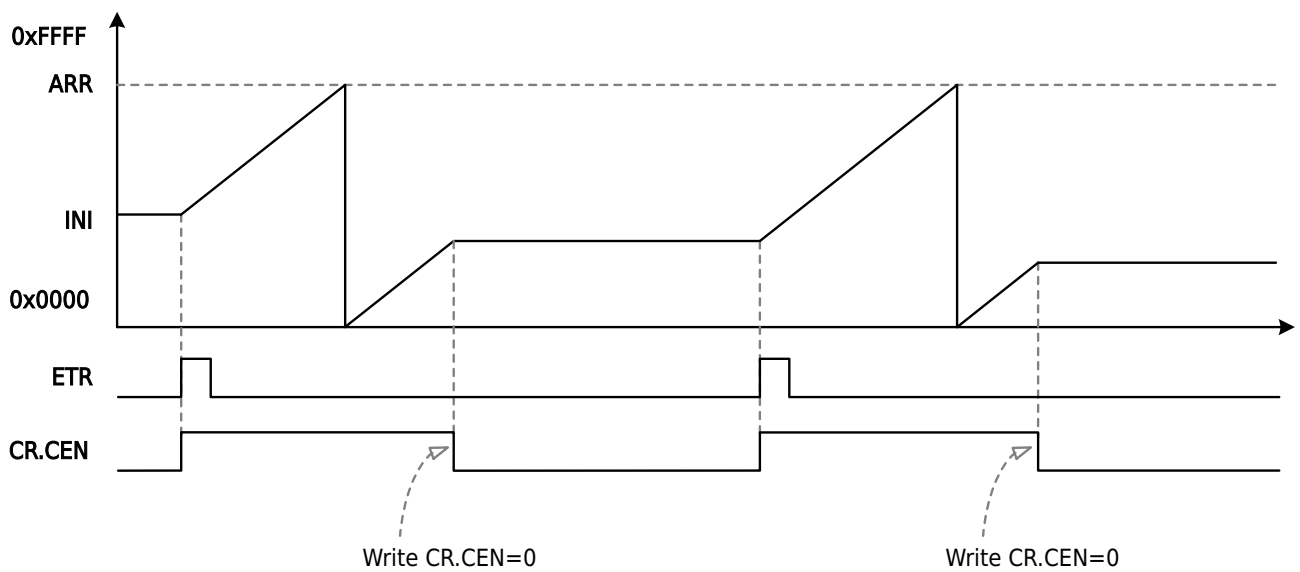


Figure 9-7 BTIM Trigger Start Mode Counting Diagram

9.3.7 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.

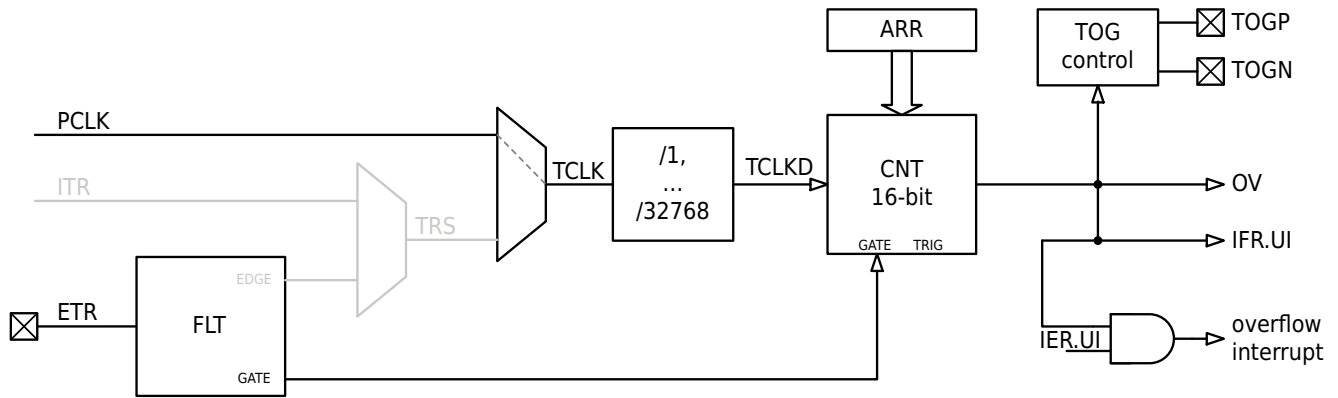


Figure 9-8 BTIM Gating Mode Block Diagram

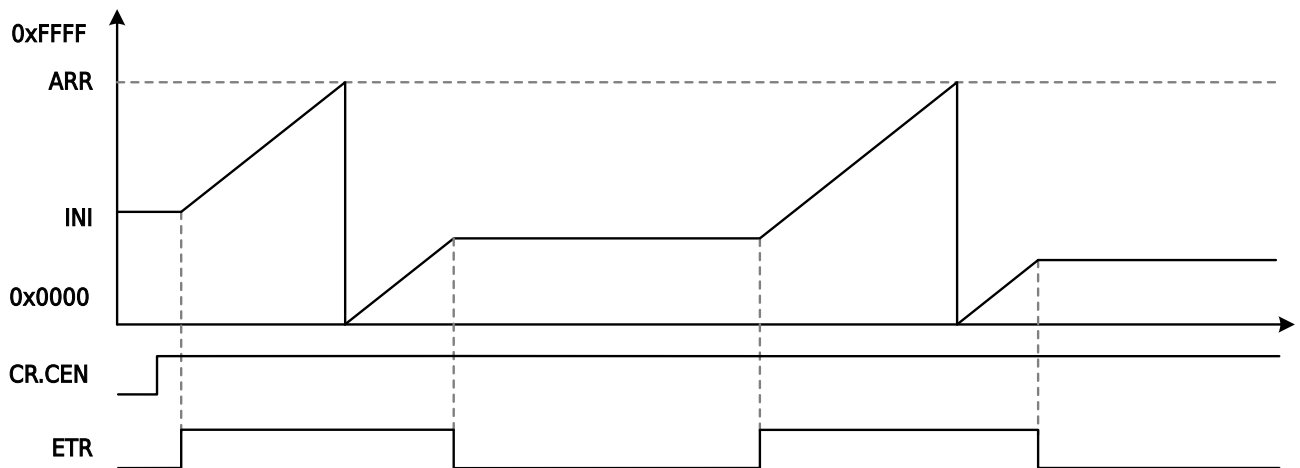


Figure 9-9 BTIM Gating Mode Counting Diagram

9.3.8 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.

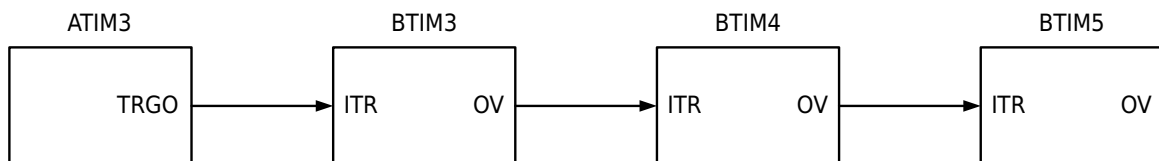


Figure 9-10 BTIM Timer Cascade Diagram

9.4 Register Description

9.4.1 Register List

- BTIM3 base address: 0x40007400
- BTIM4 base address: 0x40007500
- BTIM5 base address: 0x40007600



Note

BTIM is enabled when SYSCTRL.CR1.GTIMCFG is set to 1.

Table 9-1 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	BTIM345_AIFR	Composite Interrupt Flag Register	32
0x4C	BTIM345_AICR	Composite Interrupt Flag Clear Register	32

9.4.2 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																																

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

9.4.3 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

9.4.4 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 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

Bit	Name	Functional Description	Access
3	TOGEN	TOG pin output enable control 0b0: TOGP, TOGN output levels are both 0 0b1: TOGP, 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

9.4.5 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

9.4.6 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																																		
Access																															RO		TI	RO	UI

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

9.4.7 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																																		
Access																															WO		TI	WO	UI

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

9.4.8 Composite Interrupt Flag Register (BTIM345_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																															
Access																											RO					
TI5																																
RO																																
UI5																																
						RO						TI4																				
						RO						UI4																				
						RO						TI3																				
						RO						UI3																				

Bit	Name	Functional Description	Access
31:6	Reserved	-	-
5	TI5	BTIM5 trigger flag 0b0: No trigger event occurred 0b1: A trigger event occurred	RO
4	UI5	BTIM5 overflow flag 0b0: Counter has not overflowed 0b1: Counter overflowed	RO
3	TI4	BTIM4 trigger flag 0b0: No trigger event occurred 0b1: A trigger event occurred	RO
2	UI4	BTIM4 overflow flag 0b0: Counter has not overflowed 0b1: Counter overflowed	RO
1	TI3	BTIM3 trigger flag 0b0: No trigger event occurred 0b1: A trigger event occurred	RO
0	UI3	BTIM3 overflow flag 0b0: Counter has not overflowed 0b1: Counter overflowed	RO

9.4.9 Composite Interrupt Flag Clear Register (BTIM345_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																										T15		U15		T14		U14		T13		U13	
Access																											WO		WO		WO		WO		WO		WO	

Bit	Name	Functional Description	Access
31:6	Reserved	-	-
5	TI5	BTIM5 trigger flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
4	UI5	BTIM5 overflow flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
3	TI4	BTIM4 trigger flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
2	UI4	BTIM4 overflow flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
1	TI3	BTIM3 trigger flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
0	UI3	BTIM3 overflow flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO

10 GTIM

10.1 Introduction

GTIM is a 16-bit timer with 4-way compare capture function. BTIM3/4/5 and GTIM functions cannot be used simultaneously. They are controlled by SYSCTRL.CR1.GTIMCFG, where GTIM is active when this bit is 0.

10.2 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

10.3 Functional Description

10.3.1 Function Block Diagram

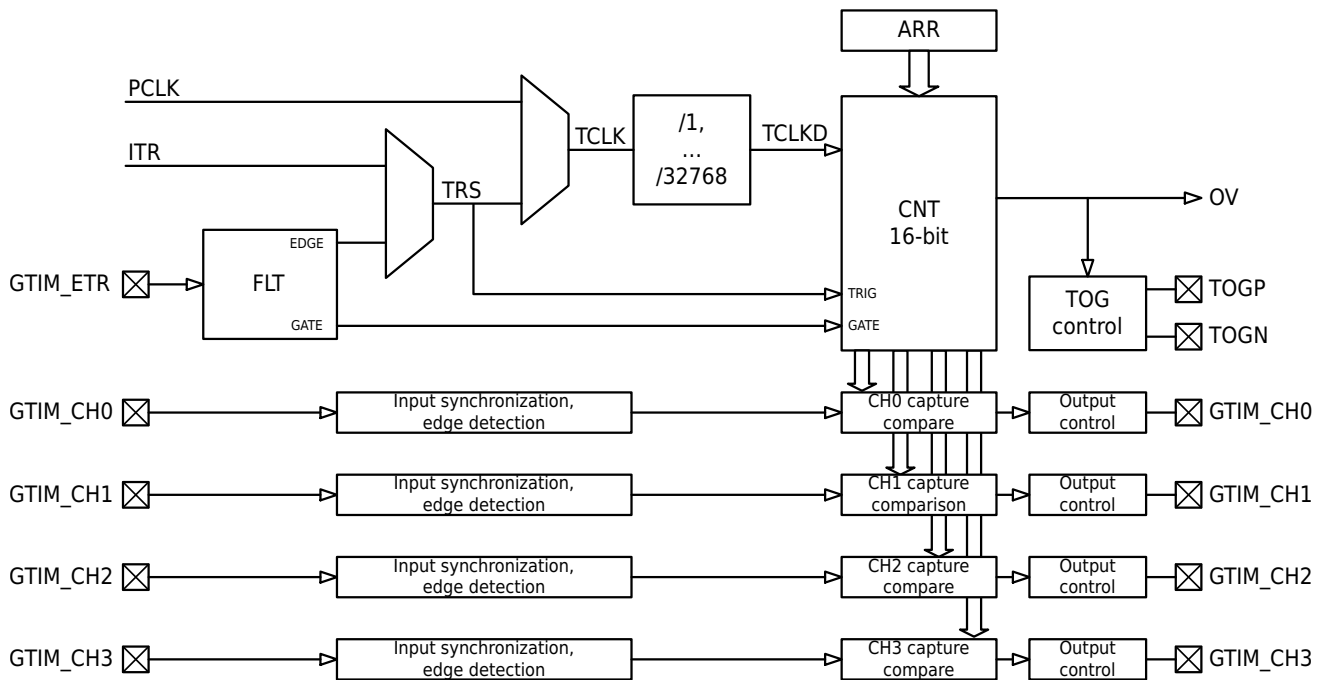


Figure 10-1 GTIM Block Diagram



Note

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

10.3.2 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.

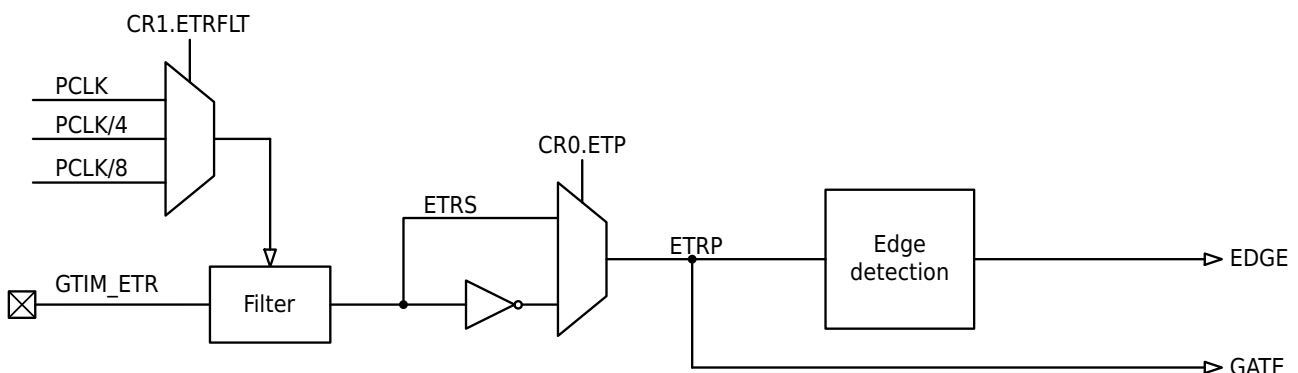


Figure 10-2 GTIM Filter Unit Diagram

10.3.3 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 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 TOGP/TOGN output level toggles once per overflow. Users can output the TOGP/TOGN signal to a pin through GPIO multiplexing function to drive external circuits.

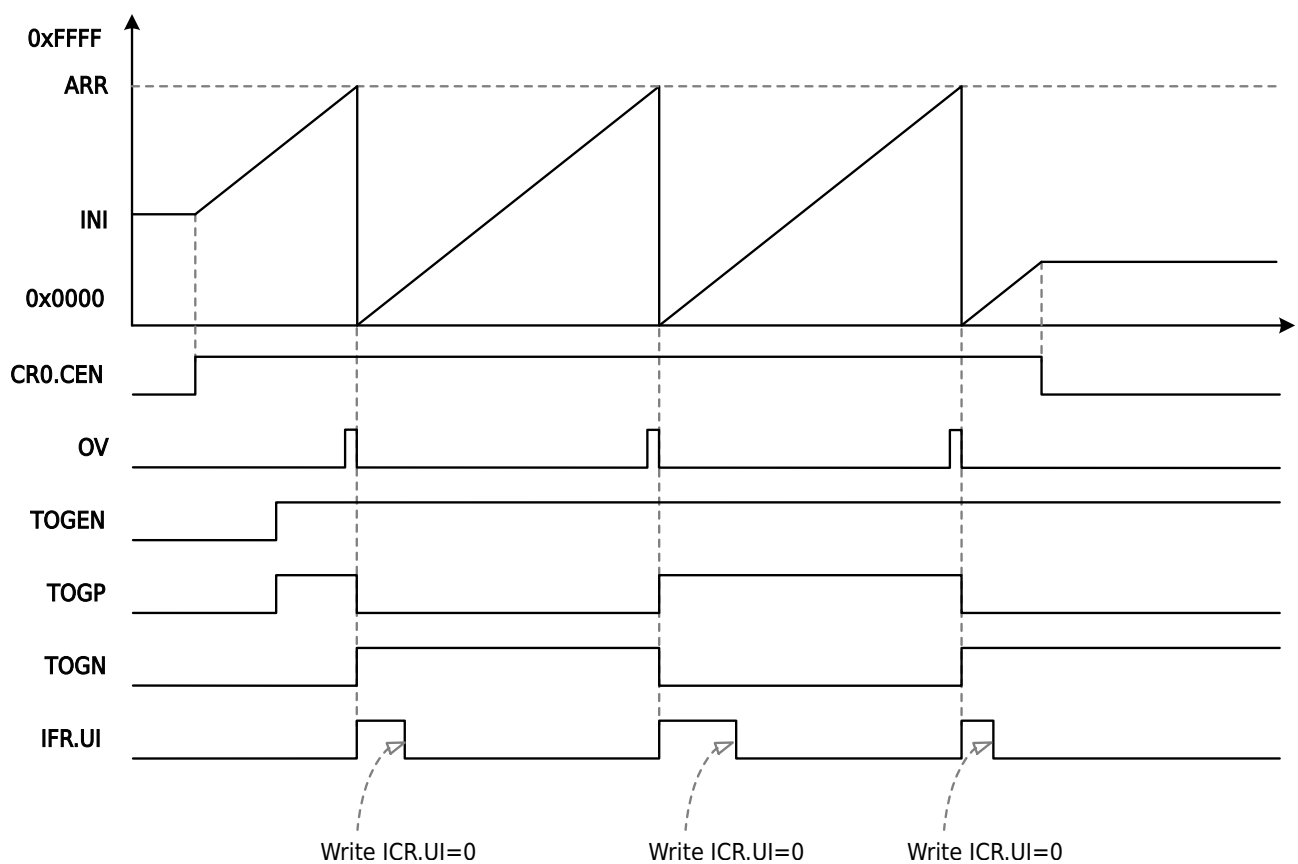


Figure 10-3 GTIM Counter Waveform

10.3.4 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.

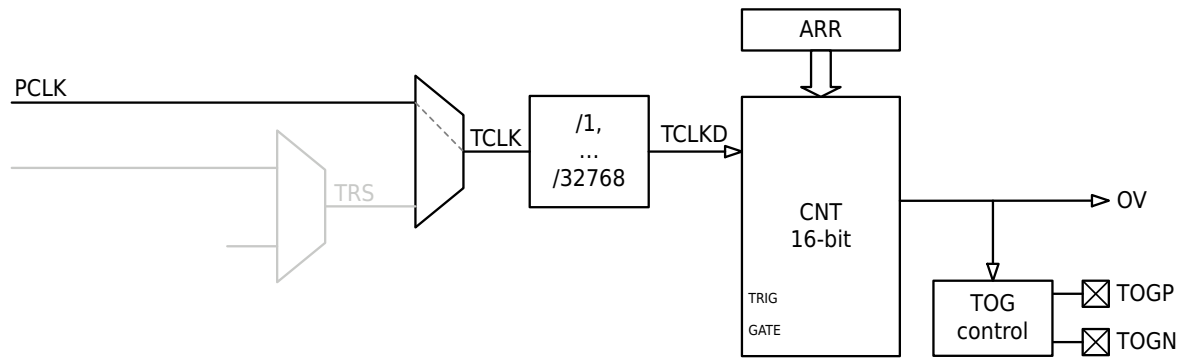


Figure 10-4 GTIM Timing Mode Block Diagram

10.3.5 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.

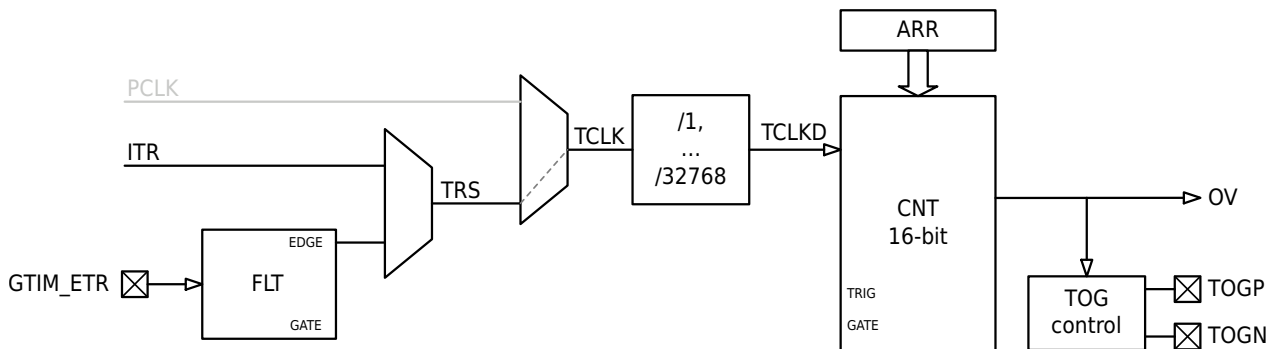


Figure 10-5 GTIM Counter Mode Block Diagram

10.3.6 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.

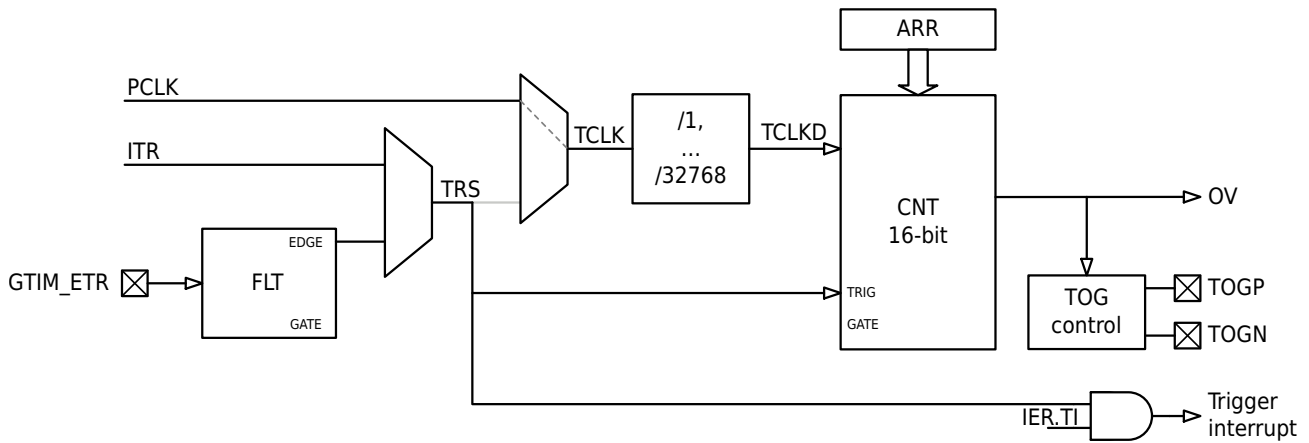


Figure 10-6 GTIM Trigger Start Mode Block Diagram

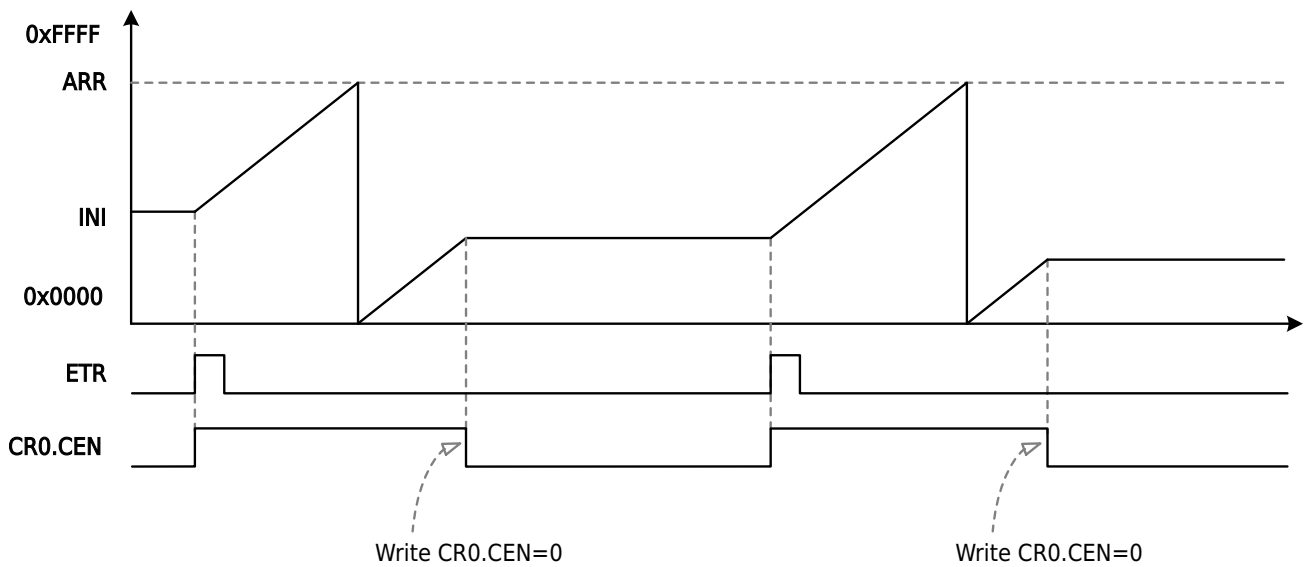


Figure 10-7 GTIM Trigger Start Mode Counting Diagram

10.3.7 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.

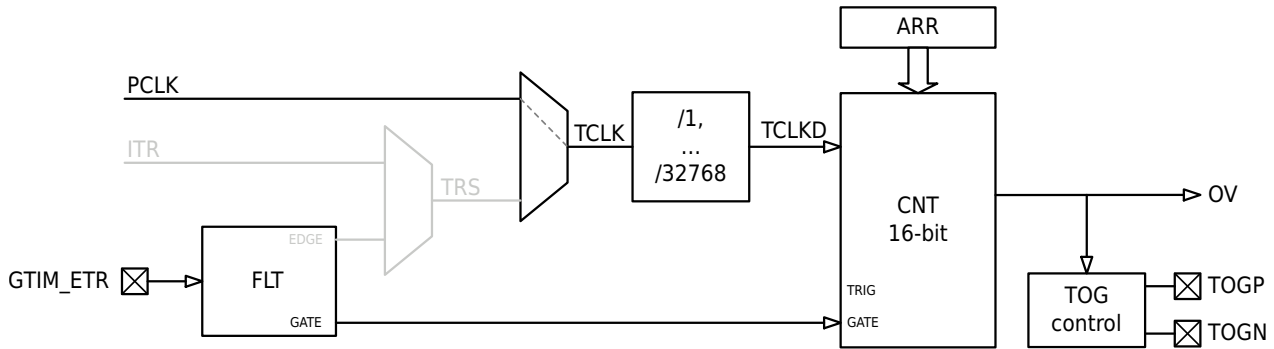


Figure 10-8 GTIM Gating Mode Block Diagram

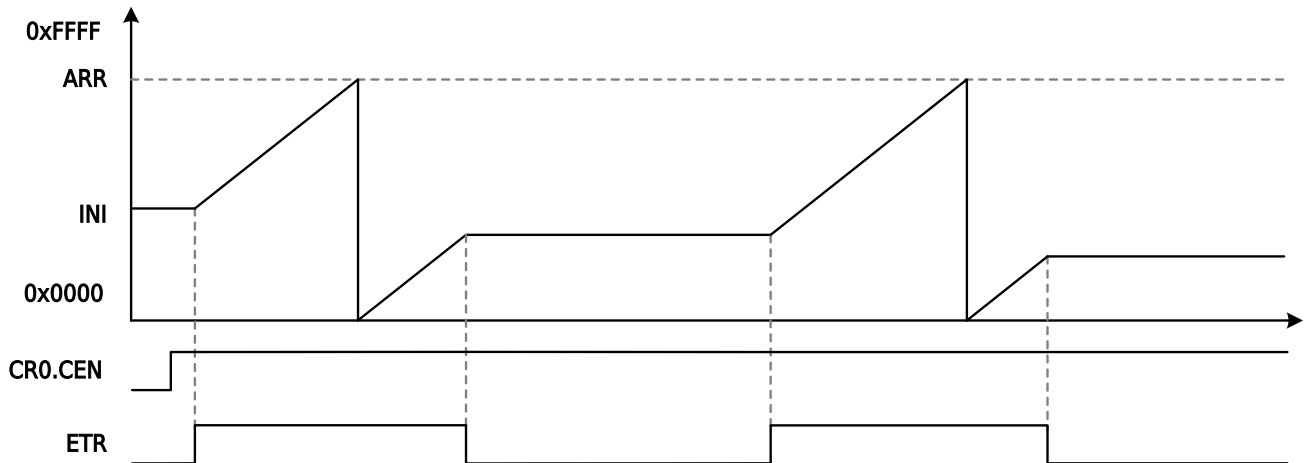


Figure 10-9 GTIM Gating Mode Counting Diagram

10.3.8 Compare Capture Function

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

10.3.8.1 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.

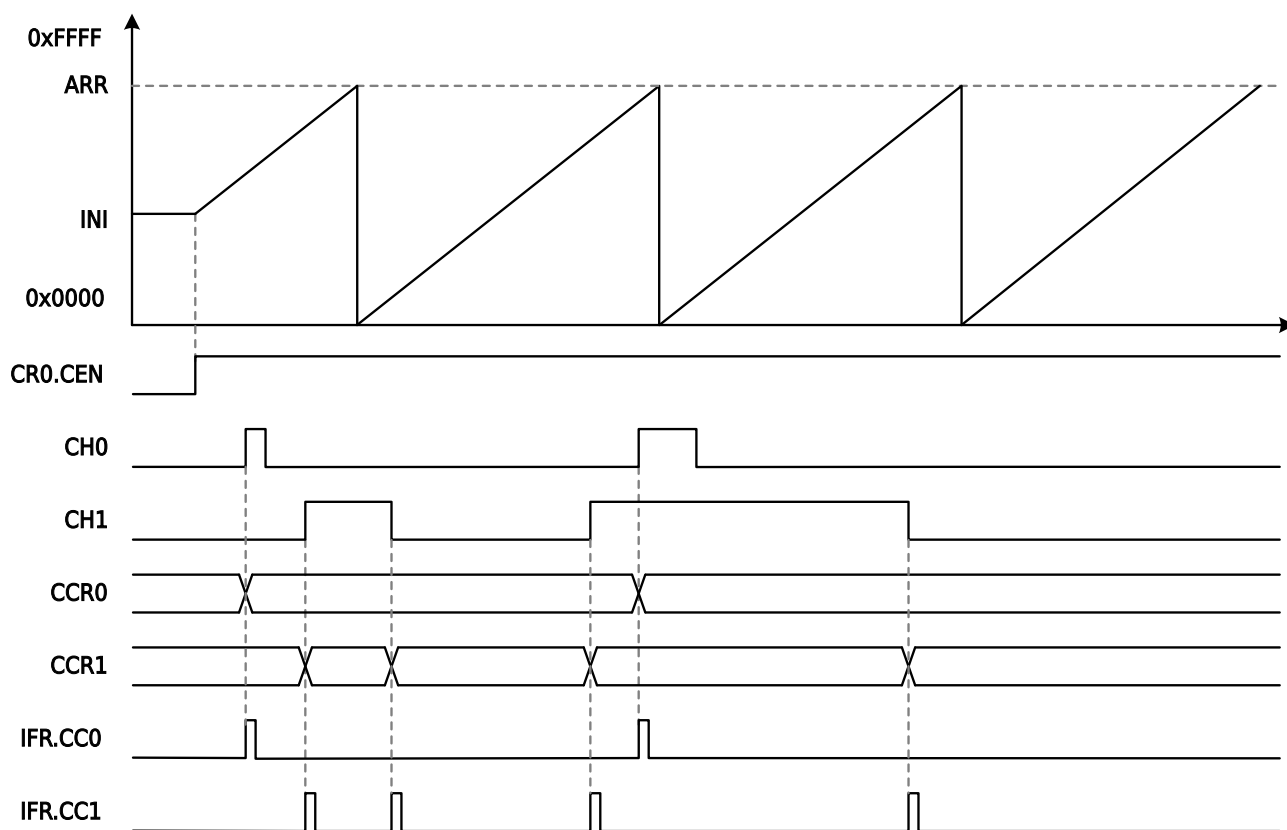


Figure 10-10 GTIM Capture Function Diagram

10.3.8.2 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 $CNT \geq CCRy$)
- 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.

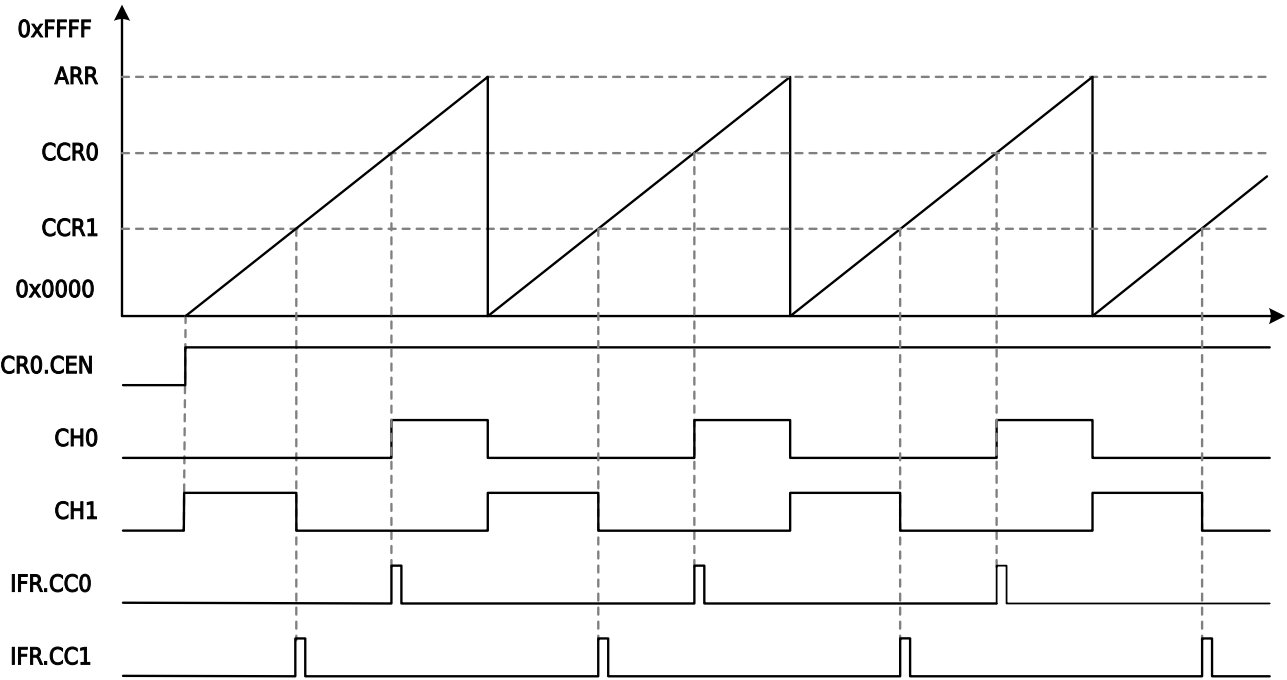


Figure 10-11 GTIM Compare Function Diagram

10.3.9 Timer Cascade

The register configuration allows ATIM3 and GTIM to be cascaded into a 32-bit timer. The ATIM3 timer uses the system clock PCLK as its timer clock, while GTIM selects the overflow signal from the preceding stage as an external pulse count and selects external counting function.

10.3.10 On-Chip Peripheral Interconnection

When GPIOx_CR4.GTIMCH0 is 0, the source of GTIM_CHx signal is determined by GPIO_AFRL; when GPIOx_CR4.GTIMCH0 is 1-3, the source of GTIM_CHx signal is determined by the following table.

GPIOx_CR4.GTIMCH0	GTIM_CH0 Source
0b01	LPUART0_RX
0b10	LPUART1_RX
0b11	LVD_OUT

10.4 Register Description

10.4.1 Register List

GTIM base address: 0x40007400



Note

GTIM is enabled when SYSCTRL.CR1.GTIMCFG is set to 0.

Table 10-2 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
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

10.4.2 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																																

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

10.4.3 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

10.4.4 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
31:9	Reserved	-	-
8:7	Reserved	All bits read as 0, Write as 0	-
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: $F_{\text{sample}} = \text{PCLK}$, $N=2$ 0b010: $F_{\text{sample}} = \text{PCLK}$, $N=4$ 0b011: $F_{\text{sample}} = \text{PCLK}$, $N=6$ 0b100: $F_{\text{sample}} = \text{PCLK}/4$, $N=4$ 0b101: $F_{\text{sample}} = \text{PCLK}/4$, $N=6$ 0b110: $F_{\text{sample}} = \text{PCLK}/8$, $N=4$ 0b111: $F_{\text{sample}} = \text{PCLK}/8$, $N=6$	RW
3:0	Reserved	-	-

10.4.5 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	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: Signal input from ITR pin (ATIM3_TRGO) 0b1X: Setting prohibited  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

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: TOGP, TOGN output levels are both 0 ● 0b1: TOGP, 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 from TRS selection signal ● 0b10: Trigger start mode, counting clock is PCLK; TRS signal triggers counter start ● 0b11: Gating mode, counting clock is PCLK, TRS 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

10.4.6 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																																
Access																											RW	RW	RW	RW	RW	RW	

Bit	Name	Functional Description	Access
31:6	Reserved	-	-
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
1	TI	Trigger interrupt enable control 0b0: Disable 0b1: Enable	RW
0	UI	Overflow interrupt enable control 0b0: Disable 0b1: Enable	RW

10.4.7 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																																					
Access																											RO	CC3	RO	CC2	RO	CC1	RO	CC0	RO	TI	RO	UI

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

10.4.8 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																															
Access																											WO	CC3	WO	CC2	WO	CC1

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

10.4.9 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 $CNT \geq CCR0$) 0b111: PWM inverted output (outputs high level when $CNT < CCR0$)	RW

10.4.10 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

11 ATIM

11.1 Introduction

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.

11.2 Features

- 6 independent PWM outputs CH0A, CH0B, CH1A, CH1B, CH2A, CH2B
- 3 complementary PWM outputs (CHxA, CHxB)
- Up to 6 capture inputs
- Pulse width measurement
- Quadrature encoder counting function
- Single pulse mode
- External input counting function
- Dead-time control
- Braking control
- Edge alignment, symmetric center alignment and asymmetric center alignment PWM output

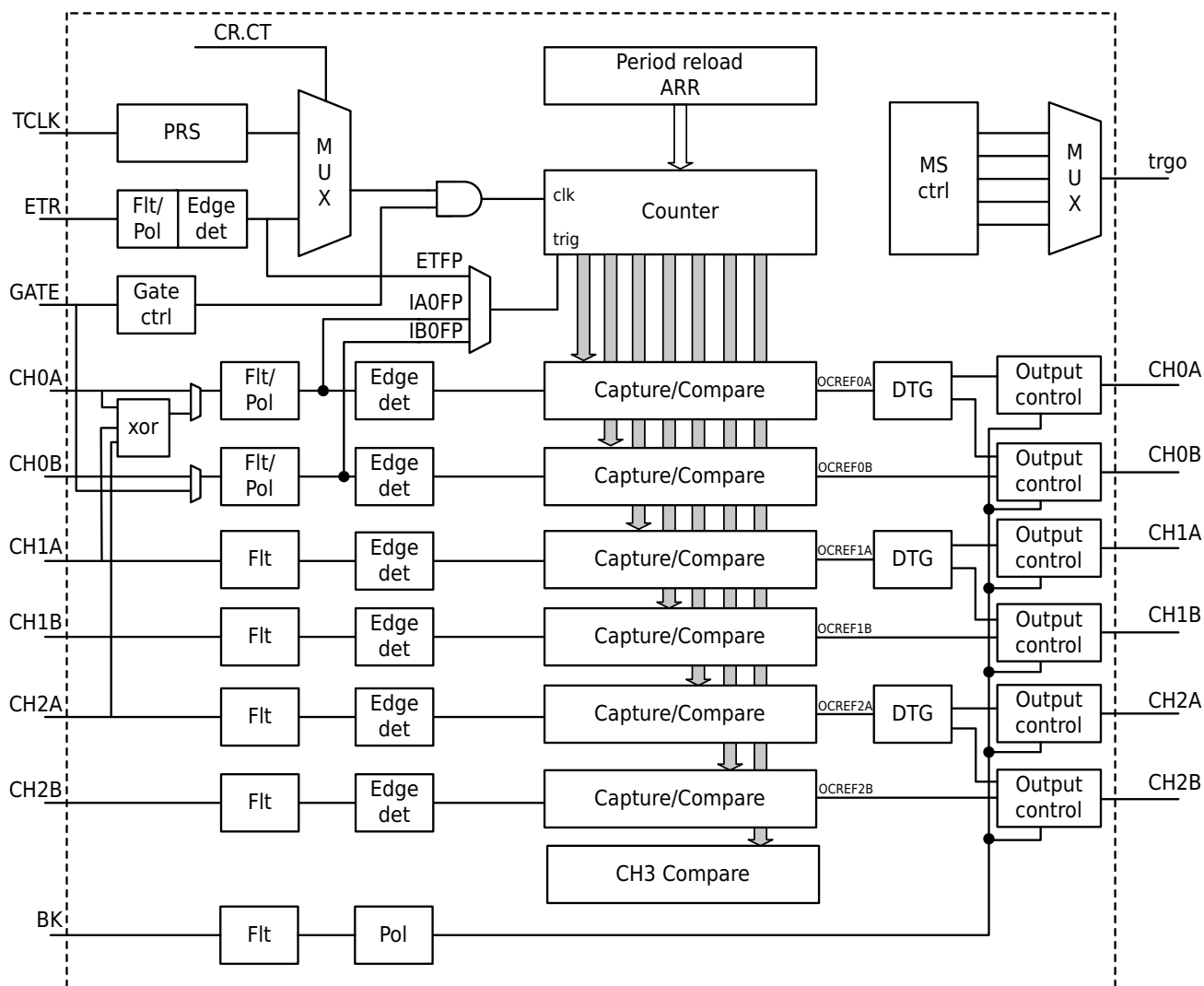


Figure 11-1 ATIM3 Block Diagram

11.3 Functional Description

11.3.1 Timer Clock

The timer uses PCLK as the timer clock, and the timer clock is marked with TCLK below.

11.3.2 Timer Counter

The main module of advanced 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.

11.3.3 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
Frequen- cy divi- sion 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.

11.3.4 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. 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.

Function Block Diagram

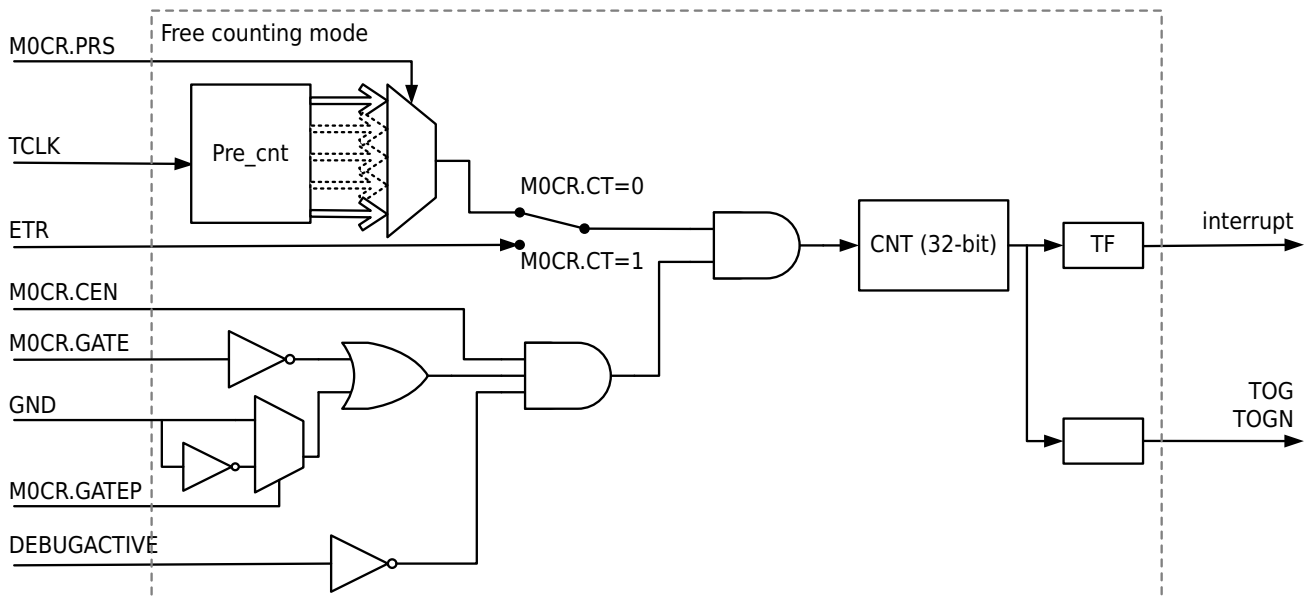


Figure 11-2 Mode0 Count Timer Free Counting Block Diagram

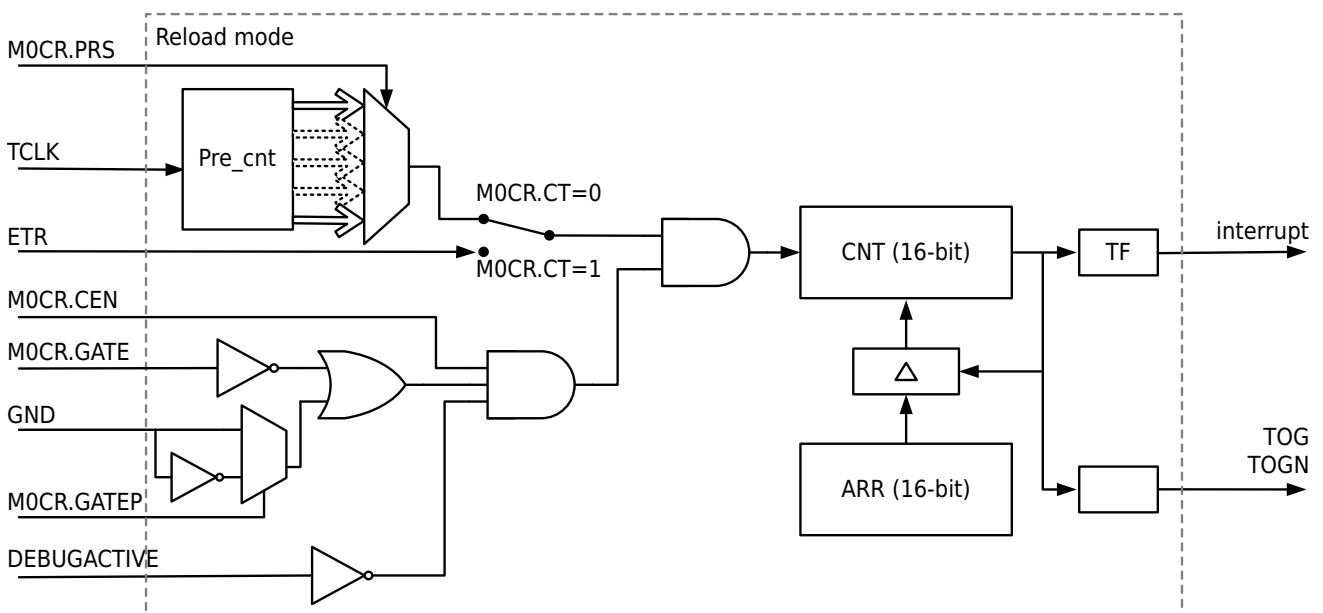


Figure 11-3 Mode0 Count Timer Reload Counting Block Diagram

**Note**

Mode 0 does not support gating function. Keep the GATE/GATEP bits at their reset values and do not modify them.

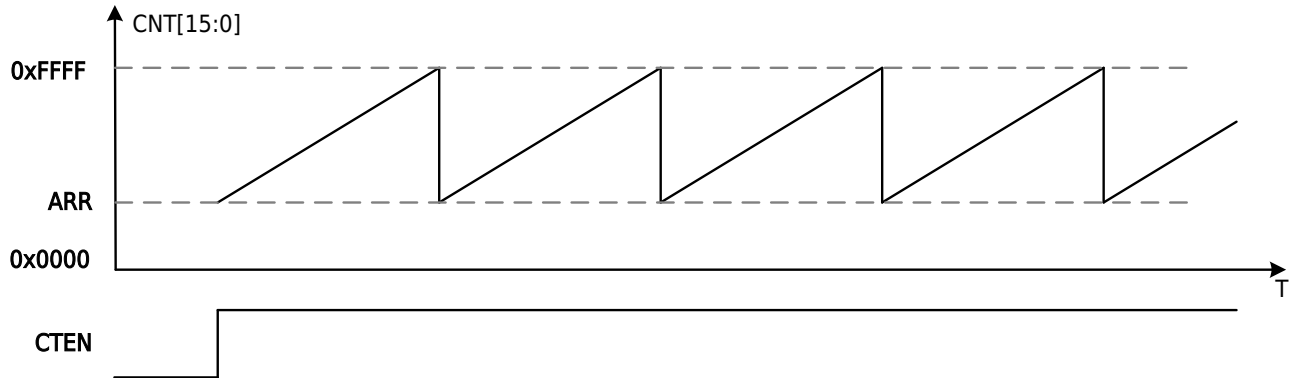
Count Waveform

Figure 11-4 Mode0 Count Timer 16-Bit Reload Counting Waveform

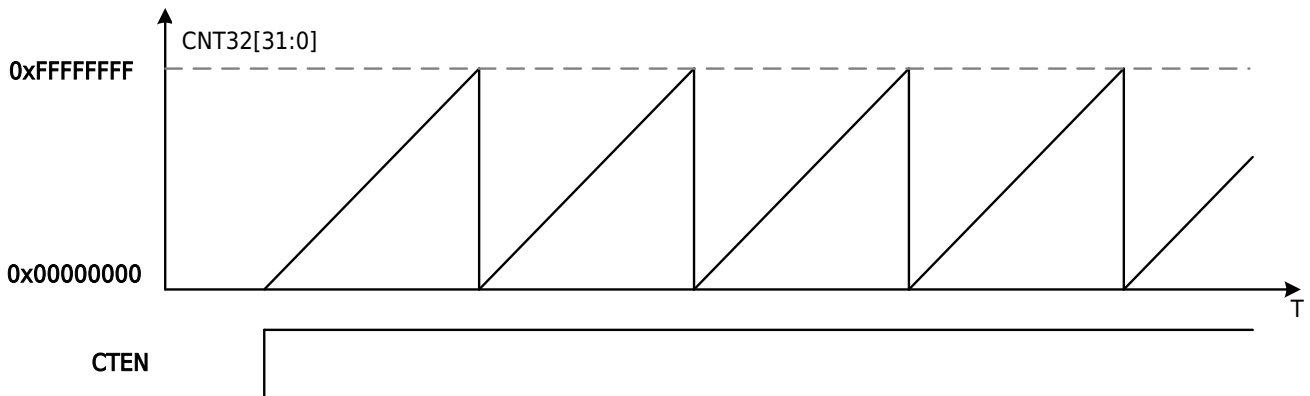


Figure 11-5 Mode0 Count Timer 32-Bit Free Counting Waveform

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.

Timing Diagram

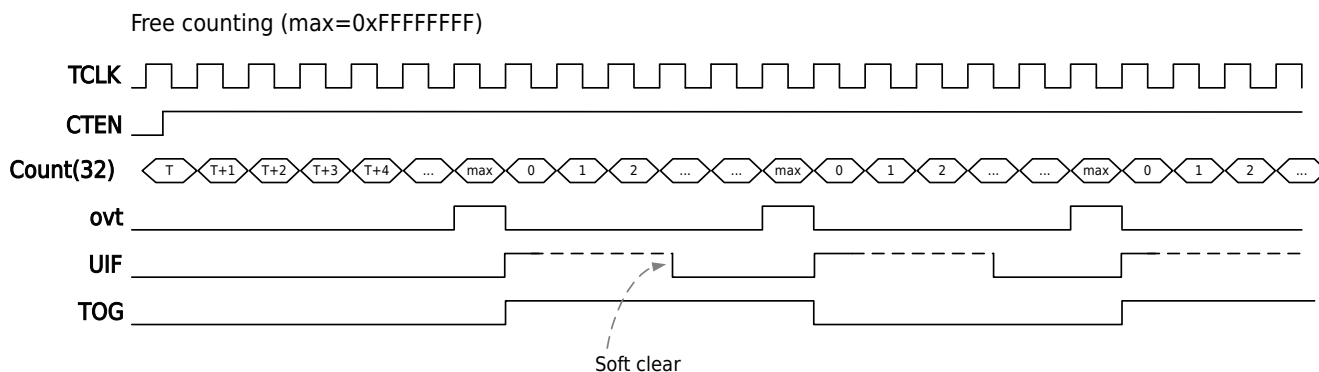


Figure 11-6 Mode0 Count Timer Free Counting Timing Diagram

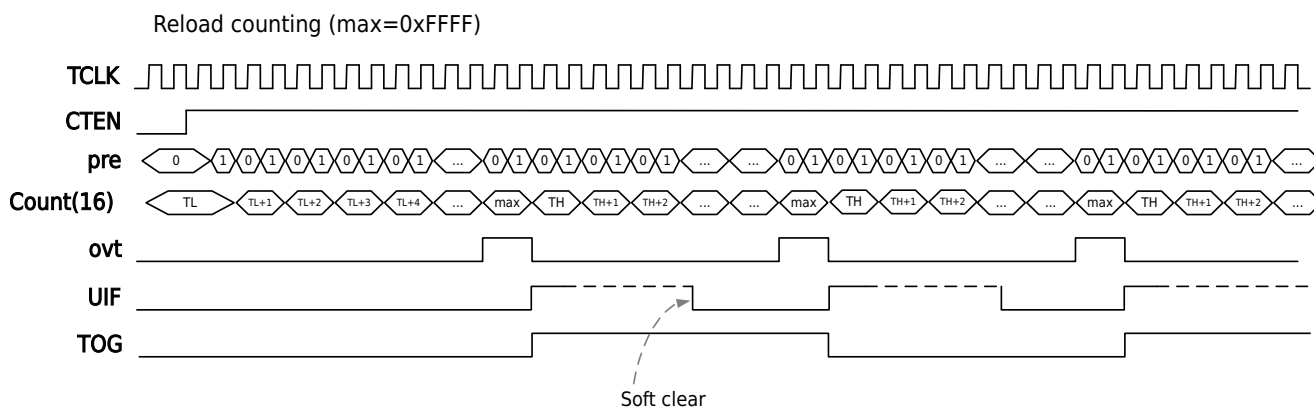


Figure 11-7 Mode0 Count Timer Reload Counting Timing Diagram (Prescaler=2)

Buzzer Function

The Buzzer function can be realized through the toggling output function of the timer. To use toggle output, the DTR.MO control bit must be enabled. Setting M0CR.TOG 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

11.3.5 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

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

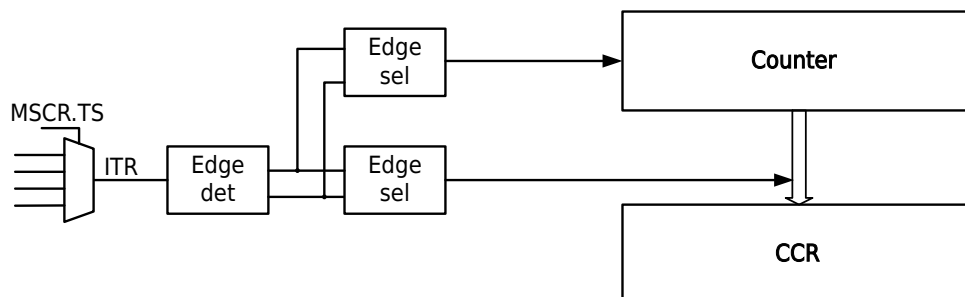


Figure 11-8 Mode1 Pulse Width Measurement PWC Function Block Diagram

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)

For TRGO output, refer to the register MSCR description.

Timer	ITR0	ITR1	ITR2	ITR3
ATIM3	-	-	GTIM_OV	-

PWC Waveform Measurement Timing Diagram

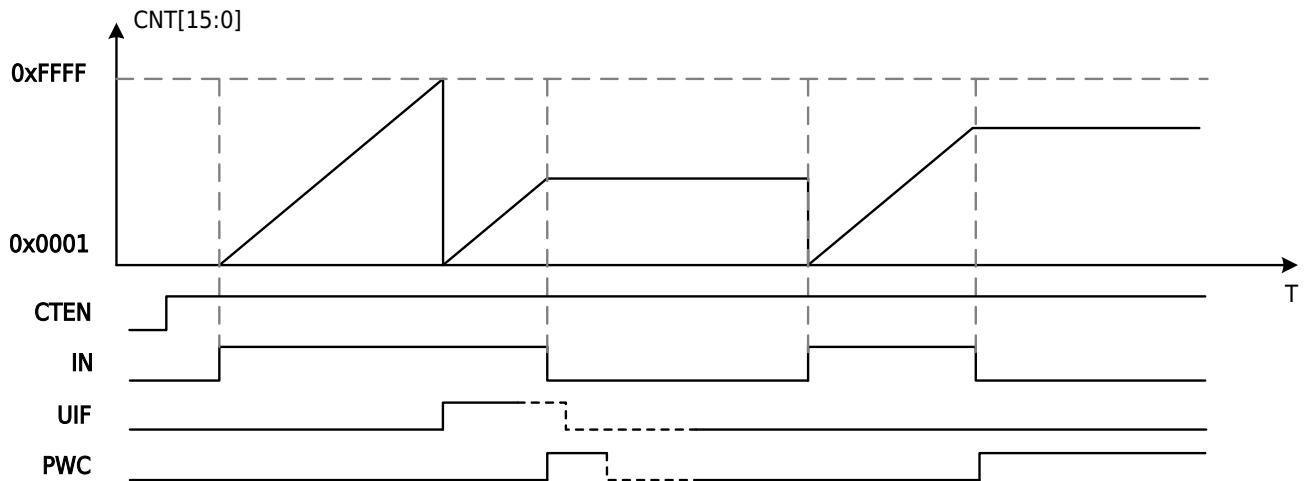


Figure 11-9 Mode1: Pulse Width Measurement PWC High-Level Pulse Width Measurement

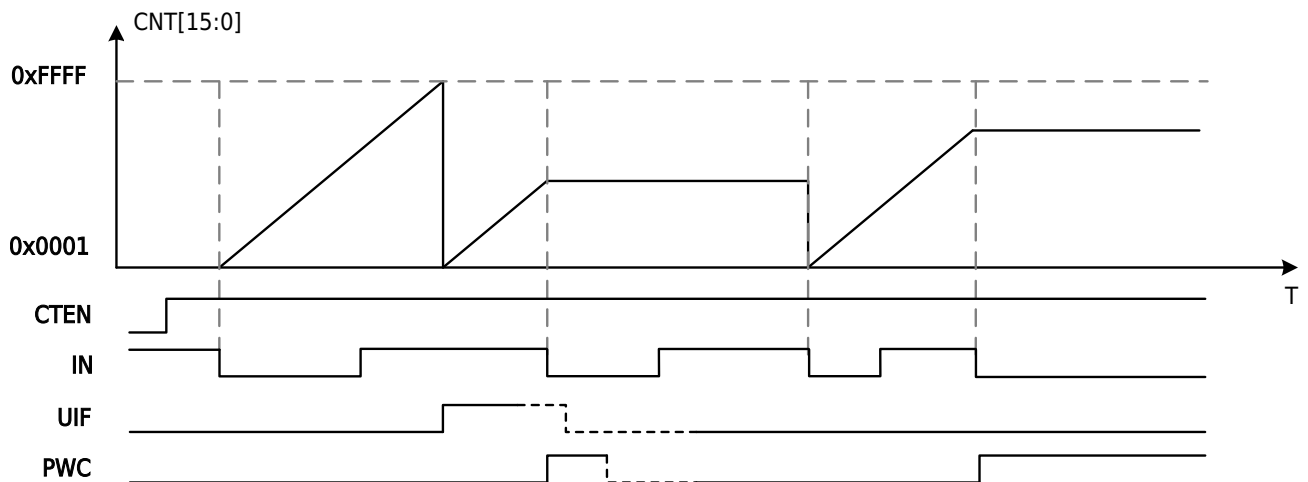


Figure 11-10 Mode1: Pulse Width Measurement PWC Falling Edge to Falling Edge Period Measurement

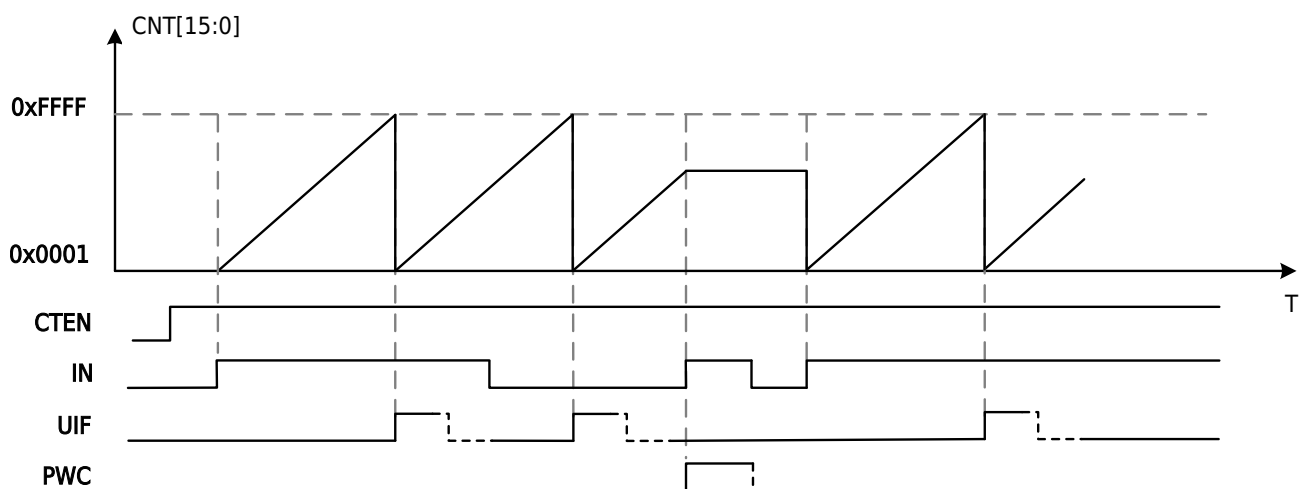


Figure 11-11 Mode1: Pulse Width Measurement PWC Rising Edge to Rising Edge Period Measurement

PWC Single trigger Mode

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

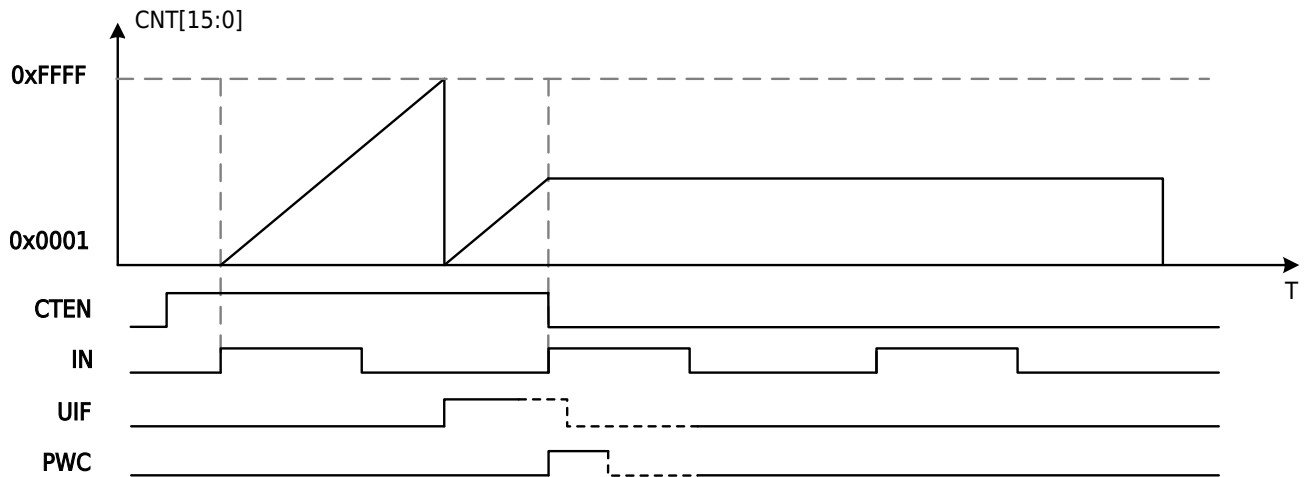


Figure 11-12 Model1: Pulse Width Measurement PWC Rising Edge to Rising Edge Period Measurement Single Mode

11.3.6 Mode2/3 Compare Capture Mode

11.3.6.1 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.

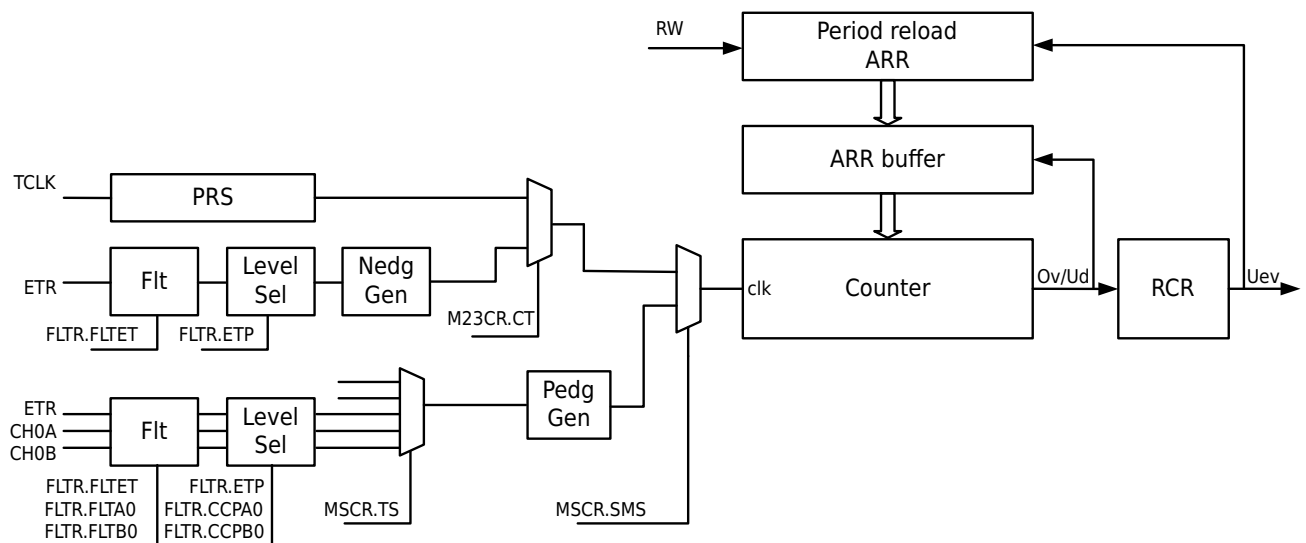


Figure 11-13 Mode 2/3 Compare Capture Mode Counter Block Diagram

The basic counter unit includes:

- Counter register CNT
- Prescaler register M23CR.PRS
- Auto-reload register ARR
- Repeat count register RCR
- Clock selection control registers FLTR, M23CR, MSCR

The auto-loaded register has a cache function, and the reload value is updated from the buffer register to the counter after the counter generates an event update. When the counter is in the stopped state or the cache function is disabled, the auto-reload register is immediately updated to the buffer register. When the timer is in the running state and the cache function is enabled, the value written to the

auto-reload register is not immediately updated to the buffer register. The auto-reload register updates to the buffer register only after an event update occurs.

For clock selection and gating functions, trigger functions, and reset functions, refer to the relevant descriptions in the [Mode2/3 Slave Mode](#) chapter.

11.3.6.2 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 (ATIM3_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 (ATIM3_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 (ATIM3_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.

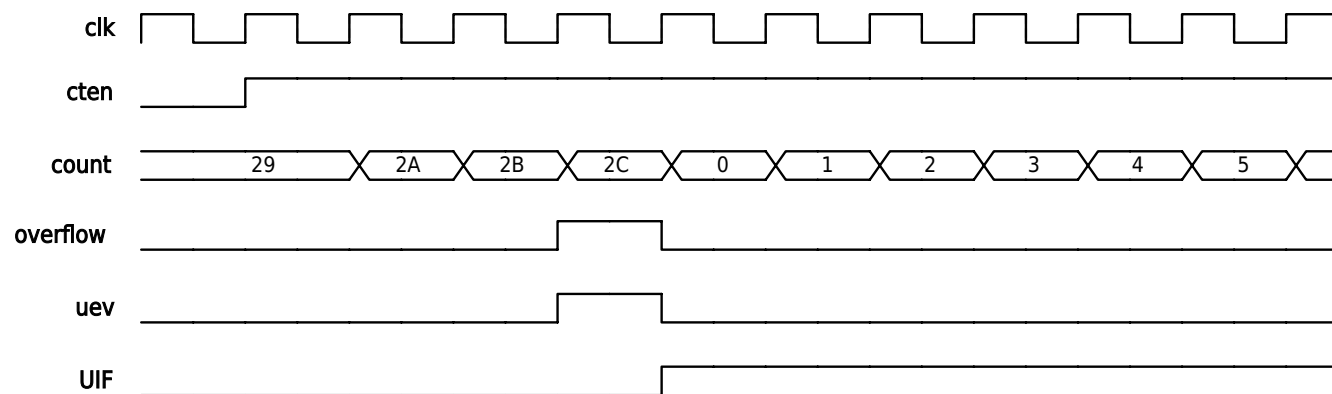


Figure 11-14 Mode 2 Compare Capture Mode Up-Counting Without Prescaler

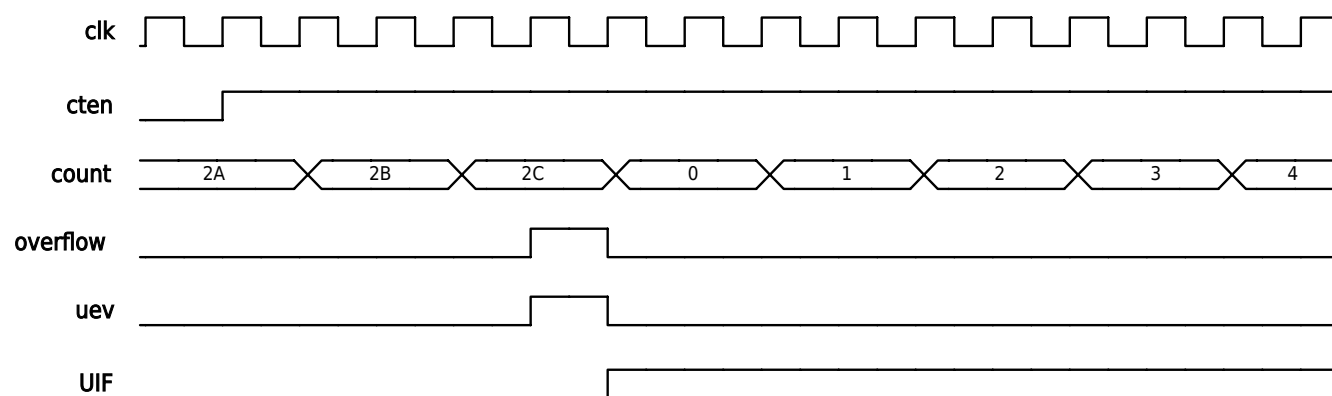


Figure 11-15 Mode 2 Compare Capture Mode Up-Counting With Prescaler

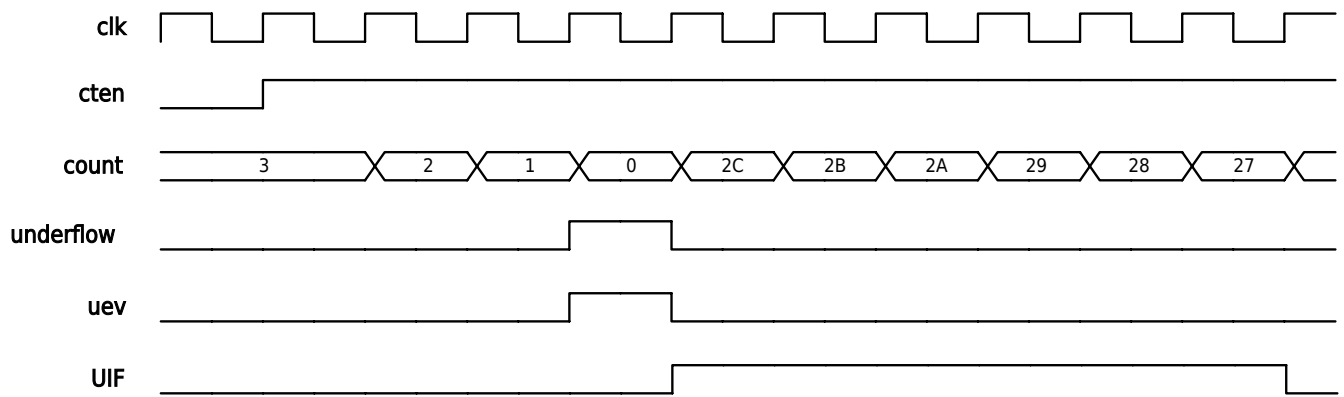


Figure 11-16 Mode 2 Compare Capture Mode Down-Counting Without Prescaler

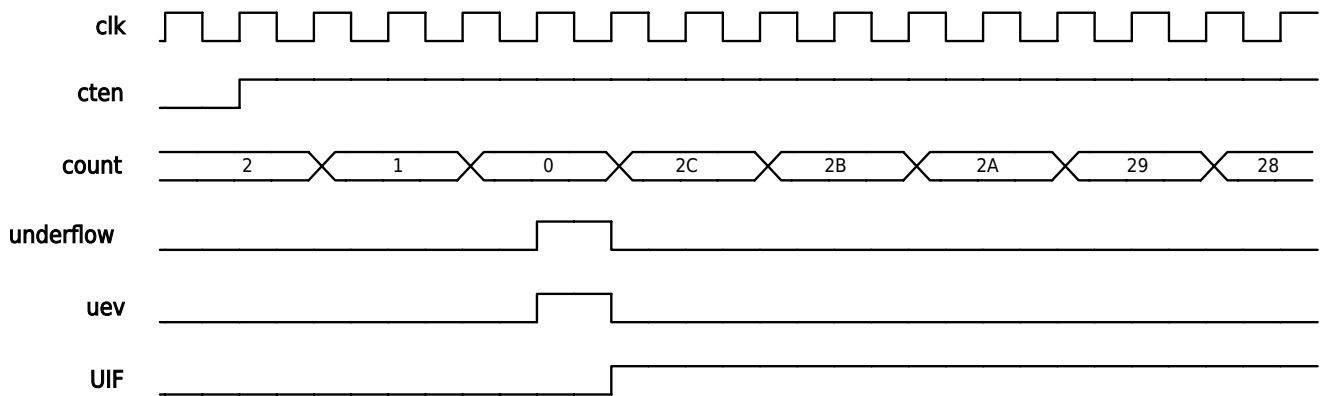


Figure 11-17 Mode 2 Compare Capture Mode Down-Counting With Prescaler

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

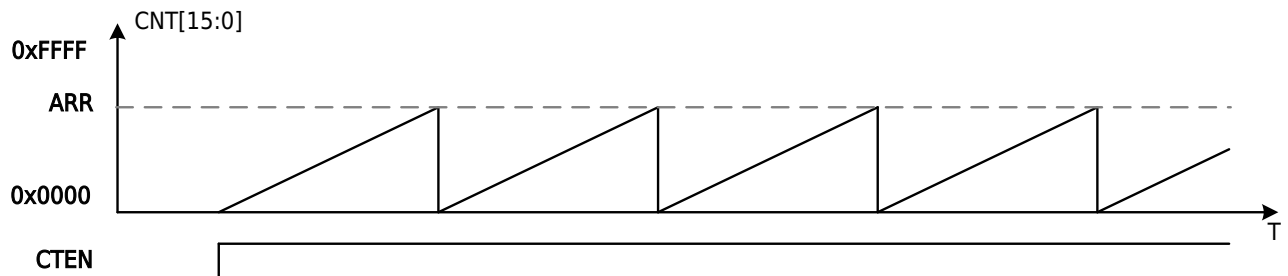


Figure 11-18 Mode2 Edge-Aligned Timer Waveform (M23CR.DIR=0)

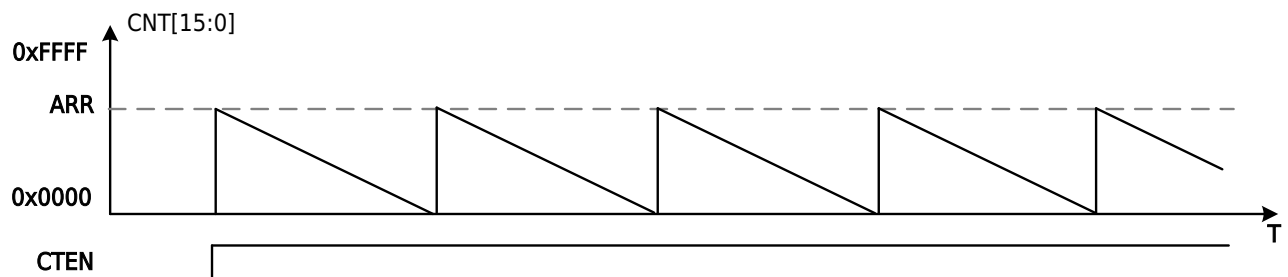


Figure 11-19 Mode2 Edge-Aligned Timer Waveform (M23CR.DIR=1)

Mode3 Triangular Counting Waveform

Mode 3 is a triangular counting waveform, the counting direction control bit is read-only, and the counting direction cannot be changed.

In center-aligned (triangular wave) mode, the M23CR.DIR direction bit is read-only, and writing to it has no effect. Switching from other modes to center-aligned mode DIR is automatically cleared to 0. DIR is automatically cleared in reset mode by software event update and external trigger in slave mode.

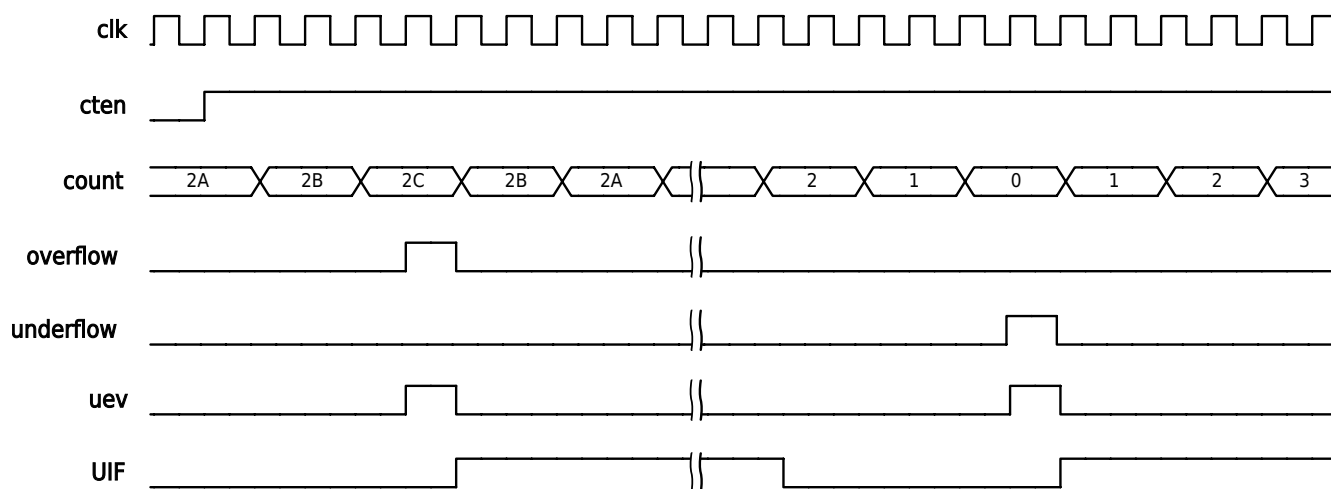


Figure 11-20 Mode3 Compare Capture Mode Up/Down-Counting with Prescaler

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

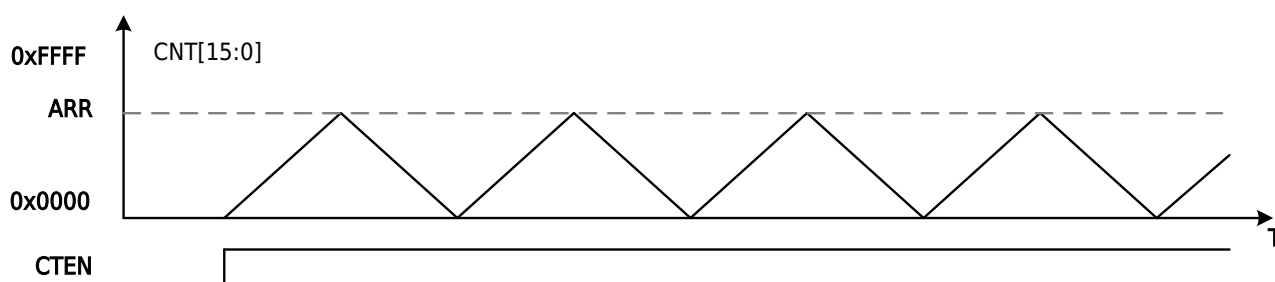


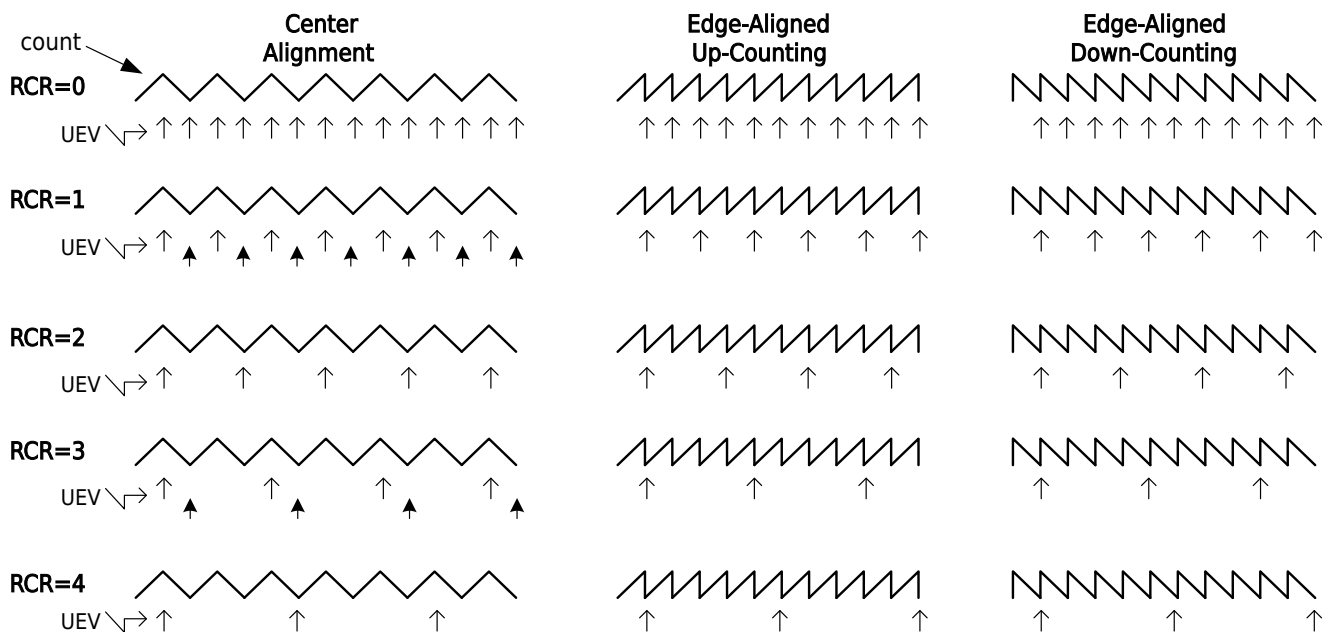
Figure 11-21 Mode3 Center-Aligned Counter Waveform

11.3.6.3 Repeat Counting

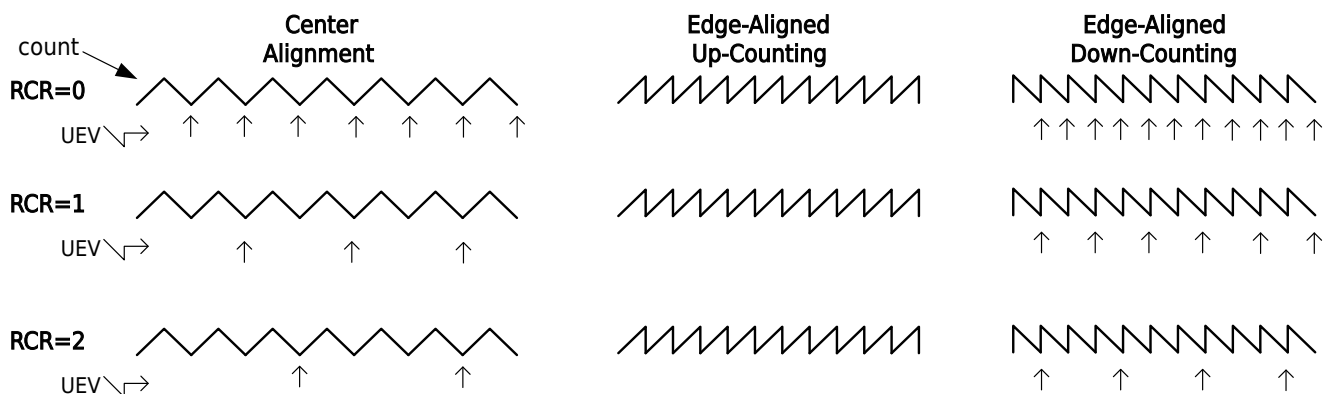
The repeat counter performs down-counting using counter overflows. When counting reaches 0, that is, when the counter overflows by the value set in the repeat register plus one. If buffer register is enabled, the period reload register updates to the period buffer register. In compare mode, the compare register value updates to the compare buffer register.

The repetition counter decrements when the following conditions are met:

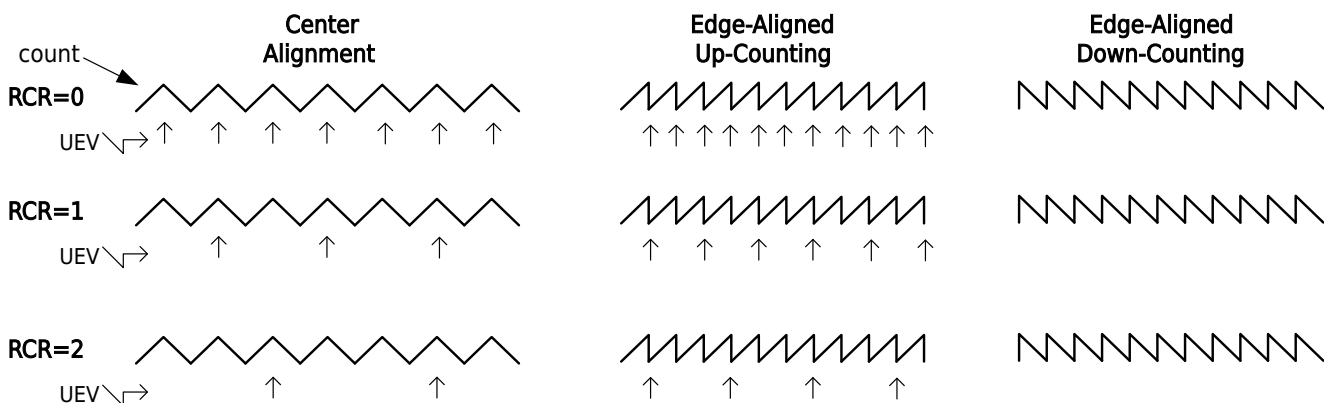
- In up-counting mode: upon each counter overflow
- In down-counting mode: upon each counter underflow
- In triangular wave mode: upon each overflow and underflow

Figure 11-22 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 11-23 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 11-24 Update Timing Generated by Repetition Counter when Underflow Is Masked ($RCR.OV=0$, $RCR.UD=1$)

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.

11.3.6.4 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:

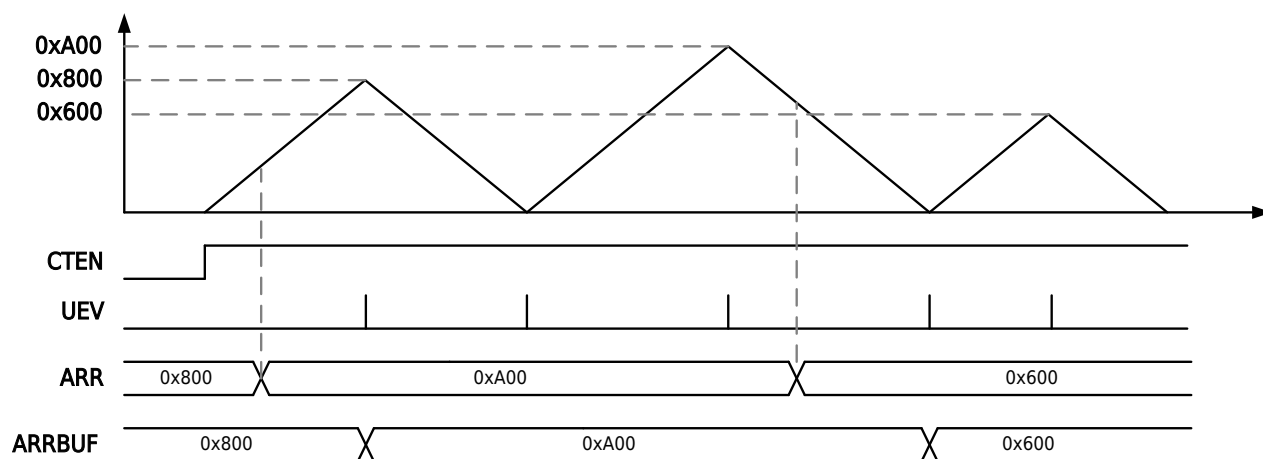


Figure 11-25 Mode 3 Triangular Wave Mode with Cache Enable

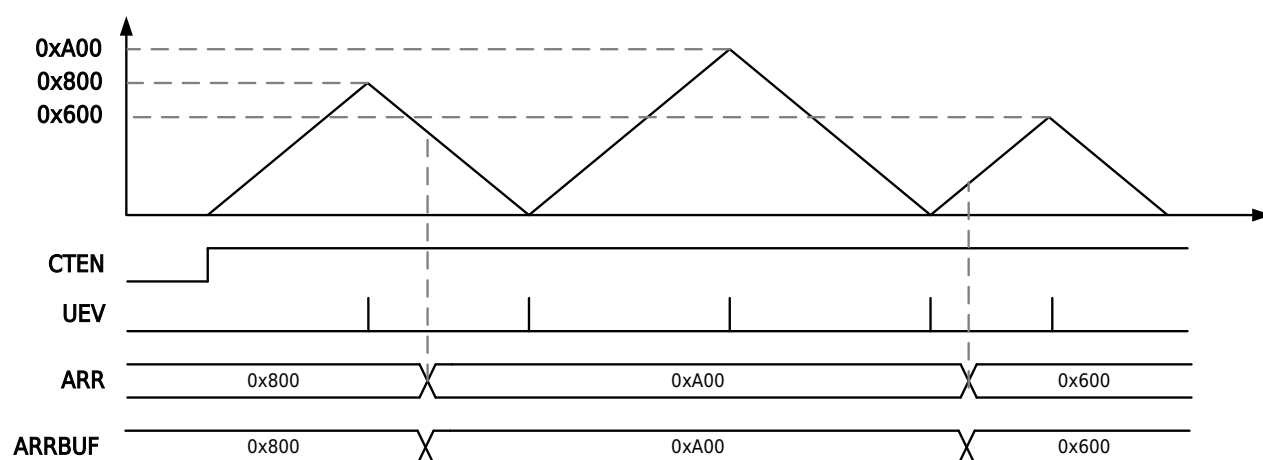


Figure 11-26 Mode 3 Triangular Wave Mode with Cache Invalid

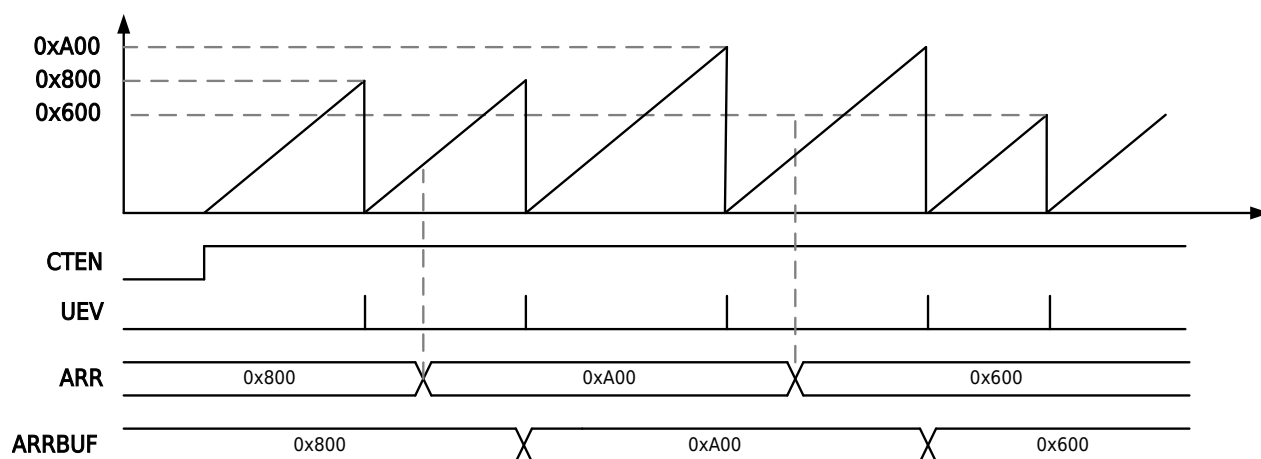


Figure 11-27 Mode 2 Sawtooth Wave Mode Up-Counting with Cache Enable

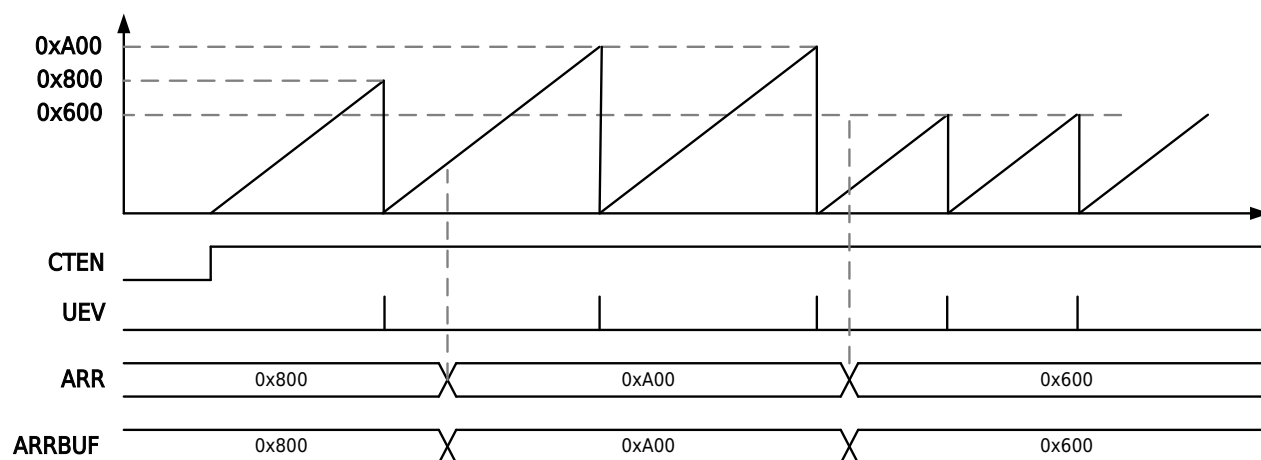


Figure 11-28 Mode 2 Sawtooth Wave Mode Up-Counting with Cache Invalid

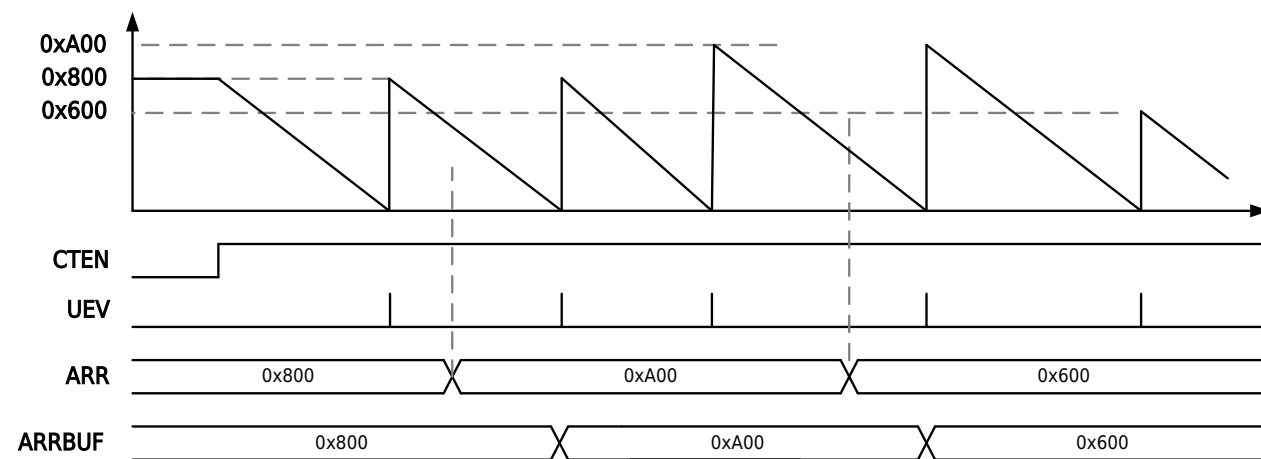


Figure 11-29 Mode 2 Sawtooth Wave Mode Down-Counting with Cache Enable

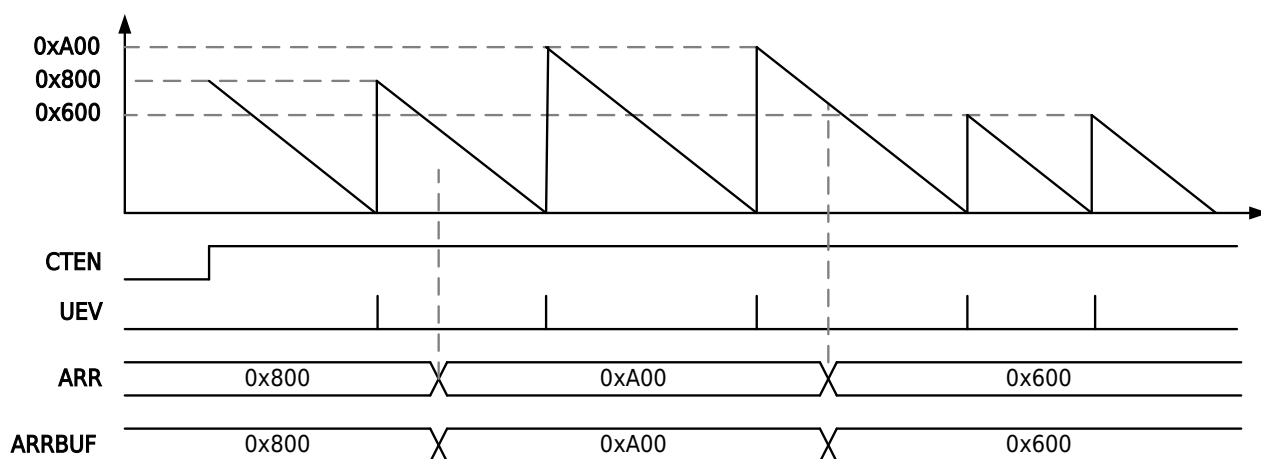


Figure 11-30 Mode 2 Sawtooth Wave Mode Down-Counting with Cache Invalid

**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.

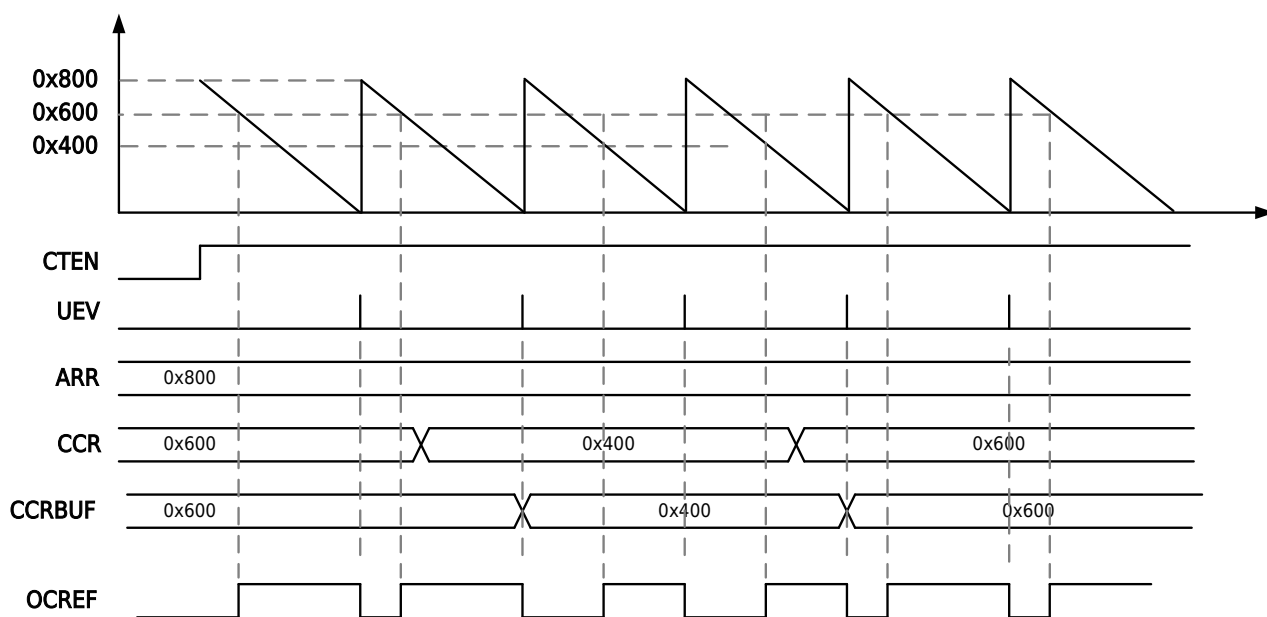


Figure 11-31 Mode2 Sawtooth Wave Down-Counting Mode with Compare Cache Enable

11.3.6.5 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.

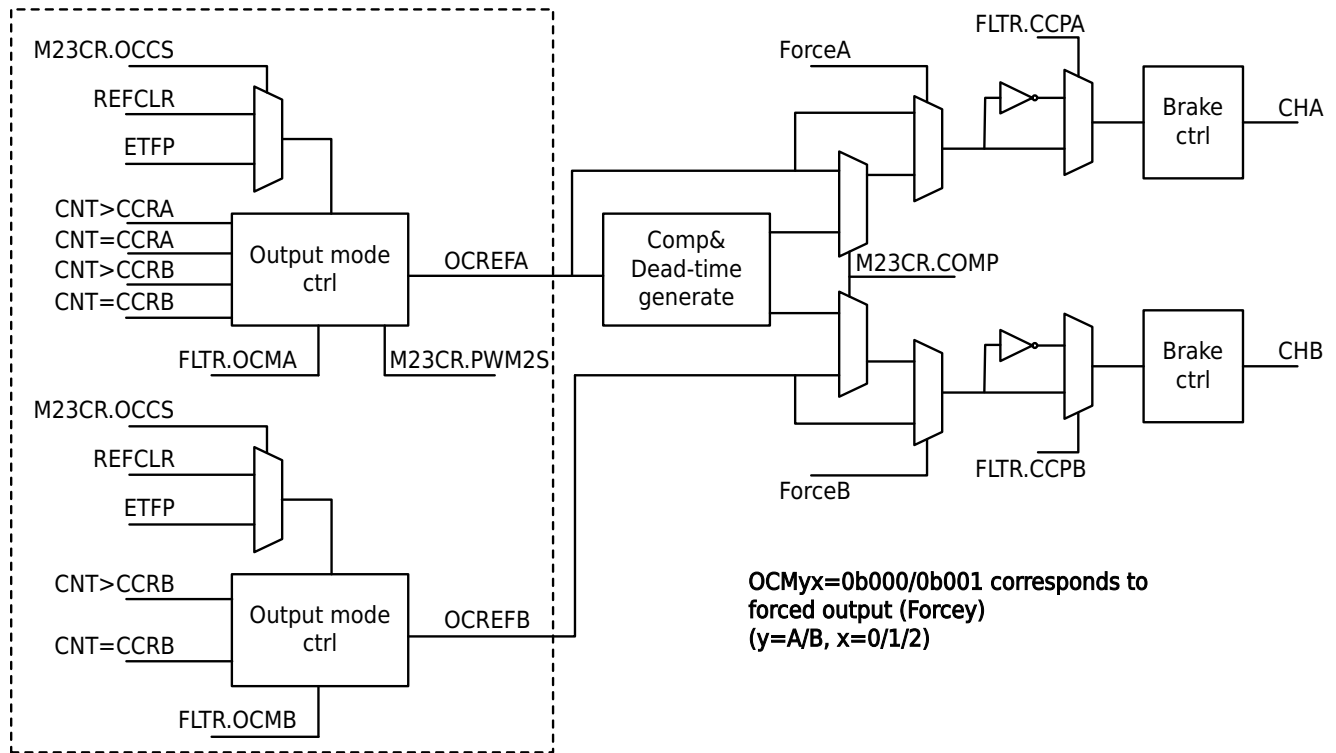


Figure 11-32 OCREF Output Block Diagram

OCREF output uses ATIM3_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 $CNT < CCR_{xy}$; during down-counting, the output is low level when $CNT > CCR_{xy}$.
 - ▶ Dual point comparison
 - ◆ For sawtooth wave up-counting, the output is low level when $CCR_{xA} < CNT \leq CCR_{xB}$
 - ◆ For sawtooth wave down-counting, the output is high level when $CCR_{xA} < CNT \leq CCR_{xB}$
 - ◆ For triangular wave up-counting, the output is high level when $CNT < CCR_{xA}$; for down-counting, the output is low level when $CNT > CCR_{xB}$
- 0b111: PWM mode 2
 - ▶ Single point comparison

During up-counting, the output is low level when $CNT < CCR_{xy}$; during down-counting, the output is high level when $CNT > CCR_{xy}$
 - ▶ Dual point comparison
 - ◆ For sawtooth wave up-counting, the output is high level when $CCR_{xA} \leq CNT < CCR_{xB}$
 - ◆ For sawtooth wave down-counting, the output is low level when $CCR_{xA} \leq CNT < CCR_{xB}$
 - ◆ For triangular wave up-counting, the output is low level when $CNT < CCR_{xA}$; for down-counting, the output is high level when $CNT > CCR_{xB}$

**Note**

Forced output has higher priority. When forced output is active, complementary output control becomes invalid.

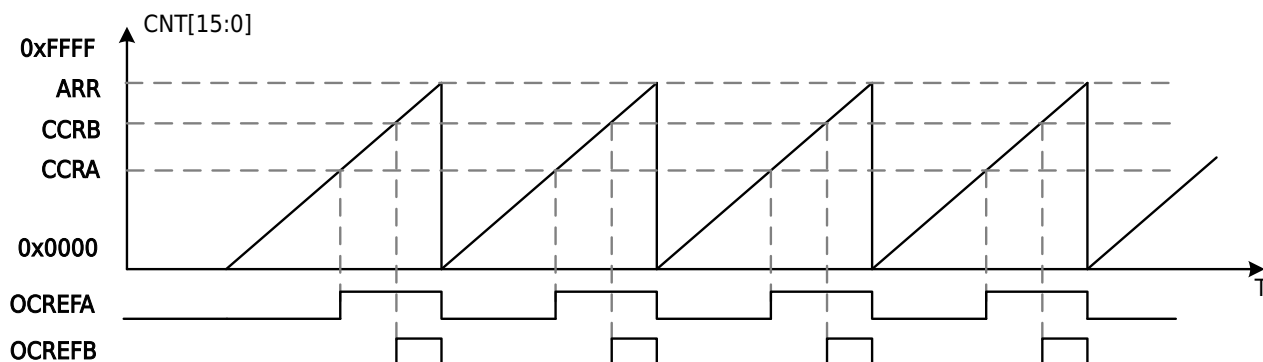


Figure 11-33 Sawtooth Counting Single Point Comparison OCREF Output Waveform (OCMyx=0b111)

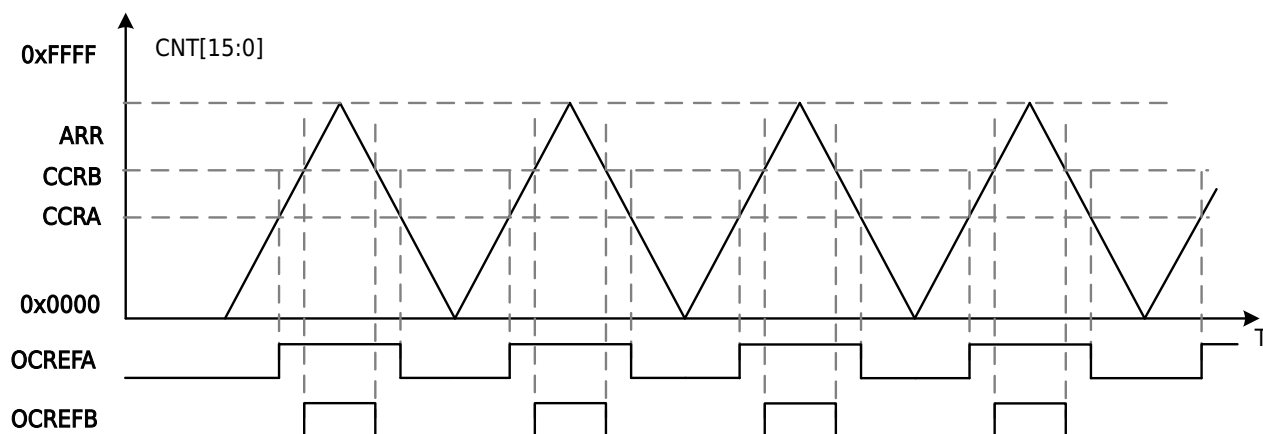


Figure 11-34 Triangular Counting Single Point Comparison OCREF Output Waveform (OCMyx=0b111)

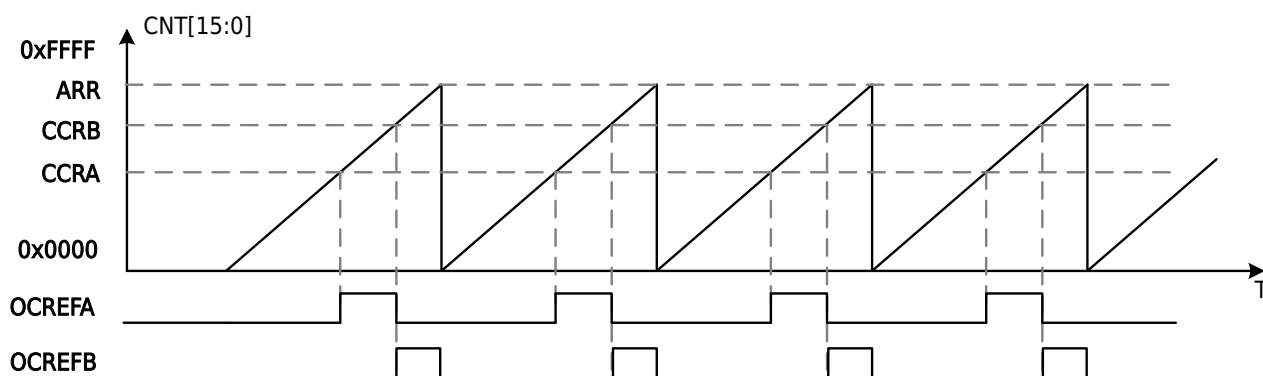


Figure 11-35 Sawtooth Counting Dual Point Comparison OCREF Output Waveform (OCMyx=0b111)

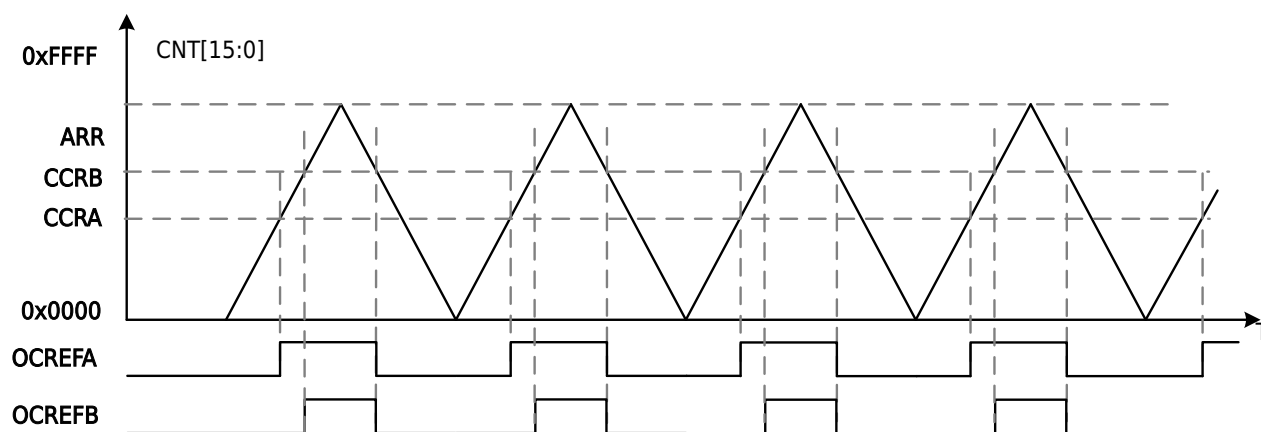


Figure 11-36 Triangular Counting Dual Point Comparison OCREF Output Waveform (OCMyx=0b111)

11.3.6.6 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.

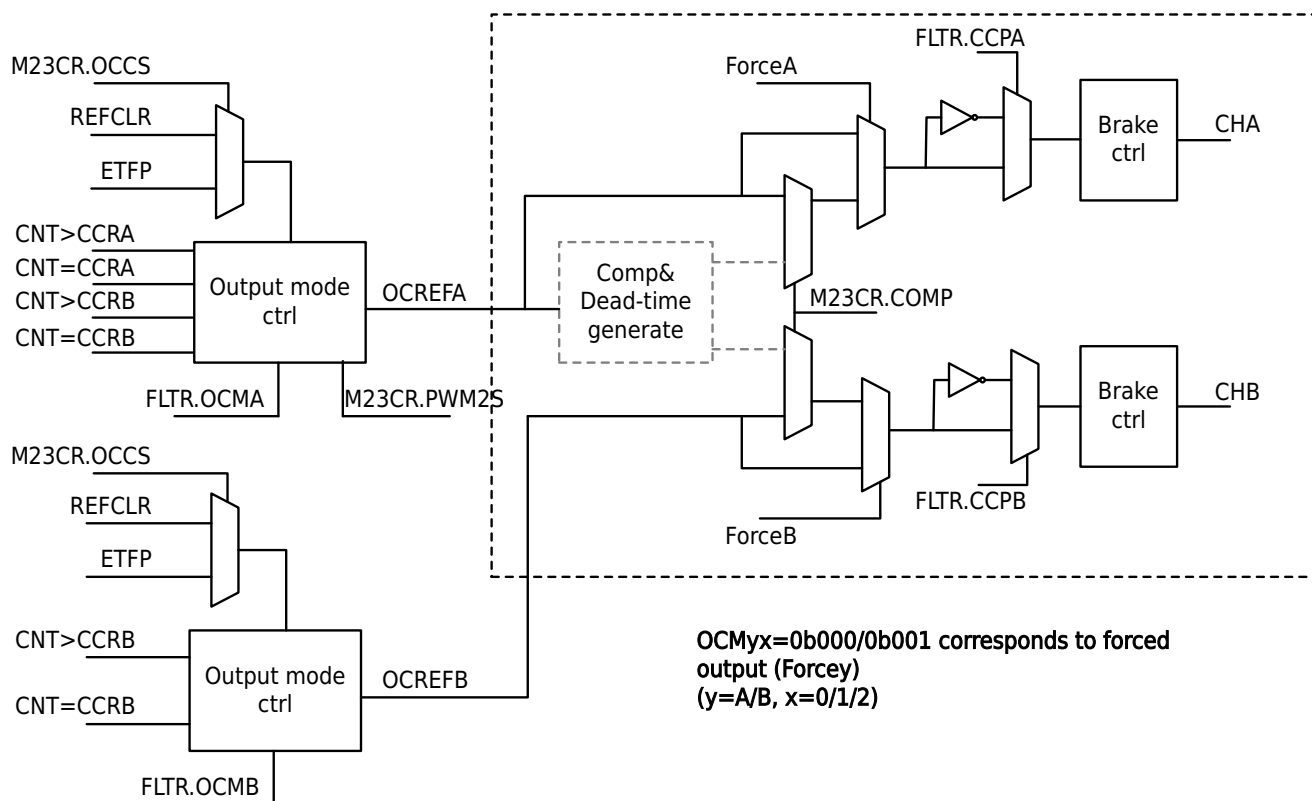


Figure 11-37 Independent PWM Output Block Diagram

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

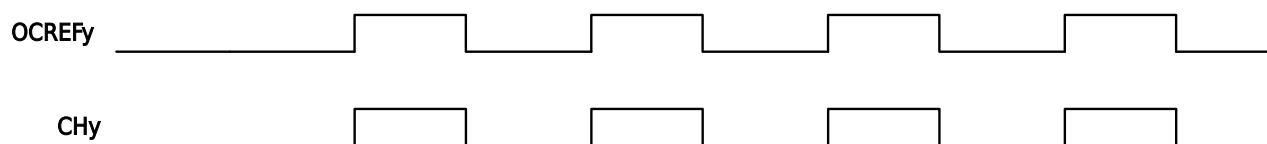


Figure 11-38 PWM Output Waveform When CCPy=0

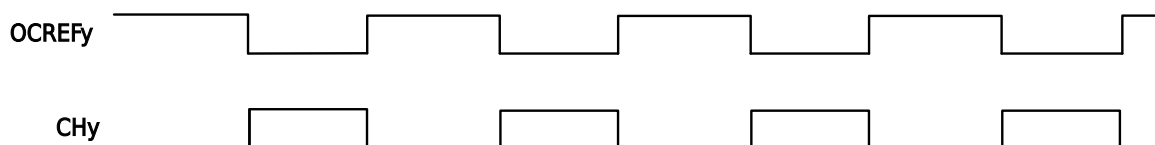


Figure 11-39 PWM Output Waveform When CCPy=1

11.3.6.7 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.

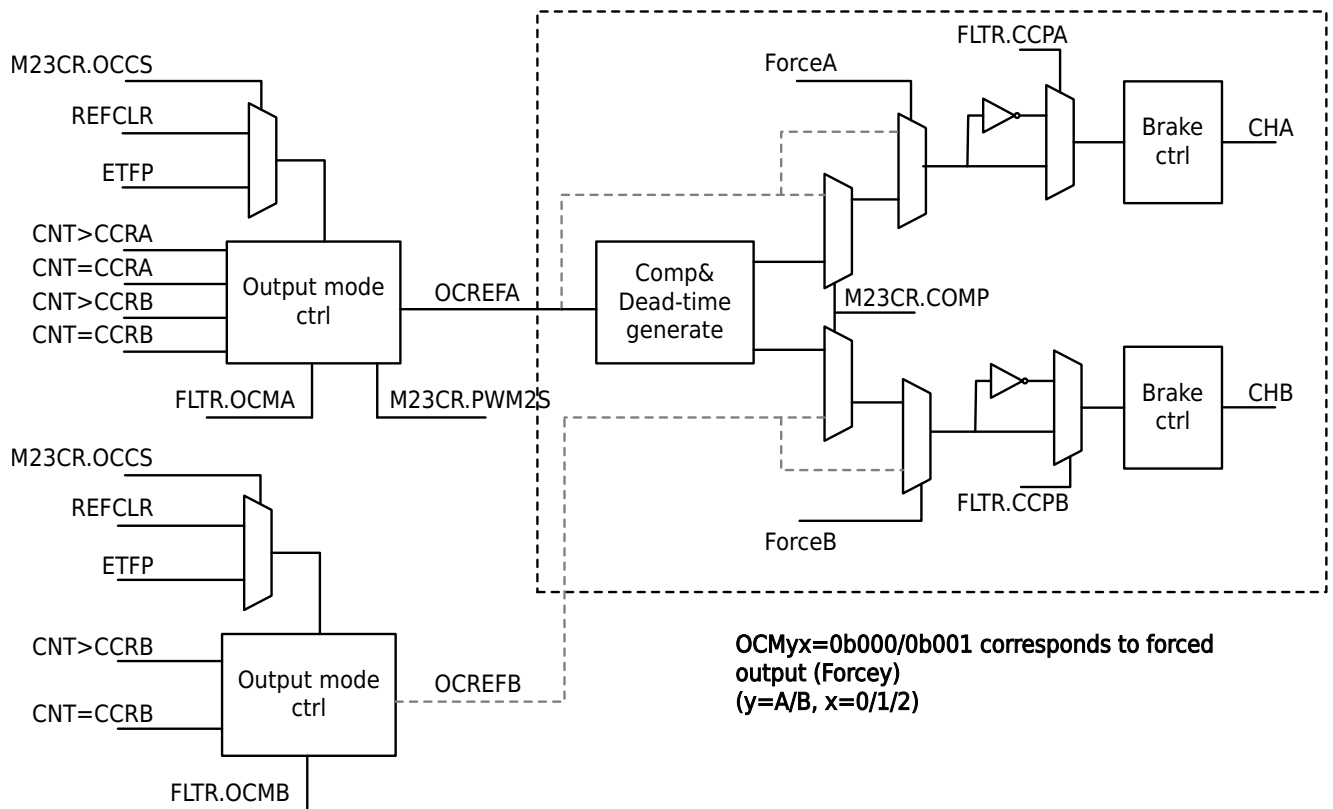


Figure 11-40 Complementary PWM Output Block Diagram

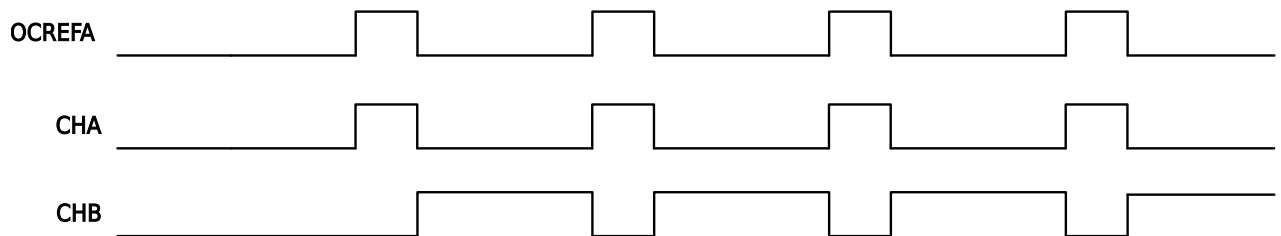


Figure 11-41 Complementary PWM Output Waveform

11.3.6.8 PWM Output with Dead Time

The dead time function can be set in complementary PWM output mode.

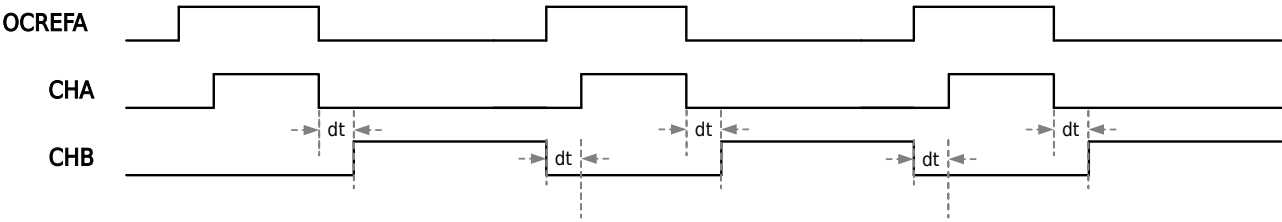


Figure 11-42 Complementary PWM Output Waveform (with Dead Time)

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

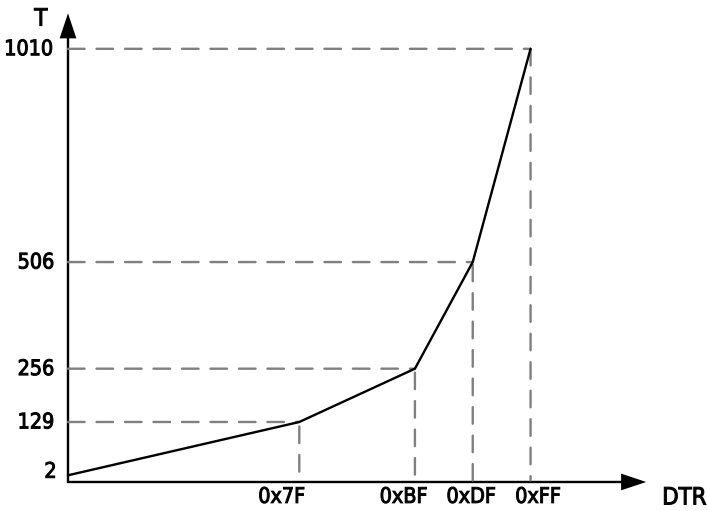


Figure 11-43 Dead Time

 **Note**

With a 100MHz clock, the dead time is adjusted from 2.56μs to 10.1μs.

11.3.6.9 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 ATIM3_M23CR.ONESHOT register to 1 enables single pulse mode. Thus, when the next update event (UEV) occurs, the counter will automatically stop.

Single pulse mode in sawtooth wave down-counting mode, the counter initial value should not be set to 0, and the count value in up-counting mode should not be set to a value greater than or equal to ARR.

Output pulse can only be correctly generated when the compare value differs from the counter initial value. Before starting (when the timer is waiting for a trigger), the following configurations must be made:

- For increment counting: $CNT < CCR_{xy} \leq ARR$ (special attention: $CCR_{xy} > 0$)
- For decrement counting: $CNT > CCR_{xy}$

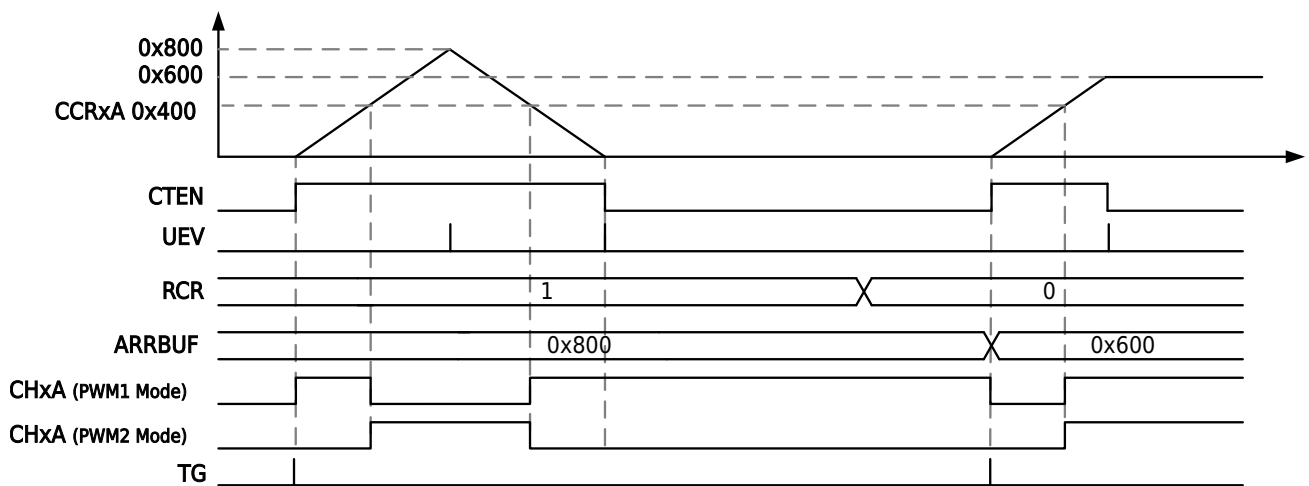


Figure 11-44 Triangular Wave Mode Single Pulse Counting

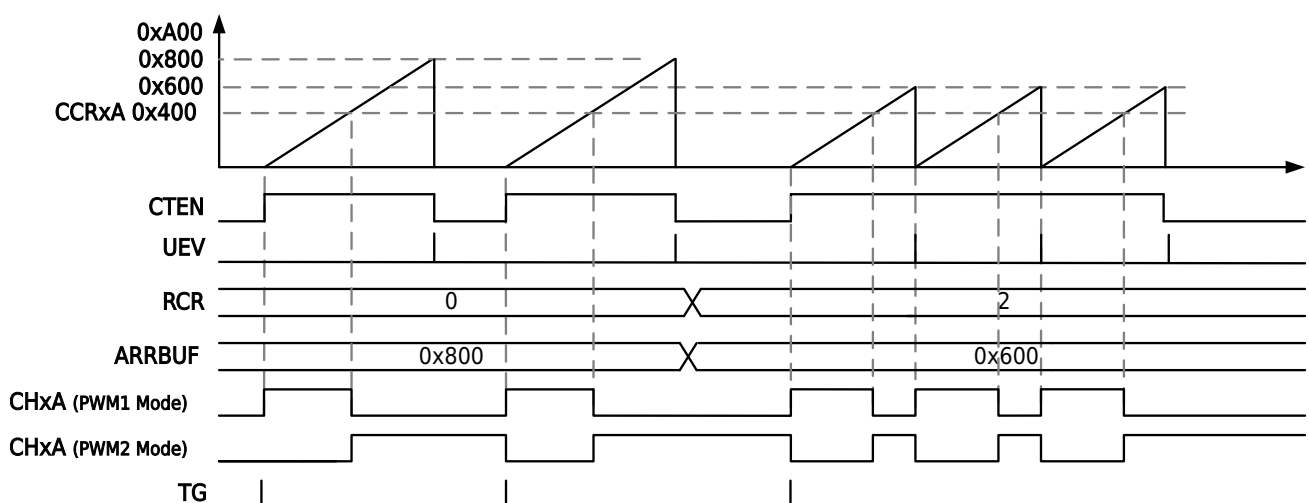


Figure 11-45 Sawtooth Up-Counting Single Pulse Mode

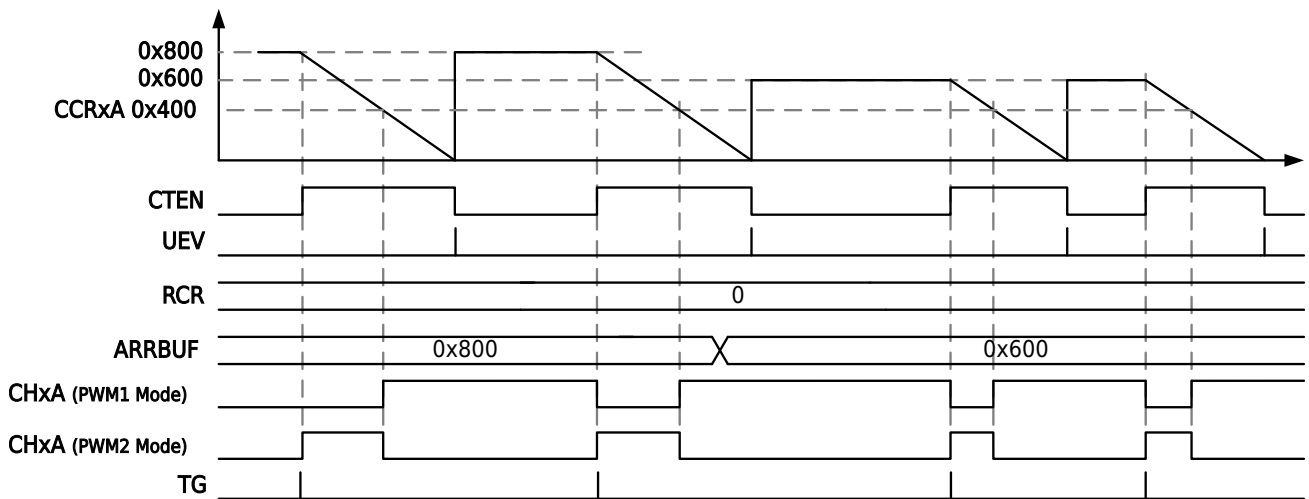


Figure 11-46 Sawtooth Down-Counting Single Pulse Mode

**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.

11.3.6.10 Compare Interrupt

Sawtooth wave compare match will set the corresponding flag in the IFR register to 1. If the interrupt enables CRCHx.CIEy (x=0/1/2; y=A/B), an interrupt will be triggered.

For triangular wave compare match, it can independently select rising count compare match, falling count compare match, or both.

For compare A match, when the count value equals CCRxA, it is uniformly controlled by M23CR.CIS:

- M23CR.CIS=0b00: no output on compare match
- M23CR.CIS=0b01: compare match during up-counting
- M23CR.CIS=0b10: compare match during down-counting
- M23CR.CIS=0b11: compare match during both up-counting and down-counting

For compare B match, when the count value equals CCRxB, each channel can be individually controlled by CRCHx.CISB:

- When CRCHx.CISB=0b00, channel x has no output on compare match.
- When CRCHx.CISB=0b01, channel x compare match during up-counting.
- When CRCHx.CISB=0b10, channel x compare match during down-counting.
- When CRCHx.CISB=0b11, channel x compare match during both up-counting and down-counting.

The independent control of channel B compare match enables more flexible ADC triggering. For more details, refer to the [Timer Trigger ADC](#) section in the documentation.

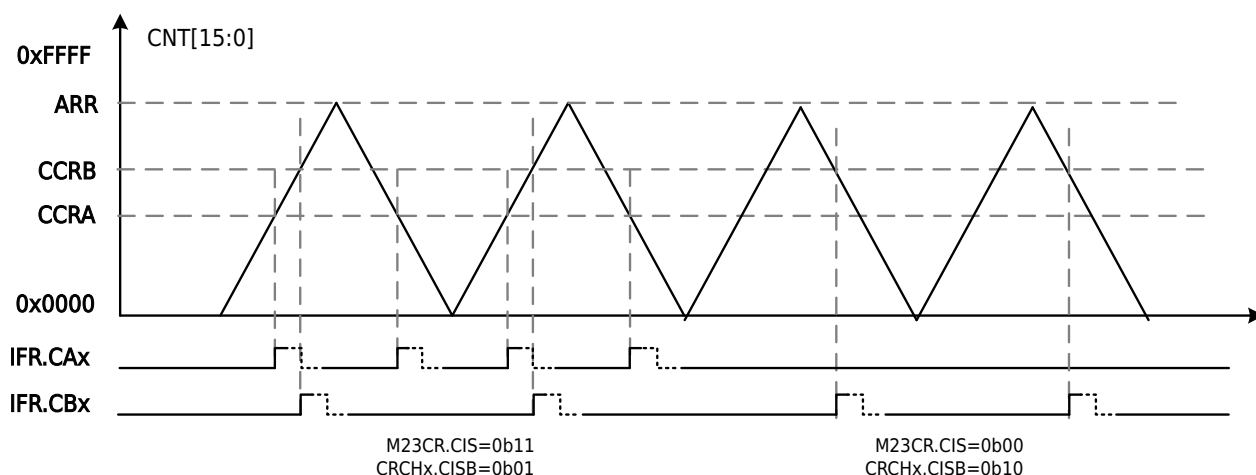


Figure 11-47 Compare Interrupt Diagram

11.3.6.11 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.

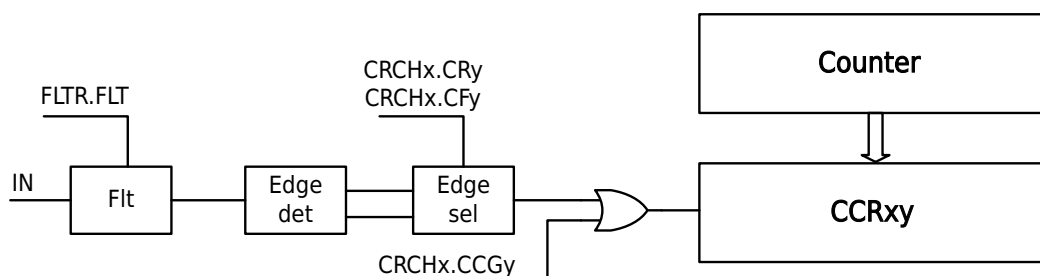


Figure 11-48 Capture Function Block Diagram

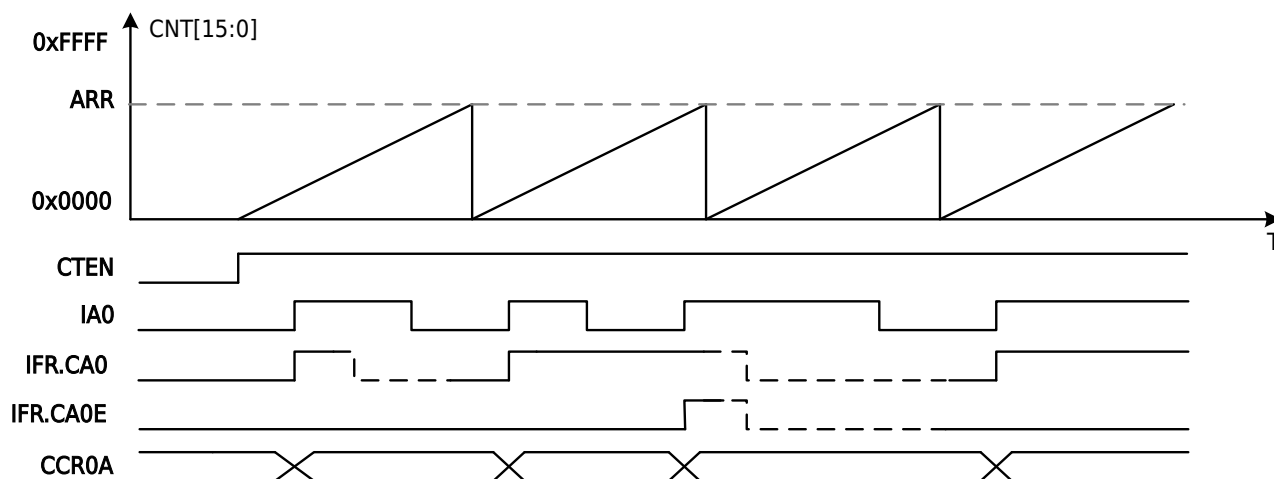


Figure 11-49 Capture Timing Diagram

CH0A Capture Input

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

Table 11-7 CH0A Capture Input Signal

MSCR.IA0S	ATIM3
0b0	CH0A
0b1	CH0A CH1A CH2A XOR Input

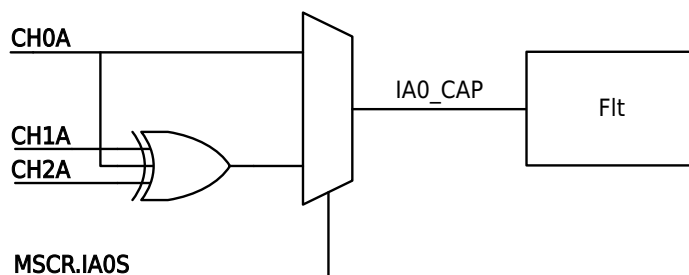


Figure 11-50 ATIM3 CH0A Capture Input Port Selection

CH0B Capture Input

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

MSCR.IB0S	ATIM3																		
0b0	CH0B																		
0b1	Internal trigger MSCR.TS select signal <table border="1"> <thead> <tr> <th>MSCR.TS Value</th><th>Channel B Capture Input</th></tr> </thead> <tbody> <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> </tbody> </table>	MSCR.TS Value	Channel B Capture Input	0b000	ETR filtered phase selection output signal, filter phase selection configurable	0b001	Internal interconnection ITR0	0b010	Internal interconnection ITR1	0b011	Internal interconnection ITR2	0b100	Internal interconnection ITR3	0b101	CH0A edge	0b110	CH0A filtered output signal, filter function configurable	0b111	Reserved
MSCR.TS Value	Channel B Capture Input																		
0b000	ETR filtered phase selection output signal, filter phase selection configurable																		
0b001	Internal interconnection ITR0																		
0b010	Internal interconnection ITR1																		
0b011	Internal interconnection ITR2																		
0b100	Internal interconnection ITR3																		
0b101	CH0A edge																		
0b110	CH0A filtered output signal, filter function configurable																		
0b111	Reserved																		

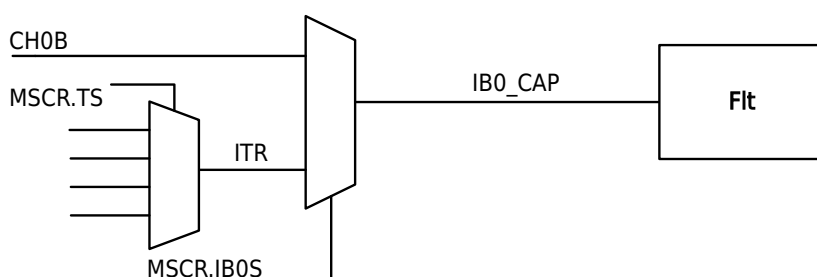


Figure 11-51 CH0B Capture Input Port Selection

11.3.7 Mode2/3 Slave Mode

The timer can be synchronized to an external trigger in several modes: reset mode, gated mode and triggered mode.

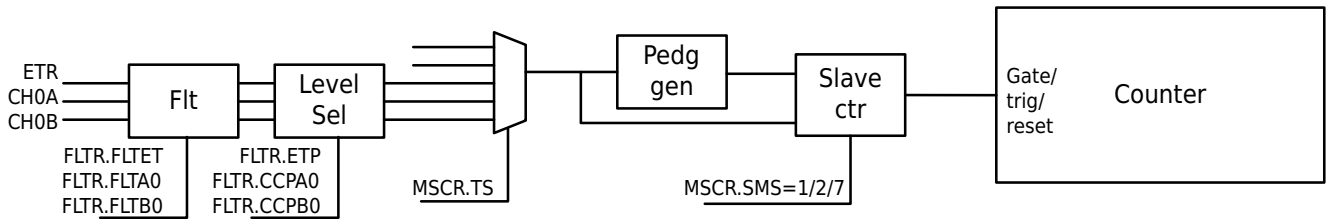


Figure 11-52 Mode2/3 Slave Mode Diagram

The slave mode function is selected via MSCR.SMS:

	Value	Description
MSCR.SMS	0b001	Slave mode function selection: reset mode
	0b010	Slave mode function selection: trigger mode
	0b111	Slave mode function selection: gating mode
MSCR.TS	0b000	The filtered phase selected signal ETP 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.FLTAE=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.

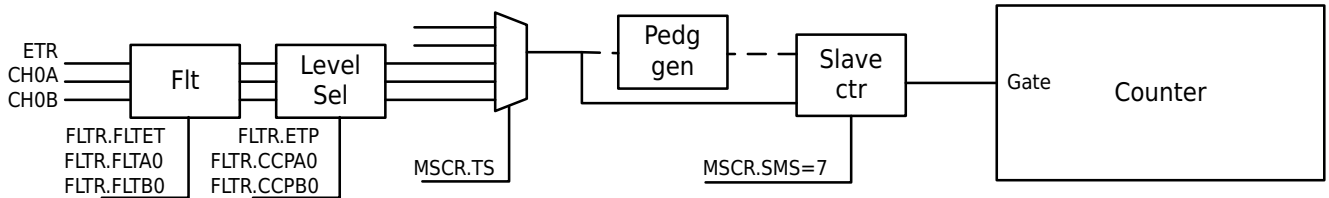


Figure 11-53 Mode2/3 Slave Mode Gating Function

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:

- 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.

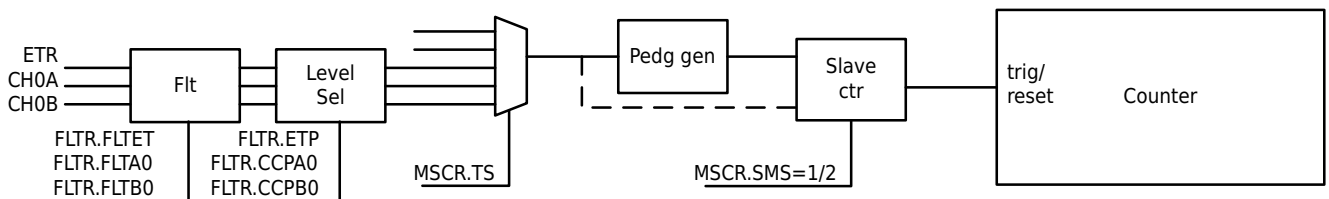


Figure 11-54 Mode2/3 Slave Mode Trigger and Reset Function

11.3.8 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.

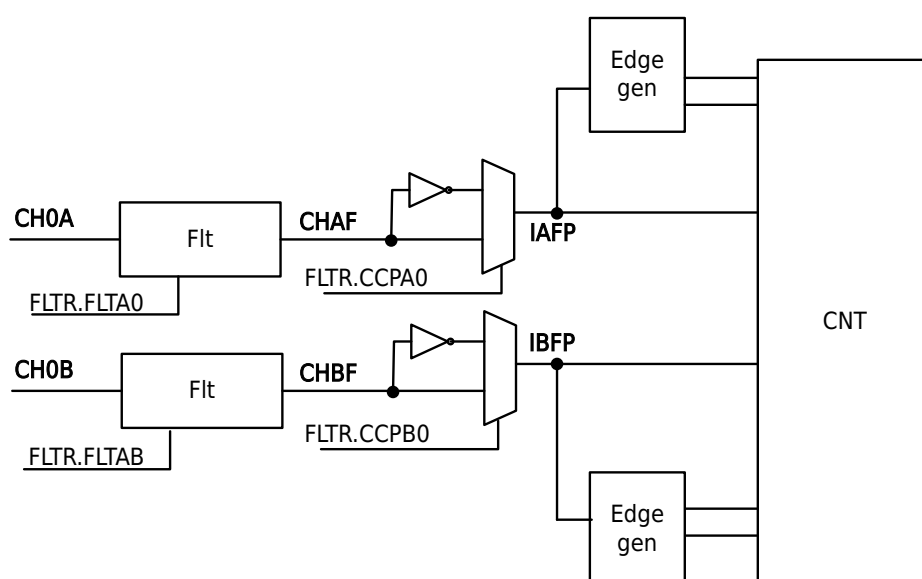
Mode 1 uses the edge count of CH0A. Mode 2 uses the edge count of CH0B. Mode 3 uses both the edges count of CH0A and CH0B.

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 Signal IBFP Edge Condition	
	IBFP	IAFP	Rising Edge	Falling Edge	Rising Edge	Falling Edge
MOD1	High level	-	Down-counting	Up-counting	-	-
	Low level	-	Up-counting	Down-counting	-	-
MOD2	-	High level	-	-	Up-counting	Down-counting
	-	Low level	-	-	Down-counting	Up-counting
MOD3	High level	-	Down-counting	Up-counting	-	-
	Low level	-	Up-counting	Down-counting	-	-
	-	High level	-	-	Up-counting	Down-counting
	-	Low level	-	-	Down-counting	Up-counting

CHAF/CHBF is the signal filtered by port CH0A/CH0B, and IAFP/IBFP is the signal after port filter phase selection.



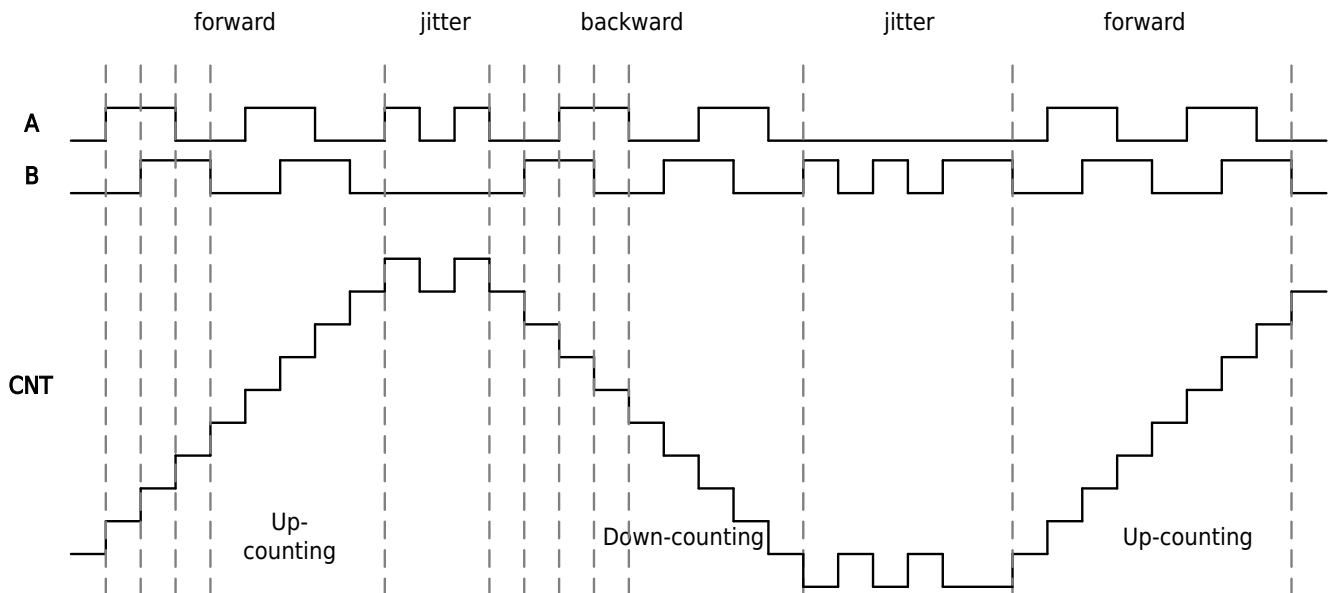


Figure 11-56 Quadrature Encoder Counting Mode3 Counting Diagram

11.3.9 Timer Triggers ADC

UEV Triggers ADC

When a update event (UEV) is generated and both ADTR.UEV and ADTR.ADT 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 match of CCR3 can also be configured to trigger the ADC (If channel 3 is used to trigger the ADC, no separate channel trigger enable needs to be set in the ADTR register; only enabling ADTR.ADT is required).

- Mode3: The comparison match conditions for channel 3 are controlled by the CRCH3.CIS register. Options include triggering on up-counting comparison match, down-counting comparison match, or both up-counting and down-counting comparison matches.
- Mode2: No need to configure CRCH3.CIS.

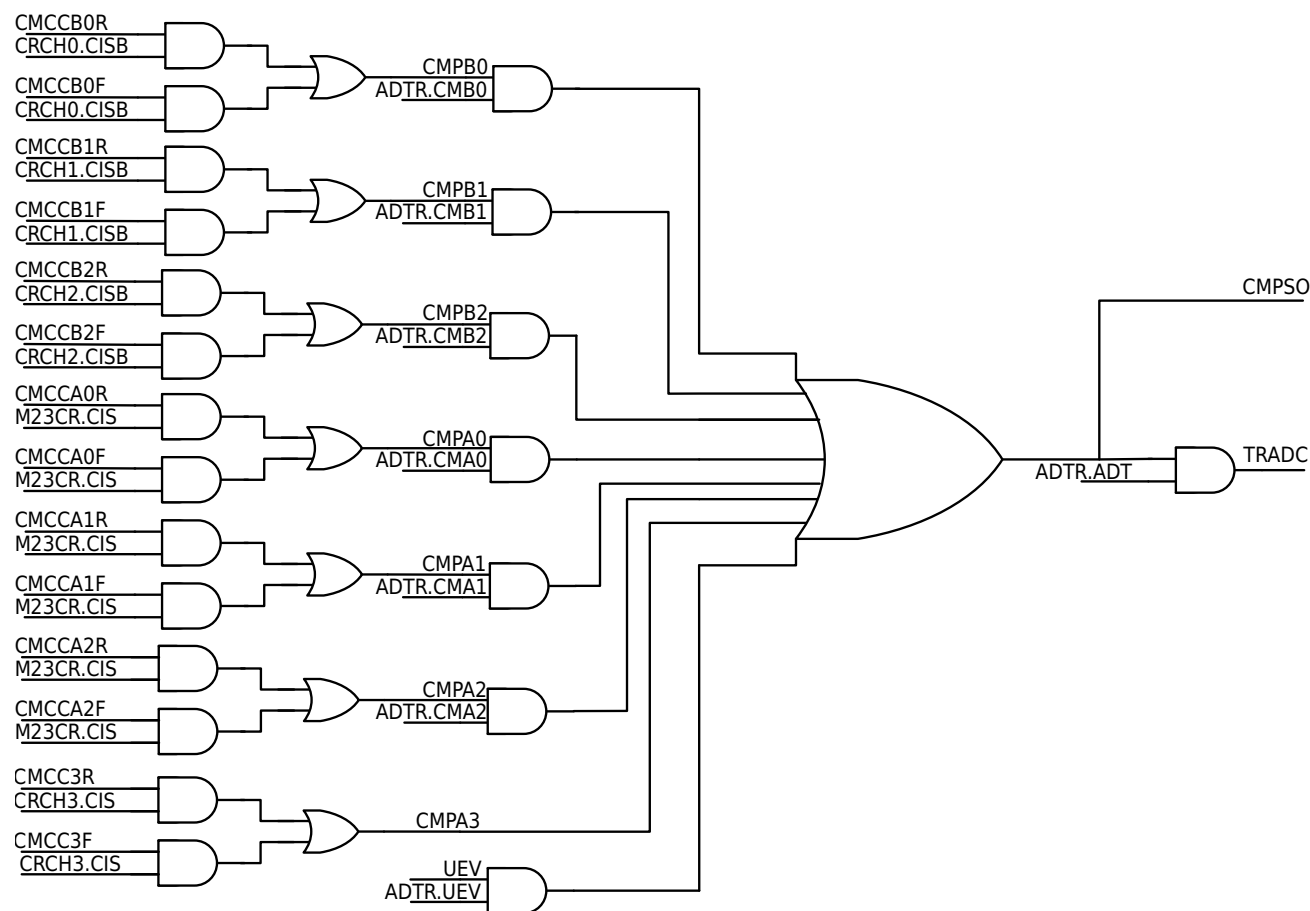


Figure 11-57 Timer Triggers ADC

11.3.10 Brake Control

The external BK port input can control the brake, and the system fail can control the brake function. The software brake function can be implemented through M23CR.BG to control the output port to the set state.

11.3.11 Timer Interconnection

The TRGO output signal can be connected to the ITR signal of other timers. The connection relationship is as follows:

Timer	ITR0	ITR1	ITR2	ITR3
ATIM3	-	-	GTIM_OV	-

11.3.12 Input Interconnection

The CH0B input of ATIM3 can be directly input from a port or selected via the port function register GPIO_CR4 to connect to other modules or ports.

When ATIM3_CHy=0x0, the capture input is the port input selected by GPIOx_AFRL. When ATIM3_CHy=0x1-0x3, it connects to the input or output of other modules.

ATIM3	CH0B
0b000	GPIOx_AFRL
0b001	LPUART0_RXD

ATIM3	CH0B
0b010	LPUART1_RXD
0b011	LVD_OUT

11.4 Operation Example

11.4.1 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.UI;
 - Step6.** Enable reload mode M0CR.MD;
 - Step7.** Start timer M0CR.CEN.
- **BUZZER Output Control**
 - Step1.** Set appropriate ARR value based on output frequency;
 - Step2.** Configure timer as reload mode, refer to [Reload Timer Settings](#);
 - Step3.** Enable output enable DTR.MO;
 - Step4.** Start another timer to control M0CR.TOG for implementing interval frequency output.

11.4.2 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.

11.4.3 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.MO;
 - Step11.** Enable timer M23CR.CTEN.
- **Center-Aligned Complementary PWM Output Setting**
 - Step1.** Set mode M23CR.MODE=3;
 - Step2.** Configure complementary output M23CR.COMP;
 - Step3.** Set PWM period value ARR;
 - Step4.** Set counter initial value (must be smaller than period value);
 - Step5.** Set PWM compare value CCRxA (CCRxB does not need to be set as PWM output is independent of this register);
 - Step6.** Set PWM output compare modes FLTR.OCMAx and FLTR.OCMBx to 6 or 7;
 - Step7.** Clear relevant interrupt flags;
 - Step8.** Enable corresponding interrupts;
 - Step9.** Configure output polarities FLTR.CCPAx and FLTR.CCPBx;
 - Step10.** Enable output DTR.MO;
 - 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 enable DTR.DT;
- Step12.** Set dead time duration DTR.DTR;
- Step13.** Enable output DTR.MO;
- 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.

11.4.4 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.ADT

11.5 Register Description

11.5.1 Register List

ATIM3 base address: 0x40005800

Table 11-14 Mode0 Timer Register Offset

Offset	Register	Description	Width
0x000	ATIM3_ARR	16-Bit Mode Reload Register/Period	32
0x004	ATIM3_CNT	16-Bit Mode Count Register	32
0x008	ATIM3_CNT32	32-Bit Mode Count Register  Note ATIM3_CNT32 is the remapping register for ATIM3_ARR and ATIM3_CNT. In order to facilitate the 32-bit timer write operation.	32
0x00C	ATIM3_M0CR	Mode0 Control Register	32
0x010	ATIM3_IFR	Interrupt Flag Register	32
0x014	ATIM3_ICLR	Interrupt Flag Clear Register	32
0x030	ATIM3_DTR	Dead Time Register	32

Table 11-15 Mode1 Pulse Width Measurement (PWM) Register Offset

Offset	Register	Description	Width
0x004	ATIM3_CNT	16-Bit Mode Count Register	32
0x00C	ATIM3_M1CR	Mode1 Control Register	32
0x010	ATIM3_IFR	Interrupt Flag Register	32
0x014	ATIM3_ICLR	Interrupt Flag Clear Register	32
0x018	ATIM3_MSCR	Master-Slave Mode Control Register	32
0x01C	ATIM3_FLTR	Input Filter Register	32
0x024	ATIM3_CR0	Control Register	32
0x03C	ATIM3_CCR0A	Compare Capture Register	32

Table 11-16 Mode2/3 Compare-Capture Mode Register Offset

Offset	Register	Description	Width
0x000	ATIM3_ARR	16-Bit Mode Reload Register/Period	32
0x004	ATIM3_CNT	16-Bit Mode Count Register	32
0x00C	ATIM3_M23CR	Mode2/3 Control Register	32

Offset	Register	Description	Width
0x010	ATIM3_IFR	Interrupt Flag Register	32
0x014	ATIM3_ICLR	Interrupt Flag Clear Register	32
0x018	ATIM3_MSCR	Master-Slave Mode Control Register	32
0x01C	ATIM3_FLTR	Output Control/Input Filter Register	32
0x020	ATIM3_ADTR	ADC Trigger Control Register	32
0x024	ATIM3_CRCH0	Channel 0 Control Register	32
0x028	ATIM3_CRCH1	Channel 1 Control Register	32
0x02C	ATIM3_CRCH2	Channel 2 Control Register	32
0x030	ATIM3_DTR	Dead Time Register	32
0x034	ATIM3_RCR	Repeat Cycle Setting Value Register	32
0x03C	ATIM3_CCR0A	Channel 0A Compare/Capture Register	32
0x040	ATIM3_CCR0B	Channel 0B Compare/Capture Register	32
0x044	ATIM3_CCR1A	Channel 1A Compare/Capture Register	32
0x048	ATIM3_CCR1B	Channel 1B Compare/Capture Register	32
0x04C	ATIM3_CCR2A	Channel 2A Compare/Capture Register	32
0x050	ATIM3_CCR2B	Channel 2B Compare/Capture Register	32
0x054	ATIM3_CCR3	Channel 3 Compare Register	32
0x058	ATIM3_CRCH3	Channel 3 Control Register	32

11.5.2 Mode0 Count Timer Register

11.5.2.1 16-Bit Mode Reload Register (ATIM3_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

11.5.2.2 16-Bit Mode Count Register (ATIM3_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

11.5.2.3 32-Bit Mode Count Register (ATIM3_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

11.5.2.4 Mode0 Control Register (ATIM3_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	UI	GATEP	GATE	Reserved	PRS			TOG	CT	MD	CEN	
Access																					RW		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	UI	Interrupt enable control ● 0b0: Disable ● 0b1: Enable	RW
9	GATEP	Reserved	RW
8	GATE	Read as 0, Write as 0	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	TOG	Toggle output enable in mode 0 ● 0b0: Toggle output disabled, CHA and CHB outputs remain low level ● 0b1: Toggle output enabled	RW
2	CT	Count clock selection ● 0b0: Internal count clock TCLK ● 0b1: External count clock ETR	RW

Bit	Name	Functional Description	Access
1	MD	Mode selection 32 timing/16 timing mode selection ● 0b0: 32-bit free counting ● 0b1: 16-bit reload counting	RW
0	CEN	Timer enable ● 0b0: Disable ● 0b1: Enable	RW

11.5.2.5 Interrupt Flag Register (ATIM3_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

11.5.2.6 Interrupt Flag Clear Register (ATIM3_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

11.5.2.7 Dead Time Register (ATIM3_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																				MO		Reserved									
Access																					RW											

Bit	Name	Functional Description	Access
31:13	Reserved	-	-
12	MO	Toggle output enable 0b0: Toggle output in input status 0b1: Toggle port in output status	RW
11:0	Reserved	-	-

11.5.3 Mode1 Pulse Width Measurement PWC Register

11.5.3.1 16-Bit Mode Count Register (ATIM3_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

11.5.3.2 Mode1 Control Register (ATIM3_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	UI		EDG2ND	EDG1ST	Reserved	PRS		Reserved	CT	Reserved	CTEN
Access																		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	UI	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		RW	
		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
7	Reserved	-	-		

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

11.5.3.3 Interrupt Flag Register (ATIM3_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		
Access																													RO			RO	UI	

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

11.5.3.4 Interrupt Flag Clear Register (ATIM3_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					

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

11.5.3.5 Master-Slave Mode Control Register (ATIM3_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  Note Setting to 1 will cause input changes upon any port transition.	RW
10:8	Reserved	-	-
7:5	TS	Trigger selection 0b000: Signal ETRFP 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	-	-

11.5.3.6 Input Filter Register (ATIM3_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	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	RW

11.5.3.7 Control Register (ATIM3_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	-	-

11.5.3.8 Compare Capture Register (ATIM3_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

11.5.4 Mode2/3 Compare Capture Mode Register

11.5.4.1 16-Bit Mode Reload Register (ATIM3_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

11.5.4.2 16-Bit Mode Count Register (ATIM3_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																																

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

11.5.4.3 Mode2/3 Control Register (ATIM3_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	UND	OV	DIR	BG	UG	TG	OCC	CIS	BI	TI	TD	URS	OCCS	CSG	ONESHOT	MODE	UD	UI	CFG	CRG	BUFP	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	RW	RW	RW	RW	RW	RW		

Bit	Name	Functional Description	Access
31:30	Reserved	-	-
29	UND	Underflow interrupt enable 0b0: Disable 0b1: Enable	RW
28	OV	Overflow interrupt enable 0b0: Disable 0b1: Enable	RW
27	DIR	Count direction Write only in sawtooth wave mode. Read only in other modes, write disable 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

Bit	Name	Functional Description	Access
23	OCC	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	BI	Brake interrupt enable 0b0: Disable 0b1: Enable	RW
19	TI	Trigger interrupt enable 0b0: Disable 0b1: Enable	RW
18	TD	<i>Reserved</i>	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: Setting prohibited 0b1: Signal after ETR port filtered phase selection When OCCE is enabled, OCREF_CLR can clear the OCREF compare output signal (valid when OCM _{yx} >1), and the compare output resumes after the next UEV event.	RW
15	CSG	<i>Reserved</i>	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
11	UD	<i>Reserved</i>	RW
10	UI	Update interrupt enable 0b0: Disable 0b1: Enable	RW
9	CFG	<i>Reserved</i>	RW

Bit	Name	Functional Description	Access
8	CRG	<i>Reserved</i>	RW
7	BUFP	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 (default value)  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
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

11.5.4.4 Interrupt Flag Register (ATIM3_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														CA3		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		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO		RO	

Bit	Name	Functional Description	Access
31:19	Reserved	-	-
18	CA3	Channel CH3 compare match flag 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 0b0: No data loss 0b1: Data loss	RO
12	CB1E	Channel CH1B capture data loss flag 0b0: No data loss 0b1: Data loss	RO
11	CB0E	Channel CH0B capture data loss flag 0b0: No data loss 0b1: Data loss	RO
10	CA2E	Channel CH2A capture data loss flag 0b0: No data loss 0b1: Data loss	RO
9	CA1E	Channel CH1A capture data loss flag 0b0: No data loss 0b1: Data loss	RO

Bit	Name	Functional Description	Access
8	CA0E	Channel CH0A capture data loss flag ● 0b0: No data loss ● 0b1: Data loss	RO
7	CB2	Channel CH2B capture/compare match flag ● 0b0: Not occurred ● 0b1: Occurred	RO
6	CB1	Channel CH1B capture/compare match flag ● 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 ● 0b0: Not occurred ● 0b1: Occurred	RO
3	CA1	Channel CH1A capture/compare match flag ● 0b0: Not occurred ● 0b1: Occurred	RO
2	CA0	Channel CH0A capture/compare match flag ● 0b0: Not occurred ● 0b1: Occurred	RO
1	<i>Reserved</i>	-	-
0	UI	Event update interrupt flag ● 0b0: Not occurred ● 0b1: Occurred	RO

11.5.4.5 Interrupt Flag Clear Register (ATIM3_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														CA3	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
31:19	Reserved	-	-
18	CA3	Channel CH3 compare match flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	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.	WO
12	CB1E	Channel CH1B capture data loss flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	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.	WO
9	CA1E	Channel CH1A capture data loss flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	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
7	CB2	Channel CH2B capture/compare match flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
6	CB1	Channel CH1B capture/compare match flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	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

Bit	Name	Functional Description	Access
4	CA2	Channel CH2A capture/compare match flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO
3	CA1	Channel CH1A capture/compare match flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	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	<i>Reserved</i>	-	-
0	UI	Event update interrupt flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO

11.5.4.6 Master-Slave Mode Control Register (ATIM3_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  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 ETRFP 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

Bit	Name	Functional Description	Access
4	MSM	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.	RW
3	CCDS	<i>Reserved</i>	RW
2:0	MMS	Master mode input selection For internal interconnection, connected to ITRx of other timers ● 0b000: Software update UG, write M23CR.UG; software update TG, write M23CR.TG ● 0b001: Timer enable CTEN ● 0b010: Timer event update UEV ● 0b011: Compare match select output CMPSO ● 0b100: Timer compare parameter output OCREFOA ● 0b101: Timer compare parameter output OCREF1A ● 0b110: Timer compare parameter output OCREF2A ● 0b111: Timer compare parameter output OCREFOB	RW

11.5.4.7 Output Control/Input Filter Register (ATIM3_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

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 $CCRxA < CNT \leq CCRxB$ - For sawtooth wave down-counting, the output is high level when $CCRxA < CNT \leq CCRxB$	RW

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 $CCRxA \leq CNT < CCRxB$ ◆ For sawtooth wave down-counting, the output is low level when $CCRxA \leq CNT < CCRxB$ ◆ For triangular wave up-counting, the output is low level when $CNT < CCRxA$; for down-counting, the output is high level when $CNT > CCRxB$	
	FLTBO	Capture function: CH0B input channel filter setup Refer to FLTBK setting	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

11.5.4.8 ADC Trigger Control Register (ATIM3_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																																																							
Access																									RW		ADT		RW		CMB2		RW		CMB1		RW		CMB0		RW		CMA2		RW		CMA1		RW		CMA0		RW		UEV	

Bit	Name	Functional Description	Access
31:8	Reserved	-	-
7	ADT	Enable ADC trigger global control 0b0: Disable 0b1: Enable	RW
6	CMB2	Channel 2B compare match trigger ADC enable 0b0: Disable 0b1: Enable	RW
5	CMB1	Channel 1B compare match trigger ADC enable 0b0: Disable 0b1: Enable	RW
4	CMB0	Channel 0B compare match trigger ADC enable 0b0: Disable 0b1: Enable	RW
3	CMA2	Channel 2A compare match trigger ADC enable 0b0: Disable 0b1: Enable	RW
2	CMA1	Channel 1A compare match trigger ADC enable 0b0: Disable 0b1: Enable	RW
1	CMA0	Channel 0A compare match trigger ADC enable 0b0: Disable 0b1: Enable	RW
0	UEV	Event update trigger ADC enable 0b0: Disable 0b1: Enable	RW

11.5.4.9 Channel x Control Register (ATIM3_CRCHx) (x=0/1/2)

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

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																																																
Access																	WO	CCGB	WO	CCGA	RW	CISB	RW	CDEB	RW	CDEA	RW	CIEB	RW	CIEA	RW	BUFEB	RW	BUFEA	RW	CSB	RW	CSA	RW	CFB_CRB/BKSB	RW	CFA_CRA/BKSA							

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	Reserved	RW
10	CDEA	Reserved	RW
9	CIEB	B capture compare trigger interrupt enable ● 0b0: Disable ● 0b1: Enable	RW
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

Bit	Name	Functional Description	Access
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	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	RW

11.5.4.10 Dead Time Register (ATIM3_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	SAFE	MO	AO	BK	DT	BKSEL	DTR						
Access																			RW	RW	RW	RW	RW	RW	RW							RW

Bit	Name	Functional Description	Access																				
31:15	Reserved	-	-																				
14	VCE	Reserved	RW																				
13	SAFE	Safety brake enable (osc fail,brown down,lockup) 0b0: Disable 0b1: Enable	RW																				
12	MO	PWM output enable 0b0: Disable 0b1: Enable	RW																				
11	AO	PWM output automatically enable 0b0: Disable 0b1: Enable	RW																				
10	BK	Port brake enable 0b0: Disable 0b1: Enable	RW																				
9	DT	Dead time control enable 0b0: Disable 0b1: Enable	RW																				
8	BKSEL	Reserved	RW																				
7:0	DTR	Dead time <table border="1"> <thead> <tr> <th>DTR</th><th>T</th><th>dt</th><th>step</th></tr> </thead> <tbody> <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> </tbody> </table>	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	RW
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																				

11.5.4.11 Repeat Cycle Setting Value Register (ATIM3_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																										RCR					
																									UD	OV						
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

11.5.4.12 Channel x Compare Capture Register (ATIM3_CCRxy) (x=0/1/2, y=A/B)

Offset: 0x03C (CCR0A), 0x040 (CCR0B), 0x044 (CCR1A), 0x048 (CCR1B), 0x04C (CCR2A), 0x050 (CCR2B)

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

11.5.4.13 Channel 3 Compare Register (ATIM3_CCR3)

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																CCR3															
Access																	RW															

Bit	Name	Functional Description	Access
31:16	Reserved	-	-
15:0	CCR3	Compare register Comparison with cache function	RW

11.5.4.14 Channel 3 Control Register (ATIM3_CRCH3)

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																										C3		CIS		CD		CI		BUF	
Access																									RW		RW		RW		RW		RW			

Bit	Name	Functional Description	Access
31:6	Reserved	-	-
5	C3	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

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

12 CTRIM

12.1 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.

12.2 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 30μs to 65536s
- Timing mode: 4 clock signals available for selection
- Timing mode: Toggle output

12.3 Functional Description

12.3.1 RCH Clock Calibration Mode

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



Note

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

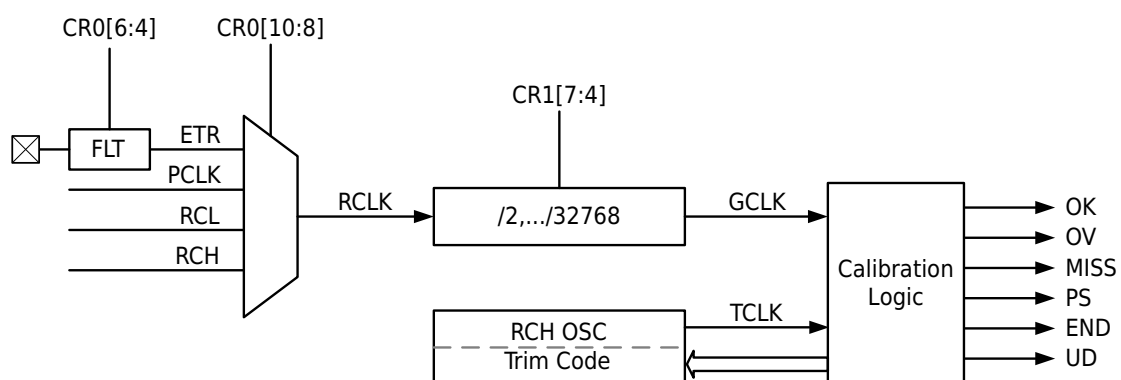


Figure 12-1 RCH Clock Calibration Operational Block Diagram

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

is down-counting, increase the TrimCode of RCH by CTRIM_CR0.STEP; If counting direction is up-counting, decrease the TrimCode of RCH by CTRIM_CR0.STEP. The calibration logic automatically reduces the value of CTRIM_CR0.STEP to achieve gradual convergence of the output frequency of RCH toward the target value.

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

12.3.2 RCL Clock Calibration Mode

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

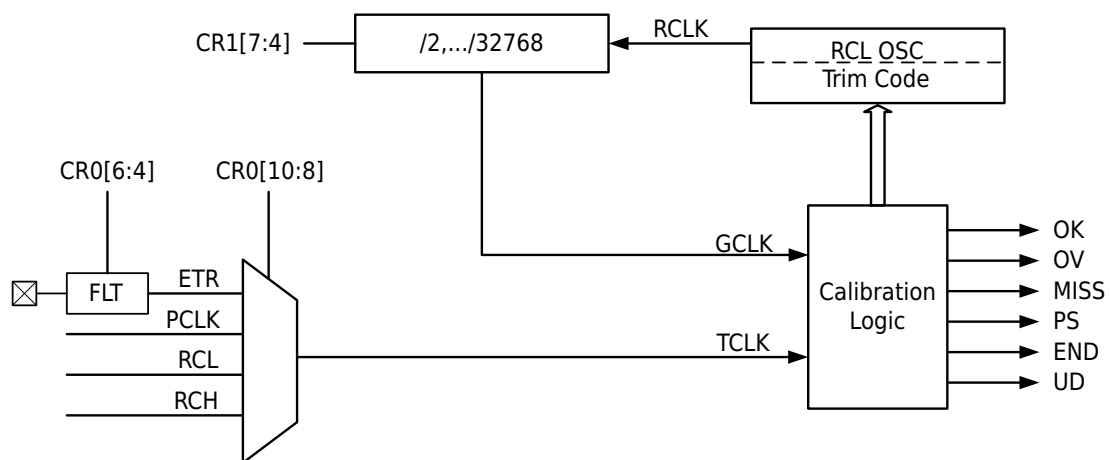


Figure 12-2 RCL Clock Calibration Operational Block Diagram

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

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

12.3.3 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 30μs-65536s.

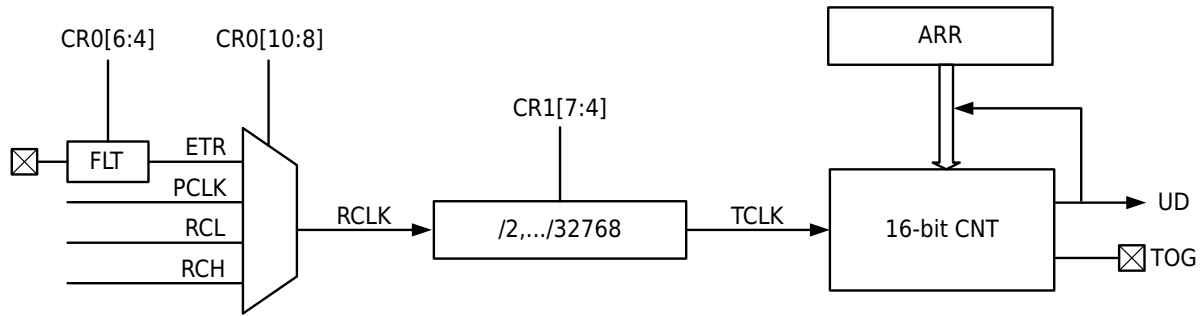


Figure 12-3 General Low Power Timer Scenario Operational Block Diagram

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

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

12.4 Operation Example

12.4.1 RCH Clock Automatic Calibration Example

Procedure

- Step1.** The CTRIM module is reset initialized using a peripheral reset.
- Step2.** Refer to the "General-Purpose I/O (GPIO)" chapter for related description of pin multiplexing configuration, set the relevant pin function to CTRIM.ETR, and supply it with a 2kHz square wave.
- Step3.** Set CTRIM_CR1.MD to 0 to select RCH clock calibration mode.
- Step4.** Set CTRIM_CR0.SRC to 0, and the external ETR pin input signal is used as RCLK.
- Step5.** Set CTRIM_CR1.PRS to 1, perform 2 frequency division on RCLK, and GCLK frequency is 1kHz.
- Step6.** Configure CTRIM_ARR to 47999, and the target count value in a GCLK count cycle is 48000.



Note

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

- Step7.** Set CTRIM_FLIM to 240, select RCH calibration accuracy of 0.5%.
 $48000 * 0.005 = 240$.
- Step8.** Set CTRIM_CR0.STEP to 4, and select the initial calibration step to be 16.
- Step9.** Set CTRIM_CR1.AUTO to 1 to enable automatic calibration.
- Step10.** Set CTRIM_CR1.EN to 1 to start clock calibration.
- Step11.** Wait for CTRIM_ISR.END to be 1, and the automatic calibration is completed.

12.4.2 RCL Clock Automatic Calibration Example

Procedure

- Step1.** The CTRIM module is reset initialized using a peripheral reset.
- Step2.** Refer to the "General-Purpose I/O (GPIO)" chapter for related description of pin multiplexing configuration, set the relevant pin function to CTRIM.ETR, and supply it with a 1MHz square wave.
- Step3.** Set CTRIM_CR1.MD to 1, select RCL clock calibration mode.
- Step4.** Set CTRIM_CR0.SRC to 0, use the external ETR pin input signal as TCLK.

- Step5.** Set CTRIM_CR1.PRS to 0x0A, perform 1024 frequency division on RCLK (RCL output frequency is), and the target frequency of GCLK is 32Hz.
- Step6.** Configure CTRIM_ARR to 31249, and the target count value in a GCLK count cycle is 31250.

**Note**

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

- Step7.** Set CTRIM_FLIM to 156, select RCL correction accuracy to 0.5%.
 $31250 \times 0.005 = 156$.
- Step8.** Set CTRIM_CR0.STEP to 4, and select the initial calibration step to be 16.
- Step9.** Set CTRIM_CR1.AUTO to 1 to enable automatic calibration.
- Step10.** Set CTRIM_CR1.EN to 1 to start clock calibration.
- Step11.** Wait for CTRIM_ISR.END to be 1, and the automatic calibration is completed.

12.5 Register Description

12.5.1 Register List

Base address: 0x40005000

Table 12-1 Clock Trim Module (CTRM) Register Offset

Offset	Register	Description	Width
0x00	CTRM_ARR	Autoload Register	32
0x04	CTRM_CNT	Error Counter Register	32
0x0C	CTRM_CR0	Control Register 0	32
0x10	CTRM_CR1	Control Register 1	32
0x14	CTRM_IER	Interrupt Enable Register	32
0x18	CTRM_ISR	Interrupt and Status Register	32
0x1C	CTRM_ICR	Interrupt Flag Clear Register	32
0x20	CTRM_FCAP	Error Result Register, Counter Value at the End of the GCLK Cycle	32
0x24	CTRM_TVAL	TRIM Result Register	32
0x28	CTRM_FLIM	Error Upper Limit Register	32

12.5.2 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																																

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

12.5.3 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																																

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

12.5.4 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: Setting prohibited ● 0b010: Setting prohibited ● 0b011: PCLK ● 0b100: RCL ● 0b101: RCH ● 0b110: Setting prohibited ● 0b111: Setting prohibited	RW
7	Reserved	-	-
6:4	ETRFLT	External input pin (ETR) filter configuration ● 0b000: No filter ● 0b001: $F_{\text{sample}} = \text{PCLK}, N=2$ ● 0b010: $F_{\text{sample}} = \text{PCLK}, N=4$ ● 0b011: $F_{\text{sample}} = \text{PCLK}, N=6$ ● 0b100: $F_{\text{sample}} = \text{PCLK}/4, N=4$ ● 0b101: $F_{\text{sample}} = \text{PCLK}/4, N=6$ ● 0b110: $F_{\text{sample}} = \text{PCLK}/8, N=4$ ● 0b111: $F_{\text{sample}} = \text{PCLK}/8, N=6$	RW
3	Reserved	-	-

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

12.5.5 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																									PRS		AUTO		MD		EN
Access																								OST								
																								RW		RW		RW		RW		RW

Bit	Name	Functional Description	Access
31:9	Reserved	-	-
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
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

Bit	Name	Functional Description	Access
0	EN	Timer enable control ● 0b0: Disable ● 0b1: Enable	RW

12.5.6 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																																OV
Access																																	RW

Bit	Name	Functional Description	Access
31:5	Reserved	-	-
4	OV	TrimCode overflow interrupt enable 0b0: Disable 0b1: Enable	RW
3	MISS	Count failure interrupt enable 0b0: Disable 0b1: Enable	RW
2	PS	Error counter completion interrupt 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

12.5.7 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																										OK		DIR		OV		MISS		PS		END		UD	
Access																											RO		RO		RO		RO		RO		RO		RO	

Bit	Name	Functional Description	Access
31:7	Reserved	-	-
6	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
5	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
4	OV	TrimCode overflow flag 0b0: The TrimCode of RCH/RCL is not 0x000 or 0x1FF 0b1: The TrimCode of RCH/RCL is adjusted to 0x000 or 0x1FF	RO
3	MISS	Counting failure flag 0b0: The error counter does not count up to ARR during one cycle of GCLK 0b1: The error counter counts up to ARR during one cycle of GCLK	RO
2	PS	Error counter completion flag 0b0: Error counter is counting 0b1: Error counter has completed one counting cycle and stopped	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

12.5.8 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																																							
Access																													WO		WO		WO		WO		WO		WO	
																													OK		OV		MISS		PS		END		UD	

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 error counter completion 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

12.5.9 Error Result Register (CTRIM_FCAP)

Offset	Bit Position																															
0x20	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000FFFF																															
Name	Reserved																FCAP															
Access																	RO															

Bit	Name	Functional Description	Access
31:16	Reserved	-	-
15:0	FCAP	Frequency error capture value	RO

12.5.10 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																																

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

12.5.11 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
31:8	Reserved	-	-
7:0	FLIM	Error tolerance	RW

13 IWDT

13.1 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 RCL), and it can still work normally even if the main clock fails.

13.2 Features

- Overflow period 132 μ s-64s @RCL 32.768kHz
- 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

13.3 Functional Description

13.3.1 Function Block Diagram

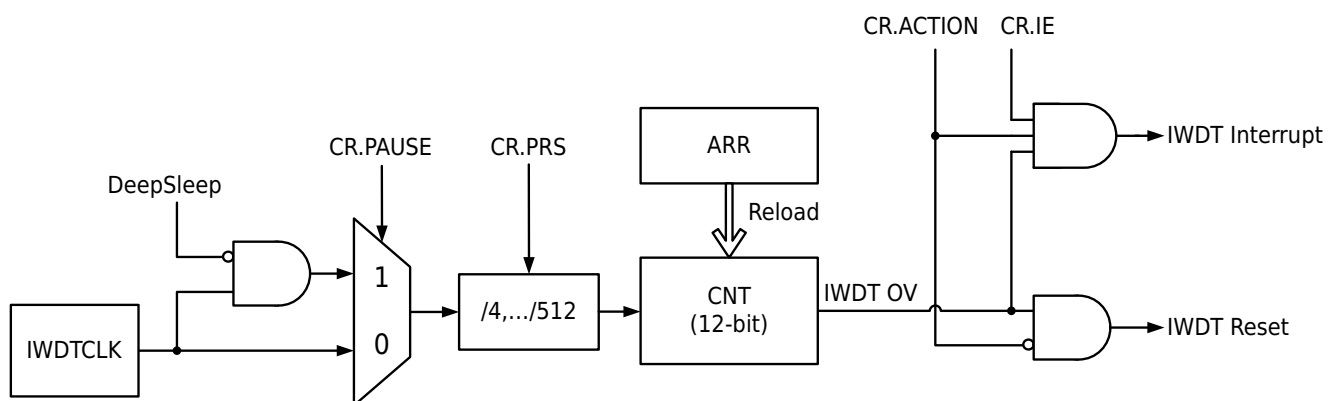


Figure 13-1 IWDT Function Block Diagram

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.

13.3.2 Window Watchdog Mode

When the value of the IWDT_WINR register is configured to be less than IWDT_ARR, IWDT works in the window watchdog mode. In this mode, there are two conditions to trigger the watchdog overflow: the count value is decremented to 0x000, and the watchdog is reloaded when the count value is greater than the window value. When changing the IWDT_WINR register value, the hardware will automatically reload the watchdog and reset the down counter value to IWDT_ARR. The user program should reload

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.

13.3.3 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.

13.3.4 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.

13.3.5 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.

13.3.6 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.

13.3.7 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.

13.4 Operation Example

13.4.1 Configure IWDT as Window Watchdog

- Step1.** Write 0xCCCC to the IWDT_KR register to enable the IWDT.
- Step2.** Write 0x5555 to the IWDT_KR register to release the write protection of the register.
- Step3.** Configure IWDT_CR, select the counting clock frequency of the watchdog, the action after overflow, and the DeepSleep feature.
- Step4.** Configure IWDT_ARR to select the overflow period of the watchdog.
- Step5.** Wait for IWDT_SR.ARRF to be 0.
- Step6.** Write the required value to IWDT_WINR to configure the window size.
- Step7.** Wait for IWDT_SR.WINRF to become 0, and automatically refresh the counter value to IWDT_ARR.
- Step8.** Before the count value is less than or equal to the IWDT_WINR register value and the count value is decremented to 0, write 0xAAAA to the IWDT_KR register to reload IWDT.

13.4.2 Configure IWDT as Independent Watchdog

- Step1.** Write 0xCCCC to the IWDT_KR register to enable the IWDT.

- Step2.** Write 0x5555 to the IWDT_KR register to release the write protection of the register.
- Step3.** Configure IWDT_CR, select the counting clock frequency of the watchdog, the action after overflow, and the DeepSleep feature.
- Step4.** Configure IWDT_ARR to select the overflow period of the watchdog.
- Step5.** Wait for IWDT_SR to be 0x10, the frequency division coefficient and reload value have been written successfully
- Step6.** Refresh the counter value to the value of IWDT_ARR(IWDT_KR=0x0000AAAA).
- Step7.** Before the count value is decremented to 0, write 0xAAAA to the IWDT_KR register to reload the IWDT.

13.5 Register Description

13.5.1 Register List

Base address: 0x40000F80

Table 13-1 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

13.5.2 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

13.5.3 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	PRS		
Access																											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
3:0	PRS	Configure the watchdog count clock and the frequency division ratio of RCL oscillator ● 0b0000: 4-frequency division ● 0b0001: 8-frequency division ● 0b0010: 16-frequency division ● 0b0011: 32-frequency division ● 0b0100: 64-frequency division ● 0b0101: 128-frequency division ● 0b0110: 256-frequency division ● 0b0111: 512-frequency division ● Others: Setting prohibited	RW

13.5.4 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																																

Bit	Name	Functional Description	Access
31:12	Reserved	-	-
11:0	ARR	IWDT reload value	RW

13.5.5 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

13.5.6 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																																

Bit	Name	Functional Description	Access
31:12	Reserved	-	-
11:0	WINR	Window comparator comparison value	RW

14 WWDT

14.1 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).

14.2 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

14.3 Functional Description

14.3.1 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.

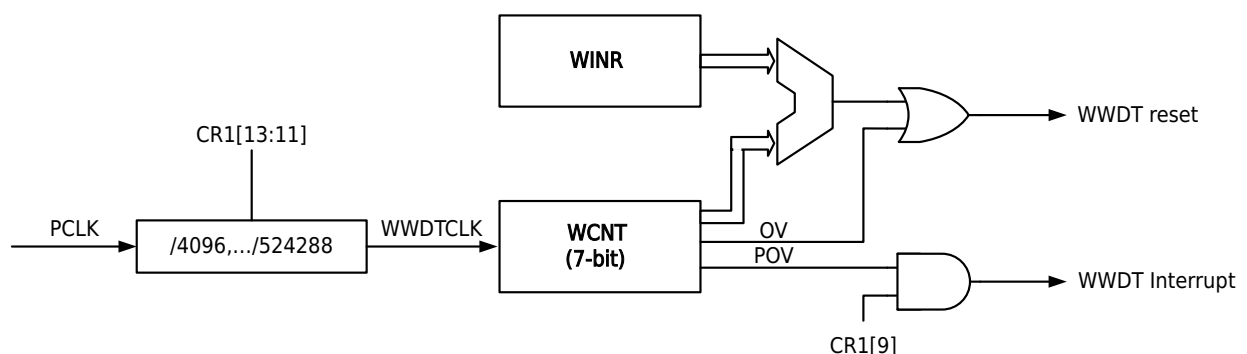


Figure 14-1 WWDT Function Block Diagram

14.3.2 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.

14.3.3 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:

$$T_{\text{WWDT}} = T_{\text{PCLK}} \times 4096 \times 2^{\text{CR1.PRS}} \times (\text{CR0.WCNT} - 0x3F)$$

14.3.4 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.

14.3.5 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.

14.4 Operation Example

14.4.1 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.

14.5 Register Description

14.5.1 Register List

Base address: 0x40006800

Table 14-1 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

14.5.2 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

14.5.3 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			Reserved																
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

14.5.4 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

15 LPUART

15.1 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.

15.2 Features

- Configuration clock PCLK
- Transmission clock SCLK (SCLK can select 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

15.3 Functional Description

15.3.1 Function Block Diagram

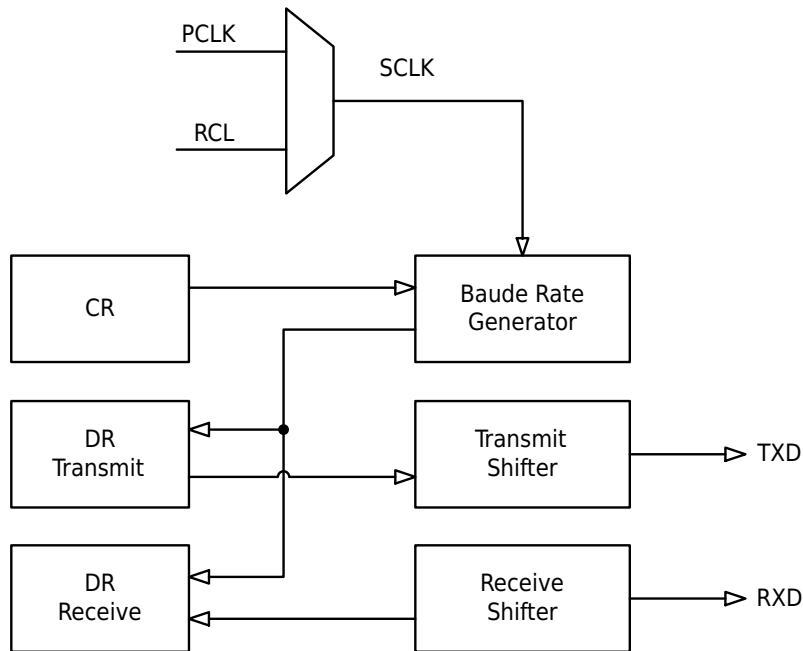


Figure 15-1 LPUART Function Block Diagram

15.3.2 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 internal low-speed clock (RCL), and the PCLK clock.

When the system enters DeepSleep mode, if SCLK selects RCL, the LPUART can still perform normal data transmission and reception without being affected by the system's DeepSleep mode.

15.3.3 Working Mode

LPUART supports multiple working modes: synchronous half-duplex mode, asynchronous full-duplex mode, and single-wire half-duplex mode.

By combining LPUART_CR.MODE with LPUART_CR.HDSEL, various required working modes can be configured.

15.3.3.1 Mode0-Mode3 Function Comparison

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

Table 15-1 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_CR introduction.
- The definition of SCNT is detailed in register LPUART_BRR introduction.
- DR8 data bit is special and has different meanings in different applications, please refer to the following table:

Table 15-2 DR8 Data Meaning

Application Scenario	LPUART_CR.AD RDET	LPUART_CR.DR 8CFG[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.

15.3.3.2 Mode0 Data Transmitting and Reception

● Mode0 Data Transmitting

When LPUART_CR.RXEN set to 0, the LPUART operates in the transmission mode. The data to be transmitted is written into the LPUART_DR 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.

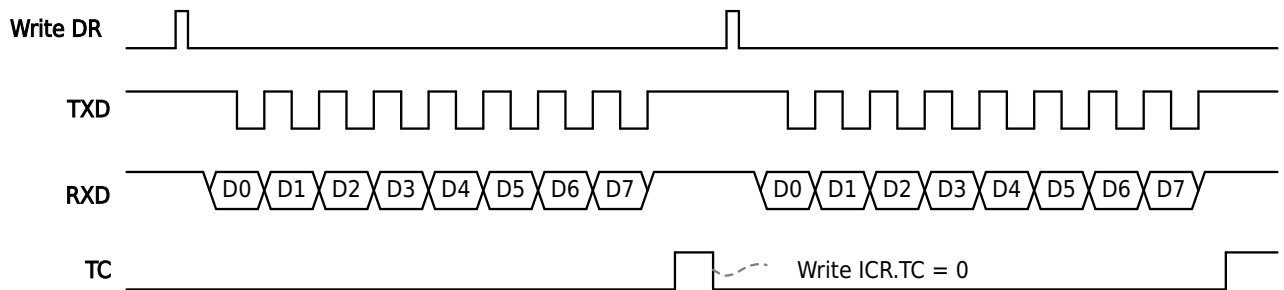


Figure 15-2 Mode0 Data Transmitting

● Mode0 Data Receiving

When LPUART_CR.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), each time the LPUART_ISR.RC flag is cleared, the next byte can be received.

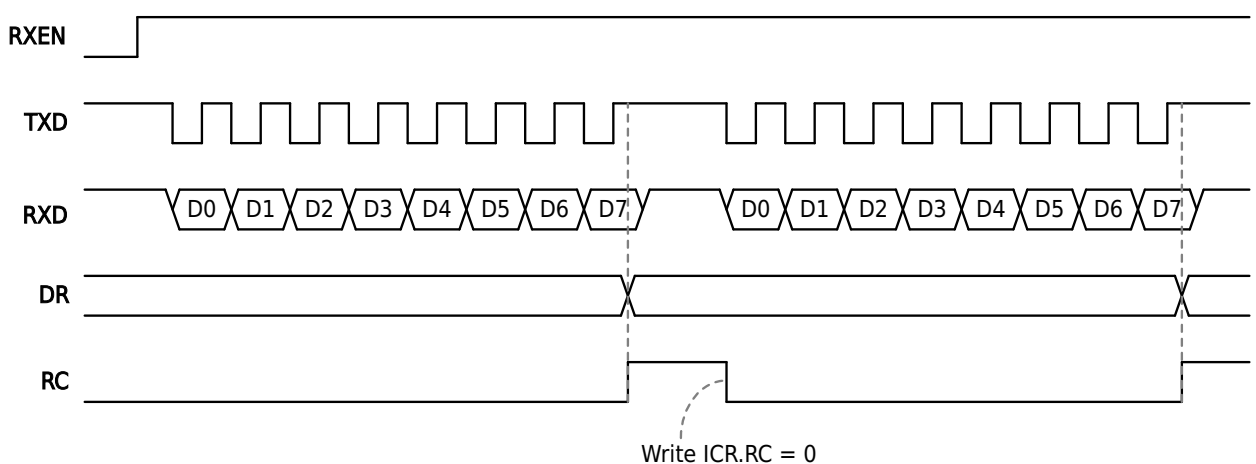


Figure 15-3 Mode0 Data Receiving

15.3.3.3 Mode1 Data Transmitting and Reception

● Mode1 Data Transmitting

Whether LPUART_CR.RXEN is 0 or 1, data can be transmitted normally.

Write data to LPUART_DR, 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_DR 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.

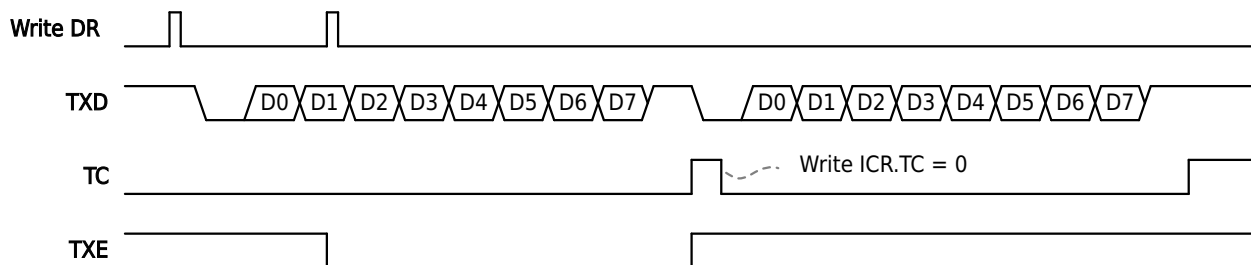


Figure 15-4 Mode1 Data Transmitting

- Mode1 Data Receiving

When receiving data, set LPUART_CR.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_DR register.

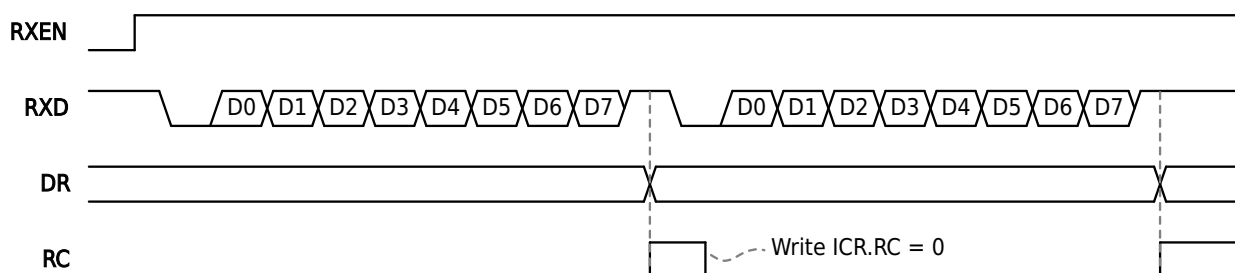


Figure 15-5 Mode1 Data Receiving

15.3.3.4 Mode2/Mode3 Data Transmitting and Reception

- Mode2/3 Data Transmitting

Whether LPUART_CR.RXEN is 0 or 1, data can be transmitted normally.

Write data to LPUART_DR, 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_DR 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.

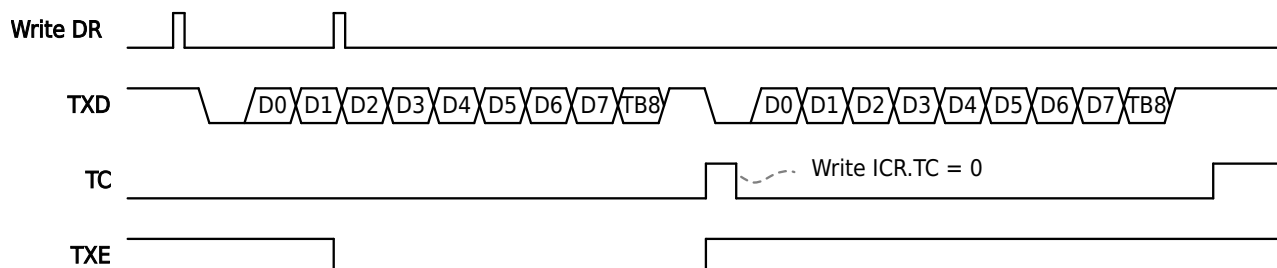


Figure 15-6 Mode2/Mode3 Data Transmitting

- Mode2/3 Data Receiving

When receiving data, set LPUART_CR.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_DR register.

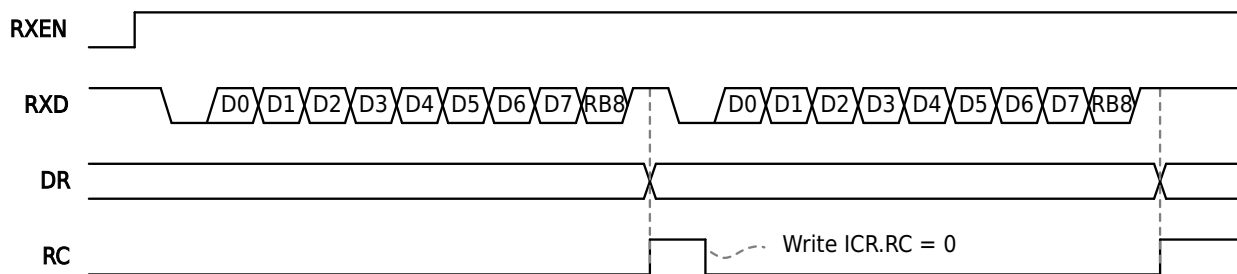


Figure 15-7 Mode2/Mode3 Data Receiving

15.3.3.5 Single-Wire Half-Duplex Data Transmitting and Reception

When the single bus function control bit LPUART_CR.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_DR register, the TXD pin changes to the output state in order to send the data written by the user to the register in LPUART_DR 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.

15.3.4 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_CR introduction.
- The definition of SCNT is detailed in register LPUART_BRR introduction.

Mode1/Mode3 Baud Rate Setting Example

Table 15-3 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%

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 15-4 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 15-5 SCLK = 16MHz Baud Rate Calculation Table

Baud Rate	SCLK=16MHz								
	OVER=4			OVER=8			OVER=16		
	SCNT	Actual Baud Rate	Error%	SCNT	Actual Baud Rate	Error%	SCNT	Actual Baud Rate	Error%
2400	1667	2399.52	-0.02%	833	2400.96	0.04%	417	2398.08	-0.08%
4800	833	4801.92	0.04%	417	4796.16	-0.08%	208	4807.69	0.16%
9600	417	9592.33	-0.08%	208	9615.38	0.16%	104	9615.38	0.16%
19200	208	19230.77	0.16%	104	19230.77	0.16%	52	19230.77	0.16%
38400	104	38461.54	0.16%	52	38461.54	0.16%	26	38461.54	0.16%
57600	69	57971.01	0.64%	35	57142.86	-0.79%	17	58823.53	2.12%
76800	52	76923.08	0.16%	26	76923.08	0.16%	13	76923.08	0.16%
115200	35	114285.71	-0.79%	17	117647.06	2.12%	9	111111.11	-3.55%
128000	31	129032.26	0.81%	16	125000.00	-2.34%	8	125000.00	-2.34%
256000	16	250000.00	-2.34%	8	250000.00	-2.34%	4	250000.00	-2.34%
1000000	4	1000000.00	0.00%	2	1000000.00	0.00%	1	1000000.00	0.00%
2000000	2	2000000.00	0.00%	1	2000000.00	0.00%	1	/	/
4000000	1	4000000.00	0.00%	1	/	/	0	/	/

Table 15-6 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 15-7 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 15-8 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%

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	/	/

15.3.5 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.

15.3.6 Multi-Machine Communication

When working in Mode2/3, set the LPUART_CR.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_DR.DR8 is 1, the address frame is transmitted, LPUART_DR.DR is the address of the target slave.

When LPUART_DR.DR8 is 0, the data frame is transmitted, LPUART_DR.DR is the data to be transmitted.

Slave Receiving Address and Data Frame

When LPUART_CR.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_DR register and LPUART_ISR.RC is set.

When LPUART_CR.ADRDET is 0, the slave only receives data frames and ignores address frames. The user program should set LPUART_CR.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_ADDR | LPUART_ADDRMASK
- User Configuration Address: LPUART_ADDR & LPUART_ADDRMASK

When an address frame is received, if the address received from the bus, after being ANDed with the value of LPUART_ADDRMASK, 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_ADDR=0x55, LPUART_ADDRMASK=0xF0, then the slave can match addresses 0xF5, 0xF7, 0xFD, 0xFF (broadcast address) and 0x50~0x5F (user-configured address).

15.3.7 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:

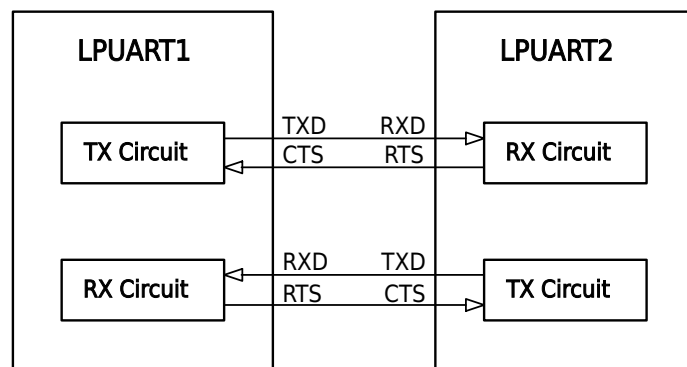


Figure 15-8 LPUART Hardware Flow Control

RTS Flow Control

When RTS flow control is enabled (LPUART_CR.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.

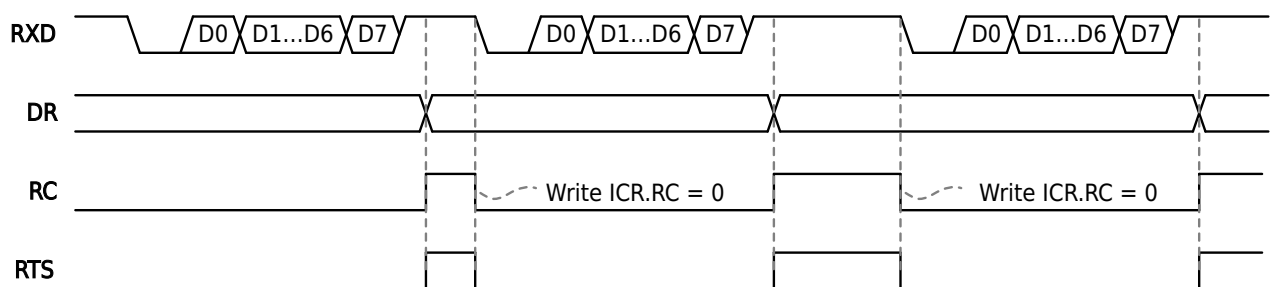


Figure 15-9 RTS Hardware Flow Control Signal

CTS Flow Control

When CTS flow control is enabled (LPUART_CR.CTSEN is set to 1), the CTS pin instructs this module to transmit the next frame of data:

- If CTS is low level, LPUART transmits the next frame of data.

- If CTS is high level, after LPUART finishes transmitting the current frame, it waits for CTS to become low level before sending the next frame.

When the CTS level changes, LPUART_ISR.CTSIF is automatically set to 1. Read LPUART_ISR.CTS to obtain the current CTS pins level value.

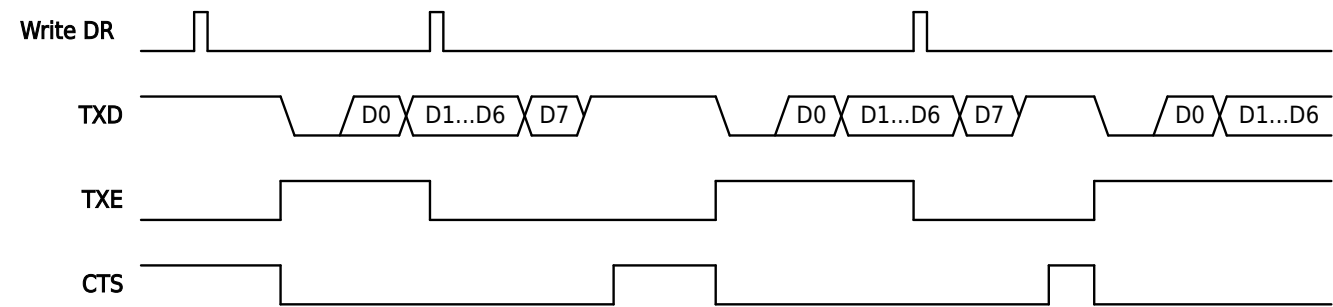


Figure 15-10 CTS Hardware Flow Control Signal

15.4 Operation Example

15.4.1 Transmit Data Example

Procedure

- Step1.** Refer to the relevant description of the port digital multiplexing function in the "General-Purpose I/O (GPIO)" chapter, map TXD to the required pin; and configure TXD to CMOS output.
- Step2.** Configure SYSCTRL module register PERICKEN0.LPUARTx to 1 to enable LPUART configuration clock and working clock.
- Step3.** Configure LPUART_CR.CLKSRC to select the transmission clock source for LPUART.
- Step4.** Configure LPUART_CR.MODE to select the working mode of LPUART.
- Step5.** Configure LPUART_CR.OVER and LPUART_BRR register to configure the communication baud rate.
- Step6.** Configure LPUART_CR.STOP, LPUART_CR.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 queried for 1, write a frame of data to be transmitted to the LPUART_DR 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.

15.4.2 Receive Data Example

Procedure

- Step1.** Refer to the relevant description of the port digital multiplexing function in the "General-Purpose I/O (GPIO)" chapter. Map the RXD to the required pin; and configure RXD as an input and enable the pull-up resistor.
- Step2.** Configure SYSCTRL module register PERICKEN0.LPUARTx to 1 to enable LPUART configuration clock and working clock.
- Step3.** Configure LPUART_CR.CLKSRC to select the transmission clock source for LPUART.
- Step4.** Configure LPUART_CR.MODE to select the working mode of LPUART.
- Step5.** Configure LPUART_CR.OVER and LPUART_BRR register to configure the communication baud rate.
- Step6.** Configure LPUART_CR.STOP, LPUART_CR.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_CR.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_DR 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.

15.5 Register Description

15.5.1 Register List

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

Table 15-9 Low Power Universal Asynchronous Receiver/Transmitter (LPUART) Register Offset

Offset	Register	Description	Width
0x00	LPUART_DR	Data Register	32
0x04	LPUART_CR	Control Register	32
0x08	LPUART_ADDR	Address Register	32
0x0C	LPUART_ADDRMASK	Address Mask Register	32
0x10	LPUART_ISR	Interrupt Flag Bit Register	32
0x14	LPUART_ICR	Interrupt Flag Bit Clear Register	32
0x18	LPUART_BRR	Baud Rate Register	32

15.5.2 Data Register (LPUART_DR)

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 the hardware parity bit is turned on, this bit is the parity bit of the received data when receiving, and the verification is performed by the hardware. If the verification error occurs, the verification error flag bit PE is set to 1; this bit is invalid when transmitting, the parity bit of the transmitted data is calculated and transmitted by the hardware. ▶ When the hardware parity bit is turned off, this bit is received data Bit8 when receiving; this bit is sent data Bit8 when transmitting.  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

15.5.3 Control Register (LPUART_CR)

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	Reserved	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	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
17:16	Reserved	-	-
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
13	PEIE	Parity error interrupt enable 0b0: Disable 0b1: Enable	RW

Bit	Name	Functional Description	Access
12:11	CLKSRC	Transmission clock SCLK source selection ● 0b00: PCLK ● 0b01: PCLK ● 0b10: RCL ● 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	ADDRDET	Multi-machine address automatic recognition enable ● 0b0: Disabled, during transmission/reception, the function of LPUART_DR.DR8 is determined by DR8CFG ● 0b1: Enabled, during transmission, LPUART_DR.DR8 is fixed to 1; 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
3:2	DR8CFG	DR8 data bit function configuration ● 0b00: Deciding by software read/write LPUART_DR.DR8 ● 0b01: Hardware even parity check ● 0b10: Hardware odd parity check ● 0b11: Setting prohibited	RW

Bit	Name	Functional Description	Access
1	TCIE	Transmit complete interrupt enable ● 0b0: Disable ● 0b1: Enable	RW
0	RCIE	Receive complete interrupt enable ● 0b0: Disable ● 0b1: Enable	RW

15.5.4 Address Register (LPUART_ADDR)

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																								ADDR							
Access																									RW							

Bit	Name	Functional Description	Access
31:8	Reserved	-	-
7:0	ADDR	Slave device address register	RW

15.5.5 Address Mask Register (LPUART_ADDRMASK)

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																								MASK							
Access																									RW							

Bit	Name	Functional Description	Access
31:8	Reserved	-	-
7:0	MASK	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

15.5.6 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 flipped	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

15.5.7 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	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

15.5.8 Baud Rate Register (LPUART_BRR)

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

16 SPI

16.1 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.

16.2 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

16.3 Functional Description

16.3.1 Function Block Diagram

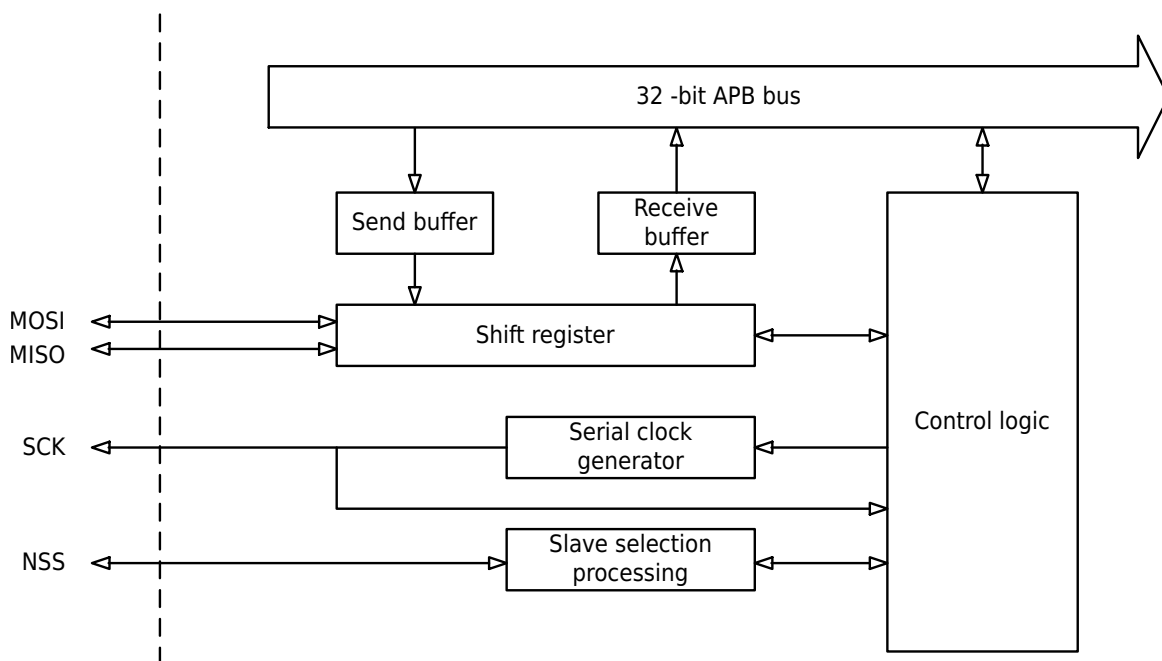


Figure 16-1 SPI Block Diagram

16.3.2 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.

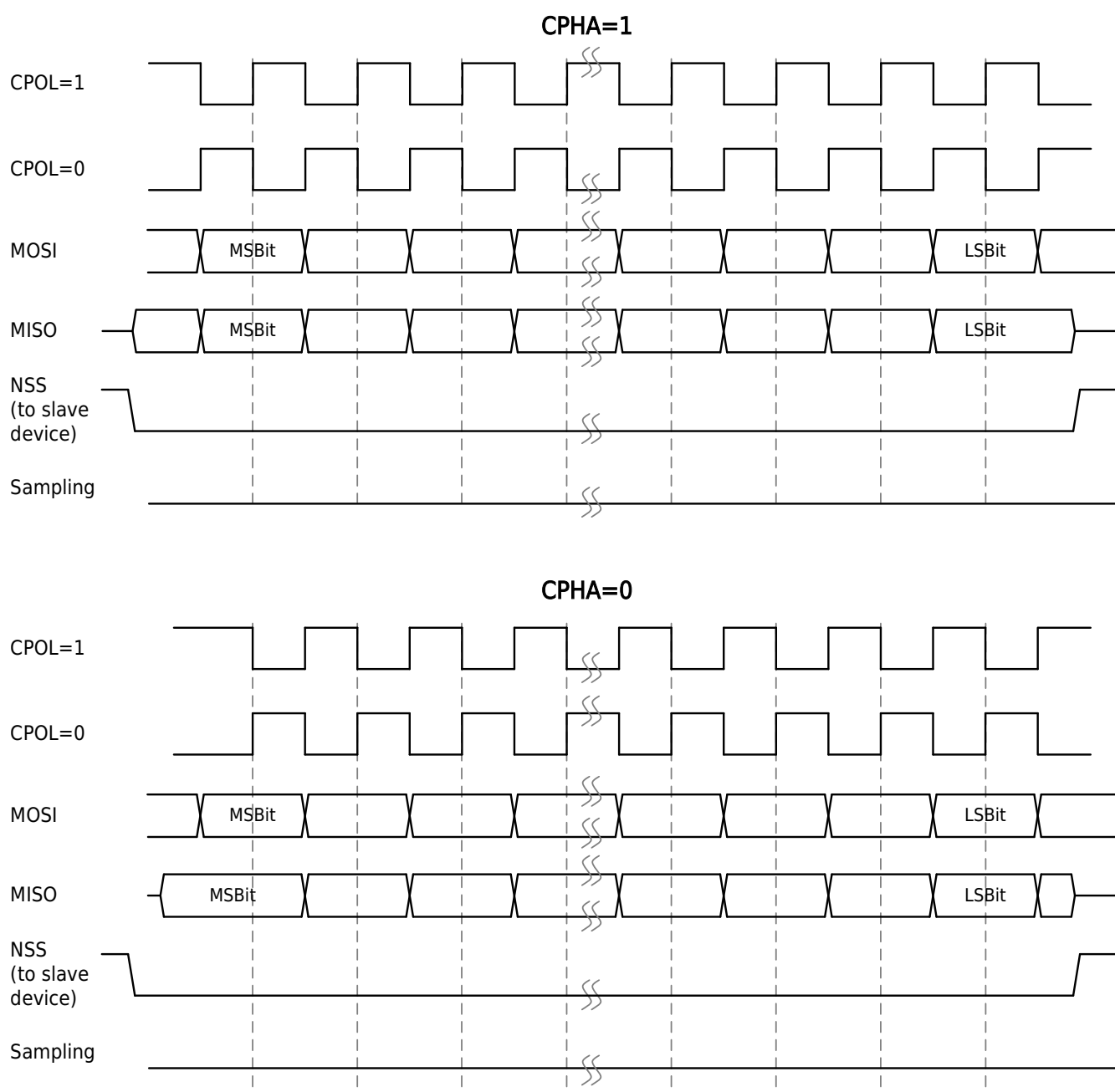


Figure 16-2 SPI Communication Format Timing Diagram

16.3.3 Slave Select Pin Configuration

This module provides a variety of slave selection methods to adapt to different application scenarios.

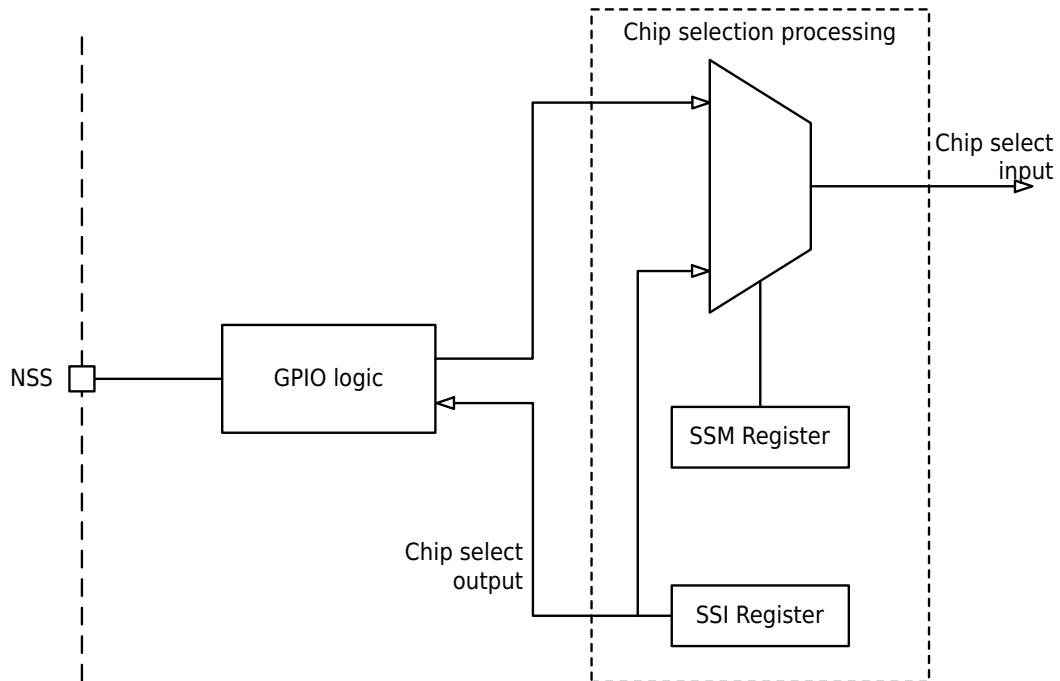


Figure 16-3 Slave Select NSS Pin Configuration

- 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.

16.3.4 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.

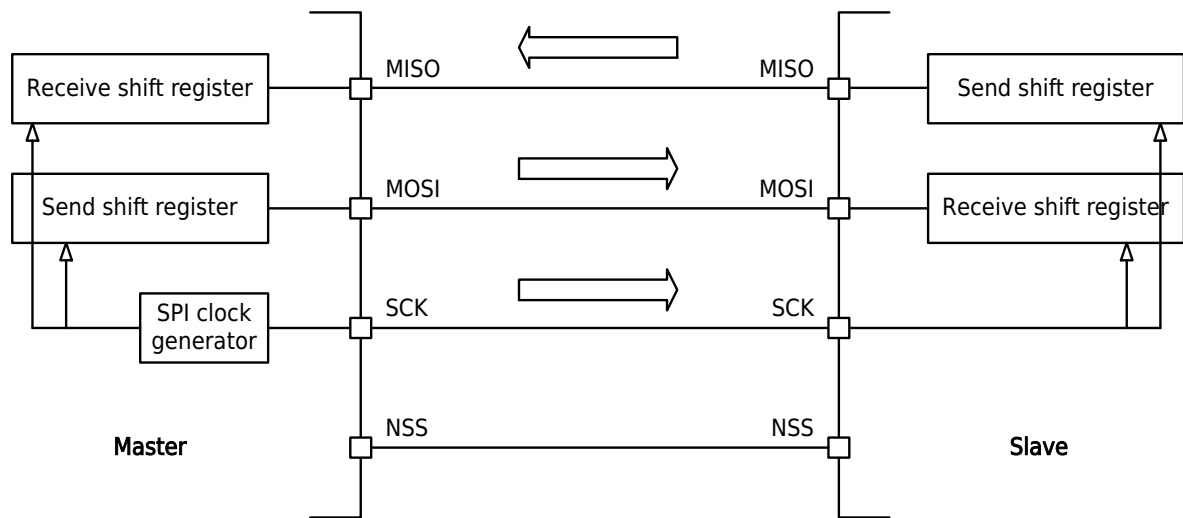


Figure 16-4 SPI Full-duplex Communication

16.3.5 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.

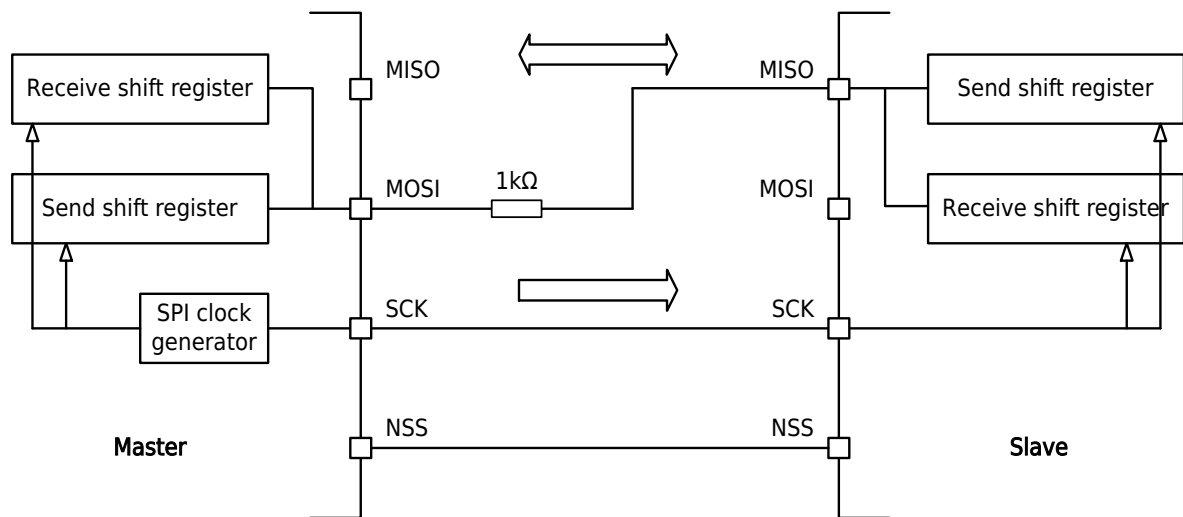


Figure 16-5 SPI Single-wire Half-duplex Communication

16.3.6 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.

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.

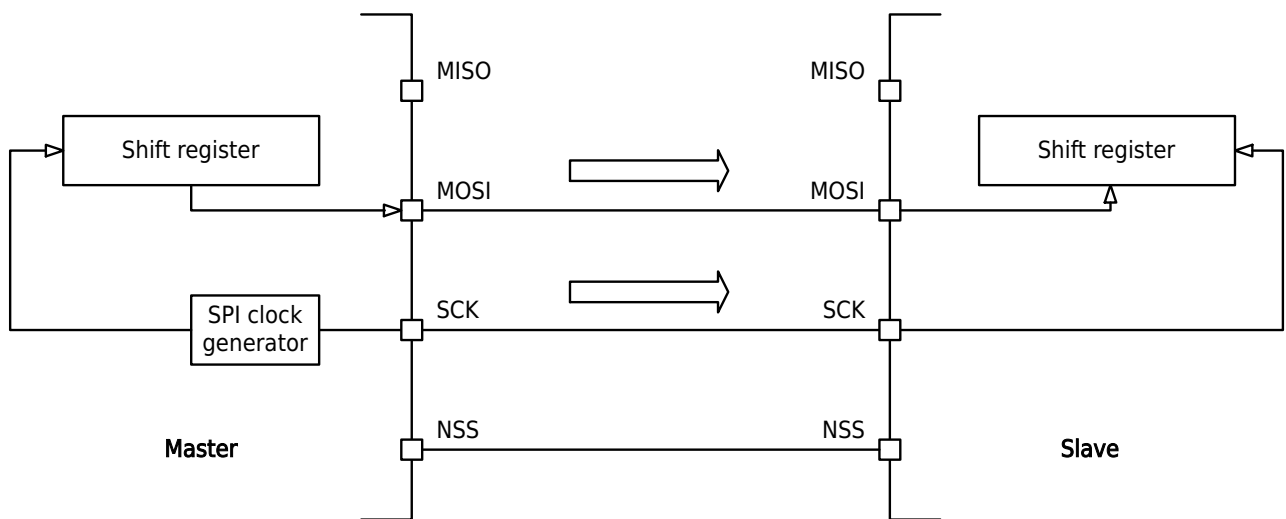


Figure 16-6 SPI Simplex Transmit Communication Diagram

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.

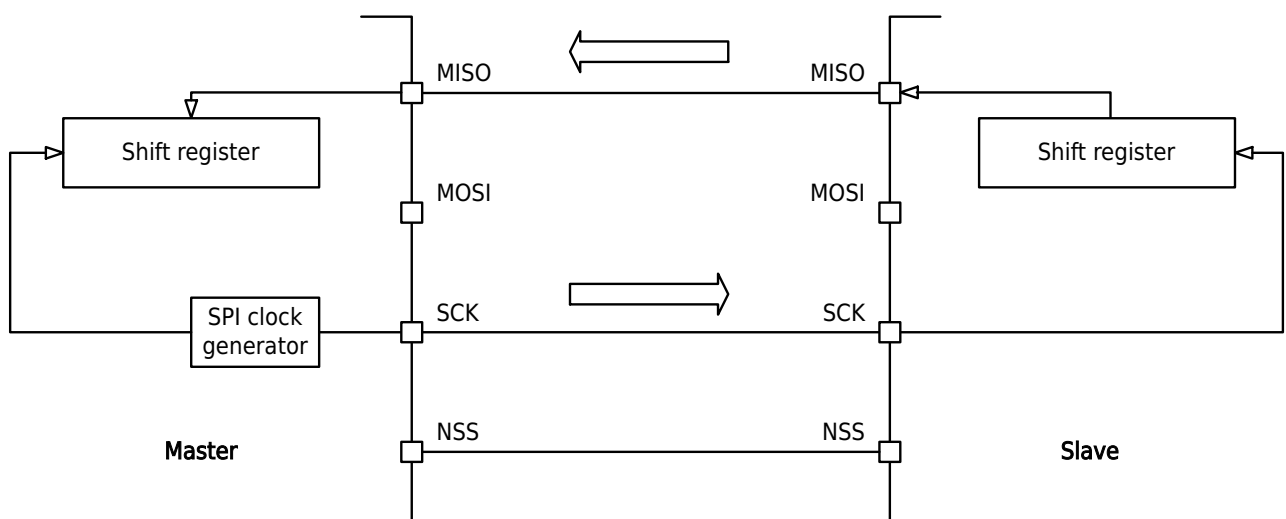


Figure 16-7 SPI Simplex Receive Communication Diagram



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.

16.3.7 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. When the master attempts to gain control of the SPI bus, it can detect bus conflicts by monitoring the mode fault (MODF) error. 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.

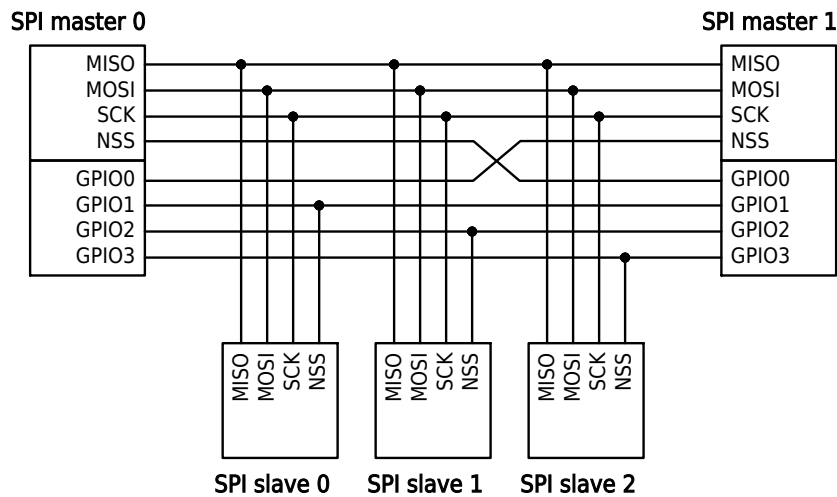


Figure 16-8 SPI Multi-machine System

16.3.8 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 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.

Master Mode Example with CPOL=1 and CPHA=1

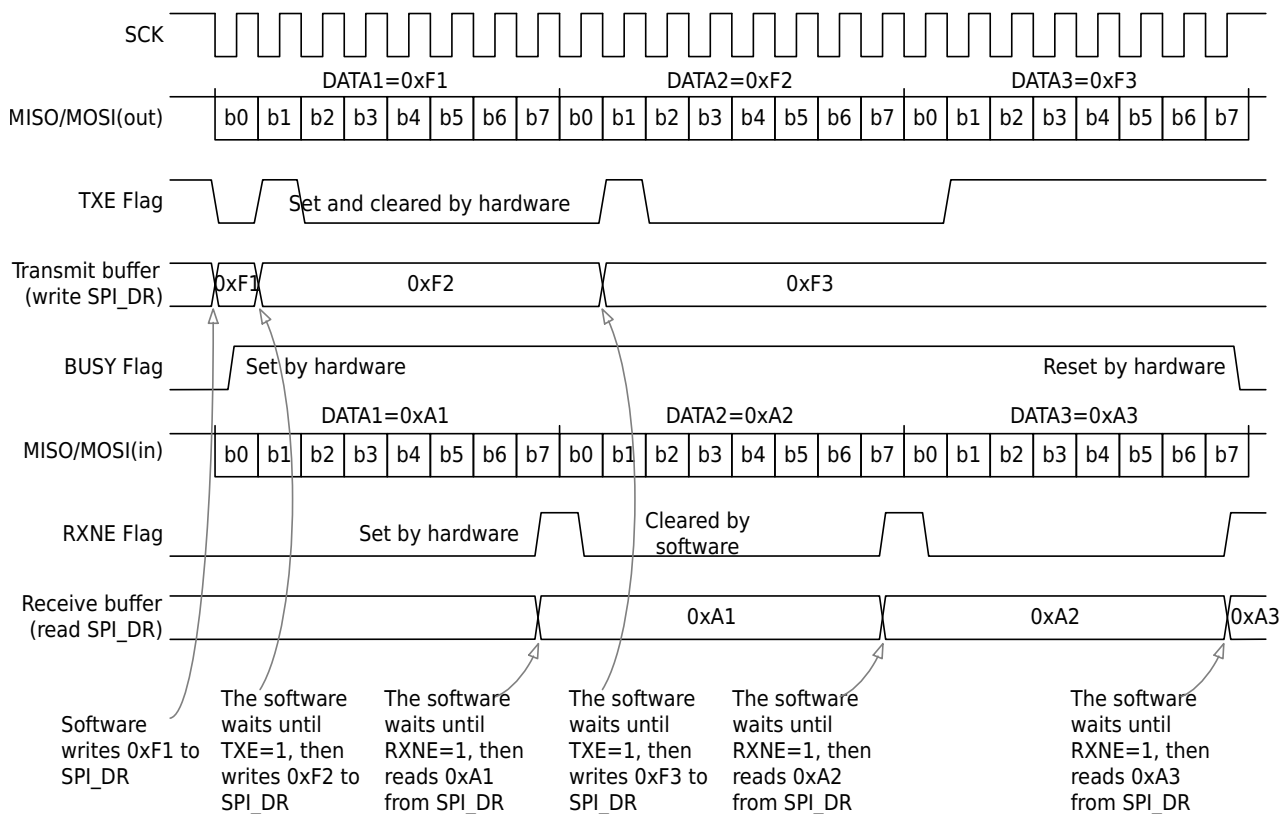


Figure 16-9 Explanation Diagram of General Flags in SPI Master Mode

**Slave mode example with
CPOL=1 and CPHA=1**

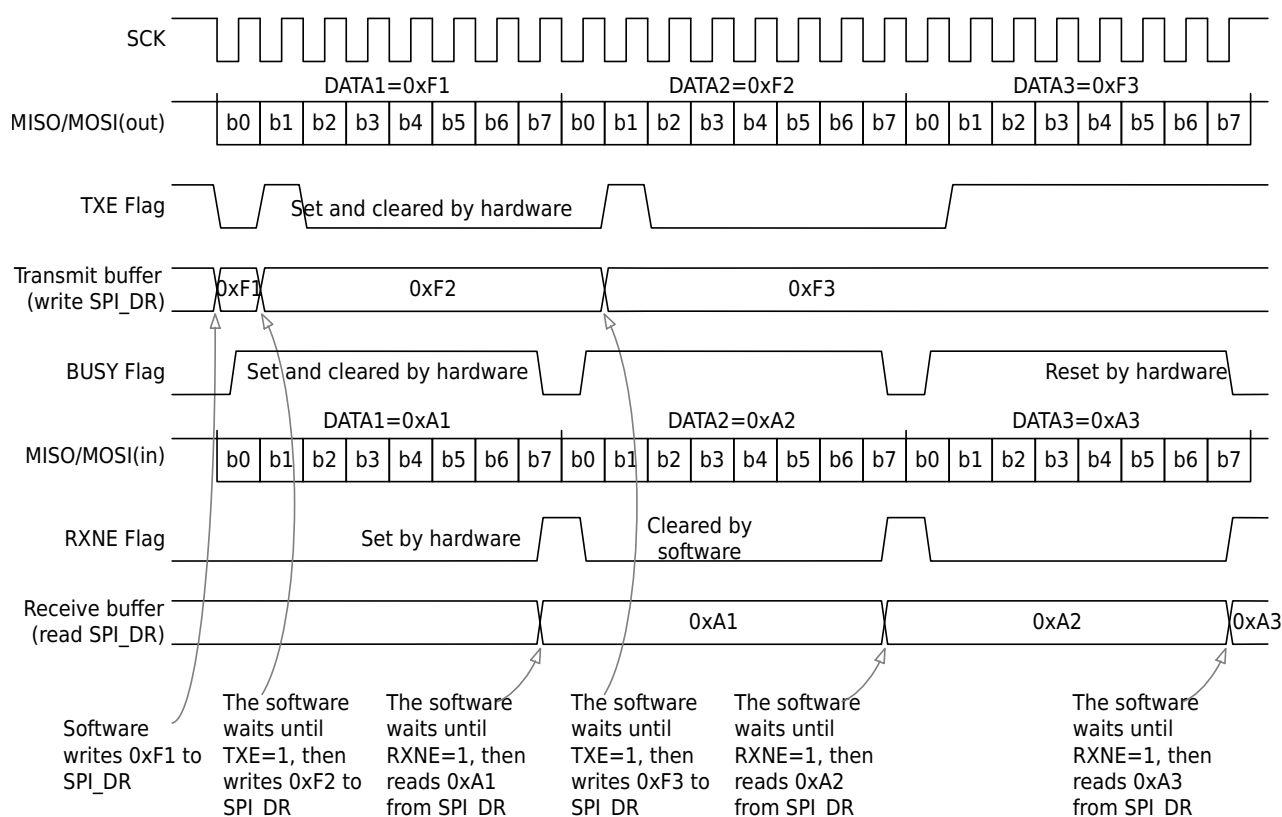


Figure 16-10 SPI Slave Mode General Flag Bit Description Diagram

16.3.9 Interrupt Generation and Clearing

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

Table 16-1 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.

16.4 Operation Example

16.4.1 Example of Sending and Receiving Data in Full-duplex by the Master

- Step1.** According to the relevant description of the pin digital multiplexing function in the chapter "General-Purpose I/O (GPIO)", map NSS, SCK, MOSI, and MISO to the required pins.
- Step2.** Configure SYSCTRL module register PERICKEN0.SPI to 1 to enable SPI 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 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.
- 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.

16.4.2 Example of Sending and Receiving Data in Full-duplex by the Slave

- Step1.** According to the relevant description of the pin digital multiplexing function in the chapter "General-Purpose I/O (GPIO)", map NSS, SCK, MOSI, and MISO to the required pins.
- Step2.** Configure SYSCTRL module register PERICKEN0.SPI to 1 to enable SPI 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.

16.4.3 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 SYCTRL module register PERICKEN0.SPI to 1 to enable SPI 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.
- 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.

16.5 Register Description

16.5.1 Register List

SPI base address: 0x40000800

Table 16-2 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

16.5.2 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																	CM	SSM	WIDTH				LSBF	MSTR	CPOL	CPHA	BRR		EN		
Access																		RW	RW	RW				RW	RW	RW	RW	RW		RW		

Bit	Name	Functional Description	Access
31:15	Reserved	-	-
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
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

Bit	Name	Functional Description	Access
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

16.5.3 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																								MODFIE	SSERRIE	OVFIE	UDFIE	SSRIE	SSFIE	RXNEIE	TXEIE		
Access																									RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Functional Description	Access
31:8	Reserved	-	-
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
0	TXEIE	Transmit buffer empty interrupt enable control 0b0: Disable 0b1: Enable	RW

16.5.4 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

16.5.5 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

16.5.6 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

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

16.5.7 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

16.5.8 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																																

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

17 I2C

17.1 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.

17.2 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)
- Slave supports 7-bit addressing function; Master supports 10-bit addressing function
- Support noise filtering function
- Support 3 slave addresses
- Support broadcast address
- Support interrupt status query function

17.3 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.

17.3.1 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).

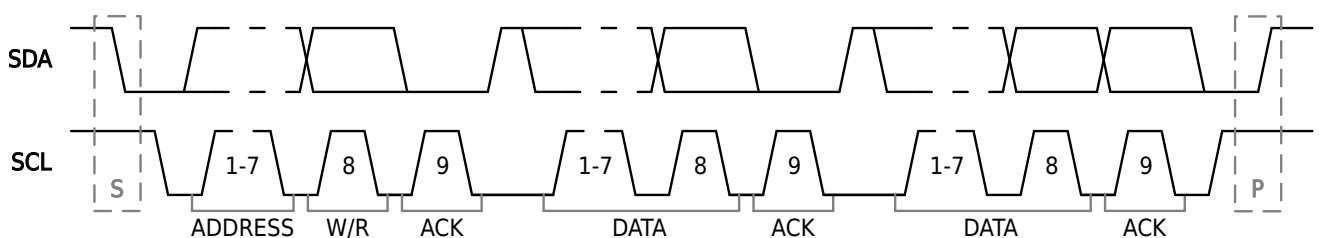


Figure 17-1 I2C Transmission Protocol

- 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.

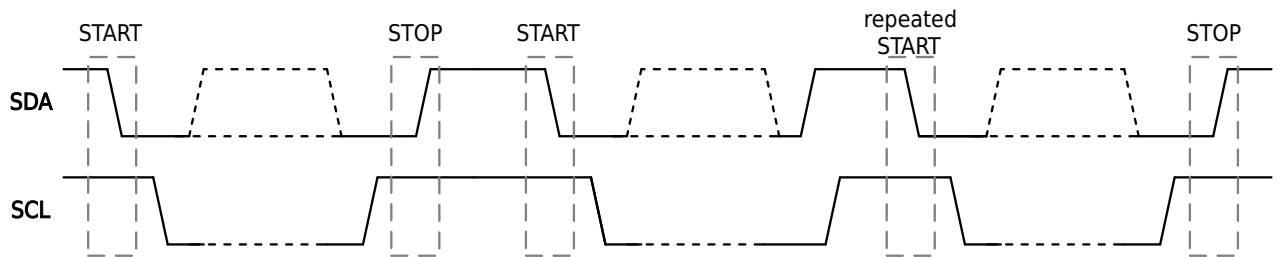


Figure 17-2 START, Repeated START and STOP

- 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.

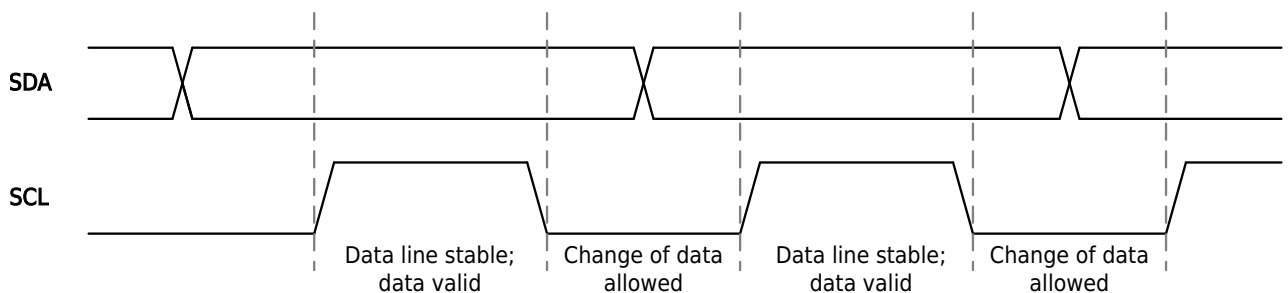


Figure 17-3 Bit Transfer on I2C Bus

17.3.2 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

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.

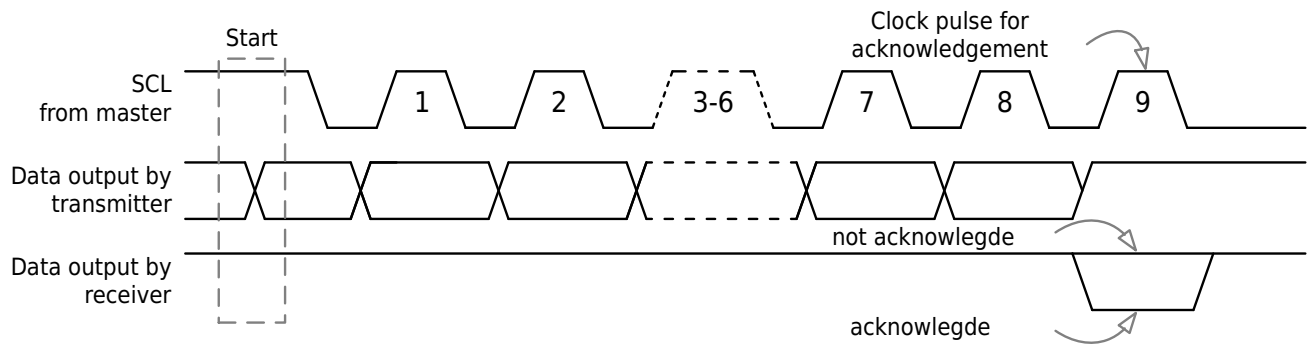


Figure 17-4 Acknowledgment Signal on I2C Bus

17.3.3 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.

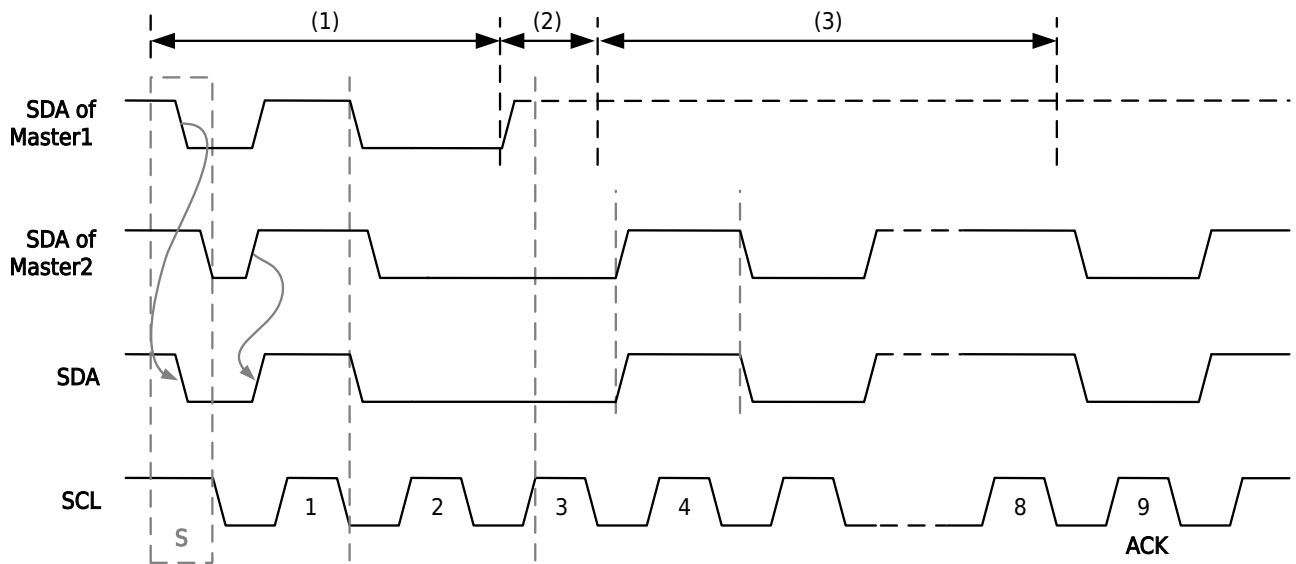


Figure 17-5 Arbitration on the I2C Bus

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 arbitration, subsequent data transfer on SDA is initiated by Master 2.

17.4 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.

17.4.1 Function Block Diagram

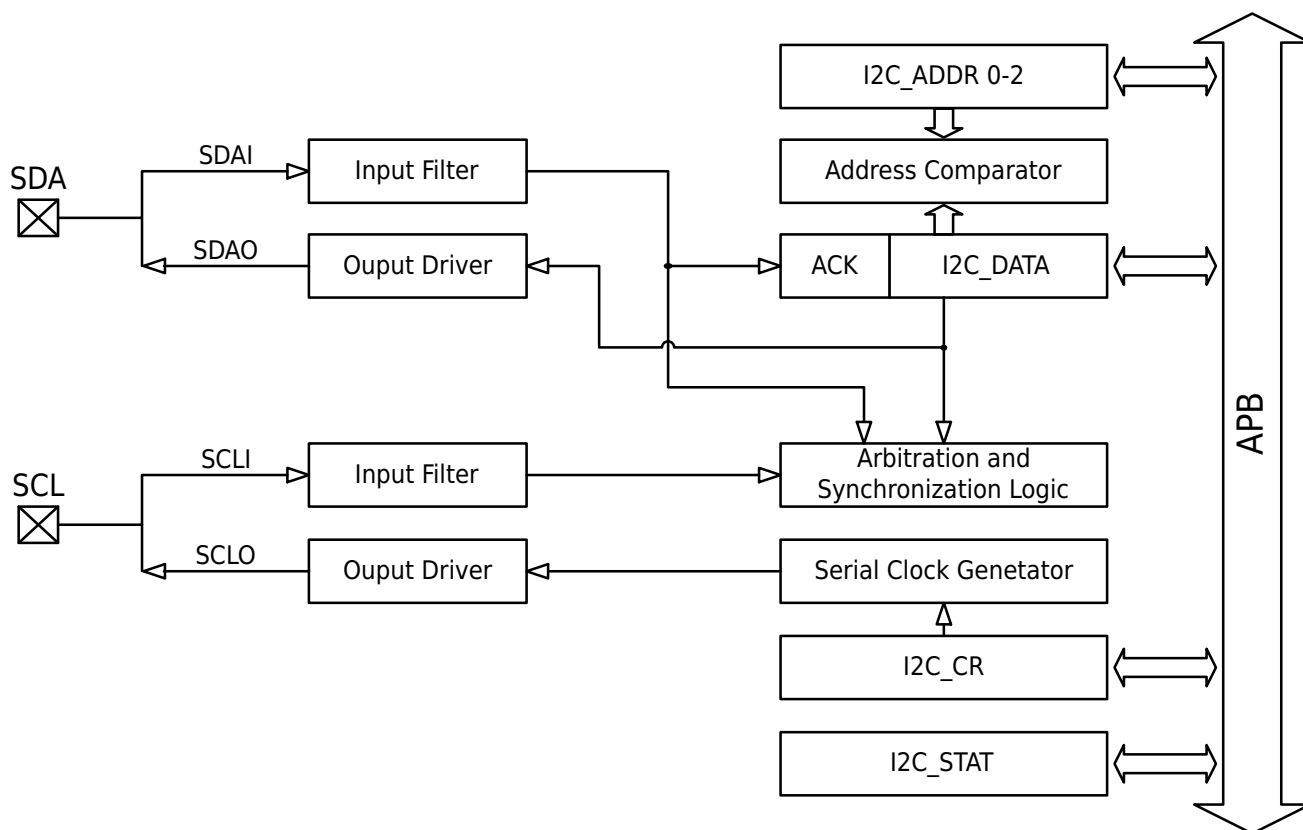


Figure 17-6 I2C Function Block Diagram

17.4.2 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 $f_{SCL} = f_{PCLK} / 8 / (I2C_BRR + 1)$, where I2C_BRR should be greater than 0.

The following table lists the output frequency value of SCL signal when PCLK frequency is combined with I2C_BRR.

Table 17-1 I2C Clock Signal Baud Rate

PCLK (kHz)	I2C_BRR						
	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

17.4.3 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.FLT, the intensity of filtering can be changed.

- When used as the master, if the value of I2C_BRR is less than or equal to 9, I2C_CR.FLT should be set to 1 to make simple filtering; if the value of I2C_BRR is greater than 9, I2C_CR.FLT 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 40, I2C_CR.FLT should be set to 1 to make simple filtering; if the ratio of PCLK to SCL frequency is greater than 40, I2C_CR.FLT should be set to 0 to make advanced filtering.

17.4.4 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.

17.4.5 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.

17.4.6 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.

17.4.7 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.

● 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_BRR. 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.

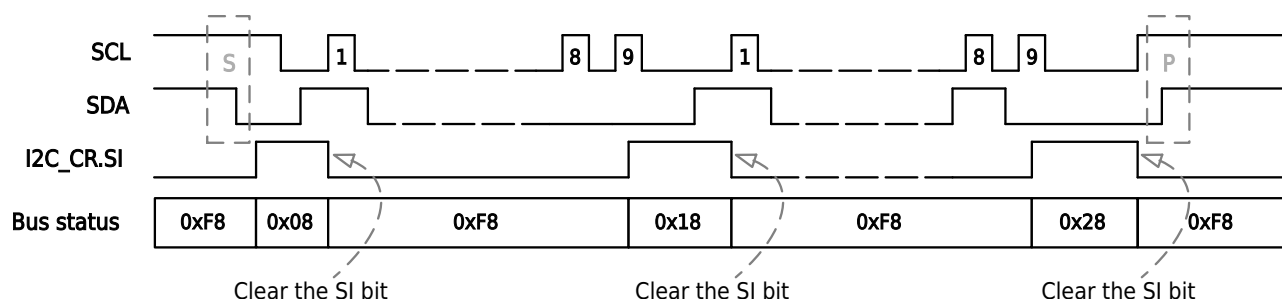


Figure 17-7 Data Synchronization Diagram of Master Sending Mode

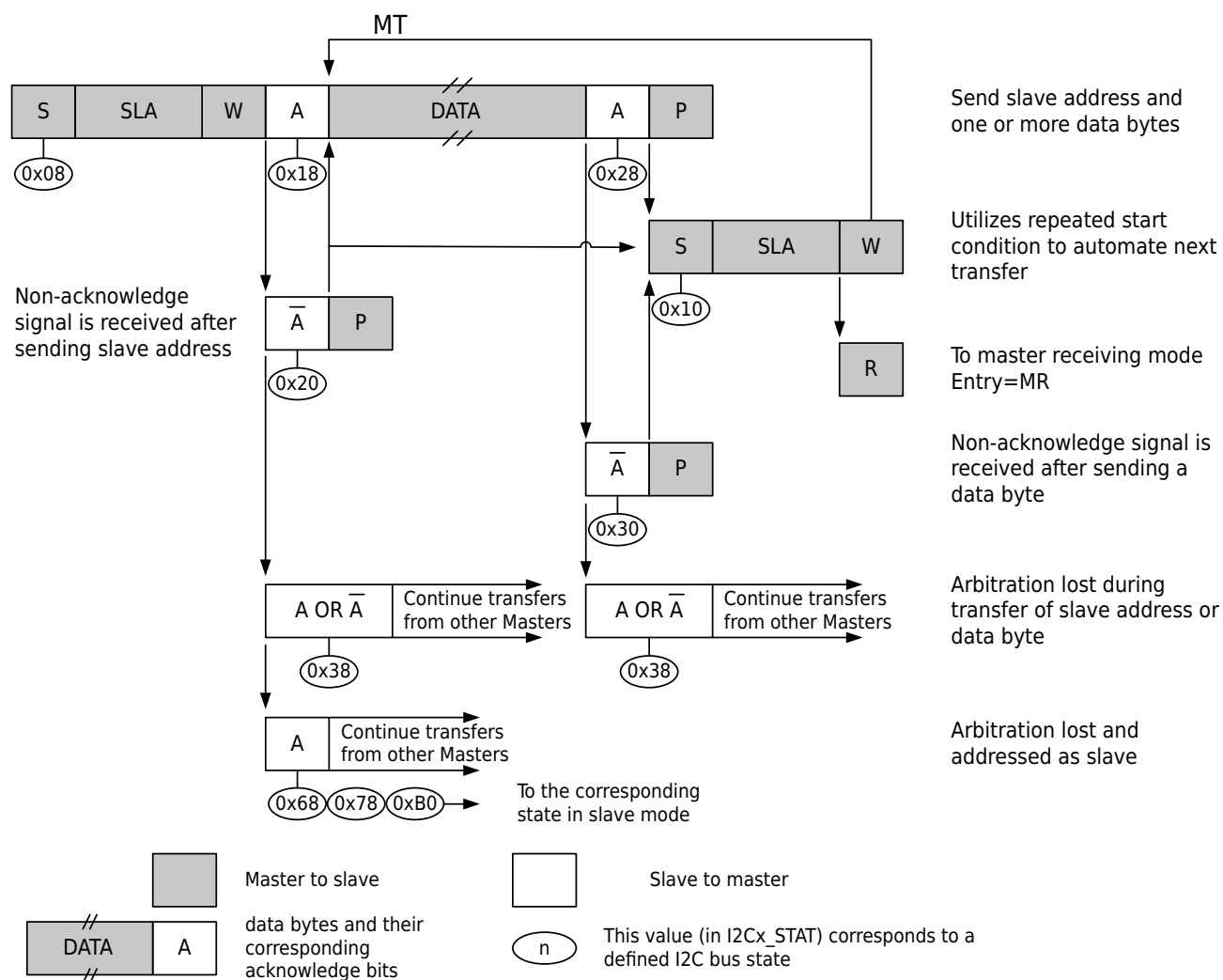


Figure 17-8 I2C Master Sending State Diagram

● 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.

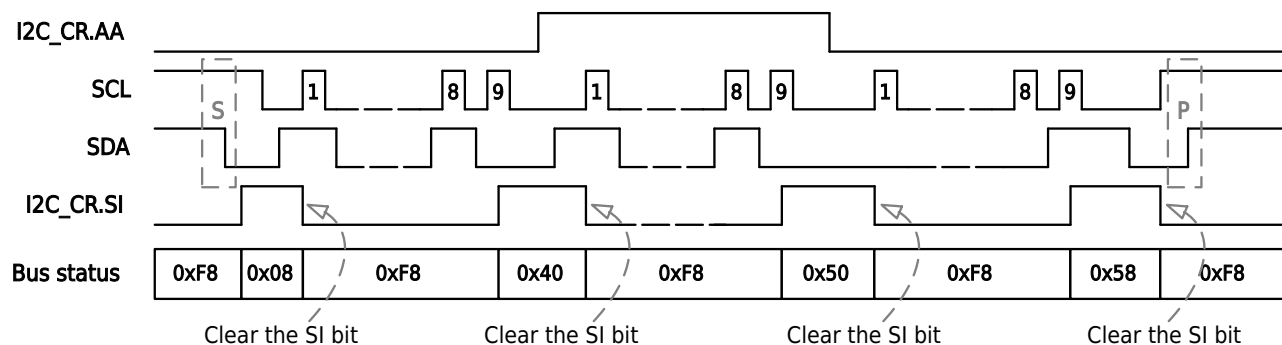


Figure 17-9 Data Synchronization Diagram of Master Receiving Mode

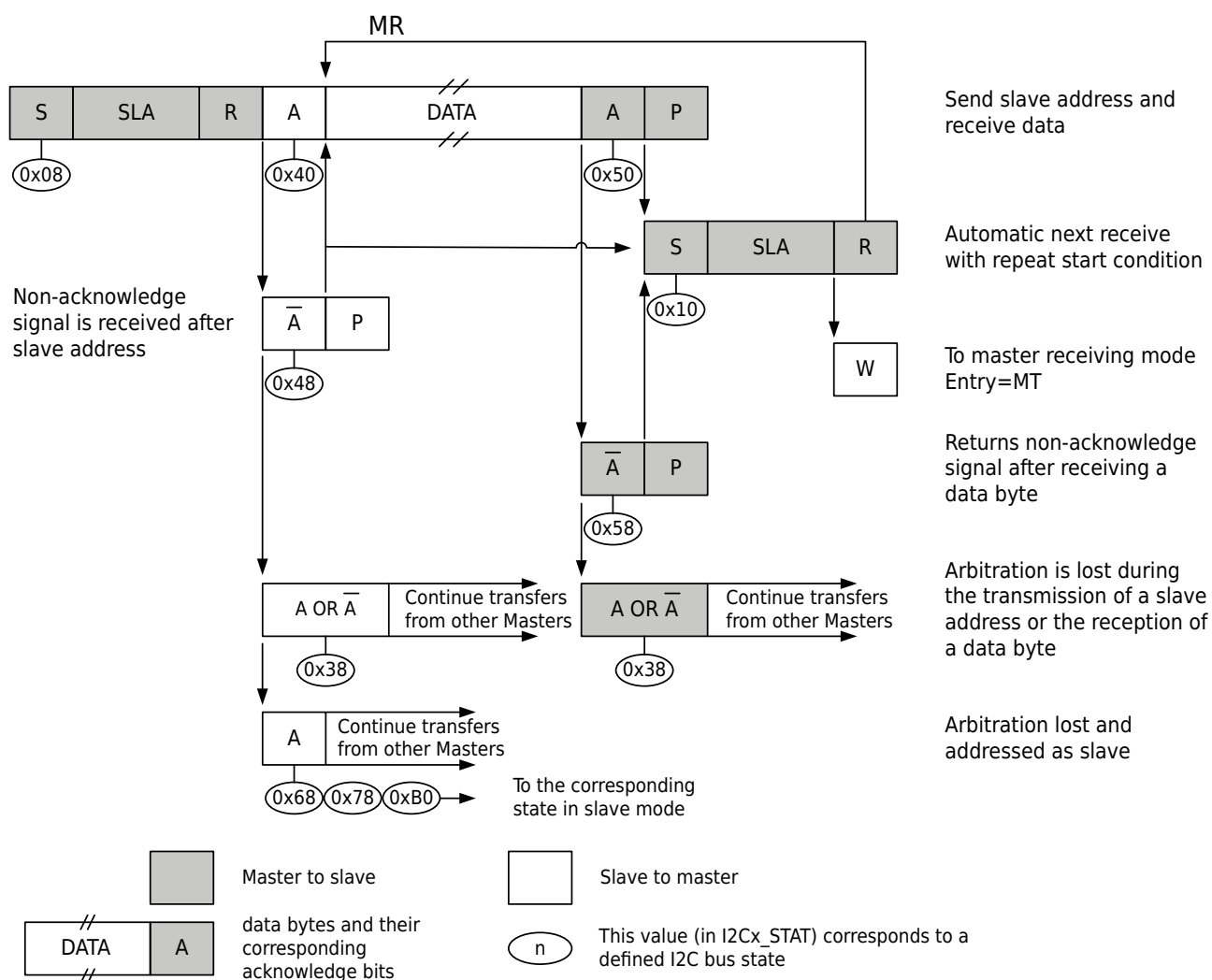


Figure 17-10 I2C Master Receiving State Diagram

● 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.

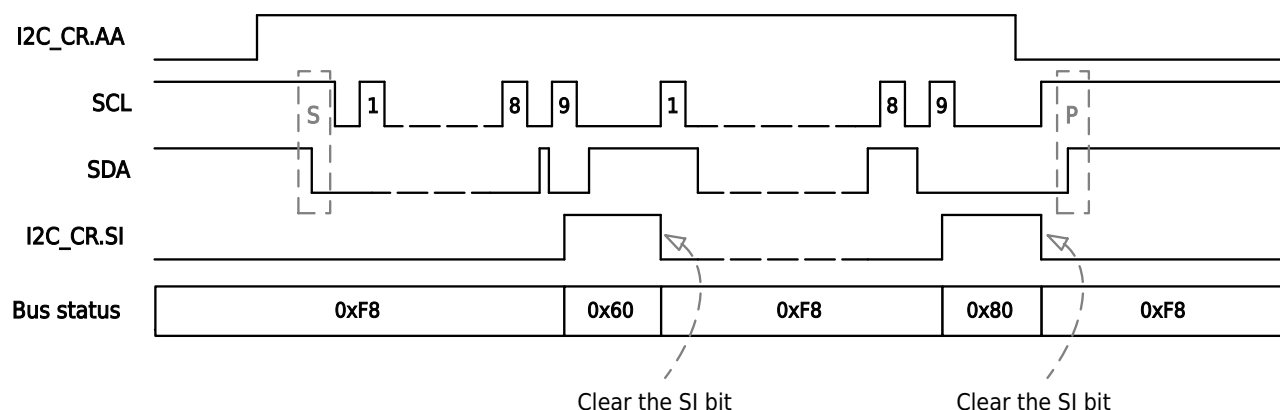


Figure 17-11 Slave Receiving Mode Data Synchronization Diagram

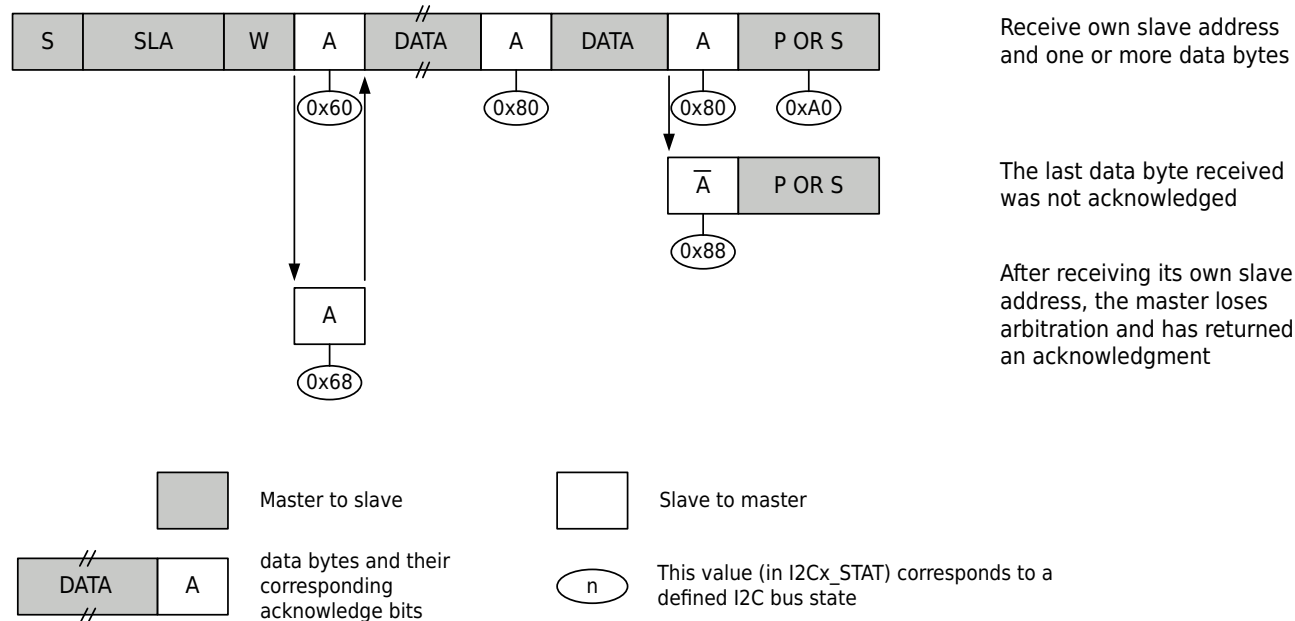


Figure 17-12 Slave Receiving State Diagram

● 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.

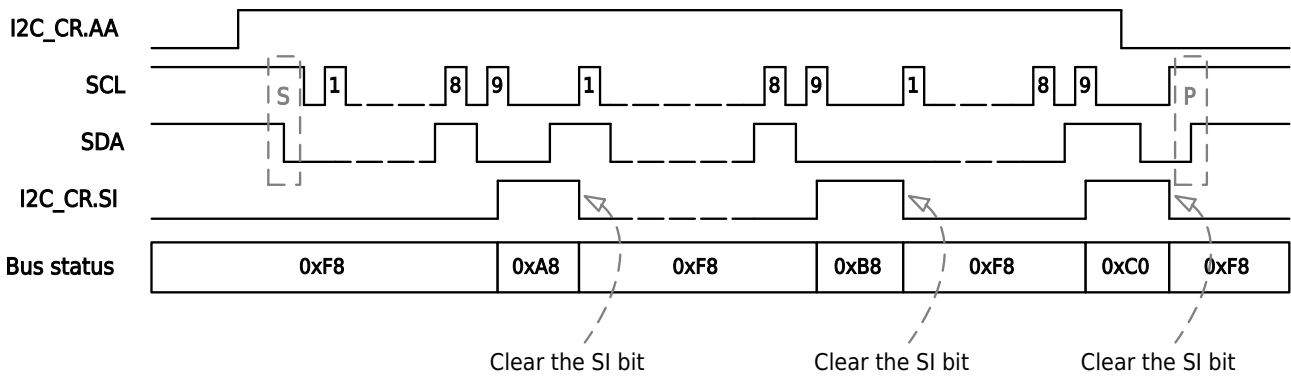


Figure 17-13 Slave Sending Mode Data Synchronization Diagram

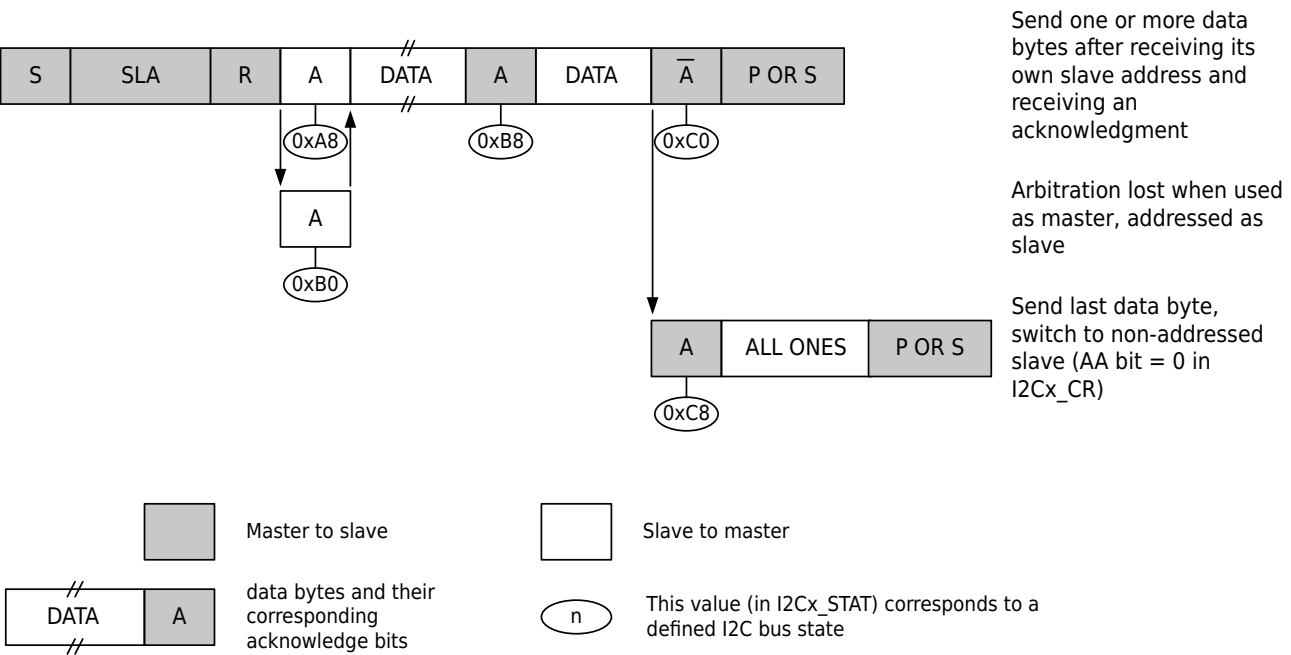


Figure 17-14 I2C Slave Sending State Diagram

● 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.

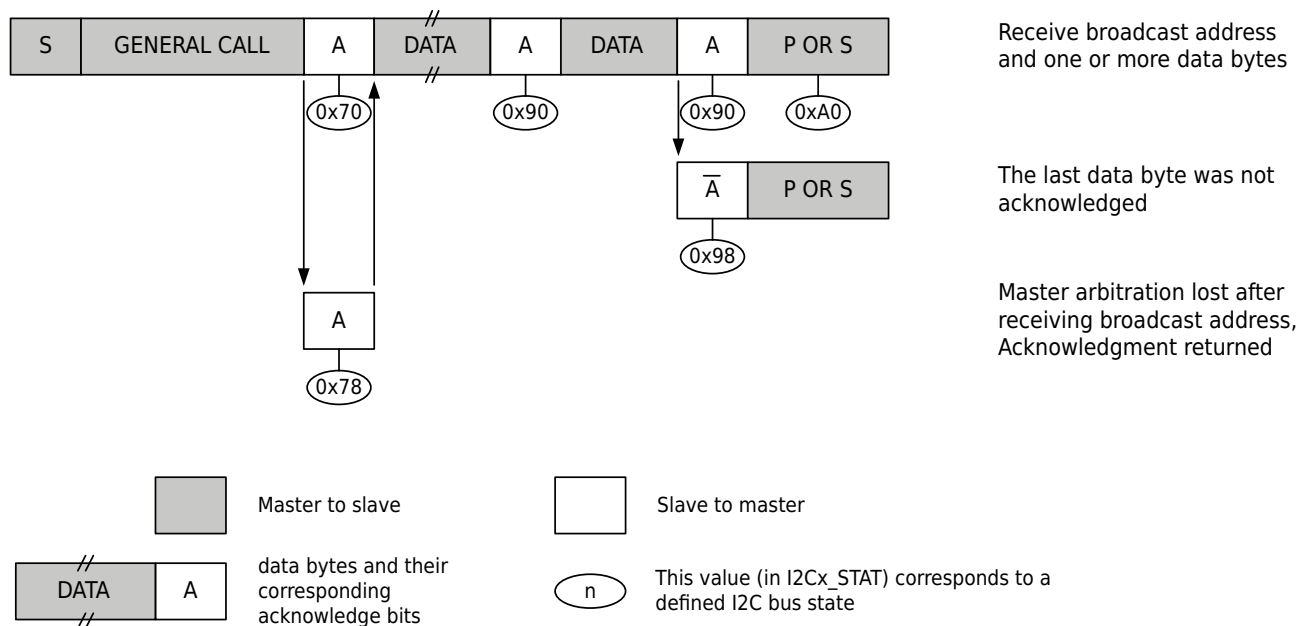


Figure 17-15 I2C Broadcast Call State Diagram

17.4.8 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 17-2 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

Status Code	Description
Master Receiving Mode	
0x08	Start condition sent
0x10	Repeated start condition sent
0x38	Loss of arbitration in NACK (that is, the master detects that SDA has been pulled low by another device when responding with ACK)
0x40	SLA+R sent, ACK received
0x48	SLA+R sent, NACK received
0x50	Data byte has been received, ACK has been returned
0x58	Data byte has been received, NACK has been returned
Slave Receiving Mode	
0x60	Has received its own SLA+W and returned ACK
0x68	When the master is in SLA+ read and write lost arbitration, has received its own SLA+W, and has returned ACK
0x80	The previous addressing used its own slave address, received data bytes, and returned ACK
0x88	The previous addressing used its own slave address, received data bytes, and returned NACK
0xA0	During static addressing, when the stop condition or the repeated start condition is received.
Slave Sending Mode	
0xA8	Received its own SLA+R and returned ACK
0xB0	When the master loses arbitration, has received its own SLA+R, and has returned ACK
0xB8	Data sent, ACK received
0xC0	Data bytes sent, NACK received
0xC8	Loaded data bytes have been sent, ACK has been received
Broadcast Call Mode	
0x70	Received broadcast address (0x00), returned ACK
0x78	When the master is in SLA+ read and write lost arbitration, the broadcast address has been received, and ACK has been returned
0x90	The previous addressing used the broadcast address, data bytes have been received, and ACK has been returned
0x98	The previous addressing used the broadcast address, data bytes have been received, and NACK has been returned
0xA0	During static addressing, when the stop condition or the repeated start condition is received.
Other Miscellaneous Status	

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

17.5 Operation Example

17.5.1 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 module register PERICKEN0.I2Cx to 1 to enable the I2C configuration clock and working clock.
- Step3.** Write 0 and 1 to RESET module register PERIRESET0.I2Cx in sequence to reset the I2C module.
- Step4.** Configure I2C_BRR, so that the clock rate of SCL meets the application requirements.
- Step5.** Set I2C_BRREN to 1 to enable the SCL clock generator.
- Step6.** Set I2C_CR.EN 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.

17.5.2 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.

**Notice**

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

- Step2.** Set SYSCTRL module register PERICKEN0.I2Cx to 1 to enable the I2C configuration clock and working clock.
- Step3.** Write 0 and 1 to RESET module register PERIRESET0.I2Cx in sequence to reset the I2C module.
- Step4.** Configure I2C_BRR, so that the clock rate of SCL meets the application requirements.
- Step5.** Set I2C_BRREN to 1 to enable the SCL clock generator.
- Step6.** Set I2C_CR.EN 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-acknowledge 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.

17.5.3 Slave Sending Example

Procedure

- Step1.** Refer to the relevant description of the port digital multiplexing function in the "General-Purpose I/O (GPIO)" chapter. Map SCL and SDA to the required pins; and configure the SCL and SDA pins to the open-drain output mode.
- Step2.** Set SYSCTRL module register PERICKEN0.I2Cx to 1 to enable the I2C configuration clock and working clock.
- Step3.** Write 0 and 1 to RESET module register PERIRESET0.I2Cx in sequence to reset the I2C module.
- Step4.** Set I2C_CR.EN to 1 to enable the I2C module.
- Step5.** Configure I2C_ADDRx as the slave address.
- Step6.** Set I2C_CR.AA to 1 to enable the acknowledge flag.
- Step7.** Wait for I2C_CR.SI to become 1 and be addressed by SLA+R;
- Step8.** Query I2C_STAT, if the value of this register is 0xA8, continue to the next step, otherwise, perform error handling.
- Step9.** Write the data to be sent to I2C_DATA, set I2C_CR.SI to 0, and send the data.
- Step10.** Wait for I2C_CR.SI to become 1, the data has been sent to the bus.

- Step11.** Query I2C_STAT, if the value of this register is 0xB8 or 0xC0, continue to the next step, otherwise, perform error handling.
- Step12.** If the data to be sent is not completed, then jump to step9 to continue execution.
- Step13.** Set I2C_CR.AA to 0 and set I2C_CR.SI to 0.

17.5.4 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 module register PERICKEN0.I2Cx to 1 to enable the I2C configuration clock and working clock.
- Step3.** Write 0 and 1 to RESET module register PERIRESET0.I2Cx in sequence to reset the I2C module.
- Step4.** Set I2C_CR.EN 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.

17.6 Register Description

17.6.1 Register List

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

Table 17-3 I2C Bus (I2C) Register Offset

Offset	Register	Description	Width
0x00	I2C_BRREN	I2C Baud Rate Generator Enable Register	32
0x04	I2C_BRR	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

17.6.2 Baud Rate Generator Enable Register (I2C_BRREN)

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																															EN
Access																																RW

Bit	Name	Functional Description	Access
31:1	Reserved	-	-
0	EN	Baud rate generator enable control 0b0: Disable 0b1: Enable	RW

17.6.3 Baud Rate Counter Configuration Register (I2C_BRR)

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																								BRR							
Access																																

Bit	Name	Functional Description	Access
31:8	Reserved	-	-
7:0	BRR	I2C bus SCL baud rate configuration $f_{SCL} = f_{PCLK}/8/(BRR+1)$, and $BRR > 0$	RW

17.6.4 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																								EN		STA		STO		SI		AA		Reserved		FLT	
Access																									RW		RW		RW		RW		RW				RW	

Bit	Name	Description	Access
31:7	Reserved	-	-
6	EN	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	-	-

Bit	Name	Description	Access
0	FLT	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

17.6.5 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																								DR							
Access																																

Bit	Name	Functional Description	Access
31:8	Reserved	-	-
7:0	DR	Data register ● In send mode, write the data to be sent ● In receive mode, read out received data	RW

17.6.6 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

17.6.7 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

17.6.8 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																								ADDR							Reserved
Access																									RW							

Bit	Name	Functional Description	Access
31:8	Reserved	-	-
7:1	ADDR	I2C slave mode address 1	RW
0	Reserved	-	-

17.6.9 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													ADDR				Reserved														
Access														RW																		

Bit	Name	Functional Description	Access
31:8	Reserved	-	-
7:1	ADDR	I2C slave mode address 2	RW
0	Reserved	-	-

17.6.10 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.

18 ADC

18.1 Introduction

External analog signals need to be converted into digital signals to be further processed by the MCU. This product integrates an 10-bit high-precision, high-speed successive approximation register (SAR ADC) module.

18.2 Features

- 10-bit conversion accuracy
- 1Msps conversion speed
- Up to 15 input channels are supported, including 14 external pin inputs, 1 internal built-in BGR 0.9V voltage channel



Note

By selecting AVCC as the reference source and measuring the BGR 0.9V, the AVCC voltage can be calculated.

- 2 reference sources: AVCC voltage, ExRef pins
- ADC voltage input range: 0-Vref
- Support single conversion and continuous conversion
- Support on-chip peripherals to automatically trigger ADC conversion, effectively reducing chip power consumption and improving real-time conversion

18.3 Functional Description

18.3.1 ADC Block Diagram

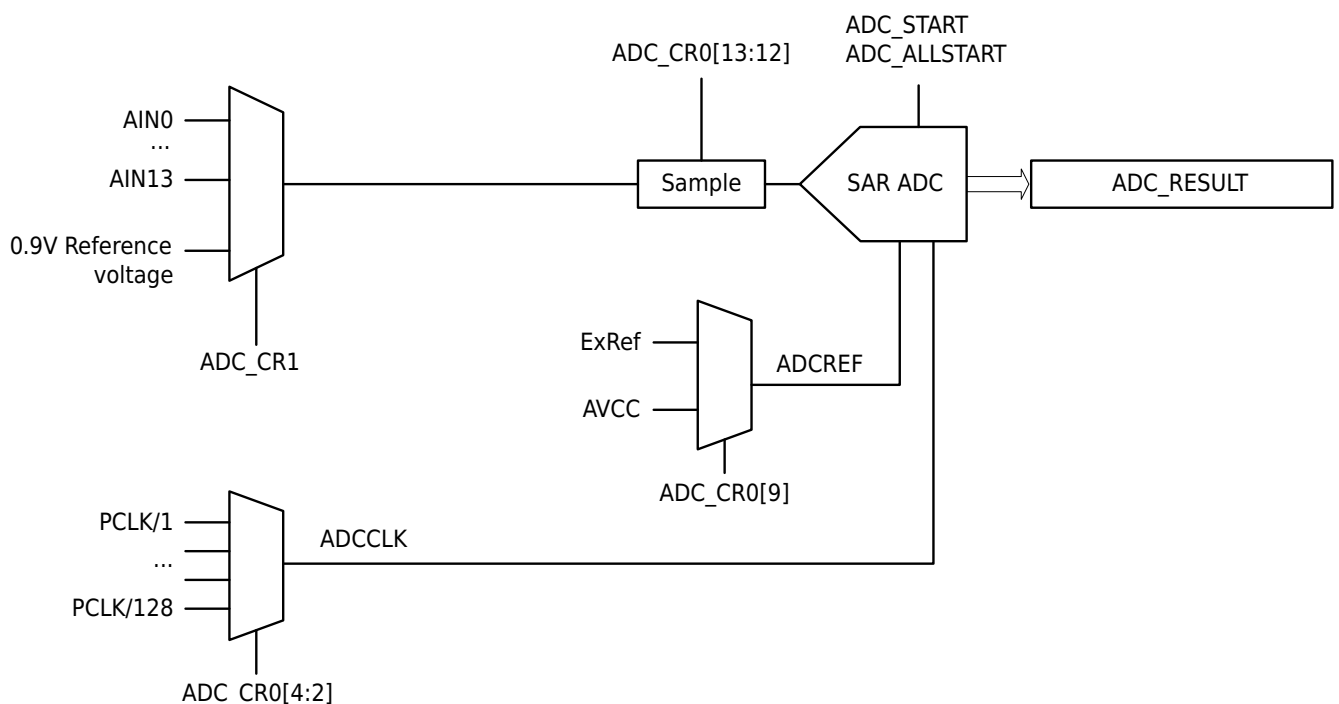


Figure 18-1 ADC Block Diagram

18.3.2 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 6-12 ADCCLK cycles, configured by ADC_CR0.SAM; the successive comparison process requires 13 ADCCLK cycles. Therefore, an ADC conversion requires a total of 19-25 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.

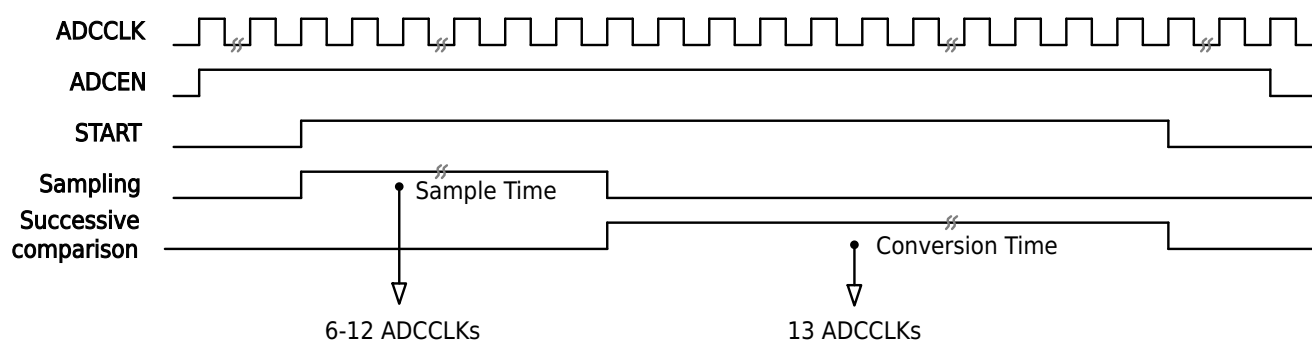


Figure 18-2 ADC Conversion Timing Diagram

The ADC conversion speed is related to the ADC reference voltage and AVCC voltage. The maximum conversion speed is shown in the following table:

ADC Reference Voltage	AVCC Voltage	Maximum Conversion Speed	Maximum ADCCLK Frequency
AVCC/ExRef	1.8-2.4V	200ksps	4MHz
AVCC/ExRef	2.4-2.7V	500ksps	16MHz
AVCC/ExRef	2.7-5.5V	1Msps	24MHz

18.3.3 Single Conversion Mode

In the single conversion mode, after ADC startup, only one conversion of the configured channel is performed. This mode can be started either by setting the ADC_START.START bit for software trigger or by configuring an external trigger via ADC_EXTTRIGGER. ADC_IFR.EOC bit is automatically set to 1, and the conversion result is saved in the ADC_RESULT register.

18.3.4 ADC Conversion External Trigger Source

ADC conversions can be initiated either by software configuration or by an external trigger.

The ADC_EXTTRIGGER registers can be configured to set the external trigger source for ADC conversions.

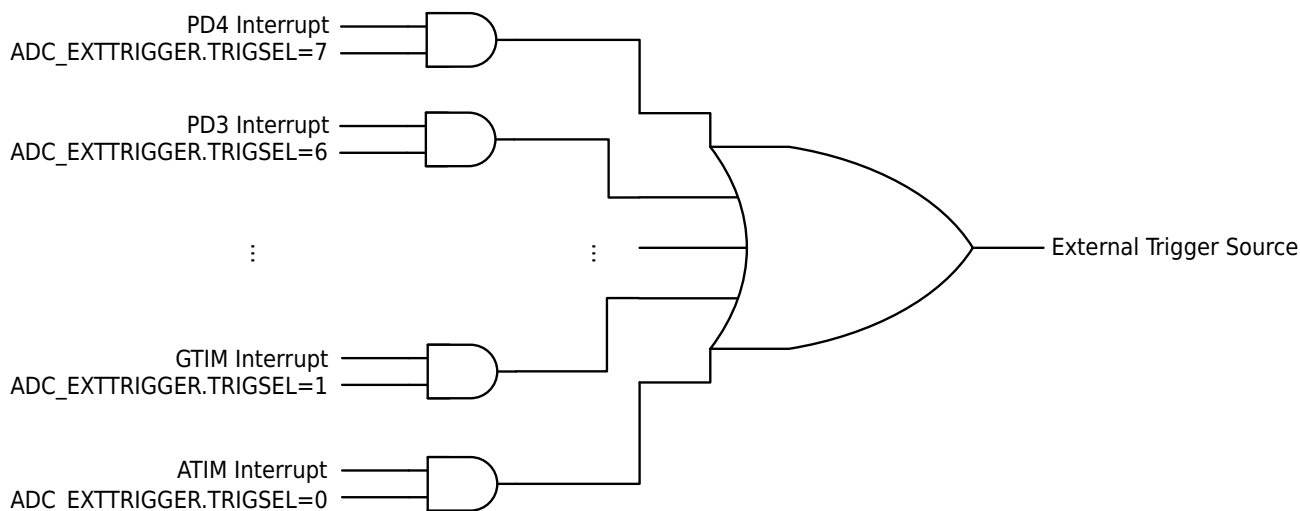


Figure 18-3 ADC External Trigger Source Diagram

18.3.5 Continuous Conversion Mode

When ADC_ALLSTART.START is set to 1, the ADC module operates in continuous conversion mode; when set to 0, the ADC module exits continuous conversion mode.

In continuous conversion mode, the ADC module continuously performs ADC conversions. After each conversion completes, the ADC_IFR.EOC bit is automatically set to 1, and the result is stored in the ADC_RESULT register. After the first conversion completes, the user program can read the ADC_RESULT register at any time to obtain the current ADC conversion result; each read of ADC_RESULT clears the ADC_IFR.EOC flag.

18.3.6 Measure AVCC Voltage

Set the ADC reference source to AVCC and configure the ADC input channel to the internal BGR 0.9V. The AVCC voltage can be calculated from the ADC conversion result. Note that under this condition, the ADC conversion speed should not exceed 200 ksps. Under this condition, the AVCC voltage calculation formula is:

$$V_{AVCC} = \frac{V_{BGR} \times 1023}{ADC_RESULT}$$

Where V_{BGR} is the internal reference voltage, with a typical value of 0.9V.

Calculation Example:

- When ADC_Result is 279, the AVCC voltage is:

$$\frac{0.9 \times 1023}{279} = 3.3V$$

- When ADC_Result is 460, the AVCC voltage is:

$$\frac{0.9 \times 1023}{460} = 2.0V$$

18.3.7 ADC Interrupt

The ADC interrupt request is shown in the table below:

Table 18-2 ADC Interrupt

Interrupt Source	Interrupt Flag	Interrupt Enable
ADC single conversion completed	ADC_IFR.EOC	ADC_IER.EOC

18.4 Operation Example

18.4.1 ADC Single Conversion Operation Flow

Procedure

- Step1.** Configure the corresponding bit of GPIO_ADS to configure the ADC channel to be converted as an analog port.
- Step2.** Set GPIOC_ADS.Bit3 to 1 to configure the ADC external reference voltage pin as an analog port.

**Note**

If the ADC reference voltage does not select an external reference voltage pin, this step can be skipped.

- Step3.** Set ADC_CR0.EN to 1 to enable the ADC module.
- Step4.** Configure ADC_CR0.REF to select the reference voltage of ADC.
- Step5.** Configure ADC_CR0.SAM and ADC_CR0.CKDIV to set the conversion speed of ADC.
- Step6.** Delay 5 μ s, wait for the ADC module to start up.
- Step7.** Configure ADC_CR1.CHSEL to select the channel to be converted.
- Step8.** Set ADC_ICR to 0 and clear ADC_IFR flag.
- Step9.** Set ADC_START.START to 1 to start ADC single conversion.
- Step10.** Wait for ADC_IFR.EOC to become 1, read the ADC_RESULT register to get the ADC conversion result.
- Step11.** To convert other channels, repeat steps7-10.
- Step12.** Set ADC_CR0.EN to 0 to turn off ADC module.

18.4.2 ADC Continuous Conversion Operation Flow

Procedure

- Step1.** Configure the corresponding bit of GPIO_ADS to configure the ADC channel to be converted as an analog port.
- Step2.** Set GPIOC_ADS.Bit3 to 1 to configure the ADC external reference voltage pin as an analog port.

**Note**

If the ADC reference voltage does not select an external reference voltage pin, this step can be skipped.

- Step3.** Set ADC_CR0.EN to 1 to enable the ADC module.
- Step4.** Configure ADC_CR0.REF to select the reference voltage of ADC.
- Step5.** Configure ADC_CR0.SAM and ADC_CR0.CKDIV to set the conversion speed of ADC.
- Step6.** Delay 5 μ s, wait for the ADC module to start up.
- Step7.** Configure ADC_CR1.CHSEL to select the channel to be converted.
- Step8.** Set ADC_ALLSTART.START to 1 to start ADC continuous conversion.
- Step9.** After the first ADC conversion completes, the ADC_RESULT register can be read at any time to obtain the real-time ADC conversion result.
- Step10.** Set ADC_ALLSTART.START to 0 to exit continuous conversion mode.
- Step11.** Set ADC_CR0.EN to 0 to turn off ADC module.

18.5 Register Description

18.5.1 Register List

Base address: 0x40002400

Table 18-3 Analog-to-Digital Converter (ADC) Register Offset

Offset	Register	Description	Width
0x004	ADC_CR0	ADC Basic Configuration Register 0	32
0x008	ADC_CR1	ADC Basic Configuration Register 1	32
0x0A0	ADC_RESULT	ADC Conversion Results Register	32
0x0B0	ADC_IFR	ADC Interrupt Flag Register	32
0x0B4	ADC_IER	ADC Interrupt Enable Register	32
0x0B8	ADC_ICR	ADC Flag Clear Register	32
0x0BC	ADC_EXTTRIGGER	ADC Conversion External Interrupt Trigger Source Configuration Register	32
0x0C0	ADC_START	ADC Conversion Start Control Register	32
0x0CC	ADC_ALLSTART	ADC Continuous Conversion Start Control Register	32

18.5.2 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	0x00002600																															
Name	Reserved																		SAM	Reserved	REF	Reserved					CKDIV		Reserved	EN		
Access																			RW		RW						RW				RW	

Bit	Name	Functional Description	Access
31:14	Reserved	-	-
13:12	SAM	ADC sampling period selection ● 0b00: 6 ADCCLK cycles ● 0b01: 8 ADCCLK cycles ● 0b10: 11 ADCCLK cycles ● 0b11: 12 ADCCLK cycles  Note The successive comparison process alone requires 13 ADCCLK cycles.	RW
11	Reserved	-	-
10:9	REF	ADC reference voltage selection ● 0bX0: External reference voltage ExRef (PC03) ● 0bX1: AVCC voltage	RW
8:5	Reserved	-	-
4:2	CKDIV	ADC clock selection ● 0b000: PCLK clock ● 0b001: PCLK clock divided by 2 ● 0b010: PCLK clock divided by 4 ● 0b011: PCLK clock divided by 8 ● 0b100: PCLK clock divided by 16 ● 0b101: PCLK clock divided by 32 ● 0b110: PCLK clock divided by 64 ● 0b111: PCLK clock divided by 128	RW
1	Reserved	-	-
0	EN	ADC enable control ● 0b0: Disable ● 0b1: Enable	RW

18.5.3 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	0x00000000																															
Name	Reserved																															CHSEL
Access																																

Bit	Name	Functional Description	Access
31:4	Reserved	-	-
3:0	CHSEL	Channel selection ● 0b0000: Channel 0, Pin PC04 ● 0b0001: Channel 1, Pin PC06 ● 0b0010: Channel 2, Pin PD02 ● 0b0011: Channel 3, Pin PD03 ● 0b0100: Channel 4, Pin PD04 ● 0b0101: Channel 5, Pin PD05 ● 0b0110: Channel 6, Pin PD06 ● 0b0111: Channel 7, Pin PA01 ● 0b1000: Channel 8, Pin PA02 ● 0b1001: Channel 9, Pin PB05 ● 0b1010: Channel 10, Pin PB04 ● 0b1011: Channel 11, Pin PB02 ● 0b1100: Channel 12, Pin PB01 ● 0b1101: Channel 13, Pin PB00 ● 0b1110: Channel 14 connected to internal 0.9V reference voltage ● Others: Setting prohibited	RW

18.5.4 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  Note Reading ADC_RESULT clears ADC_IFR.EOC.	RO

18.5.5 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																															EOC
Access																																RO

Bit	Name	Functional Description	Access
31:1	Reserved	-	-
0	EOC	ADC single conversion complete flag 0b0: ADC single conversion not complete 0b1: ADC single conversion complete	RO

18.5.6 ADC Interrupt Enable Register (ADC_IER)

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	0x00000000																																
Name	Reserved																																EOC
Access																																	RW

Bit	Name	Functional Description	Access
31:1	Reserved	-	-
0	EOC	ADC single conversion complete interrupt enable 0b0: Disable 0b1: Enable	RW

18.5.7 ADC Flag Clear Register (ADC_ICR)

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	0x00000001																																
Name	Reserved																																EOC
Access																																	WO

Bit	Name	Functional Description	Access
31:1	Reserved	-	-
0	EOC	ADC single conversion complete flag clear Write 0 to clear flag, write 1 has no effect; read always returns 1.	WO

18.5.8 ADC External Interrupt Trigger Source Configuration Register (ADC_EXTTRIGGER)

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																												TRIGSEL			
Access																													RW			

Bit	Name	Functional Description	Access
31:4	Reserved	-	-
3:0	TRIGSEL	ADC trigger source selection ● 0b0000: Trigger signal output from ATIM3; see the related description of the ATIM3_ADTR register for details. ● 0b0001: GTIM overflow signal ● 0b0010: Signal of PB03 interrupt flag rising from 0 to 1 ● 0b0011: Signal of PB04 interrupt flag rising from 0 to 1 ● 0b0100: Signal of PC03 interrupt flag rising from 0 to 1 ● 0b0101: Signal of PC04 interrupt flag rising from 0 to 1 ● 0b0110: Signal of PD03 interrupt flag rising from 0 to 1 ● 0b0111: Signal of PD04 interrupt flag rising from 0 to 1 ● Others: Setting prohibited  Note When the selected trigger source is a GPIO, the user program should promptly clear the interrupt flag of the trigger source.	RW

18.5.9 ADC Conversion Start Control Register (ADC_START)

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 conversion start control 0b0: Stop 0b1: Start ADC conversion and automatically clears after a single conversion completes	RW

18.5.10 ADC Continuous Conversion Start Control Register (ADC_ALLSTART)

Offset	Bit Position																																
0x0CC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reset	0x00000000																																
Name	Reserved																																START
Access																																	RW

Bit	Name	Functional Description	Access
31:1	Reserved	-	-
0	START	ADC continuous conversion control 0b0: Stop ADC conversion and need to wait until the end of the current conversion 0b1: Start ADC to convert all the time	RW

19 LVD

19.1 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. This module can operate normally in Active, Sleep and DeepSleep modes.

19.2 Features

- 4 monitoring sources: AVCC, PA03, PC03, PD04
- 16-stage threshold voltage, 1.8-3.3V 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

19.3 Functional Description

19.3.1 LVD Block Diagram

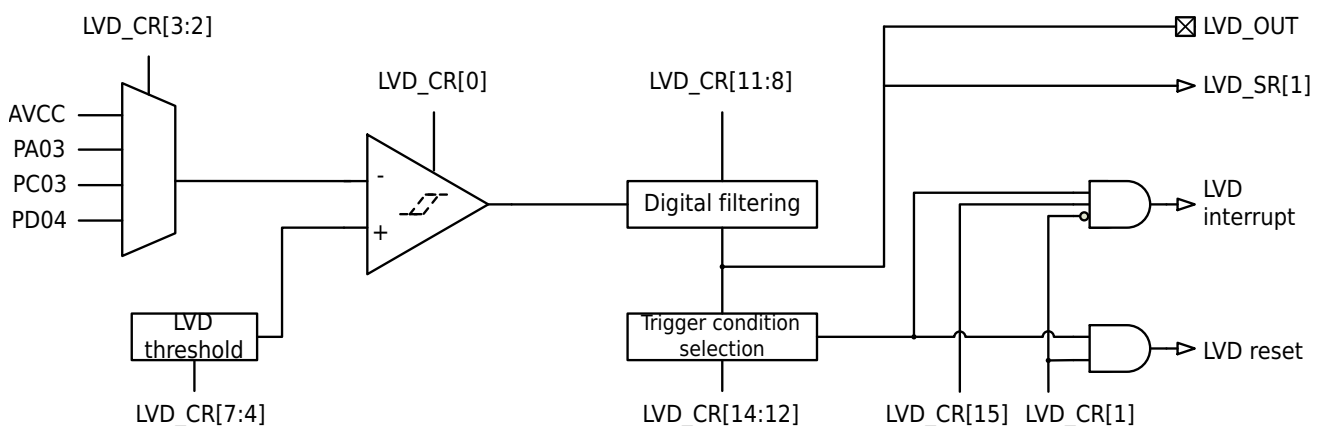


Figure 19-1 LVD Block Diagram

19.3.2 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.FLTTIME 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.

Enable the digital filter module, the filter illustration is as follows:

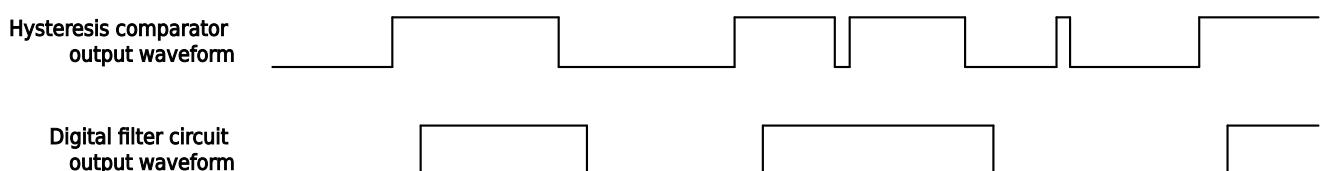


Figure 19-2 LVD Filter Output

19.3.3 Hysteresis Function

The built-in voltage comparator of LVD has a hysteresis function, which can enhance the anti-interference ability of the chip. When LVD is disabled, the output signal of LVD is low level; when LVD is enabled, 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:

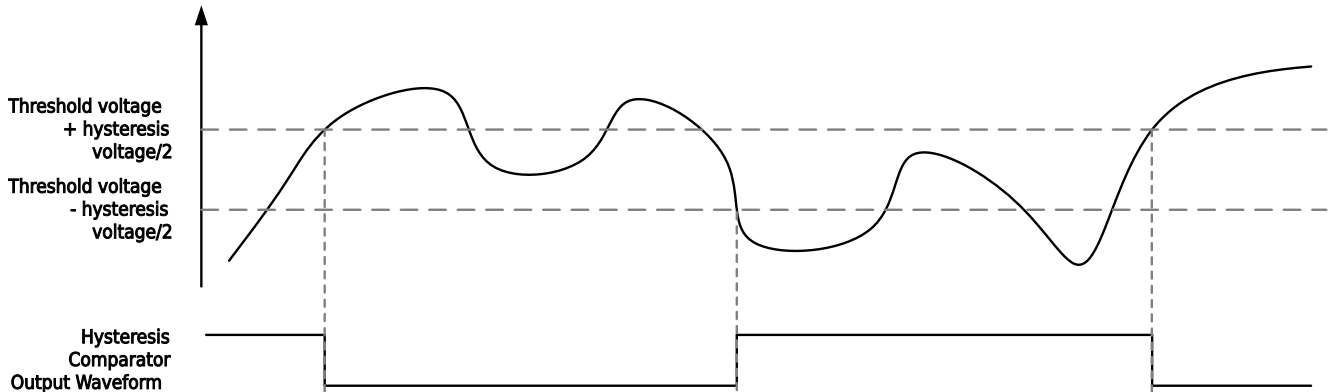


Figure 19-3 LVD Hysteresis Response

19.4 Operation Example

19.4.1 LVD Configured as Low Voltage Reset

Context

In this mode, the MCU is reset when the monitored voltage is lower than the threshold voltage.



Note

When initializing the user program, reading the reset flag register can determine whether the current reset source is LVD.

Procedure

- Step1.** Refer to the relevant description of the pin digital multiplexing function in the "General-Purpose I/O (GPIO)" chapter to configure the pin of the voltage source to be monitored as an analog function.



Note

If the voltage source to be monitored is AVCC, this step can be ignored.

- Step2.** Configure LVD_CR.SOURCE to select the voltage source to be monitored.
Step3. Configure LVD_CR.VTDS to select the threshold voltage of LVD.
Step4. Configure LVD_CR.FLTTIME to select the LVD filter time.
Step5. Set LVD_CR.FLTEN to enable LVD filtering.
Step6. Set LVD_CR.HTEN to 1, select high level to trigger LVD action.
Step7. Set LVD_CR.ACT to 1, select LVD action as reset.
Step8. Set LVD_CR.EN to 1 to enable LVD.

19.4.2 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.** Refer to the relevant description of the pin digital multiplexing function in the "General-Purpose I/O (GPIO)" chapter to configure the pin of the voltage source to be monitored as an analog function.

**Note**

If the voltage source to be monitored is AVCC, this step can be ignored.

- Step2.** Configure LVD_CR.SOURCE to select the voltage source to be monitored.
- Step3.** Configure LVD_CR.VTDS to select the threshold voltage of LVD.
- Step4.** Configure LVD_CR.FLTTIME to select the LVD filter time.
- Step5.** Set LVD_CR.FLTEN to enable LVD filtering.
- Step6.** Set LVD_CR.RTEN and LVD_CR.FTEN to 1, select level change to trigger LVD action.
- Step7.** Set LVD_CR.ACT to 0, select LVD action as interrupt.
- Step8.** Set LVD_CR.IE to 1 to enable LVD interrupt.
- Step9.** Enable the LVD interrupt in the NVIC interrupt vector table.
- Step10.** Set LVD_CR.EN to 1 to enable LVD.
- Step11.** 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.

19.5 Register Description

19.5.1 Register List

Base address: 0x40002680

Table 19-1 Low Voltage Detector (LVD) Register Offset

Offset	Register	Description	Width
0x000	LVD_CR	LVD Configuration Register	32
0x004	LVD_SR	LVD Status Register	32

19.5.2 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
Reset	0x00000000															
Name	Reserved												IE	HTEN	RTEN	FTEN
Access													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	FLTTIME	Digital filter time configuration When the filter clock is RCL 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

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 ● 0b0000: 1.8V ● 0b0001: 1.9V ● 0b0010: 2.0V ● 0b0011: 2.1V ● 0b0100: 2.2V ● 0b0101: 2.3V ● 0b0110: 2.4V ● 0b0111: 2.5V ● 0b1000: 2.6V ● 0b1001: 2.7V ● 0b1010: 2.8V ● 0b1011: 2.9V ● 0b1100: 3.0V ● 0b1101: 3.1V ● 0b1110: 3.2V ● 0b1111: 3.3V	RW
3:2	SOURCE	LVD monitoring source selection ● 0b00: AVCC power voltage ● 0b01: PA03 port input voltage ● 0b10: PC03 port input voltage ● 0b11: PD04 port input voltage	RW
1	ACT	LVD trigger action selection ● 0b0: NVIC interrupt ● 0b1: System reset	RW
0	EN	LVD enable control ● 0b0: Disable ● 0b1: Enable	RW

19.5.3 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																														FLTV		INTF
Access																															RO	RW	

Bit	Name	Functional Description	Access
31:2	Reserved	-	-
1	FLTV	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

20 Device Electronic Signature

20.1 Introduction

The electronic signature is stored in the system storage area of the flash memory module. 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.

20.2 Functional Description

20.2.1 Product Unique Identifier (UID)

Typical application scenarios for unique identifier (UID):

- Used as a serial number
- When programming internal flash memory, it can be utilized as a key or cryptographic primitive to enhance code security
- 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.

20.2.2 One-Time Programming Area (OTP Area)

The chip has a 128-byte OTP storage area with an address range of 0x00100F00-0x00100F7F. Data in this area can only be written through ISP instructions and can only be written once. Neither chip erase nor page erase operations can erase the data in this area. User programs can read the data in this area through read operations.

Based on the characteristics of the OTP area, user programs can store unchangeable information in this OTP area.

20.3 Register Description

20.3.1 Product Unique Identifier (UID) Register (80 bits)

Base address: 0x00101080

Offset	Description	Bit (80bits)								Remark
		7	6	5	4	3	2	1	0	
0	Y coordinate	UID[7:0]								0-127
1	X coordinate	UID[15:8]								0-127
2	Wafer Number	UID[23:16]								1-25
3	LOT Number 0	UID[31:24]								-
4	LOT Number 1	UID[39:32]								-
5	LOT Number 2	UID[47:40]								-
6	RFU	UID[55:48]								-
7	RFU	UID[63:56]								-
8	RFU	UID[71:64]								-
9	RFU	UID[79:72]								-

20.3.2 Product Model Register

0x00100D90-0x00100DAF 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

4843333246303032433450422D5453534F503230000000000000000000000000 is HC32F002C4PB-TSSOP20.

20.3.3 FLASH Capacity Register

Base address: 0x00100DB0

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 0x00004800 means the Flash capacity is 18K Byte	RO

20.3.4 RAM Capacity Register

Base address: 0x00100DB4

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 0x00000800 means the RAM capacity is 2K Byte	RO

20.3.5 Pin Count/Flash Sector Size Configuration Register

Base address: 0x00100DB8

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 0x0014 means the number of product pins is 20	RO
15:0	FLASHSEC-TORSIZE	Flash sector size In unit of Byte 0x0200 represents 512Byte.	RO

21 SWD Debug Interface

21.1 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't work in DeepSleep mode, please debug in Active and Sleep mode.

21.2 Functional Description

21.2.1 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.

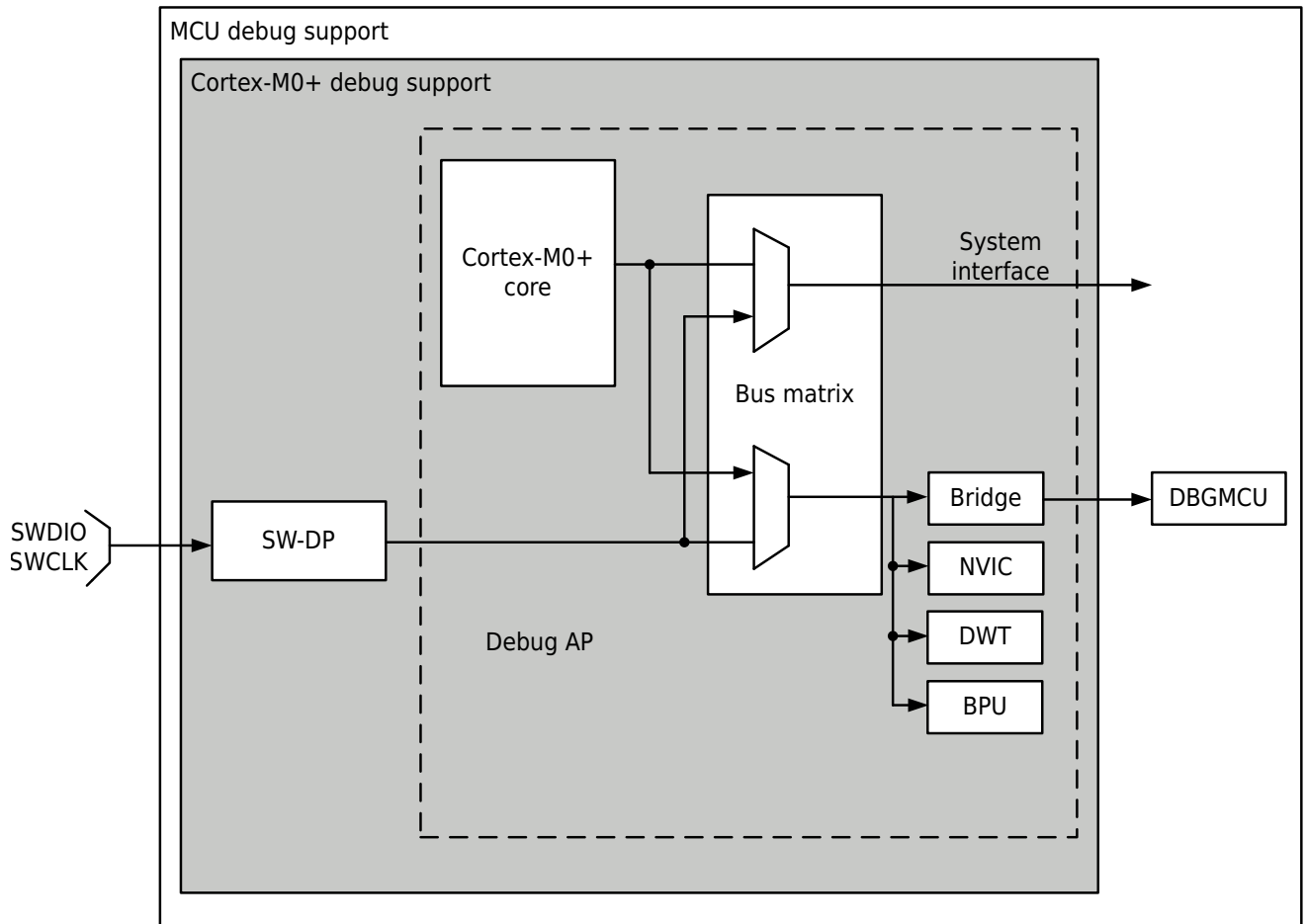


Figure 21-1 Debug Support Block Diagram

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".

21.2.2 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

21.2.3 Debug Port Pins

SWD Port Pins

The SWD interface of this series requires 2 pins, as shown in the table below.

SWD Port Name	Debugging Function	Pin Assignment
SWCLK	Serial clock	PD01
SWDIO	Serial data input/output	PC07

SW-DP Pin Assignment

If the [Level 2-3 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 or the [Level 1 Encryption Chip] option is enabled during programming, the PC07/PD01 pins will be initialized as dedicated pins for debugger use after power-on. Users can set SYSCTRL module CR1.SWDIO 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	SWDIO Configuration	PC07/PD01 Function
No encryption	0	SWD
No encryption	1	GPIO
Level 1 encryption	0	SWD (decryption only)
Level 1 encryption	1	GPIO
Level 2-3 encryption	0	-
Level 2-3 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 resistor.

21.2.4 SWD Port

21.2.4.1 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.

21.2.4.2 SWD Protocol Sequence

Each sequence consists of three phases:

1. Host transmitted data packet request (8-bit)
2. Target transmitted acknowledgment response (3-bit)
3. Data transfer phase transmitted by host or target (33-bit)

Table 21-3 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 21-4 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 21-5 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.

21.2.4.3 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+).



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".

21.2.4.4 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.

21.2.4.5 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	-

A[3:2]	RW	CTRLSEL Bit of the SELECT Register	Register	Description
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)
0b01	Read/write	1	WIRE CONTROL	Used to configure the physical serial port protocol (such as switching time duration)
0b10	Read	-	READ RESEND	Allow recovery of reads data from corrupted debug software transfers without repeating 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

21.2.4.6 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 acknowledgment)
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

Address	A[3:2] Value	Description
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).

21.2.5 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:

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)

21.2.6 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".

21.2.7 DWT (Data Watchpoint)

The Cortex-M0+ DWT implementation provides 1 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.

21.2.8 MCU Debug Component (DBG)

MCU debug component helps the debugger provide support for:

- Low power mode
- Timer and watchdog clock control during breakpoints

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.

22 SysTick Timer

22.1 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.

22.2 Operation Example

22.2.1 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.COUNT-FLAG.
 - 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

22.3 Register Description

22.3.1 Register List

Base address: 0xE000E010

Table 22-2 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_CALIB	SysTick->CALIB	SysTick Calibration Value Register	32

22.3.2 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																		

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 No reference clock is used; the system clock is always used as the SysTick clock.	RW
1	TICKINT	SysTick interrupt enable 0b0: Disable 0b1: Enable	RW
0	ENABLE	SysTick timer enable 0b0: Disable 0b1: Enable	RW

22.3.3 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	0xFFFFFFFF																															
Name	Reserved								RELOAD																							
Access									RW																							

Bit	Name	Functional Description	Access
31:24	Reserved	-	-
23:0	RELOAD	SysTick timer reload value	RW

22.3.4 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	0xFFFFFFFF																															
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

22.3.5 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	0xFFFFFFFF																																
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: Reserved 0b1: Does not support reference clock (XTL)	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

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

Revision History

Revision/Date	Changes
Rev1.30 Aug. 28, 2026	1. LPUART: In the "Multi-Machine Communication" chapter, the description of Slave Address Configuration is optimized. 2. 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. 3. SWD Debug Interface: The internal ID of SW-DP is modified to "0x0BC11477".
Rev1.22 Feb. 12, 2026	1. ATIM: In the "Mode 0 Counting Timer Function" chapter, the GATEP polarity in the functional block diagram has been modified, counting function description optimized, and ETR clock frequency constraints added; In the "Mode 2/3 Compare Capture Mode-Single Pulse Output" chapter, the pulse counting timing diagram details have been optimized with additional relevant explanations; In the "Mode 2/3 Compare Capture Mode Registers-Master/Slave Mode Control Register (TIM_MSCR)" chapter, the Bit[2:0] function description for MMS=0 now specifies ATIM3 supports "software update TG by writing M23CR.TG"; 2. 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"; 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.
Rev1.21 May 12, 2025	1. Register bit/bit field function description: The value "Reserved" is modified to "Setting prohibited". 2. SYSCTRL: The original "Reset Control" chapter content has been moved into this chapter; 3. SPI: The description regarding maximum communication speed for master/slave modes is deleted from the "Features" section. 4. ATIM: The "Reload Register/Period Mapping Address (ATIM3_ARRDM)" has been deleted. 5. ADC: The chapter title "External Trigger Conversion Mode" has been changed to "ADC Conversion External Trigger Source". 6. SysTick: In the "SysTick Calibration Value Register (CALIB)" chapter, the Bit31's 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. Storage and Architecture: A new "System Address Partition" chapter is added.

Revision/Date	Changes
Rev1.20 Dec. 25, 2024	3. Operating Modes: In the "DeepSleep mode" chapter, the description is modified from "If SysCtrl.CR1.AutoSwitch is set to 1" to "If SYSCTRL.CR1.WAKEUPCLK is set to 1". 4. SYSCTRL: In the "System Clock Introduction-Clock Types-Internal High-Speed RC Clock RCH" chapter, the description is modified from "The internal high-speed clock requires only 5μs to stabilize after wake-up" to "The internal high-speed clock stabilizes extremely quickly after wake-up (for specific timing parameters, please refer to the <Datasheet>)". 5. 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' ". 6. NVIC: In the "Interrupt Vector Table" chapter, the diagram is optimized with added interrupt #31 numbering and address information; in the "Interrupt Input and Pending Behavior" chapter, the description is modified from "The interrupt pending status can be cleared by writing to the NVIC_ICPR register, which is particularly useful when configuring peripherals as they might have generated interrupt requests prior to configuration" to "When configuring peripherals, writing to the NVIC_ICPR register clears the interrupt pending status, thereby eliminating any interrupt requests generated by the peripheral before configuration"; the diagram "Interrupt Pending Generated During Interrupt Handling Can Also Be Acknowledged" optimizes the description of interrupt request X processing; in the "Interrupt Sources" table, external interrupt source names are changed from "PORTx" to "GPIOx"; register names in the "Interrupt Structure Diagram" chapter are modified to align with register descriptions; in the "NVIC Interrupt Masking" chapter, the register is changed from "SCS_PRIMASK" to "PRIMASK". 7. FLASH: The description of register write protection in the "FLASH Security Protection" chapter is revised from "No write operations can be inserted between the two steps of writing 0x5A5A and writing 0xA5A5, otherwise the value of the target register cannot be modified." 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"; The description format of bit names in the "BYPASS Sequence Register (FLASH_BYPASS)" and "Erase/Write Protection Register (FLASH_SLOCK)" chapters has been optimized; 8. GTIM: In the "Trigger Start Mode" chapter, the description of the GTIM trigger start mode counting diagram has been changed from "Write CR.CEN=0" to "Write CR0.CEN=0"; In the "On-Chip Peripheral Interconnection" chapter, the value range of GPIOx_CR4.GTIMCH0 has been modified from "1-6" to "1-3," and the corresponding values in the table have also been updated; In the "Interrupt

Revision/Date	Changes
Rev1.20 Dec. 25, 2024	Flag Register (GTIM_IFR)" chapter, the read/write description has been changed from "RW" to "RO." 9. ATIM: In the "ATIM3 Features" chapter, the block diagram has been optimized by adding ETR's FLT/POL control and CH3 compare functionality; In the "Mode 0 Counting Timer Function" chapter, the register name CR has been changed to M0CR, and the descriptions in the "Mode 0 Counting Timer Free Counting Block Diagram" and "Mode 0 Counting Timer Reload Counting Block Diagram" have been updated from "PRS[2:0]" to "M0CR.PRS," with the output changed from "CHA/CHB" to "TOG TOGN"; In the "Mode 1 Pulse Width Measurement (PWC)" chapter, the description has been revised from "and stores the current count value in CMAR while generating a capture interrupt CAF" to "and stores the current count value in CCR0A while generating a capture interrupt PWC," and the PWC waveform timing diagram has been updated from "CAF" to "PWC"; In the "Mode 2/3 Compare Capture Mode - Counter" chapter, the counter basic unit has been updated from "clock selection control registers FLT, CR0, MSCR, CR" to "clock selection control registers FLTR, M23CR, MSCR"; In the chapters "Mode 2/3 Compare Capture Mode", "Mode 2/3 Slave Mode", "Timer Triggering ADC" and "Master-Slave Mode Control Register (ATIM3_MSCR)", the register name CR has been changed to M23CR; In the "Mode 2/3 Compare Capture Mode - Repeat Counting" chapter, the "Repeat Counter Update Timing Diagram" has been split into "Repeat Counter Update Timing Diagram with Overflow Masking" and "Repeat Counter Update Timing Diagram with Underflow Masking"; In the "Mode 2/3 Compare Capture Mode - Compare Output OCREFA, Independent PWM Output, Complementary PWM Output" chapter, the block diagram for OCREFA's Output mode control input has been updated to include "CNT=CCRB"; In the "Quadrature Encoder Counting Function" chapter, a new description has been added to the table data: "The IAFP edge condition corresponds to the IBFP level signal; the IBFP edge condition corresponds to the IAFP level signal." For example, in MOD1, when IBFP is at a high level and IAFP detects a rising edge, the counter counts down; when IBFP is at a high level and IAFP detects a falling edge, the counter counts up," with optimized diagram descriptions; The "Timer Triggering ADC" chapter description has been optimized; In the "Mode 1 Pulse Width Measurement (PWC) Setup Example" chapter, Step 2 has been expanded with the description: "Then, based on the MSCR.TS setting, configure MSCR.IA0S/IB0S (IB0S cannot be set to 1) to select the channel to be measured"; In the "Control Register (ATIM3_M0CR)" chapter, the Bit 2 function description has been changed from "Internal counting clock PCLK" to "Internal counting clock TCLK"; In the "Mode 2/3 Compare Capture Mode Registers - 16-bit Mode Counting Register (ATIM3_CNT)" chapter, the description has been revised from "When using PWM compare output, the initial CNT value must be less than the ARR value" to "When using PWM compare output in triangular wave mode or sawtooth wave mode up-counting, the initial CNT value must be less than the ARR value"; In the "Control Register (TIM_M23CR)" chapter, the Bit[22:21] name has been changed from "Center-aligned A compare match mode" to "Channel A compare match mode," with optimized value descriptions, and the Bit 16 value 0 descrip-

Revision/Date	Changes
Rev1.20 Dec. 25, 2024	tion has been updated to "Reserved"; The Bit 15/14/3/2/1/0 descriptions in the "Channel 0 Control Register (ATIM3_CRCH0)" and "Channel 1/2 Control Register (ATIM3_CRCH1/2)" chapters have been optimized. 10. CTRIM: In the "RCH Clock Calibration Mode" and "RCL Clock Calibration Mode" chapters, the signal in the working block diagram has been changed from "xxF" to "xx," and the CR1[7:4] divider setting has been updated to "2-32768"; In the "Low-Power Timer Mode" chapter, the description has been revised from "Users can output the TOG signal to a pin via the GPIO auxiliary function to drive external circuits" to "Users can output the TOG signal to a pin via the GPIO port multiplexing function to drive external circuits." The signal in the working block diagram has been changed from "xxF" to "xx," and the CR1[7:4] divider setting has been updated to "2-32768"; In the "RCH Clock Auto-Calibration Example" and "RCL Clock Auto-Calibration Example" chapters, Step 6's description has been updated from "CTRIM_CR0.FLIM" to "CTRIM_FLIM"; In the "Interrupt Status Clear Register (CTRIM_ICR)" chapter, the reset value has been changed from "0x0000001F" to "0x0000003F". 11. WWDT: In the "Features" chapter, the description has been changed from "6-bit down counter" to "7-bit down counter"; In the "Configuration Example of Watch-dog Window Overflow Period" chapter, the description has been revised from "when the down counter value is less than or equal to the window value" to "when the down counter value is less than or equal to the window value and greater than 0x3F"; In the "Control Register 0 (WWDT_CR0)" chapter, the Bit 7 description has been updated from "Cannot be disabled once enabled" to "This bit cannot be cleared by software once set to 1"; In the "Control Register 1 (WWDT_CR1)" chapter, the Bit[6:0] description has been modified from "Watch-dog window value" to "User program can only update the WWDT counter value when the counter value is less than or equal to the window value and greater than 0x3F". 12. LPUART: The "Features" chapter description has been updated from "Two-wire full-duplex communication" to "Synchronous half-duplex, asynchronous full-duplex" with the addition of "16-bit baud rate counter"; The description format has been optimized in the "Mode0 Data Transmission/Reception", "Mode1 Data Transmission/Reception", "Mode2/Mode3 Data Transmission/Reception", "Single-wire Half-duplex Data Transmission", "Frame Error Detection", and "Multi-processor Communication" chapters; In the "Data Transmission Example" chapter, Step 8 description has been modified from "Write data to be transmitted to the DR register" to "When ISR.TXE is detected as 1, write data to be transmitted to the DR register", with the original Step 10 becoming Step 11 and its description changed from "Jump to Step 7" to "Jump to Step 8", while adding a new Step 10 description. 13. SPI: In the "SPI Status Register (SPI_SR)" chapter, the reset value is changed from "0x00000001" to "0x00000201". 14. I2C: In the "Operating Modes" chapter, the slave receive mode description has been changed from "If the master loses arbitration, it will directly enter slave receive mode" to "If the master loses arbitration, it will directly enter unaddressed

Revision/Date	Changes
Rev1.20 Dec. 25, 2024	slave receive mode"; In the "Status Register (I2C_STAT)" chapter, the reset value has been modified from "0x00000000" to "0x000000F8"; In the "Slave Address 0 Register (I2C_ADDR0)" chapter, the Bit[7:1]'s name has been updated from "ADDR" to "ADDR0". 15. ADC: In the "External Trigger Conversion Mode" chapter, the description in the diagram has been changed from "ADC_Ext_Sel" to "ADC_EXTTRIGGER.TRIGSEL". 16. LVD: In the "Hysteresis Function" chapter, the description "threshold voltage 20mV" has been changed to "hysteresis voltage/2", with optimized graphical representation of hysteresis characteristics and output waveforms. 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 Jul. 1, 2024	1. SYSCTRL: The EXTCLK related content is deleted. 2. FLASH: In Table 7-1 "FLASH Capacity Allocation", the addresses of Sector33 and Sector34 are swapped; the Flash erase/write programming description in section 7.4 is modified with added operational instructions for FLASH_CR.OP; the functional description of the OP bit in the FLASH_CR register (section 7.6.2) is revised. 3. GTIM: Added prescaler value limitations for GTIM capture function, and specified that capture function cannot be used when counting with external ETR; Added note to PRS bit of GTIM_CR0 register: Prescaler coefficient for capture mode can only be set to 0, 1, 2, or 3. 4. ATIM: In Table 11-1 "Timer Register List", ATIM0 is deleted; ATIMx is changed to ATIM3; In the ATIMx_M23CR register's DIR bit description, "write invalid" is modified to "write prohibited". 5. CTRIM: In the CTRIM_CR1 register's PRS bit, "1000: 256 division" is added.
Rev1.02 Jun. 21, 2023	1. GTIM: In the "Control Register 1 (GTIM_CR1)" chapter, the filter information is modified. 2. ATIM: In the "Mode 1 Register Description" chapter, the control register (ATIMx_CR0) adds descriptions of unopened functions. 3. Device Electronic Signature: In the "Product Unique Identification Code (UID) Register" section, the base address is changed from 0x00100980 to 0x00101080; in the "Pin Count Register" section, the base address is modified from 0x001008AA to 0x00100DBA.
Rev1.01 Mar. 9, 2022	The company logo is updated.
Rev1.00 Dec. 31, 2021	First official release.